Gene Regulatory Network Discovery from Time-Series Gene Expression Data – A Computational Intelligence Approach

Nikola K. Kasabov[1], Zeke S. H. Chan[1], Vishal Jain[1], Igor Sidorov[2] and Dimiter S. Dimitrov[2]

[1]Knowledge Engineering and Discovery Research Institute (KEDRI),
Auckland University of Technology, Private Bag 92006, Auckland, New Zealand
{nkasabov, shchan, vishal.jain}@aut.ac.nz
[2]National Cancer Institute, Frederick, Washington DC, National Institute of Health, USA
{sidorovi, dimitrov}@ncifcrf.gov

Abstract. The interplay of interactions between DNA, RNA and proteins leads to genetic regulatory networks (GRN) and in turn controls the gene regulation. Directly or indirectly in a cell such molecules either interact in a positive or in repressive manner therefore it is hard to obtain the accurate computational models through which the final state of a cell can be predicted with certain accuracy. This paper describes biological behaviour of actual regulatory systems and we propose a novel method for GRN discovery of a large number of genes from multiple time series gene expression observations over small and irregular time intervals. The method integrates a genetic algorithm (GA) to select a small number of genes and a Kalman filter to derive the GRN of these genes. After GRNs of smaller number of genes are obtained, these GRNs may be integrated in order to create the GRN of a larger group of genes of interest.

1 Introduction

Gene regulatory network is one of the two main targets in biological systems because they are systems controlling the fundamental mechanisms that govern biological systems. A single gene interacts with many other genes in the cell, inhibiting or promoting directly or indirectly, the expression of some of them at the same time. Gene interaction may control whether and how vigorously that gene will produce RNA with the help of a group of important proteins known as transcription factors. When these active transcription factors associate with the target gene sequence (DNA bases), they can function to specifically suppress or activate synthesis of the corresponding RNA. Each RNA transcript then functions as the template for synthesis of a specific protein. Thus the gene, transcription factor and other proteins may interact in a manner that is very important for determination of cell function. Much less is known about the functioning of the regulatory systems of which the individual genes and interaction form a part [6], [8], [15], [20]. Transcription factors provide a feedback pathway by which genes can regulate one another's expression as mRNA and then as protein [3], [5].

The discovery of gene regulatory networks (GRN) from time series of gene expression observations can be used to: (1) Identify important genes in relation to a disease or a biological function, (2) Gain an understanding on the dynamic interaction between genes, (3) Predict gene expression values at future time points. The major approaches that deals with the modelling of gene regulatory networks involve differential equations [14], stochastic models [16], evolving connectionist systems [13], boolean networks [18], generalized logical equations [21], threshold models [19], petri nets [11], bayesian networks [9], directed and undirected graphs.

We propose here a novel method that integrates Kalman Filter [4] and Genetic Algorithm (GA) [10], [12]. The GA is used to select a small number of genes, and the Kalman filter method is used to derive the GRN of these genes. After GRNs of smaller number of genes are obtained, these GRNs may be integrated in order to create the GRN of a larger group of genes of interest. The goal of this work is develop a method for GRN discovery from multiple and short time series data of a large number of genes. The secondary goal is to apply the method as to identify the genes that co-regulate telomerase from the extracts of the U937 plus and minus series obtained in NCI, NIH. Each series contains the time-series expression of 32 pre-selected candidate genes that have been found potentially relevant, as well as the expression of the telomerase. Both the plus series and minus series contains four samples recorded at the (0, 6, 24, 48) th hour. Discovering GRN from these two series is challenging in two aspects: first, both series are sampled at irregular time intervals; second, the number of samples is scarce (only 4 samples). A third potential problem is that the search space grows exponentially in size as more candidate genes are identified in the future. Several GRNs of 3 most related to the telomerase genes are discovered, analysed and integrated. The results and their interpretation confirm the validity and the applicability of the proposed method. The integrated method can be easily generalized to extract GRN from other time series gene expression data. This paper reports the methodology and the experimental findings.

