

## Reconstruction of Gene Regulatory Networks from Temporal Microarray Data Using Order Estimation Criteria and Artificial Neural Networks

Azhar Salim, Faramarz Valafar\*  
Department of Computer Science  
San Diego State University  
5500 Campanile Drive  
San Diego, CA – 92182-7720

Email: [salim@rohan.sdsu.edu](mailto:salim@rohan.sdsu.edu), [faramarz@sciences.sdsu.edu](mailto:faramarz@sciences.sdsu.edu)

**Abstract - Gene regulatory networks allow us to study and understand genes' roles in biological processes. Among others, regulatory networks help to identify pathway initiator genes and therefore potential drug targets.**

In this paper, we discuss mining temporal microarray data for regulatory network information. For this study we have used simulated data in order to be able to verify our results. The data set was generated using a simulation model proposed by Wahde and Hertz [1]. For data mining, we have used *order estimation criteria*, in conjunction with artificial neural networks. This approach allows us to incorporate a priori biological knowledge, and thereby reduces dimensionality. This in turn produces more determined and accurate models of regulatory networks.

Our experiments show that this approach produces more determined and more accurate models of the regulatory network. This was especially true in cases where the number of genes is much greater than the number of time points at which gene expression is measured.

### 1. Introduction

---

*Gene regulatory networks* or *regulatory pathways* could be considered as the wiring diagrams for the complete biological system of an organism and are crucial to understanding the ways an organism reacts to perturbations in its environment [2]. Such networks, among others, can aid biologists in identifying potential drug targets, and in gaining insight into which genes perform multiple functions and thereby participate in different biological processes. Knowledge of the regulatory network that controls the immune system, for instance, would help us study and understand the intricacies of its operation. This could lead to genetic drug development that would fine tune the immune system to

better recognize and eliminate, for example, cancerous cells.

With recent advances in genomics, it is now possible to measure gene expression levels on a genomic scale with techniques such as microarrays. However, microarray technology is still in its adolescence. For instance, there are no standards for dissemination of data and results within the scientific community [3]. Standards such as MIAME [4] have been proposed, but have yet to be widely accepted. Furthermore, even though microarrays can measure gene expression of several thousand genes simultaneously, they are not as accurate as techniques such as RT-PCR, Northern or Southern blots that measure the expression of a few genes at a time. Ignoring human error in creating the RNA sample, the microarray technology, in itself, isn't very quantitative. For example, according to a brochure from Affymetrix (Affymetrix is a leading manufacturer of microarray chips) [5], "GeneChip arrays can detect a two-fold change of a transcript at a concentration of approximately one copy in 200,000 with a false positive rate of one percent and a detection rate of 95 percent." Therefore, changes in expression level that measure less than two-fold may not be real.

Once the microarray data has been collected and processed, the next step is the extraction of meaningful biological information. This is a challenging task that has been attempted with limited success since the invention of the technology. [3, 6] All along, the hope has been that various types of biological information (knowledge) can be extracted from microarray data. One such expectation lies in the area of reconstruction of regulatory networks from temporal microarray experiments. Many techniques have been suggested for this purpose. Examples of these are Boolean networks [7, 8], differential equations [9, 10, 11], neural networks [12, 13], evolutionary algorithms [14, 15, 16, 17], and several others [18, 19, 20]. There are also a few packages available that automate this task [21].

A major obstacle faced by all mentioned techniques is the *curse of dimensionality*. Curse of dimensionality is a term coined by Richard Bellman applied to the problem caused by the rapid increase in volume associated with adding extra dimensions to a (mathematical) space [22]. In the case of microarrays, the variables (or features) are the genes which are in the order of thousands, and the observations (the time points at which gene expression levels are measured) are typically in the range of 5-20. In order to develop a meaningful model and produce a good estimation of the regulatory network, all the above mentioned techniques require a number of time points that is in the same order as the number of genes. This is infeasible given the current prohibitive cost of microarrays. It is, therefore, the case that the models generated by the above mentioned algorithms are commonly underdetermined (i.e. there are more than one model that satisfies the given data).

