1

Computational Intelligence in Solving Bioinformatics Problems: Reviews, Perspectives, and Challenges

Aboul-Ella Hassanien[1,2], Mariofanna G. Milanova[3], Tomasz G. Smolinski[4], and Ajith Abraham[5]

[1] Information Technology Department, FCI, Cairo University
5 Ahamed Zewal Street, Orman, Giza, Egypt
[2] Information System Department, CBA, Kuwait University, Kuwait
a.hassanien@fci-cu.edu.eg, abo@cba.edu.kw
[3] Computer Science Department, University of Arkansas at Little Rock
2801 S. University Ave. Little Rock, Arkansas 72204, USA
mgmilanova@ualr.edu
[4] Biology Department, Emory University
1510 Clifton Rd. NE, Atlanta, Georgia 30322, USA
tsmolin@emory.edu
[5] Center for Quantifiable Quality of Service in Communication Systems
Norwegian University of Science and Technology,
O.S. Bragstads plass 2E, N-7491 Trondheim, Norway
ajith.abraham@ieee.org, abraham.ajith@acm.org

Summary. This chapter presents a broad overview of Computational Intelligence (CI) techniques including Artificial Neural Networks (ANN), Particle Swarm Optimization (PSO), Genetic Algorithms (GA), Fuzzy Sets (FS), and Rough Sets (RS). We review a number of applications of computational intelligence to problems in bioinformatics and computational biology, including gene expression, gene selection, cancer classification, protein function prediction, multiple sequence alignment, and DNA fragment assembly. We discuss some representative methods to provide inspiring examples to illustrate how CI could be applied to solve bioinformatic problems and how bioinformatics could be analyzed, processed, and characterized by computational intelligence. Challenges to be addressed and future directions of research are presented. An extensive bibliography is also included.

1.1 Introduction

The past few decades have seen a massive growth in biological information gathered by the related scientific communities. A deluge of such information coming in the form of genomes, protein sequences, gene expression data and so on have led to the absolute need for effective and efficient computational tools to store, analyze and interpret the multifaceted data. Bioinformatics and computational biology involve the use of techniques including applied mathematics, informatics, statistics, computer science, artificial intelligence, chemistry, and biochemistry

T.G. Smolinski et al. (Eds.): Comp. Intel. in Biomed. & Bioinform., SCI 151, pp. 3–47, 2008.
springerlink.com

to solve biological problems usually on the molecular level. Research in computational biology often overlaps with systems biology. Major research efforts in the field include sequence alignment, gene finding, genome assembly, protein structure alignment, protein structure prediction, prediction of gene expression and protein-protein interactions, and the modeling of evolution [128]. Hence, in other words, bioinformatics can be described as the application of computational methods to make biological discoveries [10]. The ultimate attempt of the field is to develop new insights into the science of life as well as creating a global perspective, from which the unifying principles of biology can be derived [5]. There are at least 26 billion base pairs (bp) representing the various genomes available on the server of the National Center for Biotechnology Information (NCBI) [27]. Besides the human genome with about 3 billion bp, many other species have their complete genome available there. Cohen [23] explained the needs of biologists to utilize and help interpret the vast amounts of data that are constantly being gathered in genomic research. He also pointed out the basic concepts in molecular cell biology, and outlined the nature of the existing data, and illustrated the algorithms needed to understand cell behavior.

Bioinformatics involve the creation and advancement of algorithms using techniques including computational intelligence, applied mathematics and statistics, informatics, and biochemistry to solve biological problems usually on the molecular level. Major research efforts in the field include sequence analysis, gene finding, genome annotation, protein structure alignment analysis and prediction, prediction of gene expression, protein-protein docking/interactions, and the modeling of evolution.

Bioinformatics and computational biology are concerned with the use of computation to understand biological phenomena and to acquire and exploit biological data, increasingly large-scale data [38]. Methods from bioinformatics and computational biology are increasingly used to augment or leverage traditional laboratory and observation-based biology. These methods have become critical in biology due to recent changes in our ability and determination to acquire massive biological data sets, and due to the ubiquitous, successful biological insights that have come from the exploitation of those data. This transformation from a data-poor to a data-rich field began with DNA sequence data, but is now occurring in many other areas of biology [27].

Computational intelligence is a well-established paradigm, where new theories with a sound biological understanding have been evolving. The current experimental systems have many of the characteristics of biological computers ("brains") and are beginning to be built to perform a variety of tasks that are difficult or impossible to do with conventional computers. Computational intelligence methods are now being applied to problems in molecular biology and bioinformatics [70]. To name a few, Tasoulis et al. [104] present an application of neural networks, evolutionary algorithms, and clustering algorithms to DNA microarray experimental data analysis; Liang and Kelemen [60] propose a time lagged recurrent neural network with trajectory learning for identifying and classifying gene functional patterns from the heterogeneous nonlinear time series fmicroarray experiments. Reader may refer to [51, 22] for an extensive review of various computational intelligence techniques applied to different bioinformatics problems. Defining computational intelligence is not an easy task. In a nutshell, which becomes quite apparent in light of the current research pursuits, the area is heterogeneous with a combination of such technologies as neural networks, fuzzy systems, evolutionary computation, swarm intelligence, and probabilistic reasoning. The recent trend is to integrate different components to take advantage of complementary features and to develop a synergistic system [51]. Hybrid architectures like neuro-fuzzy systems, evolutionary-fuzzy systems, evolutionary-neural networks, evolutionary neuro-fuzzy systems, rough-neural, rough-fuzzy, etc. are widely applied for real world problem solving [1, 2, 46].

The objective of this book chapter is to present to the computational intelligence and bioinformatics research communities the state of the art computational intelligence applications to bioinformatics processing and motivate research in new trend-setting directions. Hence, we review and discuss in the following sections some representative methods to provide inspiring examples to illustrate how CI techniques could be applied to solve bioinformatics problems and how bioinformatics could be analyzed, processed, and characterized by computational intelligence. These representative examples include (i) CI in gene expression and clustering, (ii) rough discretization of gene expression, (iii) CI in protein sequence classification, (iv) CI in gene selection, (v) CI in cancer classification and the DNA fragment assembly problem, and (vi) CI in the multiple sequence alignment problem.

To provide useful insights for CI applications in bioinformatics, we structure the rest of this chapter as follows. Section 1.2 introduces some fundamental aspects and key components of modern computational intelligence including Artificial Neural Networks (ANN) , Rough Sets (RS), Fuzzy Sets (FS), Particle Swarm Optimization (PSO), and Genetic Algorithms (GA). Section 1.3 reviews some published papers on using computational intelligence in Gene Expression. A review of the current literature on CI-based approaches in Protein Sequence Classification problems is provided in Section 1.4. Section 1.5 discusses some successful work to illustrate how CI could be applied to Gene Selection problems. Applications of computational intelligence in DNA Fragment Assembly, Multiple Sequence Alignment Problems (MSA), and Protein Structure Prediction are reviewed in Sections 1.6, 1.7 and 1.8, respectively. An example of applications of CI in the field of human genetics, in the form of genetic programming neural networks, is presented in Section 1.9. CI in Microarray Classification is discussed and reviewed in Section 1.10. Conclusions, Challenges, and Future Directions are addressed in Section 1.11.

1.2 Computational Intelligence: Overview

In the following subsections, we present an overview of selected modern computational intelligence techniques including artificial neural networks, fuzzy sets, particle swarm optimization, genetic algorithms, and rough sets.

1.2.1 Artificial Neural Networks (ANN)

Artificial neural networks have been developed as generalizations of mathematical models of biological nervous systems. In a simplified mathematical model of the neuron, synapses are represented by connection weights that modulate the effect of the associated input signals, and the nonlinear characteristic exhibited by neurons is represented by a transfer function. There are many transfer functions developed to process the weighted and biased inputs, among which four basic and widely adopted in the field transfer functions are illustrated in Figure 1.1.

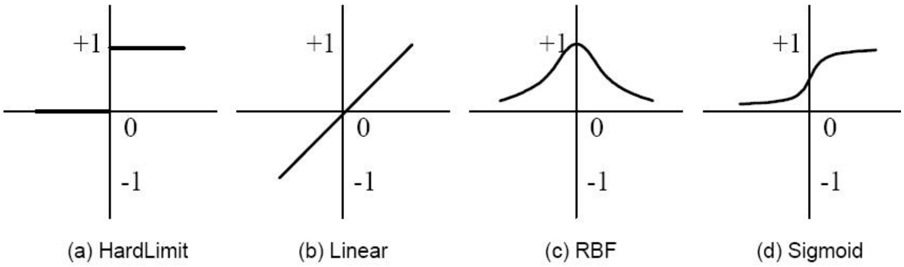


Fig. 1.1. Basic transfer functions

The neuron impulse is computed as the weighted sum of the input signals, transformed by the transfer function. The learning capability of an artificial neuron is achieved by adjusting the weights in accordance to the chosen learning algorithm. Most applications of neural networks fall into the following categories: (1) *Prediction*: Use the input values to predict some output; (2) *Classification*: Use the input values to determine the classification of the input; (3) *Data Association*: Similar to classification, but also recognizes data containing errors; and (4) *Data conceptualization*: Analyze the inputs so that grouping relationships can be inferred.

Neural Network Architecture

The behavior of the neural network depends largely on the interaction between the different neurons. The basic architecture consists of three types of neuron layers: input, hidden, and output layers.

In feed-forward networks the signal flow is from input to output units strictly in a feed-forward direction. The data processing can extend over multiple (layers of) units, but no feedback connections are present, that is, connections extending from outputs of units to inputs of units in the same layer or previous layers. Recurrent networks contain feedback connections. Contrary to feed-forward networks, the dynamical properties of such networks are important. In some cases, the activation values of the units undergo a relaxation process such that the network will evolve to a stable state in which these activations do not change anymore.

In other applications, the changes of the activation values of the output neurons are significant, such that the dynamical behavior constitutes the output of the network. There are several other neural network architectures (Elman network, adaptive resonance theory maps, competitive networks etc.) depending on the properties and requirement of the application.

Reader may refer to [13] for an extensive overview of the different neural network architectures and learning algorithms. A neural network has to be configured such that the application of a set of inputs produces the desired set of outputs. Various methods to set the strengths of the connections exist. One way is to set the weights explicitly, using a priori knowledge. Another way is to train the neural network by feeding it teaching patterns and letting it change its weights according to some learning rule. The learning situations in neural networks may be classified into three distinct sorts. These are supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, an input vector is presented at the inputs together with a set of desired responses, one for each node, at the output layer. A forward pass is done and the errors or discrepancies, between the desired and actual response for each node in the output layer, are found. These are then used to determine weight changes in the network according to the prevailing learning rule. The term 'supervised' originates from the fact that the desired signals on individual output nodes are provided by an external teacher. The best-known examples of this technique occur in the backpropagation algorithm, the delta rule, and perceptron rule. In unsupervised learning (or self-organization) an output unit is trained to respond to clusters of patterns within the input. In this paradigm the system is supposed to discover statistically salient features of the input population. Unlike the supervised learning paradigm, there is no a priori set of categories into which the patterns are to be classified; rather the system must develop its own representation of the input stimuli. Reinforcement learning is learning what to do–how to map situations to actions–so as to maximize a numerical reward signal. The learner is not told which actions to take, as in most forms of Machine Learning (ML), but instead must discover which actions yield the most reward by trying them. In the most interesting and challenging cases, actions may affect not only the immediate reward, but also the next situation and, through that, all subsequent rewards. These two characteristics, trial-and-error search and delayed reward are the two most important distinguishing features of reinforcement learning.

