ANALYZING AND OPTIMIZING GENETIC NETWORK STRUCTURE VIA PATH-FINDING

J. Gebert, M. Laetsch
Institute of Mathematics, Center for Applied Computer Science, University of Cologne, Germany
e-mail: gebert@zpr.uni-koeln.de, laetsch@zpr.uni-koeln.de

E. M. P. Quek
Advanced Biotechnology and Breeding Centre, Malaysian Palm Oil Board, Kajang, Selangor, Malaysia
e-mail: emilyq@rocketmail.com

G.-W. Weber
Institute of Applied Mathematics, Middle East Technical University, Ankara, Turkey
e-mail: gweber@metu.edu.tr

Понимание взаимодействия генов играет важную роль в анализе сложных биологических систем. С помощью дискретной аппроксимации моделируются генные сети посредством системы линейных дифференциальных уравнений с использованием microarray data дрожжей Saccharomyces cerevisiae. Эти сети анализируются с помощью алгоритма Дейкстра. Создана специальная сеть, которая состоит из восьми генов, участвующих в метаболизме гликогена в дрожжах. Построенная сеть иллюстрирует небольшой пример обработки данных microarray экспериментов. Сделан вывод о способности алгоритма Дейкстра показывать максимальное взаимодействие генов в генных сетях и выбирать ключевые гены, имеющие важнейшее значение. Сделан обзор будущих исследований.

Introduction

Advances in microarray technology have allowed mass generation of gene-expression data where gene-expression levels of thousands of genes can be measured in parallel and simultaneously in one single experiment [17, 20]. This high-throughput technique gives a detailed picture of the gene interactions based on their expression level, in contrast to the classical molecular methods such as *northern blotting* [2], *differential display* [16] and *serial analysis of gene expression* (SAGE) [24] which only enabled us to focus on one gene or a limited number of genes at one time. These well-known methods require very much laborious work and hardly provide an insight of the interactions among many genes.

© Институт вычислительных технологий Сибирского отделения Российской академии наук, 2004.

Generally, the microarray technique is based on the hybridization between complementary nucleic acids which is the core capability of molecular biology as reviewed by [22]. A microarray consists of a collection of nucleic acid sequences immobilized onto a solid support or matrix such as a glass slide. These unique and known sequences which are individually spotted on a single matrix are called *targets*. In microarray experiments, mRNA is isolated from a sample and labeled with fluorescence dye via reverse transcription. The labeled sample is called *probe*, and it consists of thousands of different labeled nucleic acid fragments. The labeled fragments in the probe are then hybridized to their immobilized complementary targets on the microarray in order to form duplexes and hence, signals are produced. The measurements of the intensity of the fluorescence signals reflect the amount of each specific fragment in the probe, which corresponds to a known sequence or gene in the target.

Typically, in one microarray experiment, two probes which represent mRNA populations of two different cells, for instance, healthy cell and cancerous cell, are labeled with different fluorescence dyes. Comparative analysis of the intensity of these fluorescence signals, which are associated with each spot on the microarray, provides the relative abundance of specific sequence or gene in each sample. Hence, the gene-expression ratios which are obtained from these analyses indicate the cell state and activity of genes.

The microarray technique provides two types of information about gene-expression patterns [8]. Firstly, the expression pattern tells about the type of tissue(s) where the expression of one gene occurs and, secondly, the relation of the expression pattern of one gene with other genes. Data accumulated from microarray experiments help biologists to elucidate the cellular response of a cell, tissue, or organism to many different environments, developmental stages and genetic stimuli. In addition, the observed gene-expression patterns help biologists to investigate the underlying gene regulatory pathways.

One of the most challenging problems in molecular and cellular biology is to infer the underlying gene regulatory networks at the system level. There are many ways to define a genetic network. Generally, a genetic network is defined as a collection of molecular components such as a group of genes in which the individual gene can influence or change the activity of other genes [14, 25]. A cellular function is then carried out collectively by the interaction between these genes.

Many attempts have been performed to infer gene regulatory networks. Various approaches are available for modeling a genetic network mathematically such as differential equation models [4, 11], linear models [23], neural networks [26], Bayesian networks [9] and Boolean networks [13]. These approaches require different levels of abstraction details.

