The recent advances of microarray technologies have made it possible to monitor simultaneously the expression pattern of thousands of genes in genomes. The challenge is to analyze effectively and interpret this large volume of data. The intrinsic nature of microarray data such as high dimensionality and small sample size calls for effective computational methods. This chapter describes several computational intelligence techniques used for analyzing gene expression data. These are either well known tools successfully used for different data analysis tasks and useful also in the area under discussion or customized computational or statistical tools developed to handle the peculiarities of gene expression data. The related methods fall in three research topics:

- Identification of non-redundant information from gene expression data
- Clustering of gene expression data
- Inference of gene regulatory network

Section 4.1 discusses statistical analysis methods used for identifying discriminative genes from microarray expression data. Section 4.2 refers to
analysis. Section 4.3 ends the chapter by presenting computational methods used for inferring gene regulatory networks from microarray data.

4.1 Statistical Methods for Identifying Differentially Expressed Genes

Among a large number of genes encoded in the microarray gene expression data, only a very small fraction of them are relevant for a certain task. A very challenging problem arises as a result – how to select informative features (genes) for performing data analysis such as diagnosis, prognosis, subtype classification of a heterogeneous disease and understanding a gene regulatory network. This selection procedure is important and sometimes necessary because of three main reasons. First, it is impossible for biologist or physicians to examine the whole feature space in the laboratory experiments at a stretch. Second, irrelevant features result in unnecessary computational difficulties. Third, especially in the case of cancer (or generally disease) classification, since the activity of only few genes are responsible for the development of the disease, it is obvious that the rest of the genes measurements are irrelevant to the task of class distinction. Such irrelevant genes act as ‘noise’ since they confuse classifiers and thus obstruct biological information within data sets.

Identifying genes or subsets of genes that are differentially expressed in disease and normal tissues using computational algorithms will obviously increase the diagnosis accuracy. Besides this, post classification analysis of the respective discriminative (or predictive) genes may reveal important
information in what concerns the dynamics of the disease. This would be highly beneficial for drug discovery and early disease prediction.

Currently, several statistical and machine learning methods have been developed for gene selection. Among them feature ranking approaches are particularly attractive because of their simplicity, stability and empirical success. In these approaches, features are scored by a certain ranking criterion and the rank of features is used as the base of selection mechanism. For example t-statistics [85], regression model [86], $\chi^2$-statistics [87] and mixture model [88] can be used for feature ranking. Some other approaches utilize machine learning approaches such as support vector machines (SVM) [89, 90, 91], decision trees [57] and genetic algorithms [92] for feature ranking and selection. Based on the rank of features, subset of significant features can be selected for further analysis.

The t-statistic, also known as the student t-test, is a well-known statistical approach frequently applied in microarray data analysis. There are several versions of two sample t-test, depending on whether the sample size is large and whether it is reasonable to assume that gene expression levels have an equal variance under the two conditions [85]. Because usually sample sizes are small and there is evidence to support unequal variance [86], microarray analysis uses t-test with two independent normal samples with unequal variance.

### 4.2 Cluster Analysis of Gene Expression Profiles

Genes that are involved in correlated functions tend to yield similar expression patterns in microarray experiments. Analyzing these data and
learning their expression pattern can therefore reveal functional association of genes. Clustering techniques are typically used to group genes with similar expression patterns based on the organization of expression data. It generally aims to identify clusters in the data based on the similarity between genes. When there is no or little priori knowledge about the data, clustering is an appropriate tool to analyze data. Clustering can be performed on both dimensions of the expression data matrix, cluster genes or samples. From the machine learning perspective cluster analysis is unsupervised since there is no desired outcome for any particular gene or experiment. From the data mining perspective, the technique is an exploratory data analysis method.

The successful clustering results may provide researchers crucial information regarding the biological role of unknown genes. This is based on the fact that genes which show similar expression patterns (co-expressed genes) are often functionally related and they share the same regulatory mechanism at the sequence level. If a novel gene of unknown function falls into cluster containing genes with known (or partially known) functionality, it is likely that this gene serves the same functions as the other members of the cluster [93]. Since co-expressed genes have a high probability to participate in the same pathway, clustering will provide crucial information for the inference of regulatory information.