2 Modelling GRN with first-order differential equations, state-space representation and Kalman Filter

2.1 Discrete-Time Approximation of First-Order Differential Equations

Our GRN is modelled with the discrete time approximation of first-order differential equations, given by:

$$\mathbf{x}_{t+1} = \mathbf{F}\mathbf{x}_t + \boldsymbol{\varepsilon}_t \tag{1}$$

where $\mathbf{x}_t = (x_1, x_2, ...x_n)'$ is the gene expression at the t-th time interval and n is the number of genes modelled, ε_t is a noise component with covariance E=cov (ε_t), and F=(f_{ij}) i=1,n, j=1,n is the transition matrix relating x_t to x_{t+1}. It is related to the continuous first-order differential equations $d\mathbf{x}/dt = \boldsymbol{\Psi}\mathbf{x} + \mathbf{e}$ by $\mathbf{F} = \tau\boldsymbol{\Psi} + \mathbf{I}$ and $\boldsymbol{\varepsilon}_t = \tau\mathbf{e}$

where τ is the time interval {note the subscript notation (t+k) is actually the common abbreviation for (t+kτ)}[7]. We work here with a discrete approximation instead of a continuous model for the ease of modelling and processing the irregular time–course data (with Kalman filter). Besides being a tool widely used for modelling biological processes, there are two advantages in using first-order differential equations.

First, gene relations can be elucidated from the transition matrix **F** through choosing a threshold value (ζ; $1>\zeta>0$). If $|f_{ij}|$ is larger then the threshold value ζ, $x_{t,j}$ is assumed to have significant influence on $x_{t+1,i}$. A positive value of f_{ij} indicates a positive influence and vice-versa. Second, they can be easily manipulated with KF to handle irregularly sampled data, which allow parameter estimation, likelihood evaluation and model simulation and prediction.

The main drawback of using differential equations is that it requires the estimation of n^2 parameters for the transition matrix **F** and n (n-1)/2 parameters for the noise covariance **E**. To minimize the number of model parameters, we estimate only **F** and fix **E** to a small value. Since both series contain only 4 samples, we avoid over-parameterization by setting n to 4, which is the maximum number of n before the number of parameters exceeds the number of training data {It matches the number of model parameters (the size of **F** is n^2=16) to the number of training data (n×4 samples =16)}. Since in our case study one of the n genes must be telomerase, we can search for a subset of size K=3 other genes to form a GRN.

To handle irregularly sampled data, we employ the state-space methodology and the KF. We treat the true trajectories as a set of unobserved or hidden variables called the *state variables*, and then apply the KF to compute their optimal estimates based on the observed data. The state variables that are regular/complete can now be applied to perform model functions like prediction, parameter estimations instead of the observed data that are irregular/incomplete. This approach is more superior to interpolation methods as it prevents false modelling by trusting a fixed set of interpolated points that may be erroneous.

2.2 State-Space Representation

To apply the state-space methodology, a model must be expressed in the following format called the *discrete-time state space representation*

$$\mathbf{x}_{t+1} = \mathbf{\Phi}\mathbf{x}_t + \mathbf{w}_t \tag{2}$$

$$\mathbf{y}_t = \mathbf{A}\mathbf{x}_t + \mathbf{v}_t \tag{3}$$

$$\text{cov}(\mathbf{v}_t) = \mathbf{R} \qquad \text{cov}(\mathbf{w}_t) = \mathbf{Q} \tag{4}$$

where, $\mathbf{x}_t$ is the system state; $\mathbf{y}_t$ is the observed data; $\mathbf{\Phi}$ is the state transition matrix that relates $\mathbf{x}_t$ to $\mathbf{x}_{t+1}$; $\mathbf{A}$ is the linear connection matrix that relates $\mathbf{x}_t$ to $\mathbf{y}_t$; $\mathbf{w}_t$ and $\mathbf{v}_t$ are uncorrelated white noise sequences whose covariance matrices are $\mathbf{Q}$ and $\mathbf{R}$ respectively. The first equation is called the *state equation* that describes the

dynamics of the state variables. The second equation is called the *observation equation* that relates the states to the observation.

To represent the discrete-time model in the state-space format, we simply substitute the discrete-time equation (1) into the state equation (2) by setting $\mathbf{\Phi}=\mathbf{F}$, $\mathbf{w}_t=\boldsymbol{\varepsilon}_t$ and $\mathbf{Q}=\mathbf{E}$ and form a direct mapping between states and observations by setting $\mathbf{A}=\mathbf{I}$. The state transition matrix $\mathbf{\Phi}$ (functional equivalent to $\mathbf{F}$) is the parameter of interest as it relates the future response of the system to the present state and governs the dynamics of the entire system. The covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ are of secondary interest and are fixed to small values to reduce the number of model parameters.

