Analysis of Microarray Data using Artificial Intelligence Based Techniques

Chapter · June 2016

DOI: 10.4018/978-1-5225-0427-6.ch011 · Source: arXiv

1 author:

Khalid Raza
Jamia Millia Islamia
147 PUBLICATIONS **1,547** CITATIONS

SEE PROFILE

Analysis of Microarray Data using Artificial Intelligence Based Techniques

Khalid Raza
Department of Computer Science
Jamia Millia Islamia (Central University), New Delhi, India
Email: kraza@jmi.ac.in

June 30, 2015

Abstract

Microarray is one of the essential technologies used by the biologist to measure genome-wide expression levels of genes in a particular organism under some particular conditions or stimuli. As microarrays technologies have become more prevalent, the challenges of analyzing these data for getting better insight about biological processes have essentially increased. Due to availability of artificial intelligence based sophisticated computational techniques, such as artificial neural networks, fuzzy logic, genetic algorithms, and many other nature-inspired algorithms, it is possible to analyse microarray gene expression data in more better way. Here, we reviewed artificial intelligence based techniques for the analysis of microarray gene expression data. Further, challenges in the field and future work direction have also been suggested.

1 Introduction

The bioinformatics is an interdisciplinary area of study where one of the objectives is to deal with the analysis and interpretation of large sets of data generated from various large-scale biological experiments. The example of one such large-scale biological experiment is measuring the expression levels of tens of thousands of genes simultaneously under some environmental condition. Microarray is one of the essential technologies used by the biologist to measure genome-wide expression levels of genes in a particular organism. As microarrays technologies have become more prevalent, the challenges

associated with collecting, managing, and analyzing the data from each experiment have essentially increased. Robust laboratory protocols, improved understanding of the complex experimental design and falling prices of commercial platforms, all these have combined to drive the field to more complex experiments, generating huge amounts of data (Brazma and Vilo, 2000).

With the help of measured transcription levels of genes under different biological conditions (e.g. at various developmental stages and in different tissues), biologists are able to develop gene expression profiles that differentiate the functionality of each gene in the genome. The gene expression profiles are organized in the form of a matrix, where rows represents genes, columns represents samples/replicas, and each cell of the matrix contains a numeric value representing the expression level of a gene in a particular sample. Generally, such as table is called gene expression matrix. If over expression of certain genes is correlated with a certain disease then researchers can discover what are other conditions affecting expression-level of these genes. Also, what are the other set of genes having similar expression profiles pattern. Hence, suitable compounds (potential drugs) can investigated that can lower the expression level of these overexpressed genes (Babu, 2004).

Many sophisticated statistical and computational tools have been developed to help biologists for the analysis of gene expression data and to identify novel targets from their experimental data (Deng et al., 2009; Debouck and Goodfellow, 1999). Among these techniques, clustering and statistical methods are most commonly used data analysis methods. Clustering generally groups the gene expression data with similar expression pattern, i.e. co-expressed genes. However, clustering approach suffers from several drawbacks (Bassett et al., 1999). The statistical methods help to analysis gene expression data and infer relationships between genes. However, it fails to provide complex regulatory relations among genes.

The chapter is organized as follows. Section 2 describes the background of Microarray experiments and data generation. Section 3 covers the applications of Microarrays and Section 4 describes artificial intelligence based techniques, and reviews its application in the analysis of Microarray data. Section 5 summarizes the chapter and presents research challenges and future work directions.

2 Microarray Technology

With the help of Microarray technology, one can measure the expression level of all genes in a genome simultaneously. By measuring and comparing the expression level of genes in an unhealthy versus healthy cell, it would be possible to identify genes which are responsible for various diseases. Due to unprecedented amount of large biological data generated out of microarray experiments, researchs focus has shifted from the generation of data to the analysis and presentation of data in the most efficient manner (Hood, 2003; Kitano, 2002a,b). With the help of these technologies, researchers can find out answer to some of the challenging questions like;

(i) What are the functions of different genes?

(ii) In what cellular processes do these genes participate?

(iii) How genes are regulated?

(iv) How genes and its products (proteins) do interact, and what are these interaction networks?

(v) How expression level of genes differs in different cell types and states?

(vi) How expressions of genes are affected by various disease or drug treatments?

Microarrays are frequently used in biomedical research to tackle a number of problems, including classification of tumors, or gene expression response to different stress conditions. A central and frequently asked question in microarray is the identification of differentially expressed genes (DEGs). The DEGs are those genes whose expression levels are associated with a response or covariate of interest (Dudoit et al., 2002). The covariates can be either polytomous (for instance, treatment/control status, cell type, drug type) or continuous (for instance, drug doses), and the responses can be, for instance, censored survival times or any other clinical outcomes (Dudoit et al., 2002; Lee et al., 2012). Scientists from different disciplines such as biology, statistics, computer science, mathematics, bioinformatics, etc. are working in this area to identify some new insight from DNA microarray data such as identification of differentially expressed genes, classification between cancerous and non-cancerous genes, identification of potential genes for drug target, identification of gene function, and so on. In the following section, various steps involved in Microarray experiments have been discussed.

2.1 Experimental Setup

Basically microarray is a solid base having grid of spots where genetic material of known sequence is arranged systematically. It is mostly made up of glass on which single stranded DNA molecules are attached at fixed positions. The size of the arrays can vary from microscope slide to square silicon chips. On an array, there can be thousands of spots and each spots contain number of identical DNA molecules. The microarray fabrication can be done in two ways: i) cDNA and ii) oligonucleotide. The steps involved in microarray experiments are shown in Fig. 1.

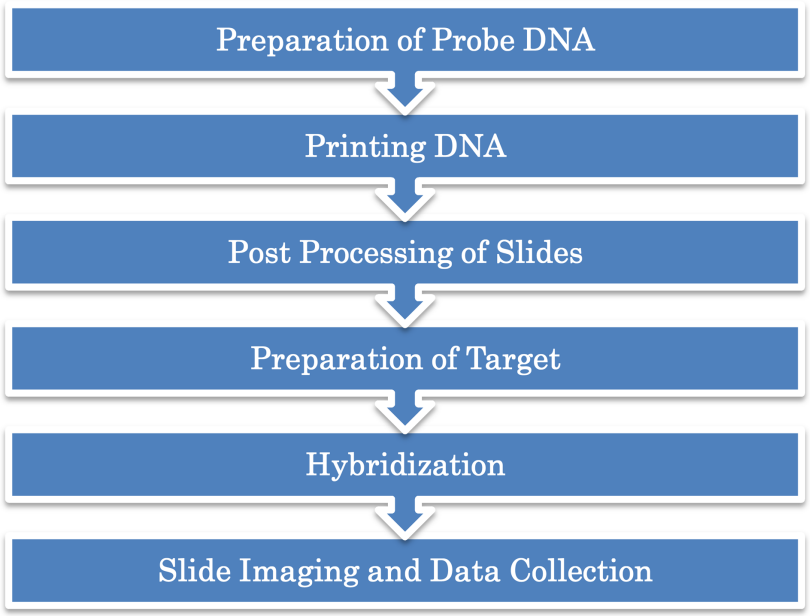

Figure 1: Step involved in microarray experiments

2.1.1 Preparation of probe DNA

For the study of large-scale expression, a specific DNA sequence is needed for all genes whose expression values are to be measured. The selection of probe is done on the basis of resources available for obtaining the representation of the genes under studies. The simplest way is to amplify every known ORF in the genome and use it as a probe. PCR is used for the amplification purpose and allows multiplication of DNA fragments by millions in just few hours. ESTs may be used to identify distinct mRNA transcripts.

2.1.2 Printing array

In cDNA array, arrays are mostly printed on Poly-L-lysine coated glass microscope slide using arraying robot. During the arraying operation, a large number of slides are placed on and secured to a platter. The samples of DNA are placed in microliter plates on the stand. The reservoir slot of each tip is filled with 1 liter of DNA solutions. The tips are then lightly tapped at identical positions on each slide leaving a small drop of DNA solution on the poly-L-lysine coated slide. In Oligonucleotide array, oligos are printed at the spots instead of cDNA. Same robotics can be applied to manufacture both types of arrays. However, the preparation of oligonucleotide array is quite different. During fabrication of array, the probes are synthesized on the chip using photolithography.