The widely adopted clustering techniques for gene expression data include hierarchical clustering [66], self-organizing maps (SOMs) [94, 63], k-means clustering [95, 96], Fuzzy C-means (FCM) [97] etc.
4.2.1 Hierarchical Agglomerative Clustering Algorithm (HAC)

The HAC algorithm is one of the earliest methods used to cluster gene expression pattern [66]. HAC successively merges clusters until all elements belong to the same cluster from an initial partition. It connects objects to form clusters based on their distance. The clustering result of HAC can be represented by a tree structure called dendrogram, where the leaves represent individual gene patterns and the internal nodes represent clusters of similar patterns. The distance at which the two clusters merge (a measure of dissimilarity between clusters) is called the threshold distance, which is measured by the height of the node from the leaf.

Based on the calculation of similarity among the non-singleton clusters, a variety of hierarchical agglomerative techniques have been proposed. Single linkage, complete linkage, and group average linkage clustering are commonly used.

- **Single Linkage**

  In single linkage (also known as nearest neighbor), the distance between two clusters is defined as the distance between the closest pair of objects that are in different clusters. In other words, the distance between two clusters is given by the length of the shortest path between the clusters.

- **Complete Linkage**

  In the complete linkage method (maximum or furthest-neighbor method), the distance between two clusters is given by the value of the longest link between the clusters.
• **Average Linkage**

  Here, the distance between two clusters is computed as the average of distances between all pairs of objects, one in each cluster.

• **Wards Method**

  In this method, cluster membership is assessed by calculating the total sum of squared deviations from the mean of the cluster. The clusters are joined in such a manner that it produces the smallest possible increase in the sum of squared errors.

  Eisenet. al. [66] used the average-link HAC algorithm to analyze the yeast cell cycle microarray expression profiles 2467 genes measured over 80 samples to discover indications of the status of cellular process. They also proposed the usage of a colored matrix to visualize the clustering result which presents a natural understanding of the clustering result of the data set. Hierarchical clustering has been used by [98, 99] for the classification of cell line, especially human cancers. They clustered the dataset on other dimension (samples) in an attempt to find new possible tumor subclasses.

  Although the HAC algorithm has been widely used for clustering gene microarray expression profiles, it has several drawbacks. First, as Tamayo et. al. [63] have noted, HAC suffers from a lack of robustness when dealing with data containing noise, and a preprocessing data is needed to filter out noise. Second, hierarchical clustering is expensive in terms of their computational time and storage requirements when dealing with larger data. Third, since HAC is unable to reevaluate the results, some clusters of patterns are based on local decisions that will produce patterns which are...
difficult to interpret when HAC is applied to a large array of data. Fourth, the number of clusters is decided by cutting the tree structure at a certain level. Biological knowledge may be needed to determine the cut.

4.2.2 K-means Clustering

The k-means algorithm is a typical partitioning based clustering technique. Given a set of N objects, a partitioning method constructs k partitions of data, where each partition represents a cluster and $k \leq N$. The k-means algorithm begins by $k$ randomly selected data objects as cluster centroid. These centroids should be placed far away from one another as much as possible to get a better result. Next, each data object is assigned to the cluster with closest centroid. Then the centroid of each cluster is recalculated as the mean of all data objects belonging to the cluster. This process iterates until no more changes occur, or the amount of change fall below a pre-defined threshold.

The k-means method is one of the most popular methods used in DNA microarray data analysis due to its high computational performance. Tavazoie et al. [100] selected the most highly expressed 3000 yeast genes from the yeast cell cycle microarray profiles and applied the k-means algorithm to cluster them into 30 clusters. They successfully found that several clusters were significantly enriched with homo functional genes.

However, k-means needs as input parameters the cluster number $k$ and initial centroids. Its result is subject to the initialization process or in other words, different runs of k-means on the same input data might produce different solutions. To avoid the local suboptimal minimum, one should run
the k-means algorithm multiple times with different initialization of centroids, and then choose the one with smallest average dissimilarity. Furthermore, since the number of clusters is not known, to determine the proper number of clusters in the dataset, one needs to run the k-means algorithm for a range of k value to determine the best k value.

### 4.2.3 Self-Organizing Maps

The self-organizing map (SOM) was developed by Kohonen [94] as a neural network based clustering method. It has been extensively used as a tool for visualizing and interpreting high dimensional data. SOM performs unsupervised learning to produce a lower-dimensional (usually 2D) representation of the input space of the training data set samples and use a neighborhood function to preserve the topological properties of the input space. As with other types of centroid based clustering, the goal of SOM is to find a set of centroids (reference vectors) and assign each object in the dataset to the centroid that provides the best approximation of that object. In neural network terminology, one neuron is associated with each centroid.