1.2.2 Rough Sets (RS)

Rough set theory [83, 84, 86, 82] is a methodology fairly new to the medical domain capable of dealing with uncertainty in data. It is used to discover data dependencies, evaluate the importance of attributes, discover the patterns of data, reduce redundant objects and attributes, seek the minimum subset of attributes, recognize and classify objects. Moreover, it is being used for extraction of rules from databases. Rough sets have proven useful for representation of vague regions in spatial data. One advantage of rough sets is creation of readable if-then rules. Such rules have a potential to reveal new patterns in the data material. Furthermore, they also collectively function as a classifier for unseen data. Unlike other computational intelligence techniques, rough set analysis requires no external parameters and uses only the information presented in the given data. One of the nice features of rough sets theory is that its can tell whether the data is complete or not based on the data itself. If the data is incomplete, the theory can suggest more information about the objects needed to be collected in order to build a good classification model. On the other hand, if the data is complete, rough sets can determine whether there is any redundant information in the data and find the minimum data needed for classification. This property of rough sets is very important for applications where domain knowledge is very limited or data collection is very expensive/laborious because it makes sure the data collected is good enough to build a good classification model without sacrificing the accuracy of the classification model or wasting time and effort to gather extra information about the objects [83, 84, 86, 82].

In rough sets theory, the data is collected in a table, called decision table. Rows of the decision table correspond to objects, and columns correspond to attributes. In the data set, we assume that class labels to indicate the class to which each example belongs are given. We call the class label the decision attribute and the rest of the attributes the condition attributes. Rough sets theory defines three regions based on the equivalent classes induced by the attribute values Lower approximation, upper approximation, and the boundary. Lower approximation contains all the objects which are classified surely based on the data collected, and upper approximation contains all the objects which can be classified probably, while the boundary is the difference between the upper approximation and the lower approximation. Thus we can define a rough set as any set represented through its lower and upper approximations. On the other hand, indiscernibility notion is fundamental to rough set theory. Informally, two objects in a decision table are indiscernible if one cannot distinguish between them on the basis of a given set of attributes. Hence, indiscernibility is a function of the set of attributes under consideration. For each set of attributes we can thus define a binary indiscernibility relation, which is a collection of pairs of objects that are indistinguishable from each other. An indiscernibility relation partitions the set of cases or objects into a number of equivalence classes. An equivalence class of a particular object is simply the collection of objects that are indiscernible to the object in question. Here we provide an explanation of the basic framework of rough set theory, along with some of the key definitions. A review of this basic material can be found in sources such as [83, 84, 86, 82, 77, 125] and many others.

1.2.3 Fuzzy Logic (FL) and Fuzzy Sets (FS)

Zadeh [121] introduced the concept of fuzzy logic to present vagueness in linguistics, and further implement and express human knowledge and inference capability in a natural way. Fuzzy logic starts with the concept of a fuzzy set. An

FS set is a set without a crisp, clearly defined boundary. It can contain elements with only a partial degree of membership. A Membership Function (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The input space is sometimes referred to as the universe of discourse. Let X be the universe of discourse and x be a generic element of X. A classical set A is defined as a collection of elements or objects $x \in X$, such that each x can either belong to or not belong to the set A, $A \sqsubseteq X$. By defining a characteristic function (or membership function) on each element x in X, a classical set A can be represented by a set of ordered pairs $(x, 0)$ or $(x, 1)$, where 1 indicates membership and 0 non-membership. Unlike conventional set mentioned above, fuzzy set expresses the degree to which an element belongs to a set. Hence the characteristic function of a fuzzy set is allowed to have value between 0 and 1, denoting the degree of membership of an element in a given set. If X is a collection of objects denoted generically by x, then a fuzzy set A in X is defined as a set of ordered pairs:

$$A = \{(x, \mu_A(x)) \mid x \in X\} \tag{1.1}$$

$\mu_A(x)$ is called the membership function of linguistic variable x in A, which maps X to the membership space M, $M = [0, 1]$, where M contains only two points, 0 and 1, A is crisp, and $\mu_A(x)$ is identical to the characteristic function of a crisp set. Triangular and trapezoidal membership functions are the simplest functions formed using straight lines. Some of the other shapes are Gaussian, generalized bell, sigmoidal, and polynomial based curves.

Figure 1.2, illustrates the shapes of two commonly used MFs. The most important thing to realize about fuzzy logical reasoning is the fact that it is a superset of standard Boolean logic.

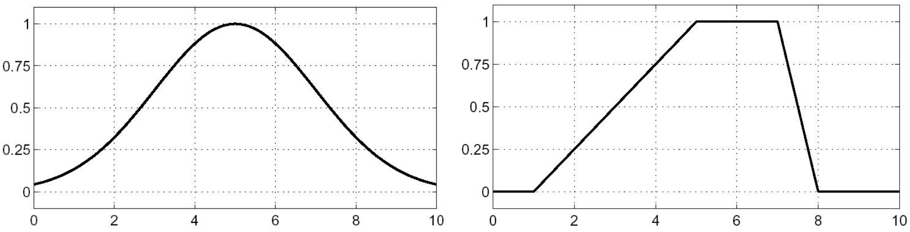


Fig. 1.2. Shapes of two commonly used MFs

1.2.4 Evolutionary Algorithms (EA)

Evolutionary Algorithms are adaptive methods, which may be used to solve search and optimization problems, based on the genetic processes of biological organisms. Over many generations, natural populations evolve according to the principles of natural selection and "survival of the fittest," first clearly stated by Charles Darwin in *The Origin of Species*. By mimicking this process, evolutionary algorithms are able to 'evolve' solutions to real world problems, if they have been suitably encoded [30]. Usually grouped under the term Evolutionary Algorithms (EA) or Evolutionary Computation (EC), we find the domains of genetic algorithms [43, 35], evolution strategies [8], evolutionary programming [32], genetic programming [57], and learning classifier systems [15]. They all share a common conceptual base of simulating the evolution of individual structures via processes of selection, mutation, and reproduction. The processes depend on the perceived performance of the individual structures as defined by the environment (problem).

EAs deal with parameters of finite length, which are coded using a finite alphabet, rather than directly manipulating the parameters themselves. This means that the search is unconstrained neither by the continuity of the function under investigation, nor the existence of a derivative function.

Genetic Algorithm (GA) is assumed that a potential solution to a problem may be represented as a set of parameters. These parameters (known as genes) are joined together to form a string of values (known as a chromosome). A gene (also referred to a feature, character or detector) refers to a specific attribute that is encoded in the chromosome. The particular values the genes can take are called its alleles. The position of the gene in the chromosome is its locus. Encoding issues deal with representing a solution in a chromosome and unfortunately, no one technique works best for all problems. A fitness function must be devised for each problem to be solved. Given a particular chromosome, the fitness function returns a single numerical fitness or figure of merit, which will determine the ability of the individual, which that chromosome represents. Reproduction is the second critical attribute of GAs where two individuals selected from the population are allowed to mate to produce offspring, which will comprise the next generation. Having selected two parents, their chromosomes are recombined, typically using the mechanisms of crossover and mutation.

There are many ways in which crossover can be implemented. In a single point crossover two chromosome strings are cut at some randomly chosen position, to produce two 'head' segments, and two 'tail' segments. The tail segments are then swapped over to produce two new full-length chromosomes. Crossover is not usually applied to all pairs of individuals selected for mating. Another genetic operation is mutation, which is an asexual operation that only operates on one individual. It randomly alters each gene with a small probability. Traditional view is that crossover is the more important of the two techniques for rapidly exploring a search space. Mutation provides a small amount of random search, and helps ensure that no point in the search space has a zero probability of being examined.

If the GA has been correctly implemented, the population will evolve over successive generations so that the fitness of the best and the average individual in each generation increases towards the global optimum. Selection is the survival of the fittest within GAs. It determines which individuals are to survive to the next generation. The selection phase consists of three parts. The first part involves determination of the individual's fitness by the fitness function. A fitness function must be devised for each problem; given a particular chromosome, the fitness function returns a single numerical fitness value, which is proportional to the ability, or utility, of the individual represented by that chromosome. For many problems, deciding upon the fitness function is very straightforward, for example, for a function optimization search; the fitness is simply the value of the function. Ideally, the fitness function should be smooth and regular so that chromosomes with reasonable fitness are close in the search space, to chromosomes with slightly better fitness. However, it is not always possible to construct such ideal fitness functions. The second part involves converting the fitness function into an expected value followed by the last part where the expected value is then converted to a discrete number of offsprings. Some of the commonly used selection techniques are roulette wheel and stochastic universal sampling. Genetic programming applies the GA concept to the generation of computer programs. Evolution programming uses mutations to evolve populations. Evolution strategies incorporate many features of the GA but use real-valued parameters in place of binary-valued parameters. Learning classifier systems use GAs in machine learning to evolve populations of condition/action rules.

1.2.5 Particle Swarm Optimization (PSO)

Swarm intelligence [54] is a collective behavior of intelligent agents in decentralized systems. Although there is typically no centralized control dictating the behavior of the agents, local interactions among them often cause a global pattern to emerge. Most of the basic ideas are derived from real swarms in the nature including ant colonies, bird flocking, honeybees, bacteria and microorganisms, etc. Ant Colony Optimization (ACO), have already been applied successfully to solve several engineering optimization problems. Swarm models are population-based and the population is initialized with a set of potential solutions. These individuals are then manipulated (optimized) over many iterations using several heuristics inspired from the social behavior of insects in an effort to find the optimal solution. Ant colony algorithms are inspired by the behavior of natural ant colonies, which solve their problems by multi agent cooperation using indirect communication through modifications in the environment. Ants release a certain amount of pheromone (hormone) while walking, and each ant prefers (probabilistically) to follow a direction, which is rich of pheromone. This simple behavior explains why ants are able to adjust to changes in the environment, such as optimizing shortest path to a food source or a nest. In ACO, ants use information collected during past simulations to direct their search and this information is available and modified through the environment. Recently ACO algorithms have also been used for clustering data sets [51].

The concept of particle swarms, although initially introduced for simulating human social behaviors, has become very popular these days as an efficient search and optimization technique. The Particle Swarm Optimization (PSO) [53], as it is called now, does not require any gradient information of the function to be optimized, uses only primitive mathematical operators, and is conceptually very simple. Since its advent in 1995, PSO has attracted the attention of many researchers all over the world resulting in a huge number of variants of the basic algorithm and many parameter automation strategies.

The canonical PSO model consists of a swarm of particles, which are initialized with a population of random candidate solutions [53]. They move iteratively through the d-dimension problem space to search for new solutions, where the fitness, f, can be calculated as the certain qualities measure. Each particle has a position represented by a position-vector $\mathbf{x}_i$ (i is the index of the particle), and a velocity represented by a velocity-vector $\mathbf{v}_i$. Each particle remembers its own best position so far in a vector $\mathbf{x}_i^{\#}$, and its j-th dimensional value is $x_{ij}^{\#}$. The best position-vector among the swarm so far is then stored in a vector $\mathbf{x}^*$, and its j-th dimensional value is x_j^*. During the iteration time t, the update of the velocity from the previous velocity to the new velocity is determined by (1.2). The new position is then determined by the sum of the previous position and the new velocity by (1.3).

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_1 (x_{ij}^{\#}(t) - x_{ij}(t)) + c_2 r_2 (x_j^*(t) - x_{ij}(t)). \tag{1.2}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1). \tag{1.3}$$

where w is called as the inertia factor, r_1 and r_2 are the random numbers, which are used to maintain the diversity of the population, and are uniformly distributed in the interval [0,1] for the j-th dimension of the i-th particle. c_1 is a positive constant, called the coefficient of the self-recognition component, c_2 is a positive constant, called the coefficient of the social component. From (1.2), a particle decides where to move next, considering its own experience, which is the memory of its best past position, and the experience of its most successful particle in the swarm. In the particle swarm model, the particle searches the solutions in the problem space with a range $[-s, s]$ (If the range is not symmetrical, it can be translated to a corresponding symmetrical range.) In order to guide the particles effectively in the search space, the maximum moving distance during one iteration must be clamped in between the maximum velocity $[-v_{max}, v_{max}]$ given in (1.4):

$$v_{ij} = sign(v_{ij}) min(|v_{ij}|, v_{max}). \tag{1.4}$$

The value of v_{max} is $p \times s$, with $0.1 \leq p \leq 1.0$ and is usually chosen to be s, i.e. $p = 1$. The end criteria are usually one of the following:

- Maximum number of iterations: the optimization process is terminated after a fixed number of iterations.
- Number of iterations without improvement: the optimization process is terminated after a fixed number of iterations without any improvement.
- Minimum objective function error: the error between the obtained objective function value and the best fitness value is less than a pre-fixed anticipated threshold.

