HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Engineering Physics and Mathematics
Mat-2.108 Independent research project in applied mathematics

cDNA Microarray Data Analysis Methods: A Review

Petri Kärkäs

57576H

Espoo, November 21, 2006

Contents

1 Introduction

Recently, the technological advances in the field of genomics have increased the amount of biological data available for analysis. In particular, the development of cDNA (complementary deoxyribonucleic acid) microarray technology allows simultaneous recording of thousands of gene expression levels. Gene expression level is a quantitative measure of the amount of mRNA (messenger ribonucleic acid) associated to that gene in a sample. The resulting data can be used to study simple organisms' genome as whole [1, 2], while in past the studies have concentrated on investigating individual genes and their function. With cDNA microarray technology, tens of thousands of human genes have been studied simultaneously [3]. In addition to studying the general large-scale physiological state of the organism at the genetic level, the microarray data can be used to investigate the interaction of individual genes, and also the effect of different conditions, e.g., disease states, on the gene expression. The use of microarray analysis is therefore directly applicable to biomedical research, e.g., cancer research [4].

The availability of microarray data has created a need for database and analysis tools and the computational methods in microarray data analysis are in rapid and continuous evolution. While studies have been published with a plethora of different methods, this work concentrates on giving a review on the most used analysis methods and their advantages and limitations. The work concentrates on clustering algorithms that are used to study the co-expression of genes and the methods to model gene regulatory networks (GRNs) [5], which map the basic functions and interactions between the genes.

The aim of this work is to give a reader a basic knowledge on the analysis of cDNA microarray data. Section 2 reviews the biological principles of genetics, the principles of cDNA microarray technology, and, most importantly, the data acquisition procedure with a microarray. The biological principles are discussed only shortly to allow an emphasis on the methodology. Readers having a more in depth interest in genomics are encouraged to read on cell biology (see, e.g., [6]). Section 3 discusses the different methods that have been widely used to analyze microarray data. Finally, Section 4 summarizes the review with some conclusions.

2 Biological Background and cDNA Microarray Technology

2.1 Biological Background

The cellular processes are controlled by a regulatory system whose instruction are coded to DNA situated in the chromosomes in the nucleus of a cell. Human DNA consists of four different nucleotides that bind pairwise by hydrogen bonds (Figure 1). The DNA information is the same in all the cells of a living organism with only few exceptions. Most importantly, the information in DNA controls the protein manufacturing. A gene is a sequence of the DNA that controls discrete hereditary characteristics, usually corresponding to a protein or ribonucleic acid (RNA), and occupies a specific place in a chromosome. [6]

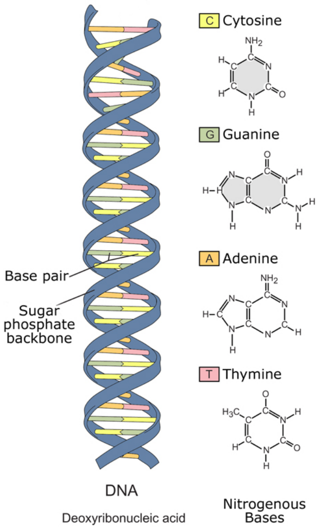


Figure 1: *DNA is constructed of sequences of four different nucleotides and a sugar phospate backbone. Adapted from* [7].

Proteins are produced by encoding DNA into messenger RNA (mRNA) which is transported to cell cytoplasm and its information is utilized in protein synthesis in the ribosomes (Figure 2). A single gene can encode several proteins. The expression level of a gene is a quantitative measure proportional to the amount of mRNA of a particular gene in the cytoplasm. The expression levels of thousands of genes can be measured with microarray technology as described in Section 2.2. [6]

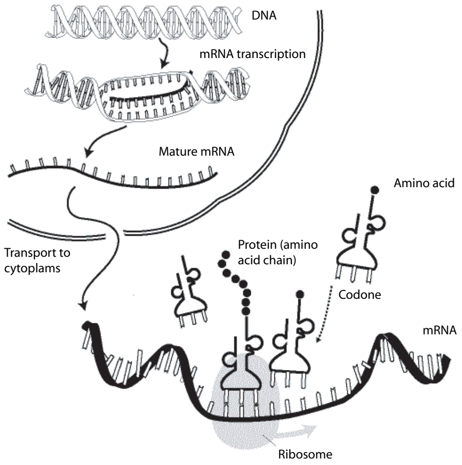


Figure 2: *mRNA is transcripted from DNA and utilized in protein synthesis. Modified from* [4].