1. Mathematical Modeling of A Genetic Network

In mathematical modeling, a genetic network can be inferred from the gene-expression data by fitting the gene-expression data into a mathematical model in order to get best-fit parameters. In this study, the mathematical model, given by a system of linear differential equations, is implemented to represent gene-expression data in order to generate a genetic network. Thus, data from gene-expression analysis such as microarray data are modeled mathematically using a system of linear differential equations as described in Eq. (1):

$$\dot{E}(t) = ME(t). \tag{1}$$

We remark that in [10, 11] the more general model $\dot{E}(t) = M(E)E(t)$ has been considered from both the aspects of modeling and stability. The right-hand side of Eq. (1) consists of a matrix M and a function $E(t)$. The matrix M is a coefficient matrix that describes the interactions between the genes. It is an nxn matrix, where the rows and columns represent the n genes. The entries in M, namely m_{ij}, where i, $j \in \{1, 2, \ldots, n\}$, are called the coefficients of the system of linear differential equations. These coefficients are used to generate a genetic network, and we try to investigate them in this study. Normally, these coefficients are fixed and the generated networks are static networks. An element of M, m_{ij} represents the effect or influence of gene j on gene i.

In the modeling of a genetic network, all expression-data are represented as column vectors with n genes as the elements of each vector. Thus, $E(t)$ describes the gene-expression patterns at a particular time t. The components in $E(t)$ actually represent the expression level of individual genes that can be obtained from microarray data. In other words, $E(t)$ contains the concentration of mRNA as a function of time. The left-hand side of Eq. (1) consists of the derivative of $E(t)$, i.e., $\dot{E}(t)$. Thus, $\dot{E}(t)$ represents the change of gene-expression patterns with infinitesimally small changes of time.

In order to yield best-fitting parameters, we designed a mathematical model in which the expression data from microarray experiments can be fitted well into that model. The model itself actually represents the function (problem) of linear differential equations. An agreement between gene-expression data and the model with a particular choice of parameters must be achieved. Thus, the parameters of the model are adjusted to achieve a minimum in the function. The process of this adjustment is a problem of minimization in many dimensions. The model was approximated using *discrete least squares approximation* [3, 15]. This approximation is a minimization problem of the sum of squared errors which is represented in the following form:

$$\min_{M \in R^{nxn}} \sum_{t=0}^{\ell-2} \left\| M\widehat{E}_t - \dot{\widehat{E}}_t \right\|^2 \tag{2}$$

subject to $m_{ij} \geq 0$ for all $i \neq j$ and $m_{ii} \geq \lambda$ for all i. The given value $\lambda \leq 0$ represents the natural decrease in the concentration of mRNA between two time steps. The vector norm $||.||$ can be considered as the *Euclidean distance*. The result of this minimization problem allows the generation of genetic network. (The algorithm of calculating M has been implemented in MATLAB.)

In the mathematical modeling of gene-expression data from microarray experiments, the system of linear differential equations was replaced with a system of difference quotients in order to find m_{ij} [5]. The discrete least squares approximation (or data fitting) of the finitely many difference quotients as described in Eq. (2) are a problem of minimization and represent the right-hand side of Eq. (1):

$$\dot{\widehat{E}}_k := \frac{\Delta \widehat{E}}{\Delta t} = \frac{\widehat{E}_{k+1} - \widehat{E}_k}{\hat{h}_k} = \frac{\widehat{E}_{k+1} - \widehat{E}_k}{\hat{t}_{k+1} - \hat{t}_k}. \tag{3}$$

On the right-hand side of Eq. (3), $\widehat{E}_k$ represents the measurement of expression level of all genes in one microarray experiment at time $\hat{t}_k$ for all $k \in \{0,\ 1,\ \ldots,\ l-1\}$, and l is the number of $\widehat{E}_k$ measurements. On the left-hand side of Eq. (3), $\dot{\widehat{E}}_k$ approximates $\dot{E}$ at a time $\hat{t}_k$. We consider time-step lengths $\hat{h}_k = \hat{t}_{k+1} - \hat{t}_k$ ($k \in \{0,\ 1,\ \ldots,\ l-2\}$). By inserting the finitely many measured expression vectors and difference quotients into our discrete least squares approximation and solving this approximation with respect to M, we get our system

Eq. (1) of differential equations. Its solutions are trajectories over a considered time-interval, or just over all real numbers (complete time axis). Furthermore, the right-hand side of Eq. (1), i.e., ME, also allows the generation of a network. In comparison with [5], we minimized the sum of squared errors, a function in the variable M, to get a best model that describe the gene-expression data from microarray experiment.