1.3 CI in Gene Expression

Gene expression refers to a process through which the coded information of a gene is converted into structures operating in the cell. It provides the physical evidence that a gene has been *turned on* or activated. Expressed genes include those that are transcribed into mRNA and then translated into protein and those that are transcribed into RNA but not translated into protein (e.g., transfer and ribosomal RNAs) [64, 71]. The expression levels of thousands of genes can be measured at the same time using the modern microarray technology [87, 127]. DNA microarrays usually consist of thin glass or nylon substrates containing specific DNA gene samples spotted in an array by a robotic printing device. Researchers spread fluorescently labeled mRNA from an experimental condition onto the DNA gene samples in the array. This mRNA binds (hybridizes) strongly with some DNA gene samples and weakly with others, depending on the inherent double helical characteristics. A laser scans the array and sensors to detect the fluorescence levels (using red and green dyes), indicating the strength with which the sample expresses each gene. The logarithmic ratio between the two intensities of each dye is used as the gene expression data.

In this section, we provide a substantial review of the state of the art research, which focuses on the application of computational intelligence to different bioinformatics related Gene Expression problems. We also discuss some representative methods to provide inspiring examples to illustrate how CI could be applied to resolve bioinformatics Gene Expression problems and how Gene Expression problems could be analyzed, processed, and characterized by computational intelligence.

1.3.1 Gene Expression Data Clustering

In the field of pattern recognition, clustering [48] refers to the process of partitioning a dataset into a finite number of groups according to some similarity measure. Currently, it has become a widely used process in microarray engineering for understanding the functional relationship between groups of genes. Clustering was used, for example, to understand the functional differences in cultured primary epatocytes relative to the intact liver [9]. In another study, clustering techniques were used on gene expression data for tumor and normal colon tissue probed by oligonucleotide arrays [4].

A number of clustering algorithms, including hierarchical clustering [113, 97], Principle Component Analysis (PCA) [119, 89], genetic algorithms [59], and artificial neural networks [42, 101, 107], have been used to cluster gene expression data. However, in 2002, Yuhui et al. [120] proposed a new approach to analysis of gene expression data using Associative Clustering Neural Network (ACNN). ACNN dynamically evaluates similarity between any two gene samples through the interactions of a group of gene samples. It exhibits more robust performance than the methods with similarities evaluated by direct distances, which has been tested on the leukemia data set. The experimental results demonstrate that ACNN is superior in dealing with high dimensional data (7,129 genes).

The performance can be further enhanced when some useful feature selection methodologies are incorporated. The study has shown ACNN can achieve 98.61% accuracy on clustering the Leukemias data set with correlation analysis.

Herrero et al. [42] used the Self-Organizing Tree Algorithm (SOTA) for analysis of gene expression data coming from DNA array experiments, using an unsupervised neural network. DNA array technologies allow monitoring thousands of genes rapidly and efficiently. One of the interests of these studies is the search for correlated gene expression patterns, and this is usually achieved by clustering them. The result of the algorithm is a hierarchical cluster obtained with the accuracy and robustness of a neural network. SOTA clustering confers several advantages over classical hierarchical clustering methods. The clustering process is performed from top to bottom, i.e. the highest hierarchical levels are resolved before going to the details of the lowest levels. The growing can be stopped at the desired hierarchical level. Moreover, a criterion to stop the growing of the tree, based on the approximate distribution of probability obtained by randomisation of the original data set, is provided. In addition, obtaining average gene expression patterns is a built-in feature of the algorithm. Different neurons defining the different hierarchical levels represent the averages of the gene expression patterns contained in the clusters.

Xiao et al. [116] proposed a new clustering approach based on the synergism of the PSO and Self Organizing Maps (SOM). The authors achieved promising results by applying the hybrid SOM-PSO algorithm over the gene expression data of yeast and rat hepatocytes. We will briefly discuss their approach in the following paragraphs. The idea of the SOM [56] stems from the orderly mapping of information in the cerebral cortex. With SOMs, high dimensional datasets are projected onto a one- or two-dimensional space. Typically, a SOM has a two dimensional lattice of neurons and each neuron represents a cluster. The learning process of a SOM is unsupervised. All neurons compete for each input pattern and the neuron that is chosen for the input pattern wins.

In the approach proposed by Xiao et al., PSO is used to evolve the weights for the SOM. In the first stage of the hybrid SOM/PSO algorithm, a SOM is used to cluster the dataset. Authors used a SOM with conscience at this step. Conscience directs each component that takes part in competitive learning toward having the same probability to win. Conscience is added to the SOM by assigning each output neuron a bias. The output neuron must overcome its own bias to win. The objective is to obtain a better approximation of pattern distribution. The SOM normally runs for 100 iterations and generates a group of weights. In the second stage, PSO is initialized with the weights produced by the SOM in the first stage. Then a *gbest* PSO is used to refine the clustering process. Each particle consists of a complete set of weights for the SOM. The dimension of each particle is the number of input neurons of the SOM times the number of output neurons of the SOM. The objective of PSO is to improve the clustering result by evolving the population of particles.

Microarrays have recently made it possible to monitor the activity of thousands of genes simultaneously. They offer new insights into the biology of a cell.

However, the data produced by microarrays poses several challenges to overcome. One major task in the analysis of microarray data is to reveal structures despite a large noise component in the data. Futschik and Kasabov [33] used Fuzzy C-Means (FCM) clustering to achieve a robust analysis of gene expression time-series. Authors address the issues of parameter selection and cluster validity. Using statistical models to simulate gene expression data, they show that FCM can detect genes belonging to different classes.

Chinatsu and Hanai [7] applied the Fuzzy Adaptive Resonance Theory (Fuzzy ART) [106] to gene clustering of DNA microarray data and their result indicate that the methodology may be more suitable for biological applications than most other methods including hierarchical clustering, k-means clustering, and SOM. In addition, the authors compared their technique with the fuzzy c-means clustering method and obtained comparable results.

Okada et al. [79] point out that although hierarchical clustering has been extensively used in analyzing patterns in microarray gene expression data, its biological interpretation is not easy. The authors propose a novel algorithm that automatically finds biologically interpretable cluster boundaries in hierarchical clustering by referring to gene annotations stored in public genome databases. In addition, the proposed algorithm has a new function of generating a set of clusters that are independent of each other with respect to the distributions of gene functions. The authors claim that this function would enable investigators to efficiently identify non-redundant and biologically-independent clusters.

An Evolutionary Rough C-Means Clustering

Cluster analysis [104] is one key step in understanding how the activity of genes varies during biological processes and is affected by disease states and cellular environments. In particular, clustering can be used either to identify sets of genes according to their expression in a set of samples [26, 113], or to cluster samples into homogeneous groups that may correspond to particular macroscopic phenotypes [36]. The latter is in general more difficult, but is very valuable in clinical practice.

Several clustering algorithms have been developed and applied in bioinformatics problems, however, most of them cannot process objects in hybrid numerical/nominal feature space or with missing values. In most of them, the number of clusters should be manually determined and the clustering results are sensitive to the input order of the objects to be clustered. These limit applicability of the clustering and reduce the quality of clustering. To solve this problem, an improved clustering algorithm based on rough set and entropy theory was presented by Chun-Bao et al. [19]. The approach aims at avoiding the need to pre-specify the number of clusters, and clustering in both numerical and nominal feature space with the similarity introduced to replace the distance index.

At the same time, rough sets are used to represent clusters in terms of upper and lower approximations. However, the relative importance of these approximation parameters, as well as a threshold parameter, need to be tuned for good partitioning. The evolutionary rough c-means algorithm employs GAs to tune these parameters. The Davies-Bouldin index is used as the fitness function to be minimized. Various values of c are used to generate different sets of clusters, and GA is employed to generate the optimal partitioning [100].

Lingras [62] argued that incorporation of rough sets into k-means clustering requires the addition of the concept of lower and upper bounds. Calculation of the centroids of clusters from conventional k-means needs to be modified to include the effects of lower as well as upper bounds. The modified centroid calculations for rough sets are then given by:

$$cen_j = W_{low} \times \frac{\sum_{v \in \underline{R}(x)}}{|\underline{R(x)}|} + w_{up} \times \frac{\sum_{v \in (\underline{BN_R(x)})}}{|\underline{BN_R(x)}|} \tag{1.5}$$

Where $1 \leq j \leq m$. The parameters w_{low} and w_{up} correspond to the relative importance of lower and upper bounds, and $w_{low} + w_{up} = 1$. If the upper bound of each cluster were equal to its lower bound, the clusters would be conventional clusters. Therefore, the boundary region $\underline{BN_R(x)}$ will be empty, and the second term in the equation will be ignored. Thus, the above equation will reduce to conventional centroid calculations. The next step in the modification of the k-means algorithms for rough sets is to design criteria to determine whether an object belongs to the upper or lower bound of a cluster, for more details refer to [62]. The main steps of the algorithm are provided below.

Algorithm 1. Rough C-Means Algorithm

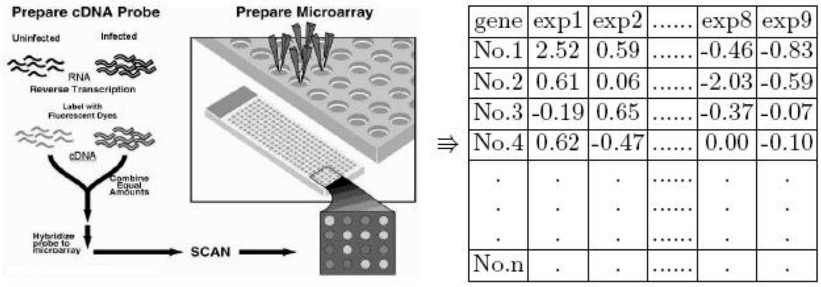
```
 1: Set $x_i$ as an initial means for the $c$ clusters.
 2: Initialize the population of particles encoding parameters threshold and $w_{low}$
 3: Initialize each data object $x_k$ to the lower approximation or upper approximation
    of clusters $c_i$ by computing the difference in its distance by:

        $diff = d(x_k, cen_i) - d(x_k, cen_j),$          (1.6)

    Where $cen_i$ and $cen_j$ are the cluster centroid pairs.
 4: if diff $< \delta$ then
 5:   $x_k \in$ the upper approximation of the $cen_i$ and $cen_j$ clusters and can not be in
      any lower approximation.
 6:   Else
 7:   $x_k \in$ lower approximation of the cluster $c_i$ such that distance $d(x_k, cen_i)$ is is
      minimum over the $c$ clusters.
 8: end if
 9: Compute a new mean using equation (1.5)
10: repeat
11:   statements 3–9
12: until convergence i.e. there is no more new assignments
```

1.3.2 Rough Sets and DNA Microarray Technology

Biological research is currently undergoing a revolution. With the advent of microarray technology the behavior of thousands of genes can be measured simultaneously. This capability opens a wide range of research opportunities in



gene	exp1	exp2	......	exp8	exp9
No.1	2.52	0.59	......	-0.46	-0.83
No.2	0.61	0.06	......	-2.03	-0.59
No.3	-0.19	0.65	......	-0.37	-0.07
No.4	0.62	-0.47	......	0.00	-0.10
.	.	.	......	.	.
.	.	.	......	.	.
.	.	.	......	.	.
No.n	.	.	......	.	.

Fig. 1.3. Microarray production process:Microarrays provide the gene expression data. A sample of 9 experiments from Synovial Sarcoma data is illustrated, n=5,520 genes in this data set [37, 96].

biology, but the technology generates a vast amount of data that cannot be handled manually. Computational analysis is thus a prerequisite for the success of this technology, and research and development of computational tools for microarray analysis are of great importance [68]. The DNA microarray technology provides enormous quantities of biological information about genetically conditioned susceptibility to diseases [11]. The data sets acquired from microarrays refer to genes via their expression levels. Microarray production starts with preparing two samples of mRNA, as illustrated by Figure 1.3. The sample of interest is paired with a healthy control sample. The fluorescent red/green labels are applied to both samples. The procedure of samples mixing is repeated for each of thousands of genes on the slide. Fluorescence of red/green colors indicates to what extent the genes are expressed. The gene expressions can be then stored in numeric attributes, coupled with, e.g., clinical information about the patients [11].