2.3 Kalman Filter (KF)

KF is a set of recursive equations capable of computing optimal estimates (in the least-square sense) of the past, present and future states of the state-space model based on the observed data. Here we use it to estimate gene expression trajectories given irregularly sampled data. To specify the operation of Kalman filter, we define the conditional mean value of the state x_t^s and its covariance P_{tu}^s as:

$$\mathbf{x}_t^s = E(\mathbf{x}_t \mid \mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_s) \tag{5}$$

$$\mathbf{P}_{tu}^s = E\left[(\mathbf{x}_t - \mathbf{x}_t^s)(\mathbf{x}_u - \mathbf{x}_u^s)' \mid \mathbf{y}_1, ..., \mathbf{y}_s\right] \tag{6}$$

For prediction, we use the KF forward recursions to compute the state estimates for ($s<t$). For likelihood evaluation and parameter estimation, we use the KF backward recursions to compute the estimates called the smoothed estimates based on the entire data, i.e. ($s=T$**;** $T>t$ is the index of the last observation), which in turn are used to compute the required statistics.

2.4 Using GA for the selection of a gene subset for a GRN

The task is to search for the genes that form the most probable GRN models, using the model likelihood computed by the KF as an objective function. Given N the number of candidates and K the size of the subset, the number of different gene combinations is $N!/K!(N-K)!$. In our case study, N=32 is small enough for an exhaustive search. However, as more candidates are identified in the future, the search space grows exponentially in size and exhaustive search will soon become infeasible. For this reason a method based on GA is proposed. The strength of GA is twofold:

1. Unlike most classical gradient methods or greedy algorithms that search along a single hill-climbing path, a GA searches with multiple points and generates new points through applying genetic operators that are stochastic in nature. These properties allow for the search to escape local optima in a multi-modal environment. GA is therefore useful for optimizing high dimensional functions and noisy functions whose search space contains many local optima points.

2. A GA is more effective than a random search method as it focuses its search in the promising regions of the problem space.

2.5 GA Design for Gene Subset Selection

In the GA-based method for gene subset selection proposed here, each solution is coded as a binary string of N bits. A "1" in the ith bit position denotes that the ith gene is selected and a "0" otherwise. Each solution must have exactly K "1"s and a repair operator is included to add or delete "1"s when this is violated. The genetic operators used for crossover, mutation and selection are respectively the standard crossover, the binary mutation and the (μ, λ) selection operators. Since there are two series – the plus and the minus series of time-course gene expression observations in our case study, a new fitness function is designed to incorporate the model likelihood in both series. For each solution, the ranking of its model likelihood in the plus series and in the minus series are obtained and then summed to obtain a joint fitness ranking. This favors convergence towards solutions that are consistently good in both the plus and the minus series. The approach is applicable to multiple time series data.

2.6 Procedures of the GA-based method for gene subset selection

Population Initialization. Create a population of μ random individuals (genes from the initial gene set, e.g. of 32) as the first generation parents.

Reproduction. The goal of reproduction is to create λ offspring from μ parents. This process involves three steps: crossover, mutation and repair.

- *Crossover*. The crossover operator transfers parental traits to the offspring. We use the uniform crossover that samples the value of each bit position from the first parent at the crossover probability p_c and from the second parent otherwise. In general, performance of GA is not sensitive to the crossover probability and it is set to a large value in the range of [0.5, 0.9] [1]. Here we set it to 0.7.
- *Mutation*. The mutation operator induces diversity to the population by injecting new genetic material into the offspring. For each bit position of the offspring, mutation inverts the value at a small mutation rate p_m. Performance of GA is very sensitive to the mutation probability and it usually adapts a very small value to avoid disrupting convergence. Here we use p_m=1/N, which has been shown to be both the lower bound value and the optimal value for many test functions [17], [1], providing an average of one mutation in every offspring.
- *Repair*. The function of the repair operator is to ensure that each offspring solution has exactly K "1" to present the indices of the K selected genes in the subset. If the number of "1"s is greater than K, invert a "1" at random; and vice-versa. Repeat the process until the number of "1"s matches the subset size K.