2. Genetic Network Analysis

In order to construct a genetic network, we consider a gene-interaction matrix, M, as our foundation. The coefficients of m_{ij} are used for network generation and they represent strong or weak interactions between genes. By solving least squares approximation as explained in Section 2, M was obtained. In [10, 11], a transformation problem into a linear model for *standard Gaussian approximation* is presented, and, an algorithm based on this transformation problem has been implemented in MATLAB to get the best-fitted model. The result of the minimization problem is our matrix M in which the coefficients are used to generate a genetic network.

By definition, a genetic network is represented as a *directed graph* (digraph). A directed graph (or network) $G = (V, A)$ consists of a finite set V of nodes (vertices) and a subset A of arcs of the cross product of $V \times V$ [1]. An arc $a \in A$ is an ordered pair of distinct nodes $(v_1, v_2) \in V \times V$ in the digraph. In a genetic network, the nodes represent genes and the arcs represent the interactions between the genes. An arc $a = (v_1, v_2)$ connects two nodes, v_1 and v_2, and it is attached with a value which represents the influence of the first node v_1 (gene 1) on the second node v_2 (gene 2) in genetic networks.

Generally, two types of information need to be stored when representing a network, i.e., (1) the network topology (node and arc structure of a network), and (2) data associated with the nodes and arcs of a network such as values on the arcs. In this study, the starting point is the matrix M which is an adjacency matrix. Each row and column in this matrix corresponds to one gene in the genetic network. Each numerical value of the matrix entries represents interaction between genes. The matrix entries with zero value corresponds to no interaction between the genes. The smaller the absolute value of the matrix entries, the less is the influence or interaction between genes.

In genetic networks, an implementation of ***Dijkstra's algorithm*** finds the *shortest path*s, the path(s) with the minimum nonnegative value of distance (weight) from one chosen gene to all genes which are connected to that chosen gene along a path in the network [7]. The path is defined as a sequence of arcs or nodes without any repetition of nodes [1]. We note that shortest paths need not uniquely exist in the digraph. In order to use Dijkstra's algorithm to search for the shortest path in the network, the adjacency matrix is transformed into another matrix M'. The transformation of the entries in the adjacency matrix into the entries in the transformed matrix is described as followed. Let us call the entries in matrix $M = (m_{ij})$ and the entries in transformed matrix $M' = (m'_{ij})$, then $m'_{ij} := 1/m_{ij}$, if $m_{ij} \neq 0$, and $m'_{ij} := \infty$, if $m_{ij} = 0$. The smaller the absolute value of the entries of the transformed matrix, the higher is the influence between interacting genes. The transformed matrix M' was used to find the shortest paths in the network via Dijkstra's algorithm.

In this algorithm, we have to choose a start node which corresponds to one gene. Let us, for example, choose gene s as the *start node* in the network. Then, Dijkstra's algorithm will iterate the search for the shortest path from s to all the genes in the network. During the iteration, s is assigned a value of zero. This algorithm will search for the connecting arcs from

s to other genes. Among these arcs, the arc with minimal value attached to it will then be selected. After one neighbouring-gene with a minimal value of the arc is selected, s is marked permanently with distance of zero while other genes in the network are marked temporarily with the chosen upper-bound value. That neighbouring-gene which has been selected will be given a new distance value. The new distance value is the summation of the value of the start-gene and the value of the arc that connects the start-gene to one or more neighbouring-genes. The value of the new distance will be chosen and updated if it is lower than the upper-bound value of that neighbouring-gene. Then, the predecessor list will store the predecessor gene to that neighbouring-gene. The neighbouring-gene is now marked permanently while the remaining genes in the network are still unmarked permanently. At each iteration, the label of a gene i is its shortest distance from s along a path whose internal genes are all permanently labeled. The internal genes are genes different from s or the gene i itself. The iteration process continues the search for the next neighbouring-gene which is marked temporarily with a minimal value of the arc. The algorithm will stop once all reachable genes from s in the network have been permanently marked or searched. The output of finding the shortest path from s to all genes in the network by Dijkstra's algorithm consists of (a) the distance of reachable gene(s) (distance-value), and, (b) the predecessor list for each gene in the network. A pseudocode of Dijkstra's algorithm is given in [1].