Most of the recent work in reconstructing gene regulatory interactions from gene expression data involves the incorporation of biological knowledge at some point into the model to reduce dimensionality and improve the accuracy (determinedness) of the model [9, 17, 20, 23, 24]. Several of these studies have accomplished this by using an order estimation criterion such as Akaike's Information Criterion (AIC) [9, 17, 23, 24].

## 2. Data

Our experiments were run on an artificially generated set of temporal microarray data. We have chosen to test our experiments on artificial data in order to be able to verify our results. The "simulated" microarray data is computer generated data that mimics temporal gene expression data similar to those recorded by microarray. We use the following equation (introduced by Wahde and Hertz [1]) to generate the simulated data:

$$S_i(t + \Delta t) = S_i(t) + \frac{\Delta t}{\tau_i} \left[ -S_i(t) + g\left(\sum_j \omega_{ij} S_j(t) + b_i\right) \right] \quad \text{Eq. 1}$$

where,  $g(z) = \frac{1}{1 + e^{-z}}$  and  $\omega_{ij}$  represents an element of the weight matrix.  $b_i$  represents the tissue-dependent bias of gene  $i$ .  $\tau_i$  represents the time constant of gene  $i$ . The time constant is introduced in order to take into account the fact that a gene's regulatory effects on other genes may extend over several measured time points.

The weight matrix  $\omega$ , bias vector  $b$ , and time constant vector  $\tau$  are randomly generated. Expression

levels for  $M$  genes are randomly generated for the initial time point. Using Eq. 1, the expression levels for the  $M$  genes is simulated for  $N$  time points. Since it is common practice to conduct microarray experiments in duplicates (sometimes triplicates) in order to improve the accuracy of the data, we have also generated triplicate sets of expression values for all genes, at all time points.

## 3. Materials and Methods

Our approach combines *order estimation criteria* and artificial neural networks. Order estimation criteria are used to estimate the set of genes that are involved in the regulation of each gene. This creates a matrix of ones and zeros that will be used as a mask. This mask is used to determine the topology of the artificial neural network. Neural networks are then built using this mask and trained to estimate the level of interactions between genes.

This approach allows us to reduce the dimensionality of the problem space and thereby infuse a priori biological knowledge into our pattern recognition system. For instance, the knowledge of known regulatory interactions could easily be incorporated into the design of a mask.

Moreover, it is known that a gene interacts with (regulates, or is regulated by) only a handful of other genes in order to accomplish its role in a biological process. Thus, the  $N$  by  $N$  matrix,  $\Xi$  that describes the interaction between the  $N$  genes of an organism would be a highly sparse matrix. In  $\Xi$ , an element in row  $i$  and column  $j$  reflects the effect of the  $j^{\text{th}}$  gene on the  $i^{\text{th}}$  gene within the genome. Our approach allows us to easily create such sparse models.

Since  $N$  is usually large (typically in the order of several thousands for highly evolved species like *Homo sapiens*), estimation of the  $N^2$  values in  $\Xi$  is a significant computational task. Since we know that  $\Xi$  is a sparse matrix, it would be helpful to know in advance which elements of  $\Xi$  are non-zero. The number of non-zero elements in row  $i$  of  $\Xi$  would represent the number of genes that interact with gene  $i$ . This value is commonly referred to as the *in-degree* for that gene. Taking into account the sparseness of the matrix will significantly simplify the pattern recognition engine that is used to estimate the regulatory network.

### Order Estimation Criterion

To determine the in-degree of the genes under study, we will use and compare the results of three different order estimation criteria. Order estimation criteria are commonly used to find appropriate values for

the parameters of models that deal with time series data. Each of these criteria attempts to select the most suitable model to fit a problem. This is done by attempting to strike a balance between reducing the magnitude of the residual variance of the estimated model and increasing the number of model parameters [25, 26].