2.1.3 Post processing of slides

This step consists of Rehydration and Blocking. The spots on the microarray are rehydrated to distribute DNA more evenly. In the blocking process, free reactive groups on the slide surface are modified to minimize their ability to bind to labeled target DNA. If these groups are not blocked, the labeled DNA target can bind to the surface of the slide.

2.1.4 Preparation of target

In this step, isolate mRNA from the samples and purify it. Since, mRNA degrades very fast; hence it is reverse-transcribed into more stable cDNA.

2.1.5 Hybridization

In hybridization process, a single stranded DNA molecule is bound to another single strand DNA molecule with a precisely matching sequence. After hybridization process, the microarray properly washed to eliminate any excess labeled sample and finally dried using a centrifuge. Sometimes two types of target mRNA samples are simultaneously hybridized on the array, called two channel microarray experiment. In that case, two types of molecules are added to targets and uses fluorescent dyes like Cy3 and Cy5, which can be separated spectrally. The Cy3 is green and Cy5 is red when excited by laser light at specific wavelength.

2.1.6 Slide imaging

Under this step, the microarray is scanned to measure the fluorescent signal emitted at every spot that determine the amount of labeled sample bound to each spot. The laser scanning confocal microscope is used for this purpose. For single channel array, the array is scanned once but for two-channel experiment, it is scanned in two phases. In the obtained image, the intensity of each spot is proportional to the amount of mRNA from the sample, matching cDNA sequence of given pot. A gene expressed in a sample labeled with red dye and not expressed in the other sample will produce a red spot and vice versa. A gene expressed in both the samples will produced equal amount of red and green intensities and a spot is given yellow (Fig. 2).

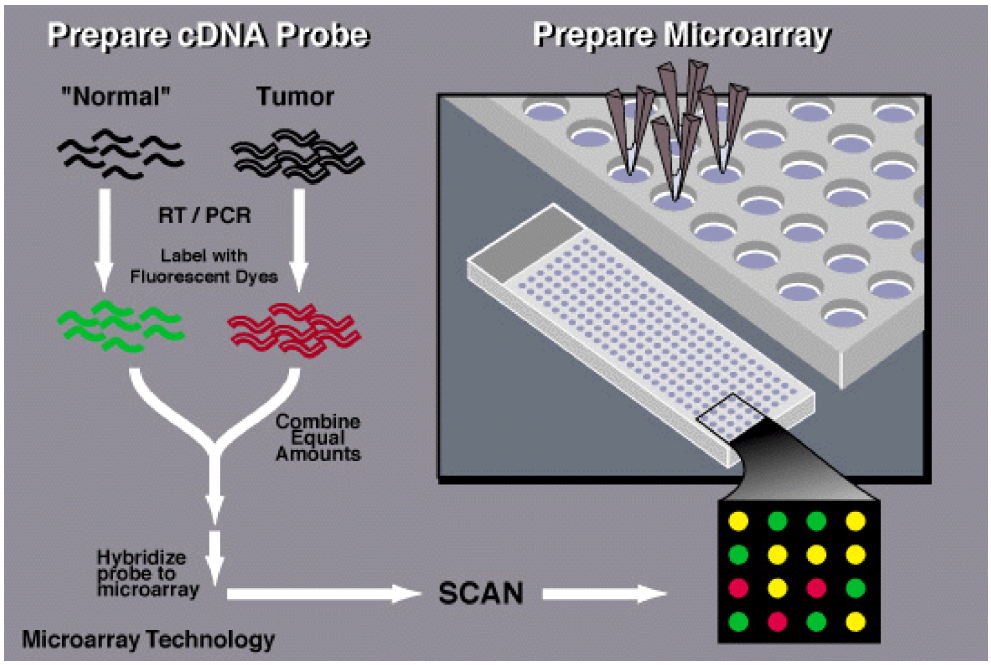

Figure 2: Preparation of cDNA probe and microarray

2.2 Quantification of Images

The images generated by scanner of microarray are the raw data. There is one image per array for a single channel microarray, while there are two images per array for two channel microarrays. The image intensity is scanned by detector at a high spatial resolution, where every probe spots are repre-

sented by several pixels. The intensity values of each probe are identified and these intensities are quantified to numeric values. Image quantification process involves following steps:

(i) Identification of position of sports on the array

(ii) For every spot on the array, pixel identification on the image

(iii) For every spot on the array, identifying pixels so that it may be used for background calculation

(iv) Computation of numeric value for the intensity of the spot, intensity of background image and quality control information.

There are several methods for segmentation and quantification which are available in software packages but they differ in their robustness. To a microarray project, quantification of image involves the transition of workflow from wet-lab procedure to computational (dry-lab). During the computation of numeric information at the microarray spot, image processing software provides a number of measures such as mean, median and standard deviation of signal and background, along with diameter and number of pixels. Among these measures, the most important measure is hybridization intensity for each spot that can be either mean or median of the pixel intensities. The second important information is signal standard deviation that helps in computation of coefficient of variation for spot as well as for the background. Once the spotted image and other statistics are computed then it is suggested that quality of the array and individual spots on the array are assessed because sometimes array may have a few spots as defected.

The obtained microarray gene expression data is presented in a tabular form, called gene expression matrix, as shown in Fig. 3. The first column specifies the identity of genes and first rows represents *samples* (replicas), *condition* or *time-series* observations. The element of the matrix gives expression values of gens under various conditions or samples.

2.3 Data Preprocessing and Normalization

Once the spotted images have been quantified to generate datasets, these datasets should be preprocessed before its analysis and interpretation. In this step, meaningful characteristics are extracted or enhanced and prepare the dataset for its analysis and interpretation. In data preprocessing step, generally two issues are addressed: i) to adjust background intensities, and ii) to transform data into scale suitable for analysis and interpretation. A

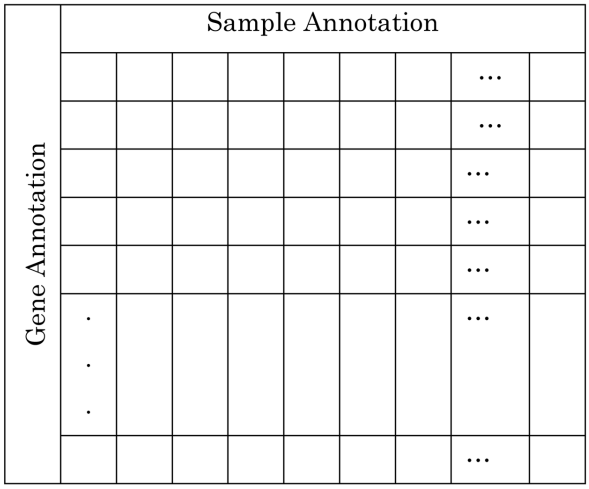

Figure 3: Preparation of cDNA probe and microarray

simple example of preprocessing microarray data is taking the Log of the raw intensity values. The main purpose of normalization is to ensure that variation in the expression values are because of biological differences between the mRNA samples and not because of experimental artifacts.

The adjustment of background intensities are needed because despite of washing done after hybridization in microarrays, there are chances of genes annealing in the background of the spot and during scanning time, it may give rise to background intensity. Another issue in the gene expression data which need to be addressed is the difference between the data generated by two microarray technologies (cDNA and oligonucleotide microarrays). The cDNA reports differences in gene expression, while oligonucleotide microarray report absolute expression values (Butte, 2002). Hence, same normalization techniques may not be applied to these different microarray technologies. In a given experiment, most genes do not changes their expression levels and if equal numbers of genes are upregulated and downregulated, then differential expression measurements might found to be normally distributed.