Two important features of Dijkstra's algorithm are the starting point or gene and the value of the arc. The starting point determines the shortest path(s) from one fixed node to all existing nodes in the network. Analysis of genetic network by Dijkstra's algorithm can be executed when changing the start-gene. If the start-gene is changed, the search output may be changed as well. Dijkstra's algorithm finds the shortest path(s) from the changed start-gene to all reachable genes connected with it in the same network. In other words, the shortest path(s) connecting the start-gene to all genes in the network changes as well. Thus, different shortest paths would be obtained.

Another analysis of genetic network by Dijkstra's algorithm is to delete one node or gene from the network and look for any existing path(s) in the network. In this case, we have to look at the *connectivity,* or *topology,* of the network. If one gene is deleted from the network, the connectivity of the network has to be checked whether the network exists as one or more connected components. If the network is still connected after deletion of one gene, then the deleted gene does not play a vital role or has little effect in the network. If the network is split into several connected components, the deleted gene may have a greater effect in the network. In order to use Dijkstra's algorithm in the network with one deleted gene, the row and column corresponding to the deleted gene have to be deleted from transformed matrix.

3. Discussion

The inference of genetic networks using gene-expression data has been performed using many mathematical formalisms such as Boolean, Bayesian, linear equations, differential equations and neural network. These inferences of genetic network provide an insight into the underlying gene interactions and gene regulation of metabolic pathways in the living organism. It also shows how manipulation of data can be done using various approaches. These data are accumulated from the high-throughput microarray technique. Hence, sophisticated computational tools can be developed to facilitate rapid and comprehensive analysis of large amounts of biological data.

Gene-expression data is rich of information that could be used to discover the higher-order structures of an organism and even interpret its behavior. Many computational tools have been developed to extract useful and functional information from the gene-expression data. In this study, a model based on a system of linear differential equations has been used to analyze gene-expression data. The building of genetic network via this model has been represented by a directed graph consisting of nodes connected by arcs. The nodes correspond to the genes in the network while the arcs correspond to the gene-gene interactions.

It is known that often gene regulatory networks are sparse networks in which most coefficients are expected to be zero [5]. With the many zeros in the network, we can reduce a bigger network into smaller networks, called *subnetworks.* We propose to investigate a given real and big network by firstly analyzing a number of smaller networks contained in the big network. These smaller networks are found by discrete least squares approximation, reveal a high connectedness. In many cases, subnetworks actually are considered as subnetworks of the big network. Often subnetworks actually represent *connected components* in the big network (or connected components up to the deletion of a few arcs in that given network). Due to the connectedness, Dijkstra's algorithm can be applied to these subnetworks.

The value of the arc plays a role in determining the type of the network, i.e., *static* and *dynamical.* If the value of the arc is fixed, the produced network is static. A dynamical network is only obtained when the value of the arc is not fixed. In this study, a static network is generated due to the consistency of the matrix M which is independent of E. Thus, the value of the arc in the generated network is fixed and it represents the influence of one gene towards the other adjacent genes. For dynamical networks, where the data on the arcs change in time, a dynamic version of Dijkstra's algorithm is helpful. Such a version is, e.g., used in analogous problems of traffic regulation and implemented at the Center for Applied Computer Science (ZAIK), University of Cologne. The utilization of dynamic algorithms in our field of bioinformatics is left for future research.

In this study, temporal pattern of gene-expression data has been used to infer a genetic network. These gene-expression data are obtained from microarray experiments of yeast *Saccharomyces cerevisiae* which are involved in the diauxic shift from anaerobic to aerobic metabolism upon depletion of glucose [6]. During this diauxic shift process from fermentation to respiration, the cell states of yeast were correlated to the changes in the expression levels of many genes in fundamental cellular processes such as protein synthesis, carbohydrate metabolism and glycogen metabolism. We infer a genetic network using microarray data from yeast [6]. The inference of genetic networks involves modeling of expression-data using a system of linear differential equations and the transformed matrix is computed in MATLAB.