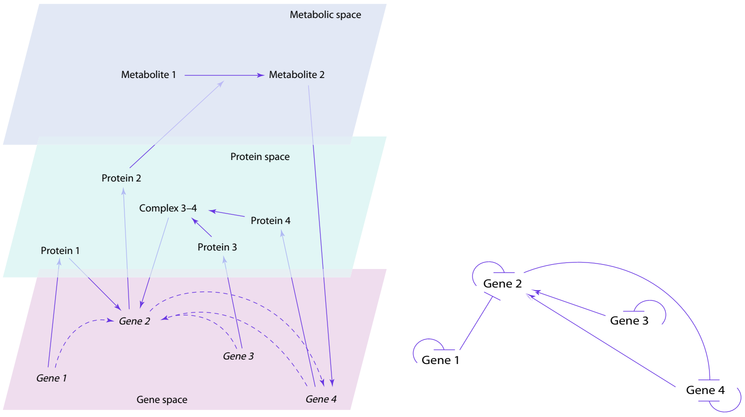


Figure 3: *Left: a hypothetical example of a biochemical network. The nodes of the network are organized in three levels: mRNA space, protein space, and metabolite space. Solid arrows indicate the interactions while the GRN obtained by observing the gene expression levels is presented on the right hand side. Modified from* [8]

Genes performing similar tasks have correlating expression levels because of complex biochemical interactions. In these biochemical networks the genes do not interact directly with each other, but the interactions are present between different proteins and metabolites. The metabolites and proteins can interact with each other or directly with the gene expression levels. Therefore, it is possible to observe the interaction of the genes, i.e., the GRNs, by measuring gene expression levels while the interactions seen are modulated through protein pathways (Figure 3).

2.2 cDNA Microarrays

Today, the most frequently utilized method for measuring the gene expression levels is the cDNA microarray [9]. In this work, the term microarray is adapted to refer to cDNA microarray, although also other forms of microarrays exist, such as oligonucleotide microarrays [10]. This review concentrates solely on the cDNA microarray technology, since it is the most abundant method and does not differ significantly from other array methods. In cDNA microarray technology, double stranded cDNAs are planted on a glass slide. The planted cDNAs act as probes for the genes whose expressions, i.e., the amount of mRNA in the sample, are to be studied. Therefore, the first problem in the microarray design is to decide which set of genes is to be used in the array. Only with some very simple organisms, the array can be prepared to include the whole genome.

In its simplest form cDNA microarray technology is used to measure the gene expression levels of two conditions, e.g., from diseased tissue and healthy tissue. mRNA samples from the conditions are extracted after which the samples are transcribed to cDNA and labeled with two fluorescent dyes. The labeled cDNA samples are then mixed and hybridized on the microarray. During the hybridization the cDNA of the sample bind to their counterparts on the array. The array is then scanned in to an image showing various intensities of the colors of the fluorescent dyes (Figure 4). The intensities correspond to the gene expression levels and are extracted from the figure with an image analysis software. It is common practice that the studied condition is labeled with a red dye and the reference condition with a green dye. Therefore, the upregulation of a gene is shown with a red spot, downregulation with a green spot, and equal expressions are denoted with a yellow spot. The analyzed conditions can range from different time points of any biological process to distinct drug treatments, and a study usually consist of more than two conditions. [11]

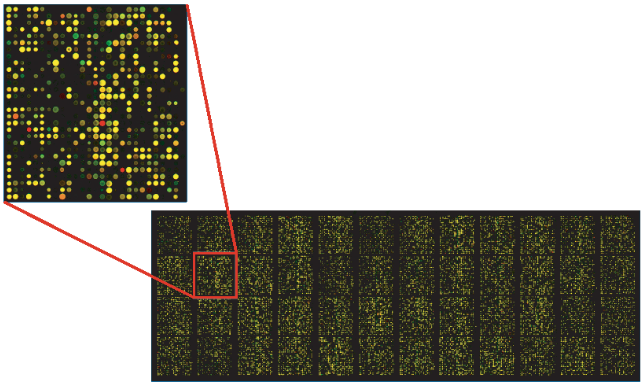


Figure 4: *An example of the image produced by scanning a cDNA microarray. Modified from* [12].

The data extraction from the scanned images involves three phases: gridding, i.e., coordinate assignment for each spot, segmentation, i.e., classification of the pixels as background or foreground image, and intensity extraction. Various factors such as distinct efficiencies in the labeling process and the differences in the initial amount of mRNA in the samples cause biases in the results. The intensity values must be rescaled accordingly before proceeding with the analysis. This process, referred to as normalization, is based on some reference point in the measurement. A typical reference would be the use of a gene whose expression is known not to change between the conditions, although also other forms of references are used [11]. The normalized data is usually log-transformed, because the upregulation and downregulation are reflected in a symmetrical scale after this transform. Finally, the data on the expression levels that do not change between the studied conditions are omitted. [11]

The vast amount of information produced by the microarray studies are saved to data storages. Some examples of popular public human DNA repository databases are ArrayExpress [13] of the European Bioinformatics Institute and ArrayDB [14] of the National Human Genome Research Institute.