Fitness Evaluation. Here λ offspring individuals (solutions) are evaluated for their fitness. For each offspring solution, we obtain the model likelihood in the both the plus and the minus series and compute their ranking (lower the rank, higher the likelihood) within the population. Next, we sum the rankings and use the negated sum as fitness estimation so that the lower the joint ranking, the higher the fitness.

Selection. The selection operator determines which offspring or parents will become the next generation parents based on their fitness function. We use the (μ, λ) scheme that selects the fittest μ of λ offspring to be the next generation parents. It is worth comparing this scheme to another popular selection scheme (μ+λ) that selects the fittest μ of the joint pool of μ parents and λ offspring to be the next generation parents, in which the best-fitness individuals found are always maintained in the population, convergence is therefore faster. We use the (μ, λ) scheme because it offers a slower but more diversified search that is less likely to be trapped in local optima.

Test for termination. Stop the procedure if the maximum number of generations is reached. Otherwise go back to the reproduction phase.

Upon completion, GA returns the highest likelihood GRNs found in both the plus and the minus series of gene expression observations. The proposed method includes running the GA-based procedure over many iterations (e.g. 50) thus obtaining different GRN that include possibly different genes. Then we summarize the significance of the genes based on their frequency of occurrence in these GRNs and if necessary we put together all these GRNs thus creating a global GRN on the whole gene set.

3 Experiments and Results

The integrated GA-KF method introduced above is applied to identify genes that regulate telomerase in a GRN from a set of 32 pre-selected genes. Since the search space is small (only C_3^{32}=4960 combinations), we apply exhaustive search as well as GA for validation and comparative analysis.

The experimental settings are as follows. The expression values of each gene in the plus and minus series are jointly normalized in the interval [-1, 1]. The purpose of the joint normalization is to preserve the information on the difference between the two series in the mean. For each subset of *n* genes defined by the GA, we apply KF for parameter estimation and likelihood evaluation of the GRN model. Each GRN is trained for at least 50 epochs (which is usually sufficient) until the likelihood value increases by less than 0.1. During training, the model is tested for stability by computing the eigenvalues of (Φ-I) [2], [7]. If any of the real part of eigenvalues is positive, the model is unstable and is abandoned.

For the experiments reported in this paper a relatively low resource settings are used. Parent and offspring population sizes (μ, λ) are set to (20, 40) and maximum

number of generations is set to 50. These values are empirically found to yield consistent results over different runs. We run it for 20 times from different initial population to obtain the cumulated results. The results are interpreted from the list of 50 most probable GRNs found in each series (we can lower this number to narrow down the shortlist of significant genes). The frequencies of each gene being part of the highest likelihood GRNs in the plus and in the minus series are recorded. Next, a joint frequency is calculated by summing the two frequencies. The genes that have a high joint frequency are considered to be significant in both minus and plus series.

For exhaustive search, we simply run through all gene combinations of 3 genes plus the telomerase; then evolve through KF a GRN for each combination and record the likelihood of each model in both the plus and minus series. A similar scoring system as GA's fitness function is employed. We obtain a joint ranking by summing the model likelihood rankings in the plus series and the minus series, and then count the frequency of the genes that belong to the best 50 GRNs in the joint ranking. The top ten highest scoring genes obtained by GA and exhaustive search are tabulated in Table 1.

Table 1. Significant genes extracted by GA and through an exhaustive search from 32 selected genes

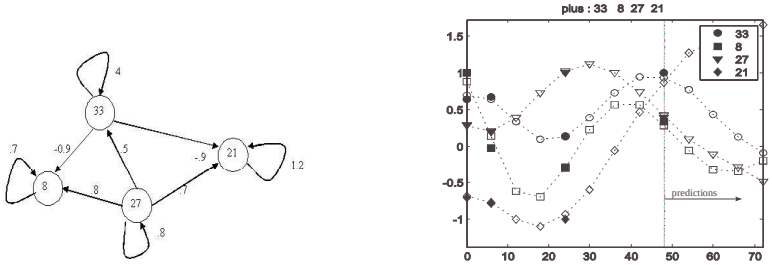
Rank	Indices of significant genes found by GA (Freq. of occurrence in Minus GRNs, Freq. of occurrence in Plus GRNs) and their accession numbers in Genbank	Indices of significant genes found by exhaustive search (gene Index)
1	27 (179,185) X59871	20 M98833
2	21 (261,0) U15655	27 X59871
3	12 (146, 48) J04101	32 X79067
4	32 (64, 118) X79067	12 J04101
5	20 (0, 159) M98833	6 AL021154
6	22 (118, 24) U25435	29 X66867
7	11 (0, 126) HG3523-HT4899	5 D50692
8	5 (111, 0) D50692	22 U25435
9	18 (0, 105) D89667	10 HG3521-HT3715
10	6 (75, 0) AL021154	13 J04102