Unlike k-means, SOM imposes a topographic ordering on the centroids. During the training process, SOM uses each data point to update the closest centroid and the centroids that are nearby. In this way, SOM produces an ordered set of centroids for any given dataset. In the SOM grid, neighboring centroids will be closely related than those are farther away. This makes it easy to find cluster relationship during the visualization of the SOM. Furthermore, SOM does not keep track of the current cluster membership of an object, and unlike k-means, if an object switches clusters,
there is no explicit update of old cluster centroid. Of course, the old cluster may well be in the neighborhood of the new cluster and thus may be updated for that reason. The processing continues until some predetermined limit is reached or when the centroids are not changing considerably.

Tomayo et al. [63] employed SOM to analyze yeast genes from the SaccharomycesCerevisiae dataset and different human cell culture microarray expression profiles. The SOM was able to identify the predominant gene expression pattern in these microarray expression profiles. Although, SOM proved robust performance, it does not overcome the problems of k-means such as cluster number determination and sub-optimization. Moreover, its convergence is controlled by various user supplied parameters such as learning rate, grid topology of neurons and the neighborhood function.

4.2.4 Fuzzy Clustering

In partitioned clustering algorithms such as k-means or self-organizing maps, each gene belongs to exactly one cluster. However, genes are often highly correlated with the patterns of more than one cluster. Fuzzy clustering appears to be a good candidate to reflect the genes multi-cluster participation since it can assign genes degrees of membership to a cluster. The membership value can vary between zero and one. This feature enables fuzzy clustering to provide more information about the structure of gene expression data.
The most popular fuzzy clustering approach in gene expression analysis is that of Fuzzy C-means (FCM) [97]. It allows one data object to belong to more one cluster at the same time.

It is based on minimization of the following objective function:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \| x_i - c_j \|^2, \quad 1 \leq m < \infty$$

where $m$ is a real valued number which controls the fuzziness of the resulting clusters, $u_{ij}$ is the degree of membership of gene $x_i$ in the cluster $j$, $x_i$ is the $i^{th}$ of d-dimensional measured data, $c_j$ is the d-dimension center of the cluster, and $\| \cdot \|$ is any norm expressing the similarity between any measured data and the center. Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership $u_{ij}$ and the cluster centers $c_j$ by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left( \frac{\| x_i - c_j \|}{\| x_i - c_k \|} \right)^{\frac{m-1}{m}}}$$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m \cdot x_i}{\sum_{i=1}^{N} u_{ij}^m}$$

This iteration will stop when

$$\max_{ij} \left\{ \left| u_{ij}^{(k+1)} - u_{ij}^{(k)} \right| \right\} < \varepsilon$$

where $\varepsilon$ is a termination criterion between 0 and 1, whereas $k$ is the number of the iterations.

Dembele and Kastner [101] applied FCM clustering approach on a variety of dataset comprising the serum data set [102] consist of 517 genes whose expressions vary in response to serum concentration in human
fibroblasts, the yeast dataset [103] and a human cancer data set represents gene expression patterns of 9703 genes in 60 human cancer cell lines [104]. Their analysis proved that the overall expression pattern for a given gene may correspond to the superimposition of distinct patterns, each corresponding to a given mode of regulation.

Besides the predetermination of the number of clusters in the data set, the FCM method requires to choose $m$, the fuzziness parameter. The optimal values for $m$ vary widely from one data set to another.

Several techniques that improve the classical clustering algorithms has been proposed and presented in section 4.4 along with the contribution of this study presented in chapter 5.

4.2.5 Cluster Validation

For gene expression data, cluster analysis partitions the given dataset into groups of co-expressed genes, groups of samples with a common phenotype, or groups of genes and samples involved in specific biological processes. However, different clustering algorithms, or even a single clustering algorithm using different parameters, generally result in different sets of clusters. Therefore, it is important to compare various clustering results and select the best one that fits dataset. The process of assessing the quality and reliability of the clustering results obtained from various clustering process is called cluster validation. Validation can be either statistical or biological. Statistical validation can be done by assessing the cluster compactness and separation, by examining the predictive power of the clusters, or by testing the robustness of a cluster result against the
addition of noise. A common method to biologically validate cluster outputs is to search for enrichment of functional categories within a cluster.