3 Methods for Analyzing the Microarray Data

3.1 Formulation of the Study Questions

The study questions in microarray experiments are usually distinguished into questions comparing two conditions, typically one reference condition and one condition of interest, and questions comparing many conditions, e.g., time behavior of expression levels during a biological process. In two condition studies the analysis aims to investigate which expression levels change significantly between the conditions. Multi-condition studies aim to produce answers for more complex questions such as studying the interactions of genes by producing GRNs (see, e.g., [15]).

The data of a microarray study is presented with a gene expression matrix (GEM), in which the cells show the normalized expression value, the genes correspond to rows, and the conditions correspond to columns:

$$\mathbf{M} = \begin{pmatrix} M_{11} & \cdots & M_{1k} \\ \vdots & \ddots & \vdots \\ M_{n1} & \cdots & M_{nk} \end{pmatrix}, \tag{1}$$

where the number of conditions is k and the number of genes studied is n; usually $n >> k$. If there are only a few genes under study, the expression level matrix can be visualized conveniently as shown in Figure 5. The hypothetical data presented in the figure shows a clear correlation between expression levels of the genes 3 and 4.

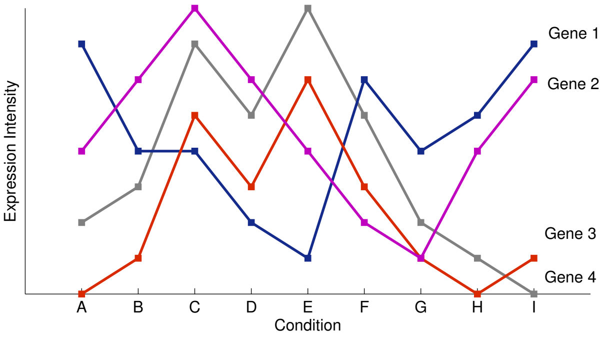


Figure 5: *A hypothetical gene expression data with four genes and multiple conditions*

The comparison of two conditions is a simple task accomplished by statistical tests such as t-test or Mann-Whitney test; however, some articles study only the absolute changes as pointed out by Dopazo [11]. The comparison of two conditions is not mathematically interesting and will not be discussed further. The first step in the multi-condition data analysis is to find sets of genes that have similar expression profiles or sets of conditions that have similar expression values. Different clustering algorithms are used to accomplish this first step (Section 3.2). The next steps of multi-condition data depend on the study question, but one interesting and general task is to study the interactions between genes by modeling the GRN. The modeling of GRNs is discussed in Section 3.3.

3.2 Clustering Algorithms

The clustering algorithms can be used to classify either the rows or columns of the GEM. In most studies the number of genes is excessively larger than the number of conditions, which makes the clustering of rows and columns somewhat different tasks. The clustering algorithm has to be chosen in respect to the dimensions of the data.

The clustering algorithms group similar data to a same group and do not require any information on the classes of the data; therefore, the clustering methods are sometimes referred to as unsupervised classification methods. The clustering algorithms can be divided into hierarchical and partitional, with the difference that the hierarchical algorithms do not need any *a priori* information. The level of separation, and therefore the number of clusters, can be decided after the hierarchical clustering results. [16]

All clustering algorithms are based on a given distance function $d(i, j)$ that tells the distance of the vectors i and j. Most widely used distance measures in microarray data analysis are the Euclidean distance and Pearson's correlation coefficient. It should be noted that Euclidean distance should only be used after removing the mean of the data and dividing the data with the variance. In the case of internal correlation in the data, a distance measure that takes the covariance of the data into account should be used. One such measure is the Mahalanobis distance, defined for vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ as

$$d_M(i, j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \Sigma^{-1} (\mathbf{x}_i - \mathbf{x}_j)}, \tag{2}$$

where Σ is the covariance matrix calculated over all the vectors $\mathbf{x}$. Mahalanobis distance can be viewed as a normalized Euclidean distance.

The choice of a distance function is dependent on the data. Pearson's correlation coefficient is suitable for most cases and can also detect the negatively correlated genes. The use of the Mahalanobis distance is preferred to the basic Euclidean distance since the biological data almost always includes a high degree of internal correlation. [11]

3.2.1 Hierarchical Clustering

Hierarchical clustering is the most usual clustering method in microarray data analysis, mainly because it allows to study higher order relationships between the expression profile clusters than the partitional clustering methods. This is because partitional clustering methods divides the profiles to a predefined number of clusters without any information on other possibilities.