One application of microarray technology is cancer studies, where supervised learning may be used for predicting tumor subtypes and clinical parameters. Herman et al. [68] present a general rough set approach for classification of tumor samples analyzed with microarrays. This approach is tested on a data set of gastric tumors, and authors develop classifiers for six clinical parameters. This research included only 2,504 genes out of a total of at least 30,000 genes in the human genome. Some of the genes that were not included in their study may have a connection to the parameters. In addition, their results show that it is possible to develop classifiers with a small number of tumor samples, and that rough set based methods may be well suited for this task. They believe that rough set based learning combined with feature selection may become an important tool for microarray analysis.

Rough Discretization

Microarray measurements are real numbers that have to be discretized before a learning algorithm can be applied on the them. It has been shown that the quality of a learning algorithm is dependent on the selected strategy used for real data discritization [25]. Discretization uses a data transformation procedure that involves finding cuts which divide the data values into intervals. Values lying within an interval are then mapped to the same ‘label’ value. Performing this process will lead to reduction in the size of the attributes value set and ensure that the rules that are mined are not too specific. Lots of discretization algorithms have been developed and applied in bioinformatics problems [68]. Examples of utilized discretization algorithms include frequency binning, naïve discretization, entropy-based discretization, discriminant discretization, and Boolean reasoning/rough set based discretization [68].

Here we demonstrate some reported examples of using discretization techniques in bioinformatics problems. Many successful work towards this issue has been addressed and discussed. For example, the rough sets with Boolean reasoning (RSBR) algorithm proposed by Zhong et al. [124, 40] was used for discretization of continuous-valued attributes. The main advantage of RSBR is that it combines discretization of real valued attributes and classification. The main steps of the RSBR discretization algorithm are provided below.

Algorithm 2. RSBR Discretization Algorithm

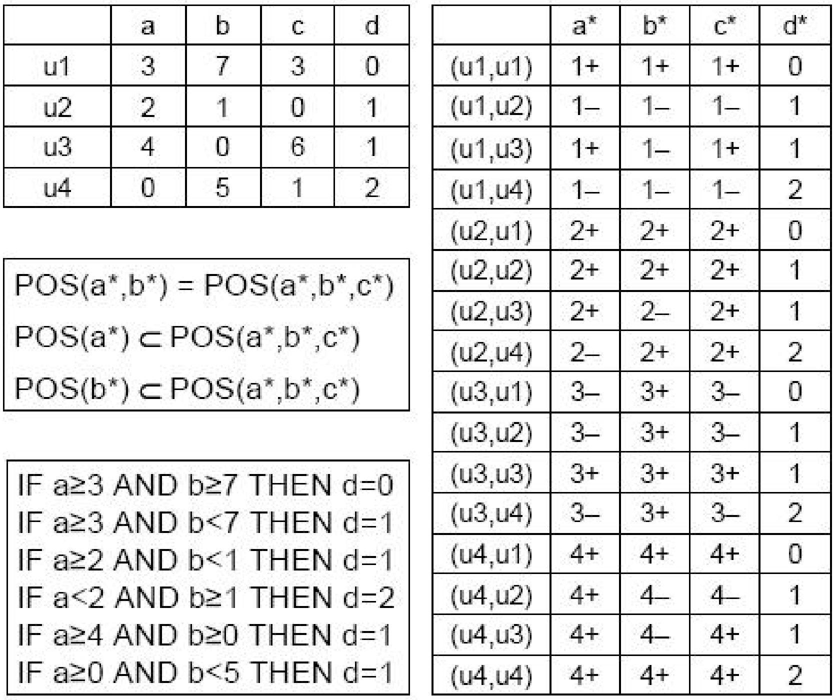
```
Input: Information system table ($S$) with real valued attributes $A_{ij}$ and $n$ is the number of inter values for each attribute.
Output: Information table ($ST$) with discretized real valued attribute
for $A_{ij} \in S$ do
    Define a set of Boolean variables as follows:
        $B = \{\sum_{i=1}^{n} C_{ai}, \sum_{i=1}^{n} C_{bi} \sum_{i=1}^{n} C_{ci}, ..., \sum_{i=1}^{n} C_{ni}\}$    (1.7)
end for
Where $\sum_{i=1}^{n} C_{ai}$ corresponds to a set of intervals defined on the variables of attributes $a$
Create a new information table $S_{new}$ by using the set of intervals $C_{ai}$
Find the minimal subset of $C_{ai}$ that discerns all the objects in the decision class $D$ using the following formula:
    $\Upsilon^u = \wedge\{\Phi(i,j) : d(x_i \neq d(x_j)\}$    (1.8)
Where $\Phi(i,j)$ is the number of minimal cuts that must be used to discern two different instances $x_i$ and $x_j$ in the information table.
```

Among further research directions, there is hybridization of rough set reduction framework with gene clustering. For example, in [37] authors used self-organizing maps to calculate the entropy distance for roughly discretized data. In another example, Ślęzak and Wróblewski [95] adapt the rough set-based approach to deal with gene expression data, where the problem is a huge amount of genes (attributes) $a \in A$ versus small amount of experiments (objects) $u \in U$. They perform gene reduction using standard rough set methodology based on approximate decision reducts applied against specially prepared data. In addition, the authors used rough discretization algorithm - Every pair of objects $(x, y) \in U \times U$ yields a new object, which takes values "$\geq a(x)$" if and only if $a(y) \geq a(x)$; and "$\leq a(x)$" otherwise; over original genes-attributes $a \in A$. In this way: 1) They work with desired, larger number of objects improving credibility of the obtained reducts; 2) They produce more decision rules, which vote during classification of new observations; 3) They avoid an issue of discretization of real-valued attributes, difficult and leading to unpredictable results in case of any data sets having much more attributes than objects. The authors illustrated their method by analysis of gene expression data related to breast cancer.

Another example given by Ślęzak and Wróblewski [96] extends the standard rough set-based approach to deal with huge amounts of numeric attributes versus a small amount of available objects. The authors transform the training data using a novel way of non-parametric discretization, called roughfication (in contrast to fuzzification known from fuzzy logic). Given roughfied data, they apply standard rough set attribute reduction and then classify the testing data by voting among the obtained decision rules. Roughfication enables to search for reducts and rules in the tables with the original number of attributes and far larger number of objects. It does not require expert knowledge or any kind of parameter tuning or learning. The authors illustrate it by analysis of gene expression data, where the number of genes (attributes) is enormously large with respect to the number of experiments (objects).

Given thousands of attributes against hundreds of objects, we face a *few-objects-many-attributes* problem, recognized as one of the main data mining challenges [118]. Moreover, in the case of gene expression, rough set based methods usually require discretization (cf. [76])–replacing the original values with the codes of intervals defined over attribute ranges. This additionally increases the amount of possible solutions of the optimization problem, now reformulated as searching for optimal subsets of attributes (genes) coupled with their optimal interval settings. Such a huge space of parameters, given too small samples of objects, leads to data overfitting (cf. [118]) and yields a kind of unreliability of the rough set techniques applied so far (cf. [109]). Ślęzak and Wróblewski [96] report an alternative method, illustrated by Figure 1.4. They call it rough discretization (or roughfication, compared to fuzzification).

As has been reported, e.g., [68], some discretization methods seem to work better than others for the problem of gene expression classification. Frequency binning and entropy-based discretization gave good results. Discretization based on linear discriminant analysis was also useful. The entropy-based method appeared to handle skewed class distributions better than the other methods. Boolean reasoning discretization had often a poor performance and behaved differently from the rest of the discretization methods. The AUC had a tendency to increase with additional genes. It is likely that this is due to the global nature of this method. The method considers all attributes at once when it creates cuts. The feature selection method, on the other hand, selects genes individually such that each selected gene may be a good classifier in itself. So, it is more appropriate to



	a	b	c	d
u1	3	7	3	0
u2	2	1	0	1
u3	4	0	6	1
u4	0	5	1	2

POS(a*,b*) = POS(a*,b*,c*)

POS(a*) ⊂ POS(a*,b*,c*)

POS(b*) ⊂ POS(a*,b*,c*)

IF a≥3 AND b≥7 THEN d=0
IF a≥3 AND b<7 THEN d=1
IF a≥2 AND b<1 THEN d=1
IF a<2 AND b≥1 THEN d=2
IF a≥4 AND b≥0 THEN d=1
IF a≥0 AND b<5 THEN d=1

	a*	b*	c*	d*
(u1,u1)	1+	1+	1+	0
(u1,u2)	1–	1–	1–	1
(u1,u3)	1+	1–	1+	1
(u1,u4)	1–	1–	1–	2
(u2,u1)	2+	2+	2+	0
(u2,u2)	2+	2+	2+	1
(u2,u3)	2+	2–	2+	1
(u2,u4)	2–	2+	2+	2
(u3,u1)	3–	3+	3–	0
(u3,u2)	3–	3+	3–	1
(u3,u3)	3+	3+	3+	1
(u3,u4)	3–	3+	3–	2
(u4,u1)	4+	4+	4+	0
(u4,u2)	4+	4–	4–	1
(u4,u3)	4+	4–	4+	1
(u4,u4)	4+	4+	4+	2

Fig. 1.4. Rough discretization [96]. Top: A sample with 3 numeric attributes and 3 decision classes. Right: Its roughfied version. Middle: Some positive regions for the roughfied table. Bottom: Rules induced by reduct a^*, b^*.

make cuts individually for each gene. The Boolean reasoning approach is consequently less suited for this problem, but it may yield a good performance in other situations.

1.4 CI in Protein Sequence Classification

The problem of protein sequence classification is a crucial task in the interpretation of genomic data. Many high-throughput systems were developed with the aim of categorizing proteins based only on their sequences. However, modeling how proteins have evolved can also help in the classification task of sequenced data. Hence the phylogenetic analysis has gained importance in the field of protein classification. Busa-Fekete et al. [16] provide an overview about the problem of protein sequence classification area and propose two algorithms that are well suited to this scope. The two algorithms are based on a weighted binary tree representation of protein similarity data. The first one is called TreeInsert which assigns the class label to the query by determining a minimum cost necessary to insert the query in the (precomputed) trees representing the various classes. Then the TreeNN algorithm assigns the label to the query based on an analysis of the query's neighborhood within a binary tree containing members of the known classes. The two algorithms were tested in combination with various sequence similarity scoring methods (BLAST, Smith-Waterman, Local Alignment Kernel as well as various compression-based distance scores) using a large number of classification tasks representing various degrees of difficulty. They reported that, at the expense of a small computational overhead, both TreeNN and TreeInsert exceed the performance of simple similarity search (1NN) as determined by ROC analysis, at the expense of a modest computational overhead. Combined with a fast tree-building method, both algorithms are suitable for web-based server applications.

Mapping the pathways that give rise to metastasis is one of the key challenges of breast cancer research. Recently, several large-scale studies have shed light on this problem through analysis of gene expression profiles to identify markers correlated with metastasis. Han-Yu Chuang et al. [21] apply a protein-network-based approach that identifies markers not as individual genes but as subnetworks extracted from protein interaction databases. The resulting subnetworks provide novel hypotheses for pathways involved in tumor progression. Although genes with known breast cancer mutations are typically not detected through analysis of differential expression, they play a central role in the protein network by interconnecting many differentially expressed genes. Authors find that the subnetwork markers are more reproducible than individual marker genes selected without network information, and that they achieve higher accuracy in classification of metastatic versus non-metastatic tumors.