### 4.2.5.1 Assessing Cluster Homogeneity and Separation

Since all genes within a cluster are expected to be tightly co-expressed, a clustering result can be considered reliable if the within-cluster distance is small and the cluster has an average profile well delineated from the remainder of the data set (maximal inter-cluster distance). Such criteria can be formalized in several ways, such as the silhouette coefficients [105], or Dunn’s validity index [106]. Both methods can be considered as a stand-alone tool to compare cluster results. Since these statistics only require a clustered sample and a set of dissimilarities, they can be found using any clustering method and any distance metric. Dunn’s validity index (D) recognizes compact and well separated clusters. The objective is to maximize the intercluster distances and minimize the intracluster distances. Therefore, the number of clusters that maximizes $D$ is taken as the optimal number of the clusters.

Whereas, the silhouette of a gene measures how well matched it is to the other objects in its own cluster versus how well matched it would be if it were moved to another cluster. It is a composite index reflecting the compactness and separation of the clusters. When all the elements are best classified, the silhouettes should all be close to 1, and the average silhouette width will be high. Therefore, the highest average silhouette width gives the strongest clustering structure, resulting in an output, which gives the optimal number of clusters.
4.2.5.2 Figure of Merit

The figure of merit (FOM) [107] compares the output of different clustering algorithms in terms of predictive power and homogeneity. This methodology is a combination of leave-one-out cross validation (LOOCV) and the Jackknife approach. The clustering algorithm is applied to all experimental conditions except for one left-out condition. If the algorithm performs well it is expected that the values of genes from a given cluster are highly coherent with those from the left out condition. Therefore, the FOM for a clustering result is computed by finding the root mean square deviation in the left-out condition of the individual gene expression levels relative to their cluster mean. The FOM estimates the within cluster similarity of the expression values of the removed experiment and therefore reflects the prediction power of the clustering. It is expected that removing one experiment from the data should not affect the cluster output if the output is robust. For cluster validation, each condition is used as a validation condition and the aggregate FOM over all conditions is used to compare cluster algorithms.

4.2.5.3 Cluster Sensitivity

Expression level of genes in the microarray dataset reflects the superposition of real biological signals and experimental errors. The confidence in cluster membership of a gene can be assessed by creating artificial datasets by adding to the original data a small amount of artificial noise (similar to the experimental noise in the data). Clustering is subsequently performed on the artificial data. If the biological signal is stronger than the experimental noise in the measurement of a particular
gene, adding small artificial noise (in the range of experimental noise) to the expression profile of this gene will not drastically influence its overall profile and therefore will not affect the cluster membership. Thus, it is clear that the cluster membership of that particular gene is robust with respect to the sensitivity analysis and reliable confidence can be assigned to the clustering result of that gene. However, for genes with low signal-to-noise ratio, the clustering result will be more sensitive to adding artificial variants. Through some robust statistics [108], sensitivity analysis allows us to detect which clusters are robust within the range of experimental noise and therefore trustworthy for further analysis.

The key issue in the validation method is the determination of noise level for sensitivity analysis. Bittner et al. [108] perturb the data by adding random Gaussian noise with zero mean and standard deviation that is estimated as the median standard deviation for the log ratios for all genes across the experiments. The bootstrap analysis methods proposed by Kerr and Churchill [109] uses the residual values of a linear analysis of variance (ANOVA) model as an estimate of the measurement error. The residuals are subsequently used to generate new replicates of the data set by bootstrapping (adding residual noise to estimated values).

4.2.5.4 Biological Significance Based on p-value

One way to biologically validate results from clustering algorithm is to compare the gene clusters with existing functional classification annotations from various ontology databases. In such databases, genes are assigned to one or more functional categories representing their biological functions, biochemical properties and so on. Finding clusters enriched
bygenes with similar function is a proof that a specific clustering technique produces biologically relevant results.