The results obtained by GA and exhaustive search are strikingly similar. In both lists, seven out of top ten genes are common (genes 27, 12, 32, 20, 22, 5, 6) and four out of top five genes are the same (genes 27, 12, 32 and 20). The similarity in the results supports the applicability of a GA-based method in this search problem and in particular, when the search space is too large for an exhaustive search. An outstanding gene identified is gene 27, TCF-1. The biological implications of TCF-1 and other high scoring genes are currently under investigation.

The identified GRNs can be used for model simulation and prediction. The GRN dynamics can also be visualized with a network diagram using the influential information extracted from the state transition matrix. As an example, we examine one of the discovered GRN of genes (33, 8, 27, 21) for both the plus and minus series,

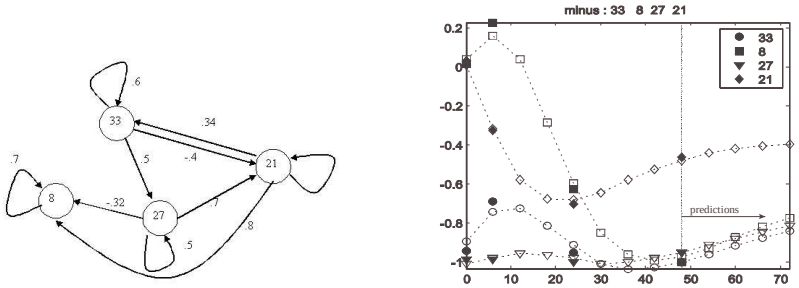
shown in Fig. 1 and Fig. 2 respectively. The network diagram shows only the components of Φ whose absolute values are above the threshold value ζ=0.3.

For the plus series, the network diagram in Fig. 1 (a) shows that gene 27 has the most significant role regulating all other genes (note that gene 27 has all its arrows out-going). The network simulation, shown in Fig. 1 (b) fits the true observations well and the predicted values appear stable, suggesting that the model is accurate and robust. For the minus series, the network diagram in Fig. 2 (a) shows a different network from that of the plus series. The role of gene 27 is not as prominent. The relationship between genes is no more causal but interdependent, with genes 27, 33 and 21 simultaneously affecting each other. The difference between the plus and minus models is expected. Again, the network simulation result shown in Fig. 2 (b) shows that the model fits the data well and the prediction appears reasonable.



(a) (b)

Fig. 1. The identified best GRN of gene 33 (telomerase) and genes 8, 27 and 21 for the plus series: (a) The network diagram (b) The network simulation and gene expression prediction over future time. Solid markers represent observations.



(a) (b)

Fig. 2. The identified best GRN of gene 33 (telomerase) and genes 8, 27 and 21 for the minus series: (a) The network diagram (b) The network simulation and gene expression prediction over future time. Solid markers represent observations.

3.1 Building a global GRN of the whole gene set out of the GRNs of smaller number of genes (Putting the pieces of the puzzle together)

After many GRNs of smaller number of genes are discovered, each involving different genes (with a different frequency of occurring), these GRNs can be put together to create a GRN of the whole gene set. Representation and illustration for the top five (fittest) GRNs from our experiment are shown in Fig3 and Table 3 respectively.

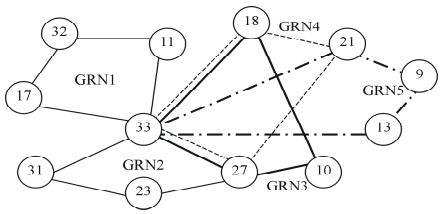


Fig. 3. The five highest likelihood GRN models found by GA in the plus series are put together.