2.3.1 Normalization for single channel experiment

Suppose that one need to find out differentially expressed genes (DEGs) under various experimental conditions, such as normal sample against cancerous sample, control tissue versus treatment, etc. in a single channel

experiment. It is generally expected that gene expression values in both the conditions are more or less similar but it is seldom found in the reality. This variation is because of many factors including different arrays are used for each samples. Hence, it is natural that one would expect distribution of expression values would more or less similar. Therefore, it is necessary to remove variation between arrays and the methods to remove variation among arrays, and is called array normalization methods. There are several methods to make empirical distribution of expression values over all arrays. Some of the methods are:

(i) Normalization by mean

(ii) Median or Q2 normalization

(iii) Q3 normalization

(iv) Quantile normalization

In normalization by mean method, the expression values are transformed so that that mean of all the arrays is same. Median method transforms the expression values so that all arrays have median same as that of some reference array. The Q3 method is defined as similar to Q2. In Q3, third quartiles of the arrays are calculated to the third quartile of the mock array. Quantile normalization method is an extension of Q2 and Q3 normalization. This method is based on transforming each array specific distributions of intensities so that they all have similar values of quantiles.

2.3.2 Normalization for two channel experiment

In a two channel microarray experiment, two different samples are labeled by two fluorescent dyes R and G, hybridized on an array. The difference of intensities between two channels gives DEGs, provided that these variations are only because of biological functioning of the genes in different conditions. Here, to identify DEGs it is necessary to compare the intensity of R and G. These methods are applied on the Log of the ratio R/G.

3 Applications of Microarrays

Microarrays have been utilized in several biomedical problems including gene discovery, disease diagnosis, pharmacogenomics, and toxicology. For instance, microarrays can be used to identify disease genes by comparing

expression patterns of genes in disease versus normal cells sample. Similarly, it can also be used to identify possible abnormal gene expression and abnormal interaction between genes for a disease.

For a majority of applications, microarrays address four broad categories of problems (Xu, 2008):

(i) Gene selection/gene filtering or identification of differentially expressed genes (DEGs)

(ii) Finding natural groupings among genes, conditions or both (clustering)

(iii) Patient classification using gene expression

(iv) Finding regulatory relationships among given set of genes

3.1 Gene Selection or Identification of DEGs

An important purpose for monitoring expression level of genes is to identify those genes which are differentially expressed across two kinds of tissue samples or samples observed under two different experimental conditions. Set of genes differentially expressed over two different samples, i.e., normal and cancerous tissue, are expected to give clues about cancer mechanism. A large variety of methods exists for finding differentially expressed genes and most of these methods are based on statistical techniques, such as fold-change (Schena et al., 1995), t-test statistics (Peck and Devore, 2011; Drăghici, 2003), ANOVA (Kerr et al., 2000), rank product (Breitling et al., 2004), Significant Analysis of Microarray (Tusher et al., 2001), Random Variance Model (Wright and Simon, 2003), Limma (Smyth, 2004), and so on. Review on the various methods can be found in: (Pan, 2002; Jeffery et al., 2006).

Fold change is one of the simplest ad-hoc methods often used in microarray analysis. A fold change is a measure that describes how much expression level of a gene changes over two different samples (conditions) or groups. To calculate a fold change, the average of expression values for each probes are calculated across the samples in each group, and then ratios of these average are taken. The levels of fold change are observed and genes under or above a thresholds are selected. For example, fold change below 0.5 is considered as down-regulated and fold change above 2.0 is considered as up-regulated. The Rank Products method is based on the statement that an experiment examining for n genes in m replicas, will have probability to be ranked first of $1/nm$, if the list values were totally random. Hence, it is improbable that single gene to have top position in all the given replicas, if given gene was not expressed differentially. Then, genes can be sorted based on likelihood

of observing their rank product values (Jeffery et al., 2006). The two-sample t-test is widely used parametric hypothesis testing method for the identification of DEGs. The t-statistics gives a probability value (p-value) for each gene. A small p-value indicates that genes are differentially expressed under the hypothesis that there is no differential expression, which is not true. The t-statistic is calculated as the difference in the means over the standard deviation. Raza and Mishra (2012) proposed an anticlustering gene algorithm for the identification of genes as drug target, where they applied a combination of statistical techniques.

3.2 Clustering Genes, Samples or Both

Clustering is a means of analysing set of objects by grouping them into different clusters based on some similarity measures. Basically clustering is an unsupervised technique that groups the similar objects into clusters. Researchers can apply clustering techniques to cluster gene expression data. Hence, genes belonging to a particular cluster are supposed to share common properties. If gene expression profiles (genes) are clustered, one may discover set of genes co-regulated in a certain samples. Similarly, gene expression can be grouped by clustering its samples. Sample clustering is done when it is needed to identify subgroups of certain condition (for instance, disease). The third means of grouping the gene expression data is to cluster both rows (genes) as well as columns (samples), which are known as co-clustering or bi-clustering. This kind of clustering helps us to find groups of genes associated with group of samples (patient). Some of the most popularly used clustering techniques are k-means, hierarchical, SOM, fuzzy c-means, non-Euclidean relational fuzzy c-means.

Using Microarray gene expression data, distance between two expressed genes can be computed so that it can be known that whether genes are interrelated or not, and placed in same cluster. Euclidean distance, Manhattan distance and Pearson correlation distance are some of the commonly applied distance measures. In Raza (2014), four different clustering techniques viz., k-means, Hierarchical clustering, density based clustering and Euclidean method based clustering, have been applied on five different types of cancer gene expression data (lung cancer, prostate cancer, colon cancer, breast cancer and ovarian cancer). In all these five datasets, there are large numbers of genes compared to numbers of samples. Hence, for better learning on machine learning techniques and to avoid the curse of dimensionality problem, after data normalization, attribute reduction using t-test has been done at a significance level of 0.001. There is no single clustering algorithm

that can work well in all the situations. Selection of a particular clustering approach depends on the problem at hand and the dataset under study.

3.3 Patient Classification

Gene expression data can be used to train a classifier so that it can recognize a given condition (e.g. class label such as normal or cancerous). The advantage of this kind of classification is that once a classifier is trained with gene expression profiles to recognize a patient class, then it can recognized a class of unknown patient for which the classifier has not been trained. There are several supervised techniques available for patient classification, such as Bayesian networks, M5 model tree, k-nearest neighbourhood, Random forest, neural networks, and support vector machines.

In Raza and Hasan (2013), authors have done a comparatively evaluation of various machine learning techniques for their accuracy in class prediction of prostate cancer based on Microarray dataset. As per their evaluation, Bayes Net gave the best accuracy for prostate cancer class prediction with an accuracy of 94.11%. Bayes Net is followed by Navie Bayes with an accuracy of 91.17%. The objective of evaluating various machine learning techniques is to come up with the best technique in terms of prediction accuracy and to reveal a good procedure for meaningful attribute reduction. A similar kind of process may be used to classify other types of cancers. One of the biggest challenges is to develop a single universal classifier which would be capable of classifying all types of cancer gene expression data into meaningful number of classes.

3.4 Finding Regulatory Relationship among Genes

A gene regulatory network (GRN) is a network of interaction among genes, where node represents genes and interconnection between them represents their regulatory relationship. Today, one of the most exciting problems in systems biology research is to decipher how the genome controls the development of complex biological system. Microarrays have been widely used to find out new (unknown) regulatory mechanism. The discovery of GRN using gene expression data is known as reverse-engineering of GRN. The GRNs help in identifying the interactions between genes and provide fruitful information about the functional role of individual genes in a cellular system. They also help in diagnosing various diseases including cancer.