At the start of agglomerative, i.e., bottom-up, hierarchical clustering each data vector forms its own cluster. The clusters are then merged iteratively based on the distance measure $d(i,j)$ and the distance measure between the clusters $D(\mathcal{S},\mathcal{R})$, where $\mathcal{S}$ and $\mathcal{R}$ are sets comprised of the studied vectors. The distance between the clusters is usually determined by the average linkage, i.e.,

$$D(\mathcal{S},\mathcal{R}) = \frac{\sum_{i,j} d(i,j)}{n_R n_S}, \quad i \in \mathcal{S}, j \in \mathcal{R}, \tag{3}$$

where n_R and n_S are the number of vectors in clusters $\mathcal{R}$ and $\mathcal{S}$, respectively. However, also single linkage (Eq. 4) and complete linkage (Eq. 5) methods can be used

$$D(\mathcal{S},\mathcal{R}) = \min_{i,j} d(i,j), \quad i \in \mathcal{S}, j \in \mathcal{R}, \tag{4}$$

$$D(\mathcal{S},\mathcal{R}) = \max_{i,j} d(i,j), \quad i \in \mathcal{S}, j \in \mathcal{R}. \tag{5}$$

Agglomerative hierarchical clustering produces a clustering tree, i.e., a dendrogram, by changing the threshold for $D(\mathcal{S},\mathcal{R})$ that is used to merge clusters (Figure 6).

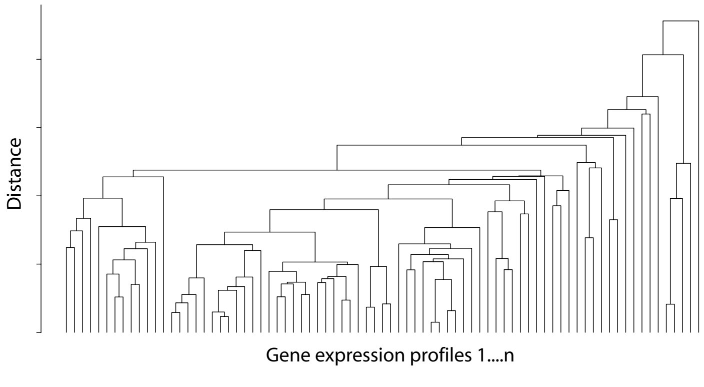


Figure 6: *A hypothetical example of an agglomerative dendrogram for gene expression data. The dendrogram shows the agglomeration at each distance level. For clustering, the user has to choose one cut-off distance value.*

From the dendrogram, the user can choose at which distance level he wants to use to form the final clusters. In divisive, i.e., top down, hierarchical clustering the procedure is similar, but at the start all the vectors are clustered in one cluster and the distance level to form clusters is reduced to produce cluster divisions. [17]

3.2.2 Partitional Clustering

Most typical partitional clustering methods used in microarray analysis include quality threshold clustering (QTC) [18], k-means clustering [19], and self organizing map (SOM) [20]. Standard partitional clustering methods such as QTC and k-means, which work by iteratively minimizing the overall within-cluster variation, have been criticized mainly because the need of prespecified number of clusters or other information and long computational times which can range from N^2 to N^4. SOM offers a faster way of partitional clustering but there exists contrary results on its accuracy (see, e.g., [4, 21]).

k-means algorithm [19] is based on calculating cluster centroids and assigning each vector to a cluster based on the distance from the centroid. The optimal centroids are searched iteratively. If the vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ are to be clustered into k clusters, an initial guess on the cluster centroids $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_k$ is done. The initial guess can be random or based on a heuristic. After this, each vector is assigned to a group whose centroid is the closest one in the sense of the distance function $d(i, j)$. After assigning, each centroid value is recalculated as

$$\mathbf{m}_i = \frac{1}{N_i} \sum_j \mathbf{x}_j, \quad j \in \mathcal{S}_i, \tag{6}$$

where $\mathcal{S}_i$ is the set of vectors assigned around group centroid $\mathbf{m}_i$ and N_i is the number of vectors in $\mathcal{S}_i$. The assigning and group centroid updating is iterative and stops after there is no change in the classification of the sample vectors. k-means clustering is one of the simplest clustering methods and is both computer intensive and inflexible.