As shown in Figure 1.5, the subnetwork markers were significantly more reproducible between data sets than were individual marker genes selected without network information (12.7 versus 1.3%). In terms of biological function, extracellular signal-regulated kinase 1 (MAPK3) was reproducible as a central node in subnetworks identified from both data sets (Figure 1.5C versus Figure 1.5D. Figure 1.5E and 1.5F illustrate two other subnetworks that were discriminative in both data sets, although there was less consistency in the expression levels of genes comprising these subnetworks. For instance, PKMYT1 is significantly differentially expressed in van de Vijver et al [110] but not in Wang et al. [112] (Figure 1.5E; diamond versus circle), whereas CD44 is significantly differentially expressed in Wang et al. [112] but not in van de Vijver et al. [110] (Figure 1.5F). However, by aggregating the expression ratios of these genes with their network neighbors, the subnetworks containing these genes are found to be significant in both data sets.

Classification of protein sequences into families is an important tool in the annotation of structural and functional properties to newly discovered proteins. Mohamed et al [72] present a classification system using pattern recognition techniques to create a numerical vector representation of a protein sequence and then classify the sequence into a number of given families. Authors introduce the

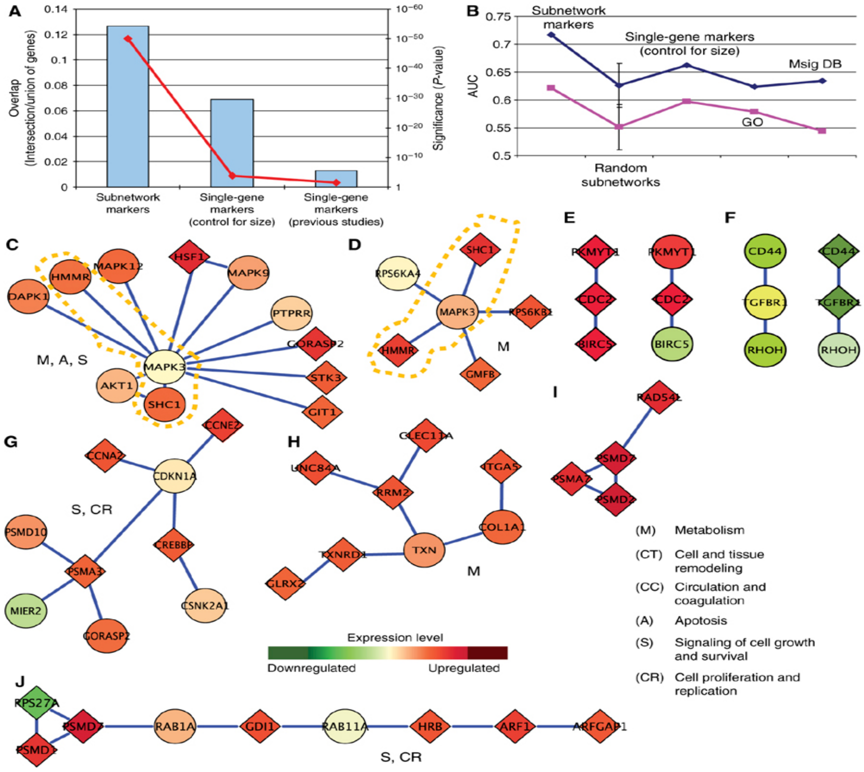


Fig. 1.5. Subnetwork markers across data sets [21]

use of fuzzy ARTMAP classifiers and show that coupled with a genetic algorithm based feature subset selection, the system is able to classify protein sequences with an accuracy of 93%. This accuracy is compared with numerous other classification tools and demonstrates that the fuzzy ARTMAP is suitable due to its high accuracy, quick training times, and ability for incremental learning.

Building improved intelligent protein sequence classification systems for effectively searching large biological database is significant for developing competitive pharmacological products. Wang et al [111] describe a methodology for constructing a neural protein classifier with various input features, rather than to train a neural classifier based on a given neural network architecture and some available data. A set of fuzzy classification rules with confidence factors can be extracted directly from the generalized radial basis function (GRBF) networks. The initial fuzzy rule set is refined using a new objective function, which compromises between misclassification rate and generalization capability, and GA programming. Their results compared favorably with other standard machine learning techniques.

1.5 CI in Gene Selection

Selecting informative and discriminative genes from huge microarray gene expression data is an important and challenging bioinformatics research topic. There have been many successful projects in this area reported in the literature. For example, Fernando et al. [29] demonstrate how a supervised fuzzy pattern algorithm can be used to perform DNA microarray data reduction over real data. The benefits of their method can be employed to find biologically significant insights relating to meaningful genes in order to improve previous successful techniques. Experimental results on acute myeloid leukemia diagnosis show the effectiveness of the proposed approach.

A new method combining correlation based clustering and rough sets attribute reduction for gene selection from gene expression data is proposed by Lijun et al [99]. Correlation based clustering is used as a filter to eliminate the redundant attributes, then the minimal reduct of the filtered attribute set is reduced by rough sets. Three different classification algorithms are employed to evaluate the performance of the proposed method. High classification accuracies achieved on two public gene expression data sets show that the introduced method is successful for selecting high discriminative genes for classification task. The experimental results indicate that rough sets based methods have the potential to become a useful tool in bioinformatics.

The approach to cancer classification based on selected gene expression data, rather than all the genes in the dataset, is important for efficient cancer diagnosis. Dingfang et al. [58] present a gene selection method, called RMIMR, which searches for the subset through maximum relevance and maximum positive interaction of genes. Compared to the classical methods based on statistics, information theory, and regression, this method led to significantly improved classification in experiments on 4 gene expression datasets.

Banerjee et al. [12] used an evolutionary rough feature selection algorithm for classifying microarray gene expression patterns. Since the data typically consist of a large number of redundant features, an initial reduction of the attributes is done to enable faster convergence. Rough set theory is employed to generate reducts, which represent the minimal sets of nonredundant features capable of discerning between all objects, in a multiobjective framework. The effectiveness of the algorithm is demonstrated on three cancer datasets.

Zhang et al. [123] present recent Support Vector Machine (SVM) classification approaches for gene selection, cancer classification, and functional gene classification, followed by analysis on the advantages and limitations of SVM on these applications.

Li et al. [59] introduced a multivariate approach that selects a subset of predictive genes jointly for sample classification based on expression data. They tested the algorithm on colon and leukemia data sets. The authors examined the sensitivity, reproducibility and stability of gene selection/sample classification to the choice of parameters of the algorithm. They used hybrid method that uses a genetic algorithms and the K-Nearest Neighbor (KNN) to identify genes that can jointly discriminate between different classes of samples (e.g. normal versus tumor). The genes identified are subsequently used to classify independent test set samples. The authors reported that the GA/KNN method is capable of selecting a subset of predictive genes from a large noisy data set for sample classification. It is a multivariate approach that can capture the correlated structure in the data.

Yuanchen et al. [41] proposed a fuzzy-granular method for the gene selection task. Firstly, genes are grouped into different function granules with the fuzzy c-means algorithm (FCM). And then informative genes in each cluster are selected with the signal-to-noise metric (S2N). With fuzzy granulation, information loss in the process of gene selection is decreased. As a result, more informative genes for cancer classification are selected and more accurate classifiers can be modeled. The simulation results on two publicly available microarray expression datasets show that the proposed method is more accurate than traditional algorithms for cancer classification.

Gene Selection Using Neural Networks

Accurate diagnosis and classification are the key issues for the optimal treatment of cancer patients. Several studies demonstrate that cancer classification can be estimated with high accuracy, sensitivity, and specificity from microarray-based gene expression profiling using artificial neural networks.

Huang and Liao [45] introduced a comprehensive study to investigate the capability of the probabilistic neural networks (PNN) associated with a feature selection method, the so-called signal-to-noise statistic, in cancer classification. The signal-to-noise statistic, which represents the correlation with the class distinction, is used to select the marker genes and trim the dimension of data samples for the PNN. The experimental results show that the association of the probabilistic neural network with the signal-to-noise statistic can achieve superior classification results for two types of acute leukemias and five categories of embryonal tumors of central nervous system with satisfactory computation speed. Furthermore, the signal-to-noise statistic analysis provides candidate genes for future study in understanding the disease process and the identification of potential targets for therapeutic intervention.

Fogel [31] highlights recent advancements in the coupling evolutionary computation with artificial neural networks for microarray class prediction and discovery. The combination of these methods holds great promise for automated feature selection and data analysis. Neural networks have been noted elsewhere in the literature as particularly useful for microarray data clustering and classification. For instance, Khan et al. [55] developed a method of classifying cancers to specific diagnostic categories based on their gene expression signatures using artificial neural networks. The authors trained the ANNs using a small, round blue-cell tumors (SRBCTs) as a model. These cancers belong to four distinct diagnostic categories and often present diagnostic dilemmas in clinical practice. The ANNs correctly classified all samples and identified genes most relevant to the classification. Expression of several of these genes has been reported in SRBCTs, but most have not been associated with these cancers. To test the ability

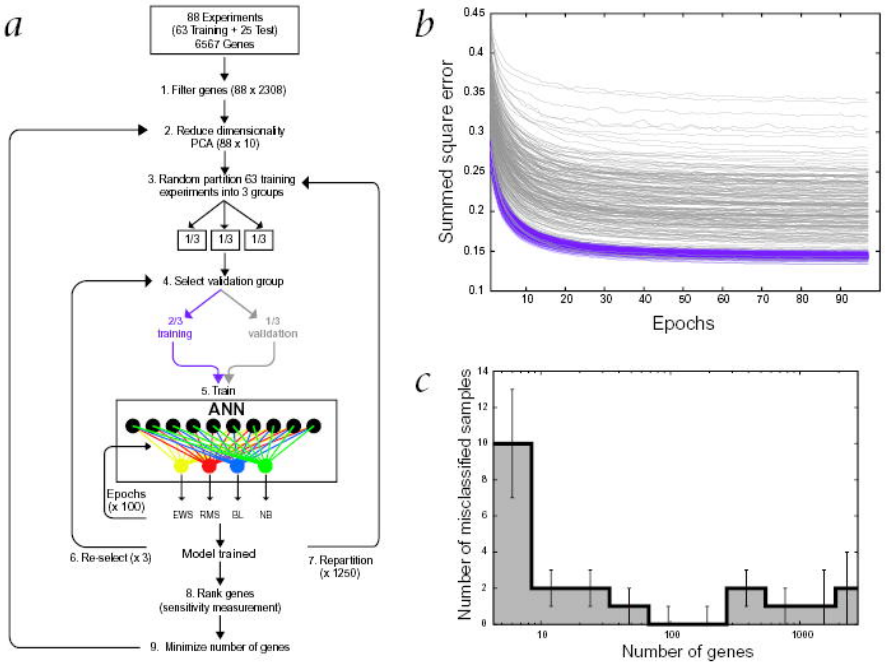


Fig. 1.6. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks [55]

of the trained ANN models to recognize SRBCTs, they analyzed additional blind samples that were not previously used for training, and correctly classified them in all cases. This study demonstrates the potential applications of these methods for tumor diagnosis and the identification of candidate targets for therapy.

As an illustrated in Figure 1.6a, the entire data-set of all 88 experiments was first quality filtered (1) and then the dimensionality was further reduced by principal component analysis (PCA) to 10 PC projections (2), from the original 6,567 expression values. Next, the 25 test experiments were set aside and the 63 training experiments were randomly partitioned into 3 groups (3). One of these groups was reserved for validation and the remaining 2 groups for calibration (4). ANN models were then calibrated using for each sample the 10 PC values as input and the cancer category as output (5). For each model, the calibration was optimized with 100 iterative cycles (epochs). This was repeated using each of the 3 groups for validation (6). The samples were again randomly partitioned and the entire training process repeated (7). For each selection of a validation group one model was calibrated, resulting in a total of 3750 trained models. Once the models were calibrated they were used to rank the genes according to their importance for the classification (8). The entire process (2–7) was repeated using only top ranked genes (9). The 25 test experiments were subsequently classified using all the calibrated models. Figure 1.6b presents monitoring of the calibration of the models. The average classification error per sample (using a summed square error function) is plotted during the training iterations (epochs) for both the training and the validation samples. A pair of lines, dark (training) and light (validation), represents one model. The decrease in the classification errors with increasing epochs demonstrates the learning of the models to distinguish these cancers. The results shown are for 200 different models, each corresponding to a random partitioning of the data. All the models performed well for both training and validation as demonstrated by the parallel decrease (with increasing epochs) of the average summed square classification error per sample. In addition, there was no sign of over-training: if the models begin to learn features in the training set, which are not present in the validation set, this would result in an increase in the error for the validation at that point and the curves would no longer remain parallel. Figure 1.6c shows minimizing the number of genes. The average number of misclassified samples for all 3,750 models is plotted against increasing number of used genes. The misclassifications were minimized to zero using the 96 highest ranked genes [31, 55].