In the last several decades, many computational methods have been proposed to discover complex regulatory interactions among genes based

on microarray data. These techniques can be clubbed into different groups, such as Boolean networks (Liang et al., 1998; Akutsu et al., 1999; Shmulevich et al., 2002; Martin et al., 2007; Raza and Jaiswal, 2013; Raza and Parveen, 2013), Bayesian networks (Friedman et al., 2000; Husmeier, 2003), Petri nets (Koch et al., 2005; Remy et al., 2006), linear and non-linear ordinary differential equations (ODEs) (Chen et al., 1999; Tyson et al., 2002; De Jong and Page, 2008), machine learning approaches (Weaver et al., 1999; Kim et al., 2000; Vohradský, 2001; Keedwell et al., 2002; Huang et al., 2003; Tian and Burrage, 2003; Zhou et al., 2004; Xu et al., 2004; Hu et al., 2006; Jung and Cho, 2007; Xu et al., 2007a,b; Chiang and Chao, 2007; Lee and Yang, 2008; Datta et al., 2009; Zhang et al., 2009; Maraziotis et al., 2010; Ghazikhani et al., 2011; Liu et al., 2011; Kentzoglanakis and Poole, 2012; Noman et al., 2013), etc.

For review of the modeling techniques and the subject, refer to (De Jong, 2002; Wei et al., 2004; Schlitt and Brazma, 2007; Cho et al., 2007; Karlebach and Shamir, 2008; Swain et al., 2010; Sîrbu et al., 2010; Mitra et al., 2011; Raza and Parveen, 2013).

4 Artificial Intelligence Techniques and Microarray Analysis

Artificial Intelligence (AI) is an interdisciplinary field of study where the goal is to create intelligence by a machine or a computer program. Most of the researchers defines AI as the study and design of intelligent agents, where an intelligent agent is a system that can perceive given environment to take actions and maximize its probability of being success. John McCarthy coined this term in 1955 and defined AI as the science and engineering of making intelligent machine. AI has a vast domain of research including reasoning, knowledge, learning, natural language processing, perception, etc. Most popularly used AI techniques for solving real-life problems, including bioinformatics problems such as analysis of microarray array data to extract fruitful knowledge, are statistical methods and computational intelligence. Among the computational intelligence, artificial neural networks, fuzzy systems, evolutionary computations and many statistical tools are mostly applied AI based approach.

In this section, few computational intelligence approaches and their applications in Microarray analysis have been briefly described.

4.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) are massively parallel computing system which is inspired by biological system of neurons. It is collection of extremely large number of simple processing elements, called neurons, having many interconnections. These elements have inputs, which are multiplied by weights and then computed by a mathematical function, called activation function, regulating the activation of the neuron. A weight value w_{ij} is assigned to each connection and hence the net input to the neuron is the weighted sum of its n input signals x_i, $i = 1, 2, .., n$. Each neuron has an activation function f (generally a sigmoidal function), which is used to compute the neurons current activation ai, and output function g (generally identity function) which is used to compute value O_i. By adjusting the weights of neurons, the desired output can be obtained from the inputs. The process of adjusting the synaptic weights is known as learning or training process. The most widely applied training algorithm is Backpropagation algorithm.

The first neural network model was proposed by McCulloch and Pitts in 1993 and since then hundreds of different models have been developed (Gershenson, 2003). The differences in the neural networks might be in their architecture, activation function, and topology, training algorithm and accepted input and output values. On the basis of architecture (connection pattern), neural networks can be broadly clubbed into two groups (Jain et al., 1996): i) feedforward networks, where there is no loop and ii) recurrent networks, which contain loops because of feedback connections. A typical network from each category is shown in Fig. 4. Feedforward networks are memory-less, i.e., their output is independent of the previous network state but recurrent networks consider network feedback paths, which are with memory, i.e., the current output is dependent on the previous state of the network. Different network architecture needs appropriate training algorithm. The learning paradigms can be categorized as: supervised, unsupervised and hybrid (reinforcement).

Vohradsky (Vohradský, 2001) applied ANN to model gene regulation by assuming that the regulatory effect on gene expression of a particular gene can be expressed in the form of ANN. Each neuron in the neural network represents a gene and connectivity between them represents regulatory interactions. Here each layer of the ANN represents the level of gene expression at time t and output of a neuron at time $t + \Delta t$ can be determined from the expression at time t. The advantage the model is that it is continuous, uses a transfer function to transform the inputs to a shape close to those observed in natural processes and does not use artificial elements.

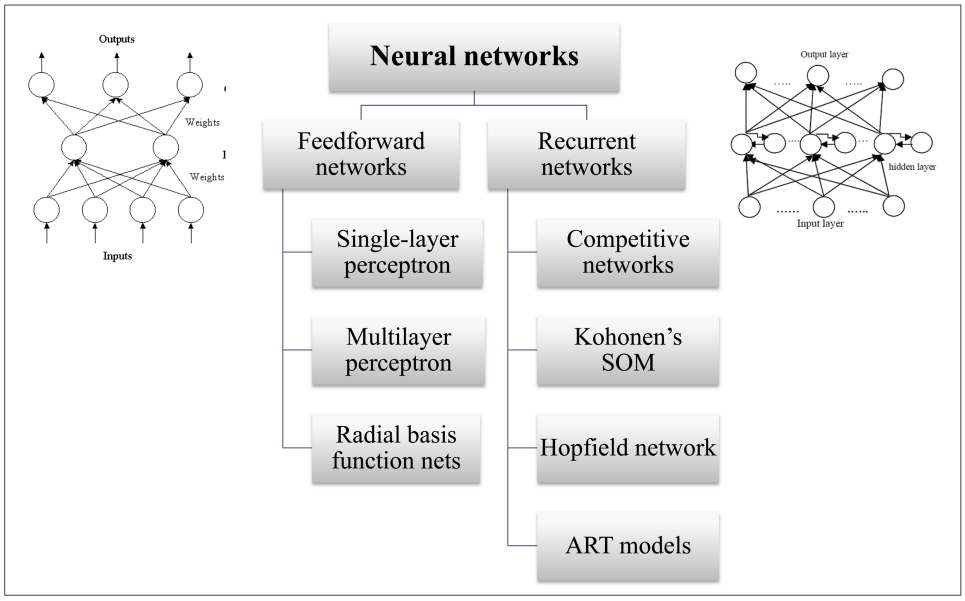

Figure 4: A taxonomy of feed-forward and recurrent network architectures

Keedwell and his collaborators (Keedwell et al., 2002) also applied ANN in the purest form for the reconstruction of GRNs from microarray data. The architecture of the neural network was quite simple when dealing with Boolean networks. Hu et al. (2006) has proposed a general recurrent neural network (RNN) model for the reverse-engineering of GRNs and to learn their parameters. RNN has been deployed due to its capability to deal with complex temporal behaviour of genetic networks. In this model, time delay between the output of a gene and its effect on another gene has been incorporated. A more recent work by Noman and his colleagues (Noman et al., 2013) proposed a decoupled-RNN model of GRN. Here, decoupled means dividing the estimation problem of parameters for the complete network into several sub-problems, each of which estimate parameters associated with single gene. This decoupled approach decreases the dimensionality problem and makes the reconstruction of large network feasible. In one of our work (Raza et al., 2014), we also proposed a RNN based hybrid model of GRN that uses extended Kalman filter to estimate and update synaptic weights using Backpropagation Through Time (BPTT) training algorithm.

The ANN approach of GRN inference works well for small size network, i.e., a network of up to 100 genes. This is because of less number of available

samples in a Microarray experiment. As the size of the network grows, the number of unknown parameters (interactions) also grows, and that requires a very large number of Microarray samples, which is rarely available in Microarray data.

4.2 Fuzzy System

Fuzzy logic is based on the concept of partial truth, i.e., truth values between *completely true* and *completely false*. For example, using fuzzy logic, propositions can be denoted with degrees of truthfulness and falsehood with the help of a membership function. L.A. Zadeh (Zadeh, 1996) was the first who introduced the concept of fuzzy logic to represent vagueness in linguistics and implement and express human knowledge and inference capability in a natural way. In broad sense, fuzzy logic is an extension of multivalue logic. In specific sense, fuzzy logic is a logic system which can be used to model approximate reasoning (Cao, 2006). Fuzzy logic has been proved to be useful in expert system and other artificial intelligence applications.