QTC [18] is a method especially designed for clustering gene expression data. In QTC the user chooses the maximum diameter for a cluster, which is the only piece of *a priori* information needed. For each vector the closest vectors inside the maximum diameter are calculated based on the distance function $d(i, j)$ and saved as candidate clusters. Thus, there is the same number of candidate clusters as there are original vectors. Next, the candidate cluster with most members is selected as a cluster for the final clustering and removed from further calculations. The process in repeated as long as there is no vectors in the candidate clusters or, alternatively, the user can specify a threshold for the minimum number of vectors in one cluster. In the latter alternative, some vectors are left unclassified. QTC is a very computer intensive technique, although the advantages of no need of *a priori* information on the amount of clusters and the possibility of unclassified vectors are considered to be greater than the disadvantages of the long computational time.

SOMs have been used in a variety of applications including gene expression clustering (see, e.g., [4]). SOM has found out to be computationally lighter ($\sim N$) than other partitional clustering methods and has also been noted to possess better visualization capabilities [4, 11]. SOM transforms multi-dimensional data to a map in one, two, or three dimensions. In gene expression clustering, one and two dimensional maps are used. For the description of the algorithm properties, a two dimensional map is considered here.

In SOM there exists a set of input samples $\mathbf{x}_i$, e.g., gene expression data, and an array of nodes in which each node k has its model vector $\mathbf{m}_k$ which is of same dimension as the input samples. The initial model vectors $\mathbf{m}_k$ can even be selected at random; however, it is more profitable to initialize the model vectors based on a heuristic, e.g., spanned by the two principal eigenvectors of the input data [20]. SOM is organized, i.e., the model vectors are updated, based on the following iteration. An input vector $\mathbf{x}_i$ is compared with all the model vectors $\mathbf{m}_k$ of the map. The model vector that minimizes the distance $d(i, j)$ is selected as the best matching unit (BMU) (Figure 7). BMU and a number of neighboring nodes are updated towards the input vector with the rules

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t)(\mathbf{x}_i - \mathbf{m}_i(t)), \quad i \in N_c(t), \tag{7}$$

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t), \quad i \notin N_c(t), \tag{8}$$

where t is the discrete time index, i.e., the amount of input vectors processed, $\alpha(t) \in [1, 0]$ is a factor that defines the size of the learning step, and $N_c(t)$ is the neighborhood function that defines which nodes around BMU are to be updated. An usual choice for the neighborhood function is a two dimensional gaussian surface. It is important to note that the neighborhood function and learning step are decreasing functions of time t to ensure the convergence of the algorithm. [20]

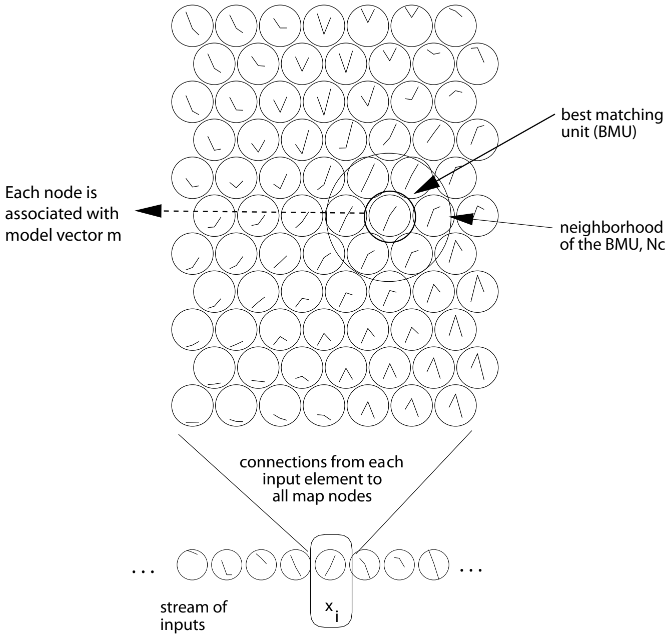

best matching unit (BMU)
Each node is associated with model vector m
neighborhood of the BMU, Nc
connections from each input element to all map nodes
...
...
stream of inputs
x_i

Figure 7: *The architecture of SOM. Each map node is visualized with a circle and the model vector of a node* $\mathbf{m}$ *is visualized with a three element plot. BMU is searched for each input vector and the map node vectors are updated accordingly. Modified from* [20].