Two popular criteria for order estimation are *Akaike's Information Criterion* (AIC) [27], and *Schwarz Information Criterion*, also known as *Bayesian Information Criterion* (BIC) [28]. When the number of fitted parameters in the model is large relative to the sample size, it has been suggested to use a second order bias-corrected version of AIC called  $AIC_c$  [29, 30, 31]. The scenario in which the number of parameters in the fitted model is larger than the sample size is common in microarray data analysis.

We use the following formula to calculate AIC:

$$AIC = N \ln\left(\frac{RSS}{N}\right) + 2 * K \quad \text{Eq. 2}$$

where,  $RSS$  is the residual sum of squares of the fitted model (we used multiple linear regression for the models),  $N$  is the sample size (or the number of time points for which gene expression is measured), and  $K$  is the number of fitted parameters (or the number of non-zero interactions) [27, 31]. We calculate BIC as follows:

$$BIC = N \ln\left(\frac{RSS}{N}\right) + \ln(N) * K \quad \text{Eq. 3}$$

which as Schwarz notes, only differs from AIC in that the dimension (or number of parameters) is multiplied by  $\frac{1}{2} * \log(N)$ , and that BIC leans more towards lower-dimensional models. He also notes that for large numbers of observations, the procedures differ markedly from each other [28].  $AIC_c$  is calculated as follows [30, 31]:

$$AIC_c = AIC + 2 * K * \frac{K+1}{N-K-1} \quad \text{Eq. 4}$$

$AIC_c$  converges to AIC for large  $N$  [30, 31].

Kuha [32], in his comparison of AIC and BIC assumptions and performance, notes that both AIC and BIC provide well-founded and self-contained approaches to model comparison, although with different motivations and partially different objectives. He notes that BIC is typically a consistent model selector, while AIC is not always consistent. This is because AIC always has a probability of selecting a model that is too large. He also mentions that adjusted versions of AIC such as Hurvich and Tsai's modified  $AIC_c$  behave much better in this respect.

After testing both AIC and BIC in several cases of examples, he concludes that when the criteria agree on the best model, it provides reassurance on the robustness of the choice [32].

### **Artificial Neural Networks (ANNs)**

For our experiments, we used single layered feed forward  $\Delta$ -rule neural networks. The input neurons and the output neurons of these ANNs correspond to the genes under study. Therefore, there are as many input and output neurons as the number of genes. The input values are the expression levels of the genes at time  $t$ , and the output values of the ANN are the expression levels of the same genes at time  $t + \Delta t$ . This type of ANN is called a regression ANN. The single layered ANN is easier to study for our experiments because the weight matrix of the ANN can be directly taken to be the matrix,  $\Xi$ , of the regulatory interactions between the genes.

As mentioned in Section 2, it is common practice in today's microarray experiments to have duplicates or triplicates of gene expression to ensure the accuracy of the data by adding redundancy. For this reason, our simulated data is also generated in triplicates. We combine the three sets of data to create training set and testing sets.

### **Estimation of Artificial Neural Network (ANN) topology**

Training fully connected ANNs to model a sparse problem is a difficult task. An appropriately sparse topology would significantly reduce the training complexity of the ANN. If we knew beforehand which set of genes interacted with others, we could customize the topology of the ANN and create partially connected ANNs that better fit the biological model. These customized ANNs would only have a connection from an input neuron/gene to an output neuron/gene if the input gene corresponding to that input neuron was shown to have an interaction with the gene corresponding to the output neuron. We could then train these ANNs to determine the magnitude of the interactions between the genes.

In this paper, we used order estimation criteria mentioned above to estimate the network topology. In this process we randomly generated several thousands of masks, each representing a different combination of gene interactions. Each mask was evaluated with all three of the above mentioned criteria. The masks that scored the lowest values for each criterion were used to create the topology of the ANNs.