A fuzzy system generally consists of three parts:

(i) fuzzy input and output variables, and their fuzzy values,

(ii) fuzzy rules, such as Zadeh-Mamdani's fuzzy rules, Takagi-Sugeno's fuzzy rules, gradual fuzzy rules and recurrent fuzzy rules,

(iii) fuzzy inference methods, which may include fuzzification and defuzzification.

The biological systems behave in a fuzzy manner. Fuzzy logic provides a mathematical framework for modeling and describing biological systems. Literature reports that fuzzy logic has been successfully used for the analysis of microarray data due to its capability to represent non-linear systems, its friendly language to incorporate and edit domain knowledge in the form of fuzzy rules (Raza and Parveen, 2013). Woolf and Wang (Woolf and Wang, 2000) proposed a fuzzy logic based algorithm for analysing gene expression data. The proposed fuzzy model was designed to extract gene triplets (activators, repressors, targets) in yeast gene expression data. The model took 200 hours to analyse the relationships between 1,898 genes on an 8-processor SGI Origin 2000 system. Later, Ressom and his colleagues (Ressom et al., 2003) extended and improved the work of Woolf and Wang (Woolf and Wang, 2000) in terms of reducing computation time and generalizing the model to accommodate co-activator and co-repressors, in addition to activators, repressors and targets. Reduction in computation time is achieved by applying clustering as a pre-processing step. The improved algorithm achieves a reduction of 50% computation time. After 3 years, Ram

and his colleagues (Ram et al., 2006) also improved the Woolf and Wangs fuzzy logic model to predict changes in gene expression values and extracted causal relationship between genes. They have improved searching for activator/repressor regulatory relationship between gene triplets in the microarray data. A pre-processing technique for the fuzzy model has also been proposed to remove redundant data present that makes the model faster. Sun and colleagues (Sun et al., 2010) applied dynamic fuzzy approach by incorporating structural knowledge to model gene regulatory networks using microarray gene expression data. This technique infers gene interactions in the form of fuzzy rules and able to reveal biological relationships among genes and their products. The distinguishing feature of this model is that (i) prior structural knowledge on GRN can be incorporated for the purpose of faster convergence of the identification process and (ii) non-linear dynamic property of the GRN can be well captured for the better prediction.

As discussed in previous section, clustering (grouping) Microarray data is also one aspect of analysing it that gives clues about set of co-regulated genes. Fuzzy based clustering algorithm, called fuzzy c-means (FCM) was first introduced by Dunn in 1973 (Dunn, 1973) but implemented by Bezdek (Bezdek, 1981). The FCM has now become most popular fuzzy clustering algorithm and considered as robust to scale the dataset (Wang et al., 2008). A major problem with FCM algorithm for clustering microarray data is the selection of the fuzziness parameter m. The work of Dembélé and Kastner (2003) shows that the commonly used value $m = 2$ is not always appropriate. The optimal value for m varies from one dataset to another. They also proposed an empirical method to estimate an adequate value for m, based on the distribution of distances between genes in the given dataset.

In addition to fuzzy-clustering hybrid, fuzzy logic has been hybridized other computational intelligence techniques, including fuzzy and neural network hybrid (called neuro-fuzzy) and fuzzy and genetic algorithm (called fuzzy-genetic). Neuro-fuzzy and fuzzy-genetic hybrid have been successfully applied for GRN inference using Microarray data. The review of the application of neuro-fuzzy and neuro-genetic for GRN inference can be found in (Mitra et al., 2011; Raza and Parveen, 2013).

4.3 Evolutionary Computation

Evolutionary computing is a collection of problem-solving techniques based on principles of biological evolution. The functional analogy of evolutionary computing is the natural evolution that relates to a particular kind of problem solving grounded on trial-and-error process. Natural selection means

that we have a population of individuals that strive for survival and reproduction. The fitness of these individuals determines how well they succeed in achieving their goals, i.e., presenting their chance for survival and reproduction. Charles Darwin formulated the theory of natural evolution. Over several generations, biological organism evolves according to the principle of natural selection like *survival of the fittest*. The history of evolutionary computing goes back to 1940s. After many decades of research in this area, researchers came up with many evolutionary computing techniques such as evolutionary programming, evolution strategies, genetic algorithm (Holland, 1975) and generic programming (Koza, 1992).

Genetic algorithms (GAs) are basically optimization techniques inspired by Darwins theory of evolution. In fact, it is a search algorithm based on the mechanism of natural selection and survival for the fittest. Here, searching in a population is done from a single point and competitive selection is done in each iterations. The solutions having high fitness are recombined with other solutions and then mutated by changing the single element of the solution. The purpose of genetic operators, such as crossover and mutation, are to generate new population of solutions for the next generation. Genetic algorithms belong to probabilistic algorithms and are different from random search algorithms because former combines elements of directed and stochastic search. Due to this reason, GAs are found to be more robust than directed search methods. Further, GAs maintain a population of potential solutions; on the other hand, other search techniques process a single point of search space (Raza and Parveen, 2012).

Noman and Iba (2005) applied decoupled S-system approach for the inference of effective kinetic parameters from time series gene expression data and applied Trigonometric Differential Evolution (TDE) for the optimization and captures the dynamics of gene expression. Later, Chowdhury and Chetty (2011) extended the work of Noman and Iba (2005) and applied GA for scoring the networks several useful features, such as a Prediction Initialization (PI) algorithm to initialize the individuals, a Flip Operation (FO) for matching values. A refinement algorithm for optimizing sensitivity and specificity of inferred networks was also proposed. Xu and colleagues (Xu et al., 2009) proposed genetic programming based method for the analysis of microarray datasets, where genetic programming performs classification and feature selection simultaneously. Maulik (2011) studied the performance of three most commonly used computational techniques such as genetic algorithm, simulated annealing and differential evolution for developing fuzzy clusters of gene expression data. Clustering is an unsupervised analysis approach for grouping co-expressed genes together. To improve results of

clustering, support vector machine (SVM) has been utilized. A review of application of evolutionary computation for Microarray analysis can be found in Sîrbu et al. (2010); Mitra et al. (2011); Raza and Parveen (2012).

4.4 Other AI Based Methods

Machine learning algorithms, like ANN, are also used to predict interactions between genes of a GRN using Microarray data. But, these algorithms are so complex and work like a black-box. Black-box model means what is happening inside the algorithm is hidden (Sîrbu et al., 2010). On the other hand, nature-inspired algorithms, in comparison to other algorithms, are simpler in nature and they have been found to be applied in various biological problems from simplest like alignment of sequences to the complex like protein structure prediction (Pal et al., 2006). One such type of nature-inspired algorithm is Genetic algorithm which has already been discussed in the previous section.

In the last two decades, several nature-inspired metaheuristic optimization algorithms have been proposed and successfully applied in many optimization problems, including microarray analysis. Fister and colleagues (Fister Jr et al., 2013) have done a survey of nature-inspired optimization algorithm and listed $\tilde{7}5$ nature-inspired algorithms proposed by different researchers, and classified these algorithms into four groups: Swarm intelligence based , Bio-inspired based, Physics-based and Chemistry-based, and Others. algorithms. Ant colony optimization (ACO) is one of the nature-inspired swarm-based optimization algorithm proposed by Marco Dorigo in 1992 (Dorigo, 1992) in his PhD thesis. ACO is a metaheuristic optimization technique where a set of artificial ants search for optimal solutions in a given optimization problem. Ants use pheromones laid by the other ants as footmarks to follow. Hence, ant reaches the food source by the shortest path using knowledge gained by the other ants. This algorithm can be used for optimization problems, including gene interaction network optimization. ACO has been applied to several bioinformatics problems including sequence alignment, drug designing, 2D protein folding and biological network optimization. Raza and Kohli (2015) applied ACO algorithm for inferring highly correlated key gene interactions in a GRN that plays an important role in identifying biomarkers for disease which further helps in drug design. The limitation of proposed algorithm by Raza and Kohli (2015) is that it can find out a total number of interactions equal to total number of genes.