The iteration is repeated for all the input vectors, and sometimes same input vectors are even used multiple times in producing the final map. At the end, each input vector is classified to the node in the map that is the closest in the sense of distance $d(i, j)$. The result of SOM is a map where similar input vectors are located near each other and dissimilar vectors at the far ends of the map. The classification of the original input vectors can be done by separating areas of the map. [20]

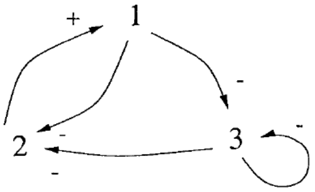
3.3 Gene Regulatory Network Models

Many different methods have been utilized to model GRNs. In a network inference, the aim is to construct a model of the interactions between the genes. The problem is a typical reverse engineering problem: the output and the behavior of the system is known and the properties of the system must be found out based on this information. Gaining an understanding of the positive and negative feedback loops that play a crucial role in genetics would be useful both academically and industrially. The various approaches to model and construct the networks include differential equation models, linear models, graph theory, Bayesian networks, and Boolean networks [5, 8]. Each of these modeling formalities requires different levels of abstractions.

This work discusses the use of directed graphs, Bayesian networks, and Boolean networks in modeling gene regulation. Reviews of other modalities have been presented by, e.g., de Jong [22] and van Sommeren et al. [23].

3.3.1 Directed Graphs

A directed graph G is a simple construct of a set of nodes V and a set of edges E. In gene regulation, each gene is modeled with a node $i \in V$ and the edges between the genes model the interactions, either negative or positive, corresponding to up regulation and down regulation, respectively. Figure 8 presents a simple example of a directed graph. There exists various generalizations of the directed graph theory to better suit the needs of genetic network modeling [22]; this issue is not discussed in this work.



$$V = \{1, 2, 3\}$$

$$E = \{\langle 2, 1, -\rangle, \langle 3, 1, -\rangle, \langle 1, 2, +\rangle, \langle 2, 3, -\rangle, \langle 3, 3, -\rangle\}$$

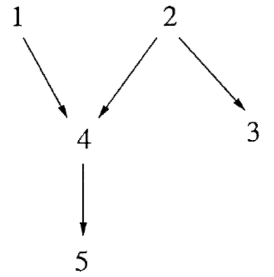
Figure 8: *A hypothetical example of a directed graph modeling a gene regulatory network. Each edge is directed and marked with notation $\langle i, j, s\rangle$ where i, j, and s are the start node, the end node, and the sign of regulation, respectively.* [22]

Directed graphs are usually constructed based on the data obtained with clustering algorithms which were discussed previously. A directed graph of a GRN can be used to search for paths between genes to reveal missing or strong regulatory interactions. Maybe the most important feature of a directed graph as a GRN model is its ability to offer information on the global connectivity of the network. This offers information on the complexity of the network and also makes it possible to study evolutionary changes in the global connectivity. [22]

3.3.2 Bayesian Networks

Bayesian network is a directed graph $G = \langle V, E\rangle$ with the nodes i, $1 \leq i \leq n$, each having a random variable X_i. In a Bayesian GRN model, node i is a gene and X_i is its expression level. In a Bayesian network, each X_i is associated with a conditional probability

$p(X_i|X_j, j \in \text{parents})$ where parents denotes the genes having a direct regulation on i. Figure 9 presents a simple example of a Bayesian network and the associated probability distributions.



$$p(X_1),\ p(X_2),\ p(X_4 \mid X_1, X_2)$$
$$p(X_5 \mid X_4),\ p(X_3 \mid X_2)$$

$$p(\boldsymbol{X}) = p(X_5 \mid X_4)\, p(X_4 \mid X_1, X_2)\, p(X_3 \mid X_2)\, p(X_2)\, p(X_1)$$

Figure 9: *A hypothetical example of a Bayesian network and the associated probability distributions.* [24]

A Bayesian network assumes that the probability of X_i is only dependable on the parents' expression levels X_j and on none other. This is also known as the Markov assumption. Under this assumption, the joint probability distribution $p(\mathbf{X})$ is defined as

$$p(\mathbf{X}) = \prod_{i=1}^{n} p(X_i|X_j, j \in \text{parents}), \tag{9}$$

which is the probability distribution for all of the genes in the model. [24]

Constructing a Bayesian network from the gene expression data involves an optimization scheme to produce a network that best matches the given data (see Heckerman et al. [25] for details). This optimization method is not guaranteed to find the global optimum. Additionally, the gene expression data available today contains information of thousands of genes under only a few conditions and therefore does not determine the network fully. [24]

Bayesian networks are suited for GRN modeling because they are statistically well defined and robust. Furthermore, the Bayesian network models the physical errors in the measurements, e.g., noise, in a natural way and can even be used with incomplete data or data with missing values. However, Bayesian networks lack information on the dynamic aspect of the gene regulation. Some generalizations to apply dynamics on the Bayesian network have been introduced. [22]

3.3.3 Boolean Networks

Boolean networks are the simplest models that express the dynamic modulation state of the genes. The states are expressed with a binary variables 1 (on) and 0 (off). The interactions in the model are represented by Boolean functions that use simple Boolean logic to calculate the state of the genes at time $t + 1$ based on the Boolean functions and the states at time t.