Similar to the approach of De Hoon et al [9], we set up a mask  $\Lambda$  which is an  $N$  by  $N$  matrix of zeros and

ones. A non-zero value in row  $i$  and column  $j$  of  $\mathbf{A}$  implies that gene  $i$  is regulated by gene  $j$ .

We evaluated each mask as follows: Using the mask under evaluation, we determined the non-zero columns in row  $i$  of the mask. These represented the genes interacting with gene  $i$ . We used multiple linear regression to determine coefficients that would model the expression of gene  $i$  at time  $t + \Delta t$  in terms of the expression of the interacting genes (according to the current mask) at time point  $t$ . Using these coefficients, we calculated the estimated gene expression at the time point  $t + \Delta t$ . We similarly calculated the estimated expression for all the genes at all the time points. We then calculated the residuals as the difference between the estimated gene expression at a time point and the actual gene expression at that time point. The sum of the squares of all the residuals was then calculated and applied in the estimation criteria. Masks that showed the best values for each of the criteria are used to customize the topologies of the ANNs. Using these masks, we could infuse the sparseness property onto the regulatory matrix.

#### 4. Experiments and Results

All experiments were coded and run in MATLAB.

*Data generation:* Triplicate gene expression levels were generated using Eq. 1 as described in the Section 2 for the following four scenarios:

1. 4 genes, 5 time points.
2. 4 genes, 10 time points.
3. 60 genes, 5 time points.
4. 100 genes, 5 time points.

*Mask generation and selection:* Several thousand masks were randomly generated for each of the above four scenarios. In generating any mask, the in-degree of genes had an upper bound of 20. This is done because, as mentioned earlier, it is known that genes only interact with a handful of other genes in any biological process.

The masks that scored the lowest for each criterion were selected for topological adjustment of ANNs. Since this is only a single layered network, the weights of the ANN correspond directly to the level of regulatory interactions between the genes.

In the case of the first two datasets, since there are four genes, the mask matrix is 4x4 dimensional. Therefore, the total number of possible masks is  $2^{(4 \times 4)} (2^{16} = 65536)$ . For these two datasets all possible masks were generated and evaluated using the three information criteria described as in Section 3. The three masks that

had the least values (best score) for the three criteria were used to create the topologies of the ANNs.

For the larger datasets, generating and evaluating every possible mask is impractical. For these cases, we generated and evaluated  $10^5$  random masks. Binary search was used to ensure that a mask is not evaluated more than once.

*ANN creation and training:* For each of the above four scenarios, three types of ANNs have been trained. For each type 100 ANNs with different random initial weights were created and trained. The three sets of experiments (types of ANNs) are:

- I. Fully connected ANNs.
- II. Partially connected ANNs created according to the topology described by the target mask.
- III. Partially connected ANNs created according to the topology described by each of the best masks chosen by the estimation criteria.

In order to test the performance of the regression networks we use ESDR (error standard deviation ratio) statistic. A significant indication of regression performance is given by the standard deviation of error during testing. We call this value TESD (testing error standard deviation) in short. Similarly, we define LESD to be the standard deviation of the error generated during learning. We denote the ratio of TESD over LESD as ESDR (Eq. 5). A value significantly below 1.0 for ESDR indicates good regression performance. A ratio below 0.1 is often said to heuristically indicate good regression [33].

$$ESDR = \frac{TESD}{LESD} \quad \text{Eq. 5}$$

Table 1, shows the results obtained from training the different ANNs for experiment set I (fully connected ANNs).