PSO has been applied for clustering and feature (genes) selection in microarray data. A k-means clustering based upon PSO has been proposed

for microarray data clustering by Deng et al. (2005). The algorithm discovers clusters in microarray data without having any prior knowledge of feasible number of clusters. Chuang et al. (2009) applied Binary PSO for feature selection in microarray data. Sahu and Mishra (2012) also proposed a PSO based feature selection algorithm for cancer microarray data. For the selection of efficient genes from thousands of genes, Chen et al. (2014) proposed an approach utilizing PSO combined with a decision tree classifier. For the biclustering of microarray data, a comparative study on three nature-inspired algorithms, such as PSO, Shuffled Frog Leaping (SFL) and Cuckoo Search (CS) algorithms, have been done on benchmark gene expression dataset by Balamurugan et al. (2014). The result reports that CS outperforms PSO and SFL for 3 out of 4 datasets. The classification accuracy of simple statistical learning techniques can be enhanced when nature-inspired algorithm are applied for the feature selection. One such study has been carried out by Gunavathi and Premalatha (2014). They performed a comparative analysis of swarm intelligence techniques, such as PSO, cuckoo search (CS), SFL, and SFL with Lévy flight (SFLLF), for feature selection in cancer classification. The k-nearest neighbour (kNN) classifier is applied to classify the samples. The result shows that k-NN classifier through SFLLF feature selection method outperform PSO, CS, and SFL. Sometimes, DEGs techniques are used for gene selection/filtering or dimension reduction in microarray data where we have a large number of genes (features). The dimension reduction is a preprocessing step whenever we use a machine learning technique for training with gene expression datasets where number of gene are larger than the available samples (generally known as *curse of dimensionality problem*).

Due to advancement in data mining algorithms and tools, it is a keen interest of the researchers to apply these tools to identify patterns of interest in the gene expression data. Association rule mining is one of the most widely used data mining technique that have been applied for gene expression mining by a number of researchers Creighton and Hanash (2003). Association rules mining may discover biologically useful associations between genes, or between different biological conditions using microarray gene expression data. An association rules are written in the form $A_1 \rightarrow A_2$, where A_1 and A_2 are disjoint sets of data items. The set A_2 is likely to occur whenever the set A_1 occurs. Here, the data items may present either highly expressed or repressed genes, or any other facts that state the cellular environment of genes (e.g. diagnosis of a disease samples) (Raza, 2015). Formal Concept Analysis (FCA), introduced by R. Wille in early 1980s, is another data mining technique based on lattice theory. It has been widely used for the

analysis of binary relational data. Like other computational techniques, FCA has also been applied in microarray analysis, gene expression mining, gene expression clustering, finding genes in gene regulatory networks, and so on. A review FAC for the analysis and knowledge discovery from gene expression and other biological data can be found in Raza (2015).

5 Conclusions, Discussions and Future Challenges

The bioinformatics is an interdisciplinary area of study where one of the objectives is to deal with the analysis and interpretation of large sets of data generated from various large-scale biological experiments, including Microarrays. Microarray technology is one of the powerful tools used to measure genome wide expression levels of genes. As microarrays technologies have become more prevalent, the challenges associated with collecting, managing, and analyzing the data from each experiment have essentially increased. With the help of these technologies, researchers can find out answer of some challenging questions like: (i) what are the functions of different genes? (ii) In what cellular processes do they participate? (iii) how genes are regulated? (iv) how genes and its products (proteins) do interact, and what are these interaction networks? (v) how expression level of genes differs in different cell types and states? (vi) how expressions of genes are affected by various disease or drug treatments?

In this chapter, four broad categories of problems have been tackled for the analysis of Microarrays:

(i) *Identification of differentially expressed genes:* It helps us in the selection of few relevant genes and elimination of irrelevant genes for further study. It also solves the dimensionality problem of machine learning techniques by filtering differentially expressed genes over various samples and training a classifier with the selected number of genes only.

(ii) *Cancer classification using gene expression:* Classification of patient based on gene expression profile is another important issue for the analysis of microarray data. The application of AI-based techniques for cancer classification based microarray data has been discussed.

(iii) *Clustering genes, conditions or both:* Another important aspect of analyzing microarray data is finding natural groupings among genes, which can be done using clustering techniques. Clustering is an unsupervised

learning technique that plays a vital role in providing a class label to unlabeled data and it can be used to identify set of co-regulated genes.

(iv) *Inferring gene interaction network:* The gene interaction network plays an important role in identifying root-cause of various diseases. Inferring gene interaction network from gene expression profiles is one of other aspect of analyzing microarray data. The application of AI-based techniques for GRN inference has been covered in length and various resources for further study has been listed.

Future Challenges

Microarray technology is a high-throughput experimental approach that measures the genome-wide expression of genes and data are produced in large-scale. Hence, analysing these data to infer useful information is big challenge. Some of the future directions for the analysis of microarray data are as follows:

(i) One of the main drawbacks of microarray technology is that data generated by these experiments contain noises and are not much reliable. Hence, before the data is analysed, we must apply sophisticated noise removal and data normalization technique.

(ii) Application of machine learning techniques in genome-wide analysis of microarray data creates the problem of dimensionality. Hence, some techniques are required to identify differentially expressed genes (DEGs). Statistical techniques, such as fold change, t-test, ANOVA, etc. dominates in the identification of DEGs. Hence, it is needed to explore the application of computational intelligence to tackle the problem of DEGs.

(iii) Another biggest challenge is to develop a single classifier which is best suitable for classification of all types of cancer gene expression data into meaningful number of classes. Nature inspired optimization techniques such as Ant Colony Optimization (ACO) (Dorigo, 1992), Artificial Bee Colony optimization (ABC) Karaboga (2005), Cuckoo Search Yang and Deb (2009), Particle Swarm Optimization (PSO) Kennedy and Eberhart (1995), Spider Monkey Algorithm (Bansal et al., 2014) and so on, are successfully being used in many challenging problems. In the future work, one can hybridize these nature inspired optimization techniques with different classifiers for better classification accuracy.

(iv) For the gene clustering problem, one can apply fuzzy based clustering techniques (such as Fuzzy C-Means) to group genes or patient or both. Even, ranked based classification techniques can be applied.

(v) Inference of gene interaction networks using gene expression profile is another open area where computational intelligence can be applied to identify interactions among given set of genes. Hybrid algorithms (for example, fusion of neural networks, genetic algorithms and/or fuzzy logic and other nature-inspired algorithms) can be applied for the said purpose.

Acknowledgements

The author acknowledges the funding received from University Grants Commission, Govt. of India through research grant 42-1019/2013(SR).

References

Akutsu, T., Miyano, S., Kuhara, S., et al. (1999). Identification of genetic networks from a small number of gene expression patterns under the boolean network model. In *Pacific Symposium on Biocomputing*, volume 4, pages 17–28. World Scientific.

Babu, M. M. (2004). Introduction to microarray data analysis. *Computational Genomics: Theory and Application*, pages 225–249.

Balamurugan, R., Natarajan, A., and Premalatha, K. (2014). Comparative study on swarm intelligence techniques for biclustering of microarray gene expression data. *International journal of computer, control, quantum and information engineering*, 8(2).

Bansal, J. C., Sharma, H., Jadon, S. S., and Clerc, M. (2014). Spider monkey optimization algorithm for numerical optimization. *Memetic computing*, 6(1):31–47.

Bassett, D. E., Eisen, M. B., and Boguski, M. S. (1999). Gene expression informaticsit's all in your mine. *Nature genetics*, 21:51–55.

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. Plenum Press, New York.

Brazma, A. and Vilo, J. (2000). Gene expression data analysis. *FEBS letters*, 480(1):17–24.

Breitling, R., Armengaud, P., Amtmann, A., and Herzyk, P. (2004). Rank products: a simple, yet powerful, new method to detect differentially regulated genes in replicated microarray experiments. *FEBS letters*, 573(1):83–92.

Butte, A. (2002). The use and analysis of microarray data. *Nature reviews drug discovery*, 1(12):951–960.