The generated network consists of eight nodes which represent the genes involved in the glycogen metabolism pathway as described in Fig. 1. These genes encode for enzyme trehalase (*NTH2*), trehalase-6-phosphate synthase (*TPS3*), UTP-glucose-1-phosphate uridylyltransferase (*UGP1*), UDP-glucose-starch glucosyltransferase (*GSY2*), 1,4-glucan branching enzyme (*GLC3*), glycogen phosphorylase (*GPH1*), phosphoglucomutase (*PGM1*), and hexokinase1 (HXK1). The generated network has then been analyzed by finding the shortest path using Dijkstra's algorithm, which has been written in Python [19]. The analysis of the generated network shows that the network is *connected* but not *complete.* The generated network consists of one connected component. A connected network is defined as a network where for every pair of nodes in the network there is a connecting path in the corresponding undirected graph. In a complete network for each pair of nodes there is a connecting arc.

Further analysis of the generated network with Dijkstra's algorithm shows that the node which corresponds to that gene which encodes enzyme glycogen phosphorylase, *GPH1*, plays an

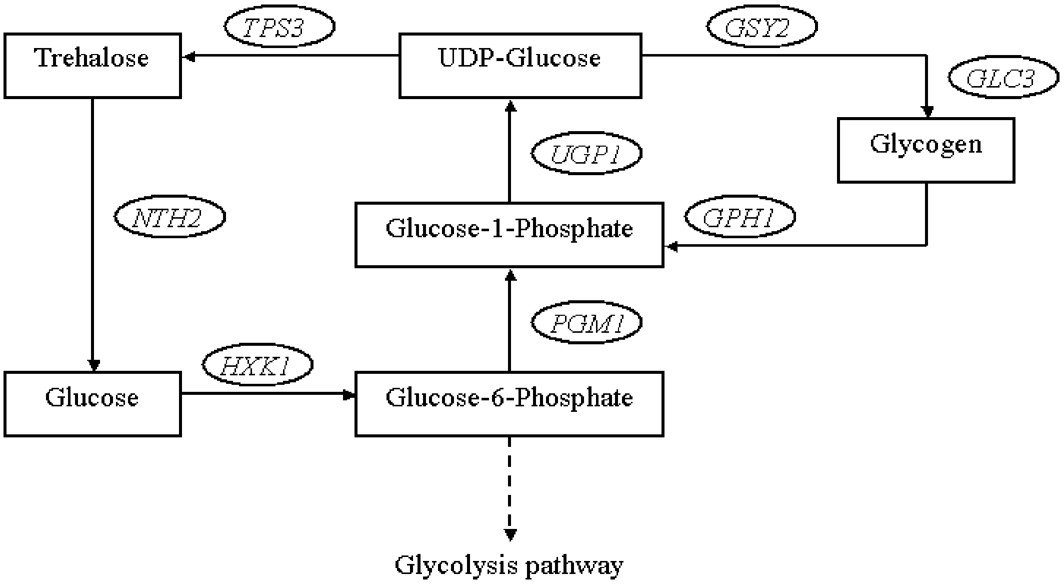

Fig. 1. Glycogen metabolism pathway in yeast *Saccharomyces cerevisiae.* The square box (□) represents the substrates (or products) involved in glycogen metabolism and circle box (○) represents genes involved in glycogen metabolism. (*Abbreviation*: *NTH2*, trehalase; *TPS3*, trehalase-6-phosphate synthase; *UGP1*, UTP-glucose-1-phosphate uridylyltransferase; *GSY2*, UDP-glucose-starch glucosyltransferase; *GLC3*, 1,4-glucan branching enzyme; *GPH1*, glycogen phosphorylase; *PGM1*, phosphoglucomutase; *HXK1*, hexokinase 1.)

important role in the glycogen metabolism pathway. Deletion of this gene causes the generated network to be disconnected and split into several connected components. Moreover, there is no path anymore that connects all the genes in the generated network when this gene is deleted. The deleted gene encodes enzyme *GPH1* which is involved in the glycogen catabolism or the breakdown of glycogen to produce glucose-1-phosphate (G1P).

The intermediate, G1P, is an important substrate for glycogen synthesis and also for the glycolysis pathway. The information deduced from the network without this gene could be used for drug targeting and also for gene therapy for those who have diseases associated with glycogen metabolism. This small network illustrates that we are able to deduce useful information from the analysis of the genetic network using Dijkstra's algorithm. We hope that Dijkstra's algorithm will be able to draw out useful information on system level from bigger networks. The glycogen metabolism pathway is just one small example of using Dijkstra's algorithm to infer information on cellular level.