While it is clear that neural network methods are well suited to microarray analysis, their proper training and optimization is a prerequisite for superior performance. A standard approach to neural network training is the use of backpropagation to optimize the weight assignments for a fixed neural network topology. This approach generally forces the user to choose the appropriate number of features to use and a fixed neural network topology. Backpropagation itself can also lead to suboptimal weight assignment if there are many local optima in the search space. Optimizing neural networks with stochastic optimization methods such as evolutionary computation, however, can outperform these classic methods by avoiding local optima and simultaneously identifying the most appropriate features to use for prediction [31].

In another study, Hwang et al. [47] applied neural networks in classification of patient samples using gene expressions levels. Here all gene expression levels are fed to the neural tree as input and the output is a binary classification. Through a structural learning process, essential genes for cancer classification are included into the neural tree and less important genes are weeded out automatically. In neural tree learning, all gene expression levels were linearly scaled into the interval [0.01, 0.99]. For the output value of neural tree learning, one was set to 0.01 and the other one to 0.99. Using this setup, their predicted accuracy was 86% and the number of genes selected was 16. Gene selection using Feed Forward Back Propagation Neural Network as a classifier is illustrated in Figure 1.7.

Francesca et al. [92] proposed a new gene selection method for analyzing microarray experiments pertaining to two classes of tissues and for determining relevant genes characterizing differences between the two classes. The new technique is based on Switching Neural Networks (SNN), learning machines that assign a relevance value to each input variable, and adopts Recursive Feature Addition (RFA) for performing gene selection. The performances of SNN-RFA are evaluated by considering its application on two real and two artificial gene

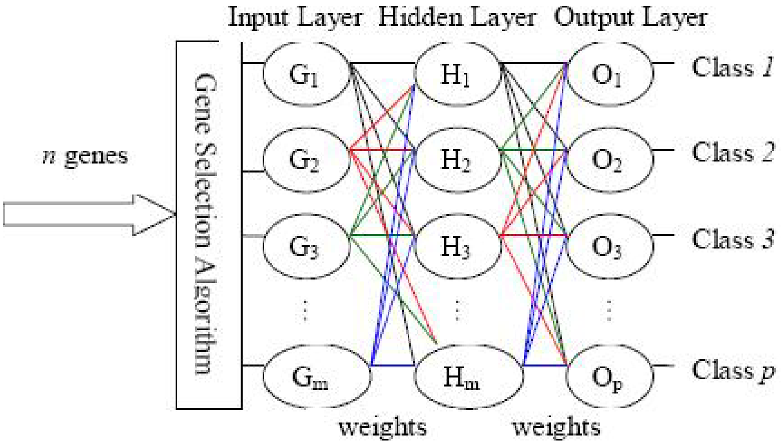


Fig. 1.7. Gene selection using Neural Network as classifere [81]

expression datasets generated according to a proper mathematical model that possesses biological and statistical plausibility.

Gene selection algorithms for cancer classification, based on the expression of a small number of biomarker genes, have been the subject of considerable research in recent years [81]. For instance, Feng et al. [20] use a t- test-based feature selection method to choose some important genes from thousands of genes. After that, authors classify the microarray data sets with a Fuzzy Neural Network (FNN). The FNN combines important features of initial fuzzy model self-generation, parameter optimization, and rule-base simplification. They applied the FNN to three well-known gene expression data sets, i.e., the lymphoma data set (with 3 sub-types), small round blue cell tumor (SRBCT) data set (with 4 sub-types), and the liver cancer data set (with 2 classes, i.e., non-tumor and hepatocellular carcinoma (HCC)). Their results in all the three data sets show that the FNN can obtain 100% accuracy with a much smaller number of genes in comparison with previously published methods. They reported that in view of the smaller number of genes required by the FNN and its high accuracy,the FNN classifier not only helps biological researchers differentiate cancers that are difficult to be classified using traditional clinical methods, but also helps biological researchers focus on a small number of important genes to find the relationships between those important genes and the development of cancers (see also [117]).

1.6 CI in DNA Fragment Assembly (FA)

The fragment assembly problem (FAP) deals with sequencing of DNA. Currently strands of DNA, longer than approximately 500 base pairs, cannot be sequenced very accurately. As a consequence, in order to sequence larger strands of DNA, they are first broken into smaller pieces. The FAP is then to reconstruct the original molecule's sequence from the smaller fragment sequences. FAP is basically a permutation problem, similar in spirit to the TSP, but with some important differences (circular tours, noise, and special relationships between entities) [94]. Meksangsouy and Chaiyaratana [67] attempted to solve the DNA fragment reordering problem with the ant colony system. The authors investigated two types of assembly problems: single-contig and multiple-contig problems. The simulation results indicate that the ant colony system algorithm outperforms the nearest neighbor heuristic algorithm when multiple-contig problems are considered.

The DNA fragment assembly is a problem to be solved in the early phases of the genome project and thus is critical since the other steps depend on its accuracy. This is an NP-hard combinatorial optimization problem which is growing in importance and complexity as more research centers become involved on sequencing new genomes. Various heuristics, including computational intelligence algorithms, have been designed for solving the fragment assembly problem, but since this problem is a crucial part of any sequencing project, better assemblers are needed. Here we demonstrated some reported examples of using the CI techniques in DNA Fragment Assembly problem.

Wannasak et al. [114] present the use of a combined ant colony system (ACS) and nearest neighbour heuristic (NNH) algorithm in DNA fragment assembly. The assembly process can be treated as combinatorial optimization where the aim is to find the right order of each fragment in the ordering sequence that leads to the formation of a consensus sequence that truly reflects the original DNA strands. The assembly procedure proposed is composed of two stages: fragment assembly and contiguous sequence (contig) assembly. In the fragment assembly stage, a possible alignment between fragments is determined where the fragment ordering sequence is created using the ACS algorithm. The resulting contigs are then assembled together using the NNH rule. Their results indicate that in overall the performance of the combined ACS/NNH technique is superior to that of a standard sequence assembly program (CAP3), which is widely used by many genomic institutions.

Angeler et al. [6] describes an alternative approach to the fragment assembly problem. The key idea is to train a recurrent neural network (RNN) to track a sequence of bases constituting a given fragment and to assign to the same cluster all sequences which are well tracked by this network. The authors make use of a 3-layer Recurrent Perceptron and examine both edited sequences from an ftp site and artificial fragments from a common simulation software. The clusters they obtain exhibit interesting properties in terms of error filtering, stability and self consistency; they define as well, with a certain degree of approximation, a metric on the fragment set. The proposed assembly algorithm is susceptible to becoming an alternative method with the following properties: (i) high quality of the rebuilt genomic sequences, (ii) high parallelizability of the computing process with consequent drastic reduction of the running time.

1.7 CI in Multiple Sequence Alignment (MSA)

Sequence Alignment (SA) refers to the process of arranging the primary sequences of DNA, RNA, or protein to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences. Given two sequences X and Y, a pair-wise alignment indicates positions of each sequence that are considered to be functionally or evolutionarily related. From a family $S = (S_0, S_1, \ldots, S_{N-1})$ of N sequences, we would like to find out common patterns of this family. Since aligning each pair of sequences from S separately often does not reveal the common information, it is necessary to perform multiple sequence alignment (MSA). A multiple sequence alignment (MSA) is a sequence alignment of three or more biological sequences, generally protein, DNA, or RNA. In general, the input set of query sequences are assumed to have an evolutionary relationship by which they share a linkage and are descended from a common ancestor.

To evaluate the quality of an alignment, a popular choice is to use the SP (sum-of-pairs) score method [63]. The SP score basically sums the substitution scores of all possible pair-wise combinations of sequence characters in one column of a multiple sequence alignment. Assuming c_i representing the i^{th} character of a given column in the sequence matrix and match (c_i, c_j) denoting the comparing score between characters c_i and c_j, the score of a column may be computed using the formula:

$$SP = (c_1, c_2, \ldots, c_N) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} match(c_i, c_j) \tag{1.9}$$

Progressive alignment is a heuristic widely used in MSA, but it does not guarantee optimality [28]. ClustalW [105] is another popular program that improved the algorithm presented by Feng and Doolittle [28]. The main shortcoming of ClustalW is that once a sequence has been aligned, that alignment can never be modified even if it conflicts with sequences added later.

Recently, Chen et al. [18] took a serious attempt to solve the classical MSA problem by using a partitioning approach coupled with the Ant Colony Optimization (ACO) algorithm. The algorithm consists of three stages. At first, a genetic algorithm is employed to find out the near optimal cut-off points in the original sequences from where they must be partitioned vertically. In this way a partitioning method is continued recursively to reduce the original problem to multiple smaller MSA problems until the lengths of the subsequences are all less than an acceptable threshold. Next, an ant colony system is used to align each small subsection derived from the previous step. The ant system consists of N ants each of which represents a solution of alignment. Each ant searches for an alignment by moving on the sequences to choose the matching characters. Let the N sequences be $S = S_0, S_1, \ldots, S_{N-1}$. In that case an artificial ant starts from $S_0[0]$, the first character of S_0, and selects one character from each of the sequences of $S_1, \ldots, S_{N-1}$ matching with $S_0[0]$. From the sequence $S_i, i = 1, 2, \ldots, n_1$,the ant selects a character $S_i[j]$ by a probability determined by the matching score with $S_0[0]$, deviation of its location from $S_0[0]$ and pheromones trail on the logical edge between $S_i[j]$ and $S_0[0]$.In addition, an ant may choose to insert an empty space according to a predetermined probability. Next, the ant starts from $S_0[1]$, selects the characters of $S_1, \ldots, S_{N-1}$ matching with $S_0[1]$ to form the second path. Similarly, starting from $S_0[2], \ldots, S_0[|S_0| - 1]$, the ant can form other paths. Here $|S_0|$ indicates the number of characters in the sequence $|S_0|$.

To evaluate an alignment represented by a set of paths, the positions of characters not selected by the ants are calculated first by aligning them to the right and adding gaps to the left. Next their SP (sum-of-pairs) score is using relation (1.9). Finally, a solution to the MSA is obtained by concatenating the results from smaller sub-alignments. The Divide-Ant-MSA algorithm outperformed the SAGA [78], a leading MSA program based on genetic algorithms, in terms of both speed and accuracy especially for longer sequences.

Rasmussen and Krink [88] focussed on a new PSO based training method for Hidden Markov Models (HMMs) in order to solve the MSA problem. The authors showed how a combination of PSO and evolutionary algorithms can generate better protein sequence alignments than with more traditional HMM training methods, such as Baum-Welch [98] and simulated annealing [39].

Genetic algorithm is one of the important and successful approaches in MSA. Zhang and Huang [122] propose an improved GA method, multiple small-popsize initialization strategy (MSPIS) and hybrid one-point crossover scheme (HOPCS) based GA, which can search the solution space in a very efficient manner. The experimental results show that this improved approach can obtain a better result compared with traditional GA approach in aligning multiple protein sequences problem.

DNA matching is a crucial step in sequence alignment. Since sequence alignment is an approximate matching process there is a need for good approximation algorithms. The process of matching in sequence alignment is generally finding longest common subsequences. However, finding the longest common subsequence may not be the best solution for either a database match or an assembly. An optimal alignment of subsequences is based on several factors, such as quality of bases, length of overlap, etc. Factors such as quality indicate if the data is an actual read or an experimental error. Fuzzy logic allows tolerance of inexactness or errors in sub sequence matching. Nasser et al. [75] propose fuzzy logic for approximate matching of subsequences. Fuzzy characteristic functions are derived for parameters that influence a match. Authors develop a prototype for a fuzzy assembler. The assembler is designed to work with low quality data, which is generally rejected by most of the existing techniques. Authors test the assembler on sequences from two genome projects, namely Drosophila melanogaster and Arabidopsis thaliana. Their results are compared with other assemblers. The fuzzy assembler successfully assembled sequences and performed similar and in some cases better than existing techniques.