Cao, Y. (2006). *Fuzzy Logic Network Theory with Applications to Gene Regulatory Networks*. PhD thesis, Duke University.

Chen, K.-H., Wang, K.-J., Tsai, M.-L., Wang, K.-M., Adrian, A. M., Cheng, W.-C., Yang, T.-S., Teng, N.-C., Tan, K.-P., and Chang, K.-S. (2014). Gene selection for cancer identification: a decision tree model empowered by particle swarm optimization algorithm. *BMC bioinformatics*, 15(1):49.

Chen, T., He, H. L., Church, G. M., et al. (1999). Modeling gene expression with differential equations. In *Pacific symposium on biocomputing*, volume 4, page 4. World Scientific.

Chiang, J.-H. and Chao, S.-Y. (2007). Modeling human cancer-related regulatory modules by ga-rnn hybrid algorithms. *BMC Bioinformatics*, 8(1):91.

Cho, K.-H., Choo, S.-M., Jung, S., Kim, J.-R., Choi, H.-S., and Kim, J. (2007). Reverse engineering of gene regulatory networks. *Systems Biology, IET*, 1(3):149–163.

Chowdhury, A. R. and Chetty, M. (2011). An improved method to infer gene regulatory network using s-system. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 1012–1019. IEEE.

Chuang, L.-Y., Yang, C.-H., and Yang, C.-H. (2009). Tabu search and binary particle swarm optimization for feature selection using microarray data. *Journal of computational biology*, 16(12):1689–1703.

Creighton, C. and Hanash, S. (2003). Mining gene expression databases for association rules. *Bioinformatics*, 19(1):79–86.

Datta, D., Choudhuri, S. S., Konar, A., Nagar, A., and Das, S. (2009). A recurrent fuzzy neural model of a gene regulatory network for knowledge extraction using differential evolution. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pages 2900–2906. IEEE.

De Jong, H. (2002). Modeling and simulation of genetic regulatory systems: a literature review. *Journal of computational biology*, 9(1):67–103.

De Jong, H. and Page, M. (2008). Search for steady states of piecewise-linear differential equation models of genetic regulatory networks. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 5(2):208–222.

Debouck, C. and Goodfellow, P. N. (1999). Dna microarrays in drug discovery and development. *Nature genetics*, 21:48–50.

Dembélé, D. and Kastner, P. (2003). Fuzzy c-means method for clustering microarray data. *Bioinformatics*, 19(8):973–980.

Deng, X., Xu, J., Hui, J., and Wang, C. (2009). Probability fold change: A robust computational approach for identifying differentially expressed gene lists. *computer methods and programs in biomedicine*, 93(2):124–139.

Deng, Y., Kayarat, D., Elasri, M. O., and Brown, S. J. (2005). Microarray data clustering using particle swarm optimization k-means algorithm. *Proc. 8th JCIS*, pages 1730–1734.

Dorigo, M. (1992). Optimization, learning and natural algorithms. *Ph. D. Thesis, Politecnico di Milano, Italy.*

Drăghici, S. (2003). *Data analysis tools for DNA microarrays.* CRC Press.

Dudoit, S., Yang, Y. H., Callow, M. J., and Speed, T. P. (2002). Statistical methods for identifying differentially expressed genes in replicated cdna microarray experiments. *Statistica sinica*, 12(1):111–140.

Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters.

Fister Jr, I., Yang, X.-S., Fister, I., Brest, J., and Fister, D. (2013). A brief review of nature-inspired algorithms for optimization. *arXiv preprint arXiv:1307.4186.*

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000). Using bayesian networks to analyze expression data. *Journal of computational biology*, 7(3-4):601–620.

Gershenson, C. (2003). Artificial neural networks for beginners. *arXiv preprint cs/0308031*.

Ghazikhani, A., Akbarzadeh, T. M. R., and Monsefi, R. (2011). Genetic regulatory network inference using recurrent neural networks trained by a multi agent system. In *Computer and Knowledge Engineering (ICCKE), 2011 1st International eConference on*, pages 95–99. IEEE.

Gunavathi, C. and Premalatha, K. (2014). A comparative analysis of swarm intelligence techniques for feature selection in cancer classification. *The Scientific World Journal*, 2014.

Holland, J. H. (1975). Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.

Hood, L. (2003). Systems biology: integrating technology, biology, and computation. *Mechanisms of ageing and development*, 124(1):9–16.

Hu, X., Maglia, A., Wunsch, D. C., et al. (2006). A general recurrent neural network approach to model genetic regulatory networks. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 4735–4738. IEEE.

Huang, J., Shimizu, H., and Shioya, S. (2003). Clustering gene expression pattern and extracting relationship in gene network based on artificial neural networks. *Journal of bioscience and bioengineering*, 96(5):421–428.

Husmeier, D. (2003). Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic bayesian networks. *Bioinformatics*, 19(17):2271–2282.

Jain, A. K., Mao, J., and Mohiuddin, K. (1996). Artificial neural networks: A tutorial. *Computer*, (3):31–44.

Jeffery, I. B., Higgins, D. G., and Culhane, A. C. (2006). Comparison and evaluation of methods for generating differentially expressed gene lists from microarray data. *BMC bioinformatics*, 7(1):359.

Jung, S. H. and Cho, K.-H. (2007). Reconstruction of gene regulatory networks by neuro-fuzzy inference systems. In *Frontiers in the Convergence of Bioscience and Information Technologies, 2007. FBIT 2007*, pages 32–37. IEEE.

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical report, Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.

Karlebach, G. and Shamir, R. (2008). Modelling and analysis of gene regulatory networks. *Nature Reviews Molecular Cell Biology*, 9(10):770–780.

Keedwell, E., Narayanan, A., and Savic, D. (2002). Modelling gene regulatory data using artificial neural networks. In *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on*, volume 1, pages 183–188. IEEE.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948 vol.4.

Kentzoglanakis, K. and Poole, M. (2012). A swarm intelligence framework for reconstructing gene networks: searching for biologically plausible architectures. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 9(2):358–371.

Kerr, M. K., Martin, M., and Churchill, G. A. (2000). Analysis of variance for gene expression microarray data. *Journal of computational biology*, 7(6):819–837.

Kim, S., Dougherty, E. R., Chen, Y., Sivakumar, K., Meltzer, P., Trent, J. M., and Bittner, M. (2000). Multivariate measurement of gene expression relationships. *Genomics*, 67(2):201–209.

Kitano, H. (2002a). Computational systems biology. *Nature*, 420(6912):206–210.

Kitano, H. (2002b). Systems biology: a brief overview. *Science*, 295(5560):1662–1664.

Koch, I., Schüler, M., and Heiner, M. (2005). Stepp-search tool for exploration of petri net paths: A new tool for petri net-based path analysis in biochemical networks. *In silico biology*, 5(2):129–138.

Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection*, volume 1. MIT press.

Lee, C.-P., Leu, Y., and Yang, W.-N. (2012). Constructing gene regulatory networks from microarray data using ga/pso with dtw. *Applied Soft Computing*, 12(3):1115–1124.

Lee, W.-P. and Yang, K.-C. (2008). A clustering-based approach for inferring recurrent neural networks as gene regulatory networks. *Neurocomputing*, 71(4):600–610.

Liang, S., Fuhrman, S., Somogyi, R., et al. (1998). Reveal, a general reverse engineering algorithm for inference of genetic network architectures. In *Pacific symposium on biocomputing*, volume 3, pages 18–29.

Liu, G., Liu, L., Liu, C., Zheng, M., Su, L., and Zhou, C. (2011). Combination of neuro-fuzzy network models with biological knowledge for reconstructing gene regulatory networks. *Journal of Bionic Engineering*, 8(1):98–106.

Maraziotis, I. A., Dragomir, A., and Thanos, D. (2010). Gene regulatory networks modelling using a dynamic evolutionary hybrid. *BMC bioinformatics*, 11(1):140.

Martin, S., Zhang, Z., Martino, A., and Faulon, J.-L. (2007). Boolean dynamics of genetic regulatory networks inferred from microarray time series data. *Bioinformatics*, 23(7):866–874.