**Table 1: Results from 100 fully connected ANNs (each with different random initial weights) trained for each dataset.**

Number of Genes	Number of Time Points	Mean ESDR	Best ESDR
4	5	2.7723	0.8284
4	10	4.3660	0.2391
60	5	1.3014	1.0578
100	5	1.8415	1.0055

For experiment set II, the results of training the partially connected ANNs are shown below:

**Table 2: Results from 100 partially connected ANNs (each with different random initial weights) trained for each dataset according to the topology described by the target mask.**

Number of Genes	Number of Time Points	Mean ESDR	Best ESDR
4	5	2.9258	2.2458
4	10	1.1421	1.1052
60	5	0.9897	0.9837
100	5	2.8848	2.3511

For experiment set III, in the case of the first dataset, all three criteria chose the same mask. For the other three datasets, the AIC and BIC criteria chose the same mask, while the best AIC<sub>c</sub> mask was different. The percentage error of the best masks chosen by the criteria when compared to the target mask is shown below:

**Table 3: Results showing the percentage of elements of the best masks that were different from their corresponding elements in the target mask.**

Number of Genes	Number of Time Points	Percentage error in best AIC/BIC mask	Percentage error in best AIC <sub>c</sub> mask
4	5	68.75	68.75
4	10	43.75	37.5
60	5	25.50	26.31
100	5	15.10	17.28

Since AIC and BIC always chose the same masks, only the masks chosen by AIC and AIC<sub>c</sub> were used to determine the topologies of the ANNs for each dataset. The results are shown in Tables 4 and 5.

**Table 4: Results from 100 partially connected ANNs (each with different random initial weights) in accordance with the mask chosen by AIC and BIC criteria.**

Number of Genes	Number of Time Points	Mean ESDR	Best ESDR
4	5	3.1331	1.7011
4	10	1.4641	1.4567
60	5	0.9827	0.9793
100	5	3.9338	2.4195

**Table 5: Results from 100 partially connected ANNs trained (each with different random initial weights) in accordance with the mask chosen by the AIC<sub>c</sub> criteria.**

Number of Genes	Number of Time Points	Mean ESDR	Best ESDR
4	5	3.1331	1.7011
4	10	1.7297	1.4471
60	5	1.2151	1.2091
100	5	3.4409	2.3930

## 5. Discussions

As mentioned before, theoretically all mentioned methods should perform better when the number of time points is on the same order or higher than the number of genes. The first dataset reflects this. When the fully connected ANNs were trained for this dataset, the mean ESDR was much greater than one, while the best ESDR was significantly lower. This large difference is indicative of the underdetermined nature of this model. In other words, while there is enough data to find a good solution, there is still not enough data to find the best solution every time.

The partially connected ANNs created using the target mask also did not display good regression for this dataset. This could be due to the lack of sufficient information to train the network.

Our results when using the information criteria to select a good mask were not very successful for this dataset as can be seen from the 68.75% error in the best masks compared to the target mask. This was confirmed by the partially connected ANNs that produced, both, mean and best ESDRs, higher than the fully connected ANNs. The lack of sufficient data points for multiple linear regression models fitting the masks could have been a factor for the low performance.

The second dataset shows the flip side to (or at least the milder case of) the curse of dimensionality. In this dataset, the number of time points is greater than the number of genes. The fully connected ANNs had a large value for the mean ESDR, but a very small value for the best ESDR indicating that a much better solution than the average case can be achieved with this dataset. Using the target mask to create the partially connected ANNs showed improvement in the mean ESDR. There was also a significant reduction in the difference between the mean and best ESDRs.

Using the estimation criteria for mask selection fared better for this dataset. Here, we observe a reduced error percentage for the best masks. However, this percentage was still high. The ANNs created using these masks had a much smaller value for the mean ESDR as compared to the fully connected ANNs. The difference between the mean and best ESDR was also lower, especially for the AIC mask. For this dataset, as expected, the ANNs created using the target mask performed the best.

The third dataset, with 60 genes, is more realistic and typical of microarray datasets with much larger number of genes than number of time points. The results when training a fully connected ANN were good considering that the mean ESDR was only slightly larger than 1. When the ANNs were trained incorporating the

target mask, both mean and best ESDRs were below 1 showing good regression from the ANNs. The models are also more determined.