For each cluster, using hyper geometric probability distribution, a p-value can be calculated, which is the probability of observing the frequency of genes in a particular functional category in a certain cluster. The p-value of observing k genes from a functional category within a cluster of size n is

\[
p = 1 - \sum_{i=0}^{k-1} \binom{f}{i} \frac{(g-f)}{\binom{g}{n-i}} = \sum_{i=k}^{\min(n,f)} \binom{f}{i} \frac{(g-f)}{\binom{g}{n-i}}
\]

where f is the total number of genes within that functional category and g is the total number of genes within the genome. The p-value can be calculated for each functional category in each cluster. The lower the p-value is, higher the biological significance of a cluster.

In addition to the validity measures mentioned, combined clustering techniques are also used to validate the clustering results. This method sometimes provides clustering output which matches better with the real biological classes (prior knowledge).

### 4.3 Inference of Gene Regulatory Networks

The success of genome sequencing projects has enabled the biologist to identify almost all the genes responsible for the biological complexity of several organisms. The next important task is to understand the complex interactions among genes and gene products to carry out specific cell functions. Since 1960, the methods from mathematics and physics have been used to describe and simulate small gene networks more stringently.
Nowadays, biological methods and high-throughput experimental technologies make it possible to study a large number of genes and proteins in parallel enabling the inference of larger gene networks. This allows simulating regulatory networks more efficiently and has led to a new discipline called Systems Biology, which combines methods from biology with methods from mathematics, physics and engineering to describe biological systems.

A number of different approaches to gene regulatory network modeling based on large scale microarray data have been introduced, including linear models [110] Bayesian networks [11, 111], graphical model [112], neural networks [69, 113, 114], differential equations [115, 116], and models including stochastic components on the molecular level [117]. The models can be static or dynamic, continuous or discrete, linear or nonlinear, deterministic or stochastic. By analyzing the data, an appropriate learning technique has to be chosen for each model to find the best fitting network structure and parameters. Following sections present some of the popular inference models for GRN and highlight the advantages and limitations of each model.

4.3.1 Boolean Network

Boolean network (BN) model, introduced by Kauffman [118, 119, 120], is a simple computational model that may provide insight into the overall behavior of genetic networks. The main objectives of boolean network modelling is to study generic coarse-grained properties of large genetic networks and the logical interactions of genes, without knowing specific
quantitative details [121]. The biological basis for the development of Boolean networks as models of gene regulatory network lies on the observation that during the regulation of its functional states the cell often exhibits switch-like behavior. In Boolean network gene expression is quantized to only two levels: ON and OFF, which are represented as “activated” and “inhibited”. The interactions between the genes are represented by Boolean functions, which determine the state of a gene on the basis of the states of some other genes. The figure 4.1 illustrates the effect of A, B, C and D on F in the form of directed graph and figure 4.2 shows the logic diagram of the activity on F as a Boolean function of 4 input variables.

![Figure 4.1](image1.png)

**Figure 4.1** A directed graph illustrating a hypothetical gene regulatory network. Arrowed lines represent activation and lines with bars in the end represent inhibition.

![Figure 4.2](image2.png)

**Figure 4.2** The logic diagram describing the activity of F in terms of 4 inputs A, B, C and D. The gates connecting A and B is an AND gate, the gate with input D is a NOT gate and the gate output is F is a NAND gate.
A Boolean network $G(V, B)$ is defined by a set of nodes (variables) representing genes $V = \{x_1, \ldots, x_n\}$ and a set of Boolean functions $B = \{f_1, \ldots, f_n\}$. Each $x_i$ represents the state of gene $i$, where $x_i = 1$ represents the fact that the gene $i$ is expressed and $x_i = 0$ means it is not expressed. Each Boolean function $f(x_{j1}(i), x_{j2}(i), \ldots, x_{jk}(i))$ with $k$ specific input nodes is assigned to node $x_i$ and is used to update its value. The gene state at time point $t+1$ is determined by the values of some other genes at previous time point $t$ using one Boolean function $f$ taken from a set of Boolean functions $B$. The values of all the nodes in $V$ are then updated synchronously.