In multiple DNA sequence alignment, some researchers used divide-and-conquer techniques to cut the sequences for the sake of decreasing complexity. Because the cutting points of sequences of the existing methods are fixed at the middle or near-middle points, the performance of sequence alignment of the existing methods is not good enough. Chen et al. [17] present a new method for multiple DNA sequence alignment using genetic algorithms and divide-and-conquer techniques to choose optimal cut points of multiple DNA sequences. Their experimental results show that the proposed method is better than the existing methods for dealing with multiple DNA sequence alignment.

The similarity judgement of two sequences is often decomposed in similarity judgements of the sequence events with an alignment process. However, in some domains like speech or music, sequences have an internal structure which is important for intelligent processing like similarity judgements. In an alignment task, this structure can be reflected more appropriately by using two levels instead of aligning event by event. This idea is related to the structural alignment framework by Markman and Gentner [34]. Weyde and Klaus [115] introduce a method to align sequences by modeling the segmenting and matching of groups in an input sequence in relation to a target sequence, detecting variations or errors. This is realized as an integrated process, using a neuro-fuzzy system. The selection of segmentations and alignments is based on fuzzy rules which allow the integration of expert knowledge via feature definitions, rule structure, and rule weights. The rule weights can be optimized effectively with an algorithm adapted from neural networks. Thus the results from the optimization process are still interpretable. The system has been implemented and tested successfully in a sample application for the recognition of musical rhythm patterns.

Hiroshi [66] proposes a new method for efficient finding of the biologically optimal alignment of multiple sequences. A key technique used in his method is *deterministic annealing* that attempts to find the global optimum in a parameter space through the annealing process. The author proposes a new simple probabilistic model for the usually time-consuming iterative process of deterministic annealing. Probabilistic parameters of his model are trained from a given sequences based on the deterministic annealing and Expectation Maximization algorithm. When a new sequence is given, this sequence is aligned by parsing it using the trained model. Experimental results show that the proposed method gives a better performance than other competing methods, like a profile hidden markov models, and is time-efficient.

1.8 CI in Protein Structure Prediction (PSP)

Protein Structure Prediction (PSP) is one of the most important goals pursued by bioinformatics and theoretical chemistry. Its aim is prediction of the three-dimensional structure of proteins from their amino acid sequences, sometimes including additional relevant information such as the structures of related proteins [128]. In other words, it deals with the prediction of a protein's tertiary structure from its primary structure. Protein structure prediction is of high importance in medicine (e.g., in drug design) and biotechnology (e.g., in the design of novel enzymes). There have been many successful research projects focusing on this problem. For example, Tang et al. [102] address a problem of predicting protein homology between given two proteins. They propose a learning method that combines the idea of association rules with their previous method called Granular Support Vector Machines (GSVM), which systematically combines a SVM with granular computing. The method, called GSVM-AR, uses association rules with high enough confidence and significant support to find suitable granules to build a GSVM with good performance. The authors compared their method with SVM by KDDCUP04 protein homology prediction data. From the experimental results, GSVM-AR showed significant improvement compared to a single SVM.

The interface between combinatorial optimization and fuzzy sets-based methodologies is the subject of a very active and increasing research. In this context, Balnco et al. [14] describe a fuzzy adaptive neighborhood search (FANS) optimization heuristic that uses a fuzzy valuation to qualify solutions and adapts its behavior as a function of the search state. FANS may also be regarded as a local search framework. The authors show an application of this fuzzy sets-based heuristic to the protein structure prediction problem in two aspects: (1) to analyze how the codification of the solutions affects the results and (2) to confirm that FANS is able to obtain as good results as a genetic algorithm. Both results shed some light on the application of heuristics to the protein structure prediction problem and show the benefits and power of combining basic fuzzy sets ideas with heuristic techniques.

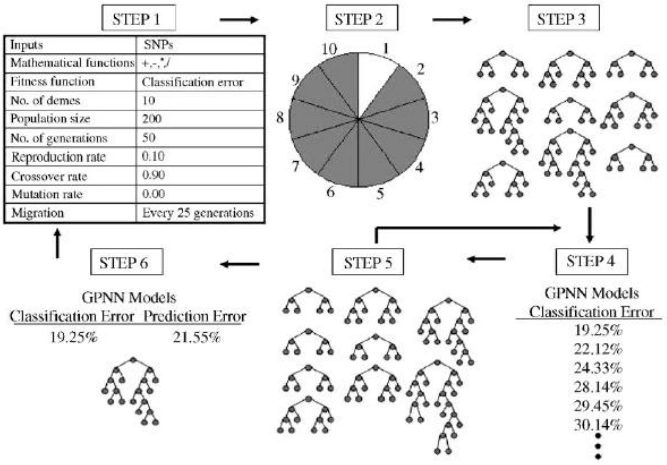
Solving the structure prediction problem for complex proteins is difficult and computationally expensive. Tantar et al. [103] propose a bicriterion parallel hybrid genetic algorithm in order to efficiently deal with the problem using a computational grid. The use of a near-optimal metaheuristic, such as a GA, allows a significant reduction in the number of explored potential structures. However, the complexity of the problem remains prohibitive as far as large proteins are concerned, making the use of parallel computing on the computational grid essential for its efficient resolution. A conjugated gradient-based Hill Climbing local search is combined with the GA in order to intensify the search in the neighborhood of its provided configurations. Authors consider two molecular complexes: (1) the tryptophan-cage protein (Brookhaven Protein Data Bank ID 1L2Y) and (2) a-cyclodextrin. The experimentation results obtained on a computational grid show the effectiveness of their approach.

Predicting the three-dimensional structure of proteins from their linear sequence is one of the major challenges in modern biology. It is widely recognized that one of the major obstacles in addressing this question is that the *standard* computational approaches are not powerful enough to search for the correct structure in the huge conformational space. Genetic algorithms, a cooperative computational method, have been successful in many difficult computational tasks. Thus it is not surprising that in recent years several studies were performed to explore the possibility of using genetic algorithms to address the protein structure prediction problem. Ron Roger [108] reviewed a general framework of how genetic algorithms can be used for structure prediction problem. Using this framework, significant studies that were published in recent years are discussed and compared. Applications of genetic algorithms to the related question of protein alignments are also mentioned. The rationale of why genetic algorithms are suitable for protein structure prediction is presented, and future improvements that are still needed are discussed.

The understanding of protein structures is vital to determine the function of a protein and its interaction with DNA, RNA, and enzymes. The information about its conformation can provide essential information for drug design and protein engineering. While there are over a million known protein sequences, only a limited number of protein structures are experimentally determined. Hence, prediction of protein structures from protein sequences using computer programs is an important step to unveil proteins' three dimensional conformation and functions. As a result, prediction of protein structures has profound theoretical and practical influence over biological study. Pan [80] shows how to use machine learning methods with various advanced encoding schemes and classifiers improve the accuracy of protein structure prediction. The explanation of how a decision is made is also important for improving protein structure prediction. The reasonable interpretation is not only useful to guide the "wet experiments," but also the extracted rules are helpful to integrate computational intelligence with symbolic AI systems for advanced deduction. The author also presents some preliminary results using SVM and decision tree for rule extraction and prediction interpretation.

1.9 CI in Human Genetics

One goal of genetic epidemiology is to identify genes associated with common, complex multifactorial diseases. Success in achieving this goal will depend on a research strategy that recognizes and addresses the importance of interactions among multiple genetic and environmental factors in the etiology of diseases such as essential hypertension [50, 73, 91]. The identification of genes that influence the risk of common, complex disease primarily through interactions with other genes and environmental factors remains a statistical and computational challenge in genetic epidemiology. This challenge is partly due to the limitations of parametric statistical methods for detecting genetic effects that are dependent solely or partially on interactions. Recently, Marylyn et al. [74] took a serious attempt to introduce a genetic programming neural network (GPNN) as a method for optimizing the architecture of a neural network to improve the identification of genetic and gene-environment combinations associated with a disease risk. This empirical studies suggest GPNN has excellent power for identifying gene-gene and gene-environment interactions. In [91] Marylyn et al. continued their study to compare the power of GPNN to stepwise logistic regression (SLR) and classification and regression trees (CART) for identifying gene-gene and gene-environment interactions. SLR and CARTare standard methods of analysis for genetic association studies. Using simulated data,authors show that GPNN has higher power to identify gene-gene and gene-environment interactions than SLR and CART. These results indicate that GPNN may be a useful



Inputs	SNPs
Mathematical functions	+,-,*,/
Fitness function	Classification error
No. of demes	10
Population size	200
No. of generations	50
Reproduction rate	0.10
Crossover rate	0.90
Mutation rate	0.00
Migration	Every 25 generations

Fig. 1.8. The steps of the GPNN algorithm [91]

pattern recognition approach for detecting gene-gene and gene-environment interactions in studies of human disease. We will briefly discuss their approach in the following paragraphs. Their method contains six steps as shown in Figure 1.8 and described in brief as follows.

- *Step-1: Set of GPNN parameters.* GPNN has a set of parameters that must be initialized before beginning the evolution of NN models. These include an independent variable input set, a list of mathematical functions, a fitness function, and finally the operating parameters of the GP. These operating parameters include number of demes (or populations), population size, number of generations, reproduction rate, crossover rate, mutation rate, and migration [90].
- *Step-2: Divide the data based on cross validation.* The data are divided into 10 equal parts for 10-fold cross-validation. Here, we will train the GPNN on 9/10 of the data to develop an NN model. They test this model on the 1/10 of the data left out to evaluate the predictive ability of the model.
- *Step-3: Generate an initial population.* Training of the GPNN begins by generating an initial population of random solutions. Each solution is a binary expression tree representation of an NN.
- *Step-4:GPNN evaluation.* Each GPNN is evaluated on the training set and its fitness recorded.
- *Step-5: The best solutions selection.* The best solutions are selected for crossover and reproduction using a fitness-proportionate selection technique, called roulette wheel selection, based on the classification error of the training data.
- *Step-6: Classification and prediction error.* Classification error is defined as the proportion of individuals where the disease status was incorrectly

specified. A predefined proportion of the best solutions are directly copied (reproduced) into the new generation. Another proportion of the solutions is used for crossover with other best solutions. The new generation, which is equal in size to the original population, begins the cycle again. T'his continues until some criterion is met at which point the GPNN stops.

Another work introduced by Alison et al [74] which developed a grammatical evolution neural network (GENN) approach that accounts for the drawbacks of GPNN. In this study, they show that this new method has high power to detect gene-gene interactions in simulated data. They also, compare the performance of GENN to GPNN, a traditional Back-Propagation Neural Network (BPNN) and a random search algorithm. GENN outperforms both BPNN and the random search, and performs at least as well as GPNN. This study demonstrates the utility of using GE to evolve NN in studies of complex human disease.

1.10 CI in Microarray Classification

A DNA microarray (also commonly known as DNA chip or gene array) is a collection of microscopic DNA spots attached to a solid surface, such as glass, plastic, or silicon chip, forming an array for the purpose of expression profiling, monitoring expression levels for thousands of genes simultaneously. Microarrays provide a powerful basis to monitor the expression of thousands of genes, in order to identify mechanisms that govern the activation of genes in an organism. Short DNA patterns (or binding sites near the genes) serve as switches that control gene expression. Therefore, similar patterns of expression correspond to similar binding site patterns. A major cause of coexpression of genes is their sharing of the regulation mechanism (coregulation) at the sequence level. Clustering of coexpressed genes into biologically meaningful groups helps in inferring the biological role of an unknown gene that is coexpressed with a known gene(s). Cluster validation is essential, from both the biological and statistical perspectives, in order to biologically validate and objectively compare the results generated by different clustering algorithms.