The mask selection error percentage was much lower for this dataset compared to the previous dataset. This is also reflected in the results of the ANNs, especially when trained with the AIC mask which had the lower error percentage. Both the mean and best ESDRs for these ANNs were less than 1. The AIC<sub>c</sub> mask had a slightly higher error percentage than the AIC mask, and this also was reflected in the mean and best ESDRs shown in Table 5. For ANNs trained with either of the two masks, the difference between the mean and best ESDRs when using the masks is very small, showing a more determined model.

The final dataset, with 100 genes, considers the scenario where the number of genes is much greater than the number of time points. The results from the fully connected ANNs showed a high mean ESDR which was expected given the high dimensionality of this dataset. The large difference between the mean and the best ESDR also shows that the model was underdetermined. The results for partially connected ANNs according to the target mask were poor. Both mean and best ESDRs were much greater than 1

This dataset displayed the smallest percentage error between the chosen masks and the target mask. However, the ANNs incorporating the AIC<sub>c</sub> mask demonstrated much higher mean and best ESDR values. This was in line with our expectations, as this dataset suffers more from the curse of dimensionality). Here, ESDR values showed poor regression in spite of using the mask.

## 6. Conclusions

---

Reconstruction of regulatory networks is a complex task with a high potential reward. In this paper we described experiments that show better performance in networks that incorporate biological knowledge such as limited interactivity between genes.

Our experiments demonstrated that incorporation of such a priori knowledge created models that were less underdetermined as compared to fully connected ANNs. The combination of order estimation criteria created ANNs that more closely resembled the underlying regulatory network. Our experiments also demonstrated that AIC and BIC criteria function in a similar fashion.

Future work in this direction is required to further develop these models so that they can discover nonlinear relationships in regulatory networks.

## 7. References

---

- [1] Wahde, M., and Hertz, J. (2001) Modeling Genetic Regulatory Dynamics In Neural Development. *Journal of Computational Biology*, 8(4):429-442.
- [2] Kurehar, M. P., Adak, S., Jhunjhunwala, S., Raghupathy, K. (2002) Genome-wide pathway analysis and visualization using gene expression data. *Pacific Symposium on Biocomputing*.
- [3] Bassett, D. E. Jr, Eisen, M. B., Boguski M. S. (1999) Gene expression informatics - it's all in your mine. *Nat Genet.* 21: 51-55.
- [4] Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., et al. (2001) Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. *Nat Genet, nature.com*
- [5] RNA Expression Analysis with the GeneChip® System, Retrieved from the Affymetrix Web site: [http://www.affymetrix.com/support/technical/other/rna\\_system\\_brochure.pdf](http://www.affymetrix.com/support/technical/other/rna_system_brochure.pdf)
- [6] Valafar F. (2002) Invited article: Pattern Recognition Techniques in Microarray Data Analysis: A Survey. *Special Issue Annals of New York Academy of Sciences. Techniques in Bioinformatics and Medical Informatics. Volume 980, 41-64.*
- [7] Akutsu, T., Miyano, S. and Kuhara, S. (1999) Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. *Pacific Symposium on Biocomputing*, 4, 17-28. <http://www.smi.stanford.edu/projects/helix/psb99/Akutsu.pdf>
- [8] Liang, S., Fuhrman, S. and Somogyi, R. (1998) REVEAL, a general reverse engineering algorithm for inference of genetic network architectures, *Pacific Symposium on Biocomputing* 3, 18-29.
- [9] De Hoon, M. J. L., Imoto, S., Kobayashi, K., Ogasawara N. and Miyano, S. (2003) Inferring gene regulatory networks from time ordered gene expression data of *Bacillus subtilis* using differential equations. *Pacific Symposium on Biocomputing.*, 8, 17-28.