Maulik, U. (2011). Analysis of gene microarray data in a soft computing framework. *Applied Soft Computing*, 11(6):4152–4160.

Mitra, S., Das, R., and Hayashi, Y. (2011). Genetic networks and soft computing. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 8(1):94–107.

Noman, N. and Iba, H. (2005). Reverse engineering genetic networks using evolutionary computation. *Genome Informatics*, 16(2):205–214.

Noman, N., Palafox, L., and Iba, H. (2013). Reconstruction of gene regulatory networks from gene expression data using decoupled recurrent neural network model. In *Natural Computing and Beyond*, pages 93–103. Springer.

Pal, S. K., Bandyopadhyay, S., and Ray, S. S. (2006). Evolutionary computation in bioinformatics: A review. *Systems, man, and cybernetics, Part c: Applications and reviews, IEEE transactions on*, 36(5):601–615.

Pan, W. (2002). A comparative review of statistical methods for discovering differentially expressed genes in replicated microarray experiments. *Bioinformatics*, 18(4):546–554.

Peck, R. and Devore, J. (2011). *Statistics: The exploration & analysis of data*. Cengage Learning.

Ram, R., Chetty, M., Dix, T., et al. (2006). Fuzzy model for gene regulatory network. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 1450–1455. IEEE.

Raza, K. (2014). Clustering analysis of cancerous microarray data. *Journal of Chemical and Pharmaceutical Research*, 6(9):488–493.

Raza, K. (2015). Formal concept analysis for knowledge discovery from biological data. *arXiv preprint arXiv:1506.00366*.

Raza, K., Alam, M., and Parveen, R. (2014). Recurrent neural network based hybrid model of gene regulatory network. *arXiv preprint arXiv:1408.5405*.

Raza, K. and Hasan, A. N. (2013). A comprehensive evaluation of machine learning techniques for cancer class prediction based on microarray data. *arXiv preprint arXiv:1307.7050*.

Raza, K. and Jaiswal, R. (2013). Reconstruction and analysis of cancer-specific gene regulatory networks from gene expression profiles. *International Journal on Bioinformatics & Biosciences*, 3(2):25–34.

Raza, K. and Kohli, M. (2015). Ant colony optimization for inferring key gene interactions. In *9th INDIACom-2015, 2nd International Conference on Computing for Sustainable Global Development*, pages 1242–1246.

Raza, K. and Mishra, A. (2012). A novel anticlustering filtering algorithm for the prediction of genes as a drug target. *American journal of biomedical engineering*, 2(5):206–211.

Raza, K. and Parveen, R. (2012). Evolutionary algorithms in genetic regulatory networks model. *Journal of Advanced Bioinformatics Applications and Research*, 3(1):271280.

Raza, K. and Parveen, R. (2013). Soft computing approach for modeling genetic regulatory networks. In *Advances in Computing and Information Technology*, pages 1–11. Springer.

Remy, E., Ruet, P., Mendoza, L., Thieffry, D., and Chaouiya, C. (2006). From logical regulatory graphs to standard petri nets: Dynamical roles and functionality of feedback circuits. In *Transactions on Computational Systems Biology VII*, pages 56–72. Springer.

Ressom, H., Wang, D., Varghese, R. S., and Reynolds, R. (2003). Fuzzy logic-based gene regulatory network. In *Fuzzy Systems, 2003. FUZZ'03. The 12th IEEE International Conference on*, volume 2, pages 1210–1215. IEEE.

Sahu, B. and Mishra, D. (2012). A novel feature selection algorithm using particle swarm optimization for cancer microarray data. *Procedia Engineering*, 38:27–31.

Schena, M., Shalon, D., Davis, R. W., and Brown, P. O. (1995). Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270(5235):467–470.

Schlitt, T. and Brazma, A. (2007). Current approaches to gene regulatory network modelling. *BMC bioinformatics*, 8(Suppl 6):S9.

Shmulevich, I., Dougherty, E. R., Kim, S., and Zhang, W. (2002). Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261–274.

Sîrbu, A., Ruskin, H. J., and Crane, M. (2010). Comparison of evolutionary algorithms in gene regulatory network model inference. *BMC bioinformatics*, 11(1):59.

Smyth, G. (2004). Statistical applications in genetics and molecular biology. *Linear models and empirical Bayes methods for assessing differential expression in microarray experiments*.

Sun, Y., Feng, G., and Cao, J. (2010). A new approach to dynamic fuzzy modeling of genetic regulatory networks. *NanoBioscience, IEEE Transactions on*, 9(4):263–272.

Swain, M. T., Mandel, J. J., and Dubitzky, W. (2010). Comparative study of three commonly used continuous deterministic methods for modeling gene regulation networks. *BMC bioinformatics*, 11(1):459.

Tian, T. and Burrage, K. (2003). Stochastic neural network models for gene regulatory networks. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, volume 1, pages 162–169. IEEE.

Tusher, V. G., Tibshirani, R., and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences*, 98(9):5116–5121.

Tyson, J. J., Csikasz-Nagy, A., and Novak, B. (2002). The dynamics of cell cycle regulation. *Bioessays*, 24(12):1095–1109.

Vohradský, J. (2001). Neural network model of gene expression. *The FASEB Journal*, 15(3):846–854.

Wang, F., Pan, D., and Ding, J. (2008). A new approach combined fuzzy clustering and bayesian networks for modeling gene regulatory networks. In *BioMedical Engineering and Informatics, 2008. BMEI 2008. International Conference on*, volume 1, pages 29–33. IEEE.

Weaver, D. C., Workman, C. T., Stormo, G. D., et al. (1999). Modeling regulatory networks with weight matrices. In *Pacific symposium on biocomputing*, volume 4, pages 112–123. World Scientific.

Wei, G., Liu, D., and Liang, C. (2004). Charting gene regulatory networks: strategies, challenges and perspectives. *Biochem. J*, 381:1–12.

Woolf, P. J. and Wang, Y. (2000). A fuzzy logic approach to analyzing gene expression data. *Physiological Genomics*, 3(1):9–15.

Wright, G. W. and Simon, R. M. (2003). A random variance model for detection of differential gene expression in small microarray experiments. *Bioinformatics*, 19(18):2448–2455.

Xu, C.-G., Liu, K.-H., and Huang, D.-S. (2009). The analysis of microarray datasets using a genetic programming. In *Computational Intelligence in Bioinformatics and Computational Biology, 2009. CIBCB'09. IEEE Symposium on*, pages 176–181. IEEE.

Xu, D. (2008). *Applications of fuzzy logic in bioinformatics*, volume 9. Imperial College Press.

Xu, R., Hu, X., Wunsch, D. C., et al. (2004). Inference of genetic regulatory networks with recurrent neural network models. In *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE*, volume 2, pages 2905–2908. IEEE.

Xu, R., Venayagamoorthy, G. K., and Wunsch, D. C. (2007a). Modeling of gene regulatory networks with hybrid differential evolution and particle swarm optimization. *Neural Networks*, 20(8):917–927.

Xu, R., Wunsch II, D., and Frank, R. (2007b). Inference of genetic regulatory networks with recurrent neural network models using particle swarm optimization. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 4(4):681–692.

Yang, X.-S. and Deb, S. (2009). Cuckoo search via lévy flights. In *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, pages 210–214. IEEE.

Zadeh, L. A. (1996). Fuzzy logic= computing with words. *Fuzzy Systems, IEEE Transactions on*, 4(2):103–111.

Zhang, Y., Xuan, J., de los Reyes, B. G., Clarke, R., and Ressom, H. W. (2009). Reverse engineering module networks by pso-rnn hybrid modeling. *BMC genomics*, 10(Suppl 1):S15.

Zhou, X., Wang, X., Pal, R., Ivanov, I., Bittner, M., and Dougherty, E. R. (2004). A bayesian connectivity-based approach to constructing probabilistic gene regulatory networks. *Bioinformatics*, 20(17):2918–2927.