The advantages of using a differential equations model for inference of a genetic network are lying in quantitativeness, causality and computational tractablility. Interactions between genes can be reflected quantitatively in terms of this system. The set of differential equations describes gene interactions in an explicitly numerical or time-discretized way. It also describes the causality relationship between genes where a coefficient of the matrix, m_{ij}, is corresponding to the effect of gene j on gene i. The inferred network can be linked to other simulation or visualisation tools, e.g., *Graph Animation Toolbox* (*GATO*) [21], and *Genomic Object Net* [18]. Thus, it is computationally tractable.

Information derived from the genetic network analysis using Dijkstra's algorithm provides an insight about the underlying interactions between genes. Dijkstra's algorithm is able to show the maximum interactions of genes in the genetic network and select groups or sets of key genes that may play an important role in certain metabolic pathways based on the shortest path. The output of the network analysis with Dijkstra's algorithm provides an idea about how influential the interacting genes are and also shows the preferred path(s) from the start-gene to all the reachable genes in the genetic network. These genes are correlated with proteins or enzymes in the metabolic pathway. We hope that the inference of genetic networks by Dijkstra's algorithm may lead to some information about the correlation between transcriptome and proteome or metabolome.

The smaller values of the output of Dijkstra's algorithm, i.e., shortest paths indicate higher interaction between genes in the network. The outputs of Dijkstra's algorithm, multiply applied, also deduces whether there may be only one or more existing paths with the same minimum distance that connect two specific genes. The highly influential genes are believed to have more control in the preferred pathways, for instance, in the glycogen metabolism pathway. Therefore, we could have an idea about which gene(s) plays an important role in that metabolism pathway based on Dijkstra's algorithm.

Further analysis of the genetic network with Dijkstra's algorithm such as deleting one gene from the generated network provides an insight about which specific gene among the highly influential genes in the selected pathway has more control for genetic manipulation or is a target for drug discovery. This deduction is based on the topology of the network once one or several genes are deleted from the generated network. If there is no connectedness among the highly influential genes, then this deleted gene may have more control than the other highly influential genes in the generated network.

The main reason for understanding or unlocking the genetic network via a path-finding algorithm, e.g., here Dijkstra's algorithm, is to provide the opportunity for pathway manipulation. Each gene or a group of genes is involved in a certain metabolism. By looking at the interaction of genes that are involved in a certain metabolic pathway, we can manipulate the pathway to overproduce needed compounds or products or use it for targeting critical steps in the metabolism pathway for drug discovery.

Conclusion

Dijkstra's algorithm is helpful for inferring the underlying interaction of genes in order to gain an insight about the pathway structure in a metabolism based on the maximum interaction of genes. Therefore, a preferred route for genetic manipulation and drug target can be generated. This approach will hopefully have a significant use in the data mining of microarray data in order to gain useful information.

At Middle East Technical University (Ankara), the mathematical modeling given by Eq. (1) is becoming generalized by admitting discontinuities in the trajectories and in the right-hand side (together with Marat Akhmet). Herewith, we want to take account of the mixed continuous-discontinuous character of nature. In future, we want to apply further mathematical modeling methods coming from statistical learning ([12]; with Hakan Öktem), represent the mutual influence between genes not only by coefficients m_{ij} but by intervals (in terms of fuzzy logics), and apply Dijkstra's algorithm in its dynamical version.

References

[1] Ahuja R.K., Magnanti T.L., Orlin J.B. Network Flow: Theory, Algorithms, and Applications. Englewood Cliffs, New Jersey, USA. Prentice Hall, 1993. P. 23–132.

[2] Alwine J.C., Kemp D.J., Stark G.R. Method for detection of specific RNAs in agarose gels by transfer to diazobenzyloxymethyl-paper and hybridization with DNA probes // Proc. Natl. Acad. Sci. USA. 1977. Vol. 74. P. 5350–5354.

[3] Bevington P.R. Data Reduction and Error Analysis for the Physical Sciences. N.Y.: McGraw-Hill, 1969.

[4] Chen T., He H., Church G. Modeling gene expression with differential equations // Pac. Symp. Biocomput. 1999. Vol. 4. P. 29–40.

[5] De Hoon M., Imoto S., Miyano S. Inferring gene regulatory networks from time-ordered gene expression data using differential equations // Discovery Sci. 2002. P. 267–274.