In a network of n nodes the states of the nodes are presented by the vector $\mathbf{x}$ with x_i showing the state of node i. Therefore, the state space of the system consists of 2^n states. The dynamical dependence of the model is regulated by the Boolean functions $\mathbf{B}$ including a function B_i for each node. Hence, the states at the time step $t + 1$ can be calculated with

$$x_i(t + 1) = B_i(\mathbf{x}(t)), \quad 1 \leq i \leq n. \tag{10}$$

The resulting state sequences form the trajectories of the system, thus allowing the study of the dynamic properties and attractors of the system. A simple example of the Boolean network and its Boolean functions is presented in Figure 10.

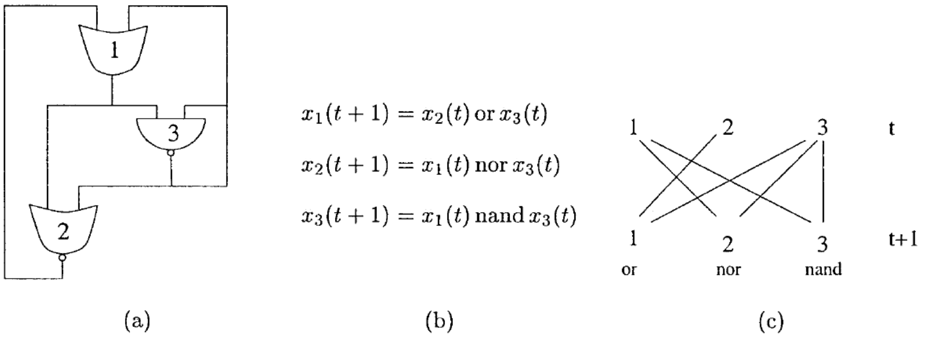

1
3
2
$x_1(t+1) = x_2(t) \,\mathrm{or}\, x_3(t)$
$x_2(t+1) = x_1(t) \,\mathrm{nor}\, x_3(t)$
$x_3(t+1) = x_1(t) \,\mathrm{nand}\, x_3(t)$
1
2
3
t
1
2
3
t+1
or
nor
nand
(a)
(b)
(c)

Figure 10: *(a) Hypothetical example of a Boolean network (b) The associated Boolean equations (c) The corresponding wiring diagram. A state vector* $[0\,0\,0]$ *at* t *would be mapped to* $[0\,1\,1]$ *at* $t+1$. *Terms nor and nand denote the logical operators of "not or" and "not and", respectively.* [22]

In short, a Boolean network to model GRN is constructed by the means of information theory to detect the connections in the experimental data and to choose the Boolean functions to present these connections. For each n at $t + 1$ all the combinations of possible inputs from time t are considered to reveal the underlying logic. A detailed view of the model construction is presented by Liang et al. [26].

Boolean networks make strong assumptions on the regulation between the genes. However, this approach has been found out to produce good results, mainly because the assumptions allow effective modeling of vast amounts of data. The assumptions include the binary gene state, i.e., the gene is either on or off and the intermediate regulation levels are mapped to either one of these binary states, and the assumption on the synchronous gene state transitions. Various situations do exits, where different modeling formalisms are needed. [22]

4 Discussion

This work specified the problem of cDNA microarray data analysis and presented an overview of usual methods utilized in this field. The models and methods described in this work do not form a comprehensive collection of methods used today for cDNA microarray data analysis, but rather aim to introduce the reader to this topic by describing the most usual analysis methods. The analysis methods presented included clustering methods of hierarchical clustering, SOM, k-means clustering, and QTC. Additionally, the work discussed the use of different approaches to model the gene regulation, including directed graphs, Boolean networks, and Bayesian networks. Other clustering methods and modeling formalities were mentioned with references.

cDNA microarray data analysis is a field with a continuously growing demand for mathematical analysis methods. Data analysis is continuously evolving and new methods are under development. During this review I have discovered that very often the methods and models used in scientific articles have not been justified or compared with other methods and have been applied to data based more on habit than on the properties of the method. Thus, in my opinion, a standardized comparison of different methods and models would be very important in this field.

The mathematical methods and models in this field have remained numerous, mainly because there is no model that would be optimal for every research question. For example, simpler GRN models are used to detect the global connectivity of the studied genes and to address questions on the amount of interactions rather than their detailed nature.