Boolean networks have proved successful in modelling real world regulatory networks [122, 123]. One of the appealing properties of BNs is that they are inherently simple, emphasizing generic network behavior rather than quantitative biochemical details, but are able to capture much of the complex dynamics of gene regulatory networks. However, their application in practice is hindered by a number of shortcomings. In particular, analysis can be problematic due to the exponential growth in Boolean states and the lack of tool support in this area. They are also unable to cope with the inconsistent and incomplete regulatory network data that often occurs in practice. Since Boolean network has no intermediate expression level, it cannot capture the details of the biochemical reactions involved in cellular processes. However, it is not the binary nature of the Boolean network model that is its greatest weakness, one even more important deficiency is its determinism. Deterministic models, such as the Boolean network, cannot represent the consequential perturbations due to external latent variables. In addition, the Boolean network model in its original formulation cannot be used to represent
biologically meaningful events, such as gene mutations. The stochastic extension of Boolean network - probabilistic Boolean network (PBN), accounts for those latent variables and gene perturbations while keeping the Boolean logic as the model for the gene-gene interactions [124].

### 4.3.2 Bayesian Network

Bayesian networks provide a language for representing joint probability distributions of many random variables. Bayesian networks have been applied extensively for modeling complex domains in different fields [125]. This success is due to both the flexibility of the models and the naturalness of incorporating expert (or prior) knowledge into the domain. Another major advantage of Bayesian model is the ability to learn from observed data. This is particularly important when the knowledge about the domain is partial; as is the case in the biological domains. Although the Bayesian network is a discrete variable model, the variables can also be continuous.

In the Bayesian Network formalism [11], the structure of GRN is modeled by a directed acyclic graph that explicitly establishes probabilistic relationships between network nodes. It is represented by a tuple $G = (V, E)$ where $V$ is the set of vertices corresponds to the random variables $X_1, X_2, \ldots, X_n$ ($X_i$ describes the expression level of gene $i$) and $E$ describes the set of conditional distributions. For each $X_i$, a conditional distribution $p(X_{ij} | \text{parents}(X_i))$ is defined, where parents ($X_i$) denotes the variables corresponding to the direct regulators of $i$ in $G$. The graph $G$ and the
Chapter 4

Conditional distributions $p(X_i \mid \text{parents}(X_i))$ encode a unique joint probability distribution $p(X)$, thus defining a Bayesian network.

The graph encodes a set of independence statements of the form: for every gene in $G$, $I(X_i; \text{nondescendants}(X_i) \mid \text{parents}(X_i))$, that is, every node is independent of its nondescendants given its parents in graph $G$. By means of the Markov assumption, the joint probability distribution over $X$ can be written as

$$p(X) = \prod_{i=1}^{n} p(X_i \mid \text{parents}(X_i))$$

Hence two Bayesian networks are said to be equivalent if they imply the same set of independencies among variables. Although Bayesian network is effective in dealing with noise, incompleteness and stochastic aspects of gene regulation, they fail to consider temporal dynamic aspects of gene regulation that are an important part of GRN modelling [11]. To effectively deal with the dynamic process of regulatory network, Dynamic Bayesian networks (DBN) were developed, and can yield more accurate models. However, their benefits are hindered by the high computational cost required in the cases where large numbers of genes are involved [126]. For this purpose, some supplementary methods, such as network decomposition (for dimension reduction), and Monte Carlo strategies using random sampling have been developed to enhance performance [127, 128, 129].

### 4.3.3 Differential Equations

Unlike the above discrete models, the models based on differential equations uses continuous variables. Several models based on differential
equations have been used to explore genetic networks, of which some uses linear ordinary differential equations (ODEs) and others nonlinear power law differential equations [130, 131, 132, 133, 134]. In general, the change in the expression level of a gene at a certain time (discrete or continuous) is characterized by rate equation that takes the regulatory influence (activation or inhibition) of other genes into account. The rate equations have the mathematical form

$$\frac{dx_i}{dt} = f_i(x_1, x_2, ..., x_n, p, u)$$

where $x_i$ ($1 \leq i \leq n$) is the expression level of gene $i$ at time $t$, $n$ is the number of genes, $p$ is parameter set of the system and $u$ is a set of transcriptional perturbation. The function $f_i$ can be linear, piecewise linear, pseudo-linear, or continuously nonlinear, each describing the system dynamics with a different level of complexity.

Compared to the discrete variable models, the differential equation models can more accurately represent the system dynamics of Gene Networks by virtue of their use of continuous variables. In particular, with nonlinear ODE models (such as S-system models), a given system’s steady-state evaluation, control analysis and sensitivity analysis can be established mathematically [135]. D’Haeseleer et al. [136] and von Dassow et al. [137] successfully applied differential equation to the modeling of regulatory networks in Drosophila, though the use of linear models or linearization of non-linear models does not limit the generality of methods and results.