- [10] Chen, T., He, H. L. and Church, G. M. (1999). Modeling gene expression with differential equations. Pacific Symposium on Biocomputing, 4, 29-40.
- [11] Perkins, T. J., Hallett, M., Glass, L., (2004) Inferring Models Of Gene Expression Dynamics. Journal of Theoretical Biology 230(3):289-299.
- [12] Wahde, M., Hertz, J. (2000). Coarse-grained reverse engineering of genetic regulatory networks. BioSystems 55, 129-136
- [13] Vohradský, J. (2001) Neural model of the genetic network. The Journal of Biological Chemistry, 276(39): 36168–36173.
- [14] Tominaga, D., Okamoto, M., Maki, Y., Watanabe, S., and Eguchi, Y. (1999) Nonlinear numerical optimization technique based on genetic algorithm for inverse problem: towards the inference of genetic networks, Proceedings. of the German Conference on Bioinformatics, 127–140.
- [15] Cumiskey, M., Levine, J., Armstrong, D. (2002) Gene Network Reconstruction Using a Distributed GA with a Backprop Local Search. Proceedings of the 1st European Workshop on Evolutionary Bioinformatics (EvoBIO 2003), Springer.
- [16] Ando, S., Sakamoto, E., Iba, H. (2002). Evolutionary modeling and inference of gene network. Information Sciences, 145(3-4):237–259.
- [17] Ando, S. and Iba, H. (2003). Construction of genetic network using evolutionary algorithm and combined fitness function. Genome Informatics, 14: 94{x}{2013}103.
- [18] D'Haeseleer, P., Liang, S., Somogyi, R. (2000) Genetic network inference: from co-expression clustering to reverse engineering. Bioinformatics, 16:707-726.
- [19] Kato, M., Tsunoda, T., and Takagi, T. (2000) Inferring genetic networks from DNA microarray data by multiple regression analysis, Genome Informatics 11 118-128.
- [20] Ong, I. M., Glasner, J. D., and Page, D. (2002) Modelling regulatory pathways in e. coli from time series expression profiles. Bioinformatics, 18(S1):S2418.
- [21] Wu, C. C., Huang, H. C., Juan, H. F., Chen, S. T. (2004) GeneNetwork: an interactive tool for reconstruction of genetic networks using microarray data. Bioinformatics, 20(18):3691–3.
- [22] Bellman, R. (1961) Adaptive Control Processes: A Guided Tour, Princeton University Press
- [23] Boscolo, R., Sabatti, C., Liao, J. C., Roychowdhury, V. P. (2005) A Generalized Framework for Network Component Analysis. IEEE/ACM Transactions on Computational Biology and Bioinformatics 2(4):289-301.
- [24] Nachman, I., Regev, A. & Friedman, N. (2004) Inferring quantitative models of regulatory networks from expression data. Bioinformatics 20 Suppl. 1, 1248{x}{2212}1256.
- [25] Priestley, M. .B. (1988) Non-Linear And Non-Stationary Time Series Analysis, Academic Press, London
- [26] Pankratz, A. (1991) Forecasting With Dynamic Regression Models, Wiley, New York
- [27] Akaike, H. (December 1974) A new look at the statistical model identification. IEEE Transactions on Automatic Control 19 (6): 716 – 723
- [28] Schwarz, G. (1978) Estimating Dimension of a Model. The Annals of Statistics, 6(2), 461-464.
- [29] Sugiura, N. (1978) Further analysis of the data by Akaike's information criterion and the finite corrections. Communications in Statistics, A7, 13-26.
- [30] Hurvich, C. M. and Tsai, C. L. (1989) Regression and time-series model selection in small samples. Biometrika, 76, 297–307.
- [31] Burnham, K.P., and Anderson, D.R. (2004) Multimodel Inference: understanding AIC and BIC in Model Selection, Amsterdam Workshop on Model Selection (available online: <http://www2.fmg.uva.nl/modelselection/presentations/AWMS2004-Burnham-paper.pdf>)
- [32] Kuha, J. (2004) AIC and BIC: Comparisons of assumptions and performance. Sociological Methods and Research, 33: 188–229.
- [33] Neural Networks, Retrieved from the StatSoft Web site: <http://www.statsoft.com/textbook/stneunet.html>
- [34] Shibata, R. (1981) An Optimal Selection Of Regression Variables. Biometrika, 68(1), 45-54.