In summary, the field of cDNA microarray data analysis is an interesting one with the need of standardized comparisons of different methods, models, and their performance. Therefore, the field is filled with promising issues for scientists interested in applied mathematics, modeling, and genetic biology.

References

[1] S.M. Arfin, A.D. Long, E.T. Ito, L. Tolleri, M.M. Riehle, E.S. Paeglei, and G.W. Hatfield. Global gene expression profiling in Escherishcia Coli K12. *J Biol Chem*, 38:29672–29682, 2000.

[2] A.A. Hill, E.L. Brown, M.Z. Whitley, G. Tucker-Kellog, and E.L. Brown. Genomic analysis of gene expression in C. elegans. *Science*, 290:809–812, 2000.

[3] L. Zhang, W. Zhou, V.E. Velculescu, S.E. Kern, R.H. Hruban, S.R. Hamilton, B. Volgestein, and K.W. Kinzler. Gene expression profiles in normal and cancer cells. *Science*, 276:1268–1272, 1997.

[4] S. Hautaniemi. *Studies of microarray data analysis with applications for human cancers*. PhD thesis, Tampere University of Technology, 2003.

[5] H. Bolouri and E.H. Davidson. Modeling transcriptional regulatory networks. *BioEssays*, 24:1118–1129, 2002.

[6] B. Albert, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell*. Garland Science, 4 edition, 2002.

[7] The National Human Genome Research Institute: www.genome.gov. 28.10.2006.

[8] P. Brazhnik, A. de la Fuente, and P. Mendes. Gene networks: how to put the function in genomics. *Trends in Biotechnology*, 20:467–472, 2002.

[9] M. Schena, D. Shalon, R.W. Davis, and P.O. Brown. Quantitative monitoring of gene expression patterns with complementary dna microarray. *Science*, 270:467–470, 1995.

[10] A. Holloway, R. van Laar, R. Tothill, and D. Bowtell. Options available - from start to finish - for obtaining data from dna microarrays. *Nature Genetics*, 32:481–489, 2002.

[11] J. Dopazo. Microarray data processing and analysis. In S. Lin and K. Johnson, editors, *Microarray data analysis II*. Kluwer Academic, 2002.

[12] Y.F. Leung and D. Cavalieri. Fundamentals of cdna microarray data analysis. *Trends in Genetics*, 19:649–659, 2003.

[13] ArrayExpress database, European Bioinformatics Institute:
www.ebi.ac.uk/arrayexpress. 22.10.2006.

[14] ArrayDB database, National Human Genome Research Institute: genome.nhgri.nih.gov/arraydb. 22.10.2006.

[15] P. D'haeseleer, S. Liang, and R. Somogyi. Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*, 16:707–726, 2000.

[16] Panel on Discriminant Analysis, Classification, and Clustering. *Discriminant Analysis and Clustering*. National Academy Press, Washington D.C., USA, 1988.

[17] J. Dopazo, E. Zanders, I. Dragoni, G. Amphlett, and F. Falciani. Methods and approaches in the analysis of gene expression data. *J Immunol Meth*, 250:93–112, 2001.

[18] L.J. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data: identification and analysis of coexpressed genes. *Genome Res*, 9:1106–1115, 1999.

[19] J.B. MacQueen. *Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1. University of California Press, Berkeley, 1967.

[20] T. Honkela. *Self-organizing maps in natural language processing*. PhD thesis, Helsinki University of Technology, 1997.

[21] A. Mateos, J. Herrero, J. Tamames, and J. Dopazo. Supervised neural networks for clustering conditions in DNA array data after reducing noise by clustering gene expression profiles. In S. Lin and K. Johnson, editors, *Microarray data analysis II*. Kluwer Academic, 2002.

[22] H. de Jong. Modeling and simulation og genetic regulatory systems: A literature review. *J Comput Biol*, 9:67–103, 2002.

[23] E.P. van Sommeren, L.F.A. Wessels, E. Backer, and M.J.T. Reinders. Genetic network modeling. *Pharmacogenomics*, 3, 2002.

[24] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using bayesian networks to analyze expression data. *J Comput Biol*, 7:601–620, 2000.

[25] D. Heckerman. A tutorial on learning with Bayesian networks. In M.I. Jordan, editor, *Learning in Graphical Models*. Kluwer Academic, 1998.

[26] S. Liang, S. Fuhrman, and R. Somogyi. REVEAL: A general reverse engineering algorithm for inference of genetic network architectures. In R.B. Altman, A.K. Dunker, L. Hunter, and T.E. Klein, editors, *Proc Pac Symp Biocomput (PSB'98)*, pages 18–29. World Scientific Publishing, 1998.