Although the differential equations models are effective in reproducing the characteristics of the target system accurately, the benefit
comes at a significant cost: an increase in computational expense. Such expense can make certain system sizes too unwieldy to be represented by anything but the simplest of differential equation models, which may lead to lower likelihood of accurately modeling the underlying physical phenomena. Also, these models depend on numerical parameters that are often difficult to establish experimentally. It must be also noted that the modeling is very sensitive to noisy and imprecise data (as in the case of data resulting from microarray experiments) [138].

### 4.3.4 Neural Networks

The other commonly used type of continuous variable model to represent the dynamics of gene regulation is the neural network model. Each neuron represents genes and the connection between nodes represents the regulatory actions of one gene over other. The influence of one gene product on the expression level of other genes of the system is defined by a weight matrix. Each layer of the network represents the expression level of genes at time $t$. The expression level of a gene at time $t+1$ can be derived from the expression levels at the time $t$ ($y_j$) and connection weights ($w_{ij}$) of all the genes connected to it. In other words, the net regulatory effect on a particular gene can be regarded as a weighted sum of expression level of all other genes capable of regulating it. Thus $g_i$, a regulatory effect to gene $i$, is

$$g_i \approx \sum_j w_{ij} \cdot y_j$$

This regulatory effect is transformed by an activation function to the interval $<0, 1>$. For sigmoid activation function regulatory effect is defined as
where $b_i$ represents an external input that can be interpreted as a reaction delay parameter. High positive or negative values of $b$ result in a low influence of factors given in the weight matrix. In principle, all genes can regulate all others (fully connected network); in reality, only a few genes control the activity of one particular gene. The state of a gene in the network is updated in a stepwise manner; the state of each gene in the next time point $t_{i+1}$ is determined by the state at time $t_i$.

The most successful neural network model is recurrent neural model (RNN) [139]. This model is biologically plausible and noise resistant. By adapting self-loops and feedback connections to their structure, recurrent networks can deal with temporal and spatial/temporal problems, both of which are used to memorize past information. In addition, its nonlinear characteristics provide information about the principles of control, as well as about the natural interactions of elements of the modeled system. There are several RNN architectures for GRN modelling ranging from restricted classes of feedback to full interconnection between nodes [140, 141, 142]. Greedy algorithms based on the gradient descent method, such as back-propagation through time (BPTT) [143], have been developed to efficiently update the relevant parameters of recurrent networks in discrete time steps. Recently, Xuet. al. [140] used an RNN combined with Particle Swarm Optimization (PSO) to capture the complex nonlinear dynamics of gene
regulatory networks. However due to its training time and computational complexity, this modelling can currently applied to only small systems.

### 4.3.5 Other Inference Approaches

Recent approaches tried to overcome the drawback of traditional methods in several ways [144]. Woolf and Wang [13] applied fuzzy rules to every possible combination of genes to find the activator/repressor relationship in a normalized subset of saccharomyces cervisae data. Although their method is intuitive, the results are consistent with the literature on genetic networks of Saccharomyces Cervisae. However this approach is slow and computationally complex. Keedwell and Narayanan [145] proposed a hybrid neuro-genetic algorithm to extract regulatory relationship among genes. Their method integrates genetic algorithms with a single layer artificial neural-network, where each chromosome of the GA selects a small number of regulating genes from the whole data set and the neural network identifies the regulatory relations among genes. However, when modelling the complex temporal dynamics of gene expression regulation, the lack of a recurrent structure and the proper training method of the ANN may pose serious problems. Also, the method is vulnerable to local minima traps.

Although RNNs are better candidates in dealing with temporal sequence production problems, it has been proved that recurrent neural fuzzy networks outperform recurrent neural networks [146, 147] in problems that involve concurrent spatial and temporal mapping like the one of regulatory networks reconstruction. Additionally, due to the high level
human like reasoning fuzzy system, fuzzy-based approaches are efficient in handling the uncertainties of modelling noisy data [148, 149]. In [148] a fuzzy data mining approach has been proposed to measure the statistically significant fuzzy dependency relationships among genes. Sokhansanj et. al. [149] demonstrated an approach with exhaustive search for possible rules describing gene interactions, under the framework of a linear fuzzy logic scheme that restricts the search space. However, both methods require prior data discretization, while [149] has the additional disadvantage of not considering temporal information.