Table 2. Illustration of top five fittest GRNs (plus series)

GRN Number	GRN identified
1	(33 32 17 11)
2	(33 31 27 23)
3	(33 27 18 10)
4	(33 27 21 18)
5	(33 21 13 9)

4 Conclusions

In this work, we propose a novel method that integrates Kalman Filter and Genetic Algorithm for the discovery of GRN from gene expression observations of several time series (in this case they are two) of small number of observations. As a case study we have applied the method for the discovery of GRN of genes that regulate telomerase in two sub-clones of the human leukemic cell line U937. The time-series contain 12,625 genes, each of which sampled 4 times at irregular time intervals, but only 32 genes of interest are dealt with in the paper. The method is designed to deal effectively with irregular and scarce data collected from a large number of variables (genes). GRNs are modelled as discrete-time approximations of first-order differential equations and Kalman Filter is applied to estimate the true gene trajectories from the irregular observations and to evaluate the likelihood of the GRN models. GA is applied to search for smaller subset of genes that are probable in forming GRN using

the model likelihood as an optimization objective. The biological implications of the identified networks are complex and currently under investigation.

References

1. Baeck, T., D. B. Fogel, et al.: Evolutionary Computation I and II. Advanced algorithm and operators. Bristol, Institute of Physics Pub (2000)
2. Bay, J. S. (ed.): Fundamentals of Linear State Space Systems, WCB/McGraw-Hill (1999)
3. Bolouri, J. M. B. a. H. (eds.): Computational modelling of Genetic and Biochemical Networks. London, The MIT Press (2001)
4. Brown, R. G. (ed.): Introduction to Random Signal Analysis and Kalman Filtering, John Wiley & Son (1983)
5. Brownstein, M. J., Trent, J.M., and Boguski, M.S., Functional genomics In M. Patterson and M. Handel (eds.): Trends Guide to Bioinformatics (1998) 27-29
6. Collado-Vides, J.: A transformational-grammar approach to study the regulation of gene expression, J. Theor. Biol. **136** (1989) 403-425
7. Dorf, R. and R. H. Bishop: Modern Control Systems, Prentice Hall (1998)
8. Fields, S., Kohara, Y. and Lockhart, D. J.: Functional genomics. Proc Natl. Acad. Sci USA 96 (1999) 8825-8826
9. Friedman, L., Nachman, Pe'er: Using Bayesian networks to analyze expression data. Journal of Computational Biology **7** (2000) 601-620
10. Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization and machine Learning Reading, MA, Addison-Wesley
11. Hofestadt, R. a. M., F.: Interactive modelling and simulation of biochemical networks Comput. Biol Med. **25** (1995) 321-334
12. Holland. H.: Adaptation in natural and artificial systems, The University of Michigan Press, Ann Arbor, MI (1975)
13. Kasabov, N. and D. Dimitrov: A method for gene regulatory network modelling with the use of evolving connectionist systems. ICONIP - International Conference on Neuro-Information Processing, Singapore, IEEE Press (2002)
14. Likhoshvai, V. A., Matushkin, Yu G., Vatolin, Yu N. and Bazan, S. I: A generalized chemical kinetic method for simulating complex biological systems. A computer model of lambda phage ontogenesis." computational technol. **5**, issue 2 (2000) 87-89
15. Loomis, W. F., and Sternberg, P.W.: Genetic networks. Science **269** (1995) 649
16. Mc Adams, H. H. a. A. A.: Stochastic mechanism in gene expression. Proc. Natl. Acad. Sci. USA **94** (1997) 814-819
17. Muhlenbein, H.: How genetic algorithms really work: I. mutation and hillclimbing. Parallel Problem Solving from Nature 2. B. Manderick. Amsterdam, Elsevier (1992)
18. Sanchez, L., van Helden, J. and thieffry, D.: Establishment of the dorso-ventral pattern during embryonic development of Drosophila melanogaster. A logical analysis. J. Theor. Biol. **189** (1997) 377-389
19. Tchuraev, R. N.: A new method for the analysis of the dynamics of the molecular genetic control systems. I. Description of the method of generalized threshold models. J. Theor. Biol. **151** (1991) 71-87
20. Thieffry, D.: From global expression data to gene networks. BioEssays **21** issue 11 (1999) 895-899
21. Thieffry, D. a. T. R.: Dynamical behaviour of biological regulatory networks-II. Immunity control in bacteriophage lambda. Bull. Math. Biol **57** (1995) 277-297