[6] De Risi J., Iyer V., Brown P. Exploring the metabolic and genetic control of gene expression on a genomic scale // Sci. 1997. Vol. 278. P. 680–686.

[7] Dijkstra E.W. A note on two problems in connection with graphs // Numerische Math. 1959. Vol. 1. P. 269–271.

[8] Duggan D.J., Bittner M., Chen Y., Meltzer P. et al. Expression profiling using cDNA microarrays // Nature Genet. 1999. Vol. 21. P. 10–14.

[9] Friedman N., Linial M., Nachman I., Pe'er D. Using Bayesian networks to analyze expression data // J. Comp. Biol. 2000. Vol. 7. P. 601–620.

[10] Gebert J., Laetsch M., Pickl S.W., Weber G.-W. et al. An algorithm to analyze stability of gene-expression patterns. Preprint. ZAIK, Univ. of Cologne, Submitted for Publication in Annals of Operations Research. 2002.

[11] Gebert J., Laetsch M., Pickl S.W., Weber G.-W. et al. Gene networks and anticipation of gene expression patterns // Proc. of Conf. on Anticipatory Systems, Liege, Belgium, Aug. 2003.

[12] Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. Data Mining, Inferences, and Prediction. Springer Series in Statistics, Springer. 2001.

[13] Huang S. Gene expression profiling, genetic networks and cellular states: an integrating concept for tumorigenesis and drug discovery // J. Mol. Med. 1999. Vol. 77. P. 469–480.

[14] Ideker T.E., Thorsson V., Karp R.M. Discovery of regulatory interaction through perturbation: inference and experimental design // Pac. Symp. Biocomput. 2000. Vol. 5. P. 302–313.

[15] Isaacson E., Keller H.B. Analysis of Numerical Methods. N.Y.: John Wiley and Sons, 1966.

[16] Liang P., Pardee A.B. Differential display of eukaryotic messenger RNA by means of the polymerase chain reaction // Sci. 1992. Vol. 257. P. 967–971.

[17] Lockhart D.J., Dong H., Byrne M.C., Follettie M.T. et al. Expression monitoring by hybridization to high-density oligonucleotide arrays // Nature Biotechnol. 1996. Vol. 14. P. 1675–1680.

[18] Matsuno H., Doi A., Hirata Y., Miyano S. XML documentation of biopathways and their simulation in Genomic Object Net // Genome Informatics. 2001. Vol. 12. P. 54–62.

[19] Quek E.M.P. Mathematical Modelling and Implementation of a Path-Finding Algorithm for a Problem in Genetic Networks. Postgraduate Dissertation Report. CUBIC, University of Cologne, Germany, 2003.

[20] Schena M., Shalon D., Davis R.W., Brown P.O. Quantitative monitoring of gene expression patterns with a complementary DNA microarray // Sci. 1995. Vol. 270. P. 467–470.

[21] Schliep A., Hochstaettler W. Developing Gato and CATBox with Python: teaching graph algoritms through visualization and experimentation // Proc. of Multimedia Tools for Communicating Mathematics (MTCM), 23–25 Nov., 2000. Lisbon, Portugal.

[22] Southern E., Mir K., Schepinow M. Molecular interactions on microarrays // Nature Genet. 1999. Vol. 21. P. 5–9.

[23] Van Someren E.P., Wessels L.F.A., Reinders M.J.T. Linear modeling of genetic networks from experimental data // Proc. of the Conf. on Intelligent Systems for Molecular Biology (ISMBOO), La Jolla, CA. AAAI Press, Menlo Park, CA. 2000. P. 355–366.

[24] Velculescu V.E., Zhang L., Vogelstein B., Kinzler K.W. Serial analysis of gene expression // Sci. 1995. Vol. 270. P. 484–487.

[25] Wagner A. Estimating coarse gene network structure from large-scale gene perturbation data // Genome Res. 2002. Vol. 12. P. 309–315.

[26] Weaver D.C., Workman C.T., Stormo G.D. Modeling regulatory networks with weight matrices // Pac. Symp. Biocomput. 1999. Vol. 4. P. 112–123.

[27] Yilmaz Ö. A Stage-Mathematical Algorithm for Predicting Secondary Structure of Proteins // MS Thesis, Graduate School of Engineering. Koç Univ., Istanbul, 2003.

Received for publication February 25, 2004