Microarray classification has a broad variety of biomedical applications. Support Vector Machines (SVM) have emerged as a powerful and popular classifier for microarray data. At the same time, there is increasing interest in the development of methods for identifying important features in microarray data. Many of these methods use SVM classifiers either directly in the search for good features or indirectly as a measure of dissociating classes of microarray samples. Peterson and Thaut [85] present study that describes empirical results in model selection for SVM classification of DNA microarray data. Authors demonstrate that classifier performance is very sensitive to the SVM's kernel and model parameters. They also demonstrate that the optimal model parameters depend on the cardinality of feature subsets and can influence the evolution of a genetic search for good feature subsets. Their results suggest that application of SVM classifiers to microarray data should include careful consideration of the space of possible SVM parameters. The results also suggest that feature selection search and model selection should be conducted jointly rather than independently.

Tasoulis et al. [104] study and compare various computational intelligence approaches such as neural networks, evolutionary algorithms, and clustering algorithms, then they demonstrate their applicability as well as their weaknesses and shortcomings to efficient DNA microarray data analysis.

Heterogeneous types of gene expressions may provide a better insight into the biological role of gene interaction with the environment, disease development, and drug effect at the molecular level. Liang and Kelemen [60] proposed a Time Lagged Recurrent Neural Network with trajectory learning for identifying and classifying the gene functional patterns from the heterogeneous nonlinear time series microarray experiments. The proposed procedures identify gene functional patterns from the dynamics of a state-trajectory learned in the heterogeneous time series and the gradient information over time. Also, the trajectory learning with back-propagation through time algorithm can recognize gene expression patterns varying over time. This may reveal much more information about the regulatory network underlying gene expressions. The analyzed data were extracted from spotted DNA microarrays in the budding yeast expression measurements, produced by Eisen et al. [26]. The gene matrix contained 79 experiments over a variety of heterogeneous experiment conditions. The number of recognized gene patterns in our study ranged from two to ten and were divided into three cases. Optimal network architectures with different memory structures were selected based on Akaike and Bayesian information criteria using two-way factorial design. The optimal model performance was compared to other popular gene classification algorithms, such as nearest neighbor, support vector machine, and self-organized maps. The reliability of the performance was verified with multiple iterated runs.

Efficient and reliable methods that can find a small sample of informative genes amongst thousands are of great importance. In this area, much research is devoted to combining advanced search strategies (to find subsets of features), and classification methods [44]. Juliusdottir et al. [49] investigate a simple evolutionary algorithm/classifier combination on two microarray cancer datasets, where this combination is applied twice–once for feature selection, and once for further selection and classification. Their contribution are: (further) demonstration that a simple EA/classifier combination is capable of good feature discovery and classification performance with no initial dimensionality reduction; demonstration that a simple repeated EA/K-NN approach is capable of competitive or better performance than methods using more sophisticated preprocessing and classifier methods; new and challenging results on two public datasets with clear explanation of experimental setup; review material on the EA/K-NN area; and specific identification of genes that their work suggests are significant regarding colon cancer and prostate cancer.

Lin et al. [61] propose a genetic algorithm with silhouette statistics as discriminant function (GASS) for gene selection and pattern recognition. The proposed method evaluates gene expression patterns for discriminating heterogeneous cancers. Distance metrics and classification rules have also been analyzed to design a GASS with high classification accuracy. Moreover, the proposed method is compared to previously published methods. Various experimental results show that their method is effective for classifying the NCI60, the GCM and the SRBCTs datasets. Moreover, GASS outperforms other existing methods in both the leave-one-out cross-validations and the independent test for novel data.

Identification of the short DNA sequence motifs that serve as binding targets for transcription factors is an important challenge in bioinformatics. Unsupervised techniques from the statistical learning theory literature have often been applied to motif discovery, but effective solutions for large genomic datasets have yet to be found. Mahonya et al. [65] present three self-organizing neural networks that have applicability to the motif-finding problem. The core system in this study is a previously described SOM-based otif-finder named SOMBRERO. The motif-finder is integrated in this work with a SOM-based method that automatically constructs generalized models for structurally related motifs and initializes SOMBRERO with relevant biological knowledge. A self-organizing tree method that displays the relationships between various motifs is also presented in this work, and it is shown that such a method can act as an effective structural classifier of novel motifs. The performance of the three self-organizing neural networks is evaluated and analyzed using various datasets.

1.11 Conclusions, Challenges, and Future Directions

Computational Intelligence (CI) has increasingly gained attention in bioinformatics research and computational biology. With the availability of different types of CI algorithms, it has become common for researchers to apply the off-shelf systems to classify and mine their databases. At present, with various intelligent methods available in the literature, scientists are facing difficulties in choosing the best method that could be applied to a specific data set. Researchers need tools, which present the data in a comprehensible fashion, annotated with context, estimates of accuracy, and explanation. The terms bioinformatics and computational biology mean about the same. Recently, however, the US National Institutes of Health (NIH) [126] came up with slightly different definitions, which for the convenience of the reader are repeated below. *Bioinformatics*: Research, development, or application of computational tools and approaches for expanding the use of biological, medical, behavioral, or health data, including those to acquire, store, organize, archive, analyze, or visualize such data. *Computational biology*: The development and application of data-analytical and theoretical methods, mathematical modeling, and computational simulation techniques to the study of biological, behavioral, and social systems.

The goal of motif finding is to detect novel, over-represented unknown signals in a set of sequences. Most widely used algorithms for finding motifs obtain a generative probabilistic representation of the over-represented signals and try to discover profiles that maximize information content score. The major difficulty for these algorithms arises from the fact that the best motif corresponds to the global maximum of a non-convex continuous function. Algorithms like Expectation Maximization (EM) and Gibbs sampling are very sensitive to the initial guesses and only converge to the nearest local maximum. A challenge here is to develop a novel optimization framework that searches the neighborhood regions of the initial alignments in a systematic manner to explore the neighborhood profiles. Algorithms like PSO could lead to new and interesting avenues of research.

The problem of cancer classification is another challenge. It has been divided into two related but separate challenges: class prediction and class discovery [31]. Class prediction refers the assignment of samples to one of several previously defined classes. Class discovery refers to defining a previously unrecognized tumor subtype(s) in expression data. Both of these tasks are challenging and require computational assistance. Class prediction via cluster analysis is typically used to infer the function of novel genes by grouping them with genes of well-known functionality in gene expression profiling. Genes that show similar activity patterns are often related functionally and are controlled by the same mechanisms of regulation. A major obstacle to the eventual utility of microarrays is the lack of efficient methods for cataloging the data into coexpressed groups. A new way of processing numeric data with large number of attributes versus low number of objects turns out to be well-suited to the gene expression data. Furthermore, tumors are not identical–even when they occur in the same organ, and patients may need different treatments depending on their particular subtype of cancer. Identification of tumor subgroups is therefore important for diagnosis and design of medical treatment. Most medical classification systems for tumors are currently based on clinical observations and the microscopical appearance of the tumors. These observations are not informative with regard to the molecular characteristics of the cancer. The genes, whose expression levels are associated with the tumor subtypes, are largely unknown. A better understanding of the cancer could be achieved if these genes were identified. Furthermore, the disease may manifest itself earlier on the molecular level than on a clinical level. Hence, gene expression data from microarrays may enable prediction of tumor subtype and outcome at an earlier stage than clinical examination. Thus microarray analysis may allow earlier detection and treatment of the disease, which again may increase the survival rate.

Most universities and companies have the same reasons for pursuing biomarker research: better diagnosis and better treatment for patients. According to Lynn Rutkowski, co-leader of clinical translational medicine at Wyeth Company (a global leader in pharmaceuticals, consumer health care products, and animal health care products), "*You need a strategy in place, so you have time to do the research you need to fill in gaps and get biomarkers you have confidence in. There are so many technologies emerging. The moment you commit to one, there is another right behind it.*" Both companies and researchers have already considered a new approach of combining imaging technology text mining and biomarkers discovery as a possible solution in future biometric research. For example, Wyeth Company is investing almost $86 million for biomarker discovery, including ten in cardiovascular and metabolic disease, four in inflammation and seven in neuroscience. This company has developed new markers using the 'combine' approach. In stroke, for example, in addition to imaging technology, Wyeth has used rehabilitation tools to measure patients' responses. A robotic instrumentation has been used for therapy that can also provide a quantitative assessment of motor-function recovery. Another example includes Alzheimer's disease (AD). AD has 11 compounds in development. One of these compounds is FK962. The company's long-term strategy involves molecular markets, structural and functional brain imaging, and physiological, behavioral, and associative learning tests.

Another challenge is to combine gene expression research with noninvasive imaging techniques. Eran Segal [93] and his collaborators hypothesized that the global gene expression patterns of human cancers may systematically correlate with their dynamic imaging features [93]. To address the challenges of relating gene expression to imaging, the researches followed a three step methodology and created an association map between imaging features on tree-phase contrast enhanced CT scans and gene expression patterns of 28 human hepatocellular carcinomas (HCC). First, the researchers defined and quantified 138 *units of distinctiveness* named *traits* present in one or more HCCs. Second, the module networks algorithm was implemented. The algorithm systematically search for associations between expression levels of 6,732 well-measured genes determined by mycroarary analysis and combinations of imaging traits. Third, the statistical significance of the association map was validated by comparison with permuted data sets, and by testing the prediction of the association map in an independent set of tumors.

Paralleling the diversity of genetic and protein activities pathologic human tissues also exhibit diverse radiographic features. It is proven that dynamic imaging trails in noninvasive computer tomography (CT) systematically correlate with the global gene expression profiles. For example: the association map of imaging traits and gene expression revealed that a large fraction of the gene expression program can be reconstructed from a small number of image trails. The expression variation in 6,732 genes was captured by 116 gene modules, each of which was associated with specific combination of imaging trails. For each module, the presence or absence of combination of imaging traits explained the aggregate expression level of genes within the module. The combinations of relevant imaging trials are depicted in decision trees: each split in the tree is specified by variation of an imaging trait, each terminal leaf in the tree is a cluster of samples that share a similar expression pattern of module genes. Thus the association map allowed the user to reconstruct the relative expression level of a gene (by mapping it to a module) in a given HCC sample (by mapping it to a cluster) Across all 116 gene modules capturing 6,732 genes in the presented model, the difference in the level of expression of member genes from their cognate module averages is 1.36- 1.33 fold. Thus the expression level of individual genes can be reconstructed from imaging features with an average deviation of about twofold, within the experimental determination level allowed by microarray analysis. The experiment shows that only 8 imaging traits are sufficient to reconstruct the variation of all 116 gene modules [93].

The term cyber-infrastracture has been established by US National Science Foundation (NSF) to address the needs for new mechanisms of information handling and exchange. Eric Neumann, Director of Clinical Semantics Group at MIT, has presented the following project as an example of text mining research: NeuroCommons is a project within Science Commons at MIT. This project is using text mining to extract neuro-molecular relations from text mining, representing them as RDF (Resource Description Framework). SWAN (Semantic Web Applications in Neuromedicine) is an NIH-funded project that allows scientists to directly annotate knowledge onto findings using RDF. The user interface consists of a SPARQL–a query page that permits a wide variety of questions regarding genes, neurological diseases, microanatomy, and publications. Examples include: "Find all publications with neural dendrites in their description;" "Show all genes expressed in brain region CA1 involved in signal transduction;" "Find all papers on Parkinson's disease that involve gene products localized in the nucleus;" etc. Results can be formatted as tables. In RDF additional tools can process the data for enhanced scientific view. Tool such as Google can also be applied to the output from a query. The future of cyberinfrastarcture for bioinformatics and biomedical research is becoming a reality: a connected research community more effectively utilizing data and computational resources from different areas.

Also, intelligent support is essential for managing and interpreting this great amount of information. One of the well-known constraints specifically related to microarray data is the large number of genes in comparison with the small number of available experiments. In this context, the ability of design methods capable of overcoming current limitations of state-of-the-art algorithms is crucial to the development of successful applications.

A combination of computational intelligence techniques in application to bioinformatics and computational biology has become one of the most important areas of research in intelligent information processing [24]. Neural networks show their strong ability to solve complex problems for many bioinformatics problems. From the perspective of specific rough sets approaches that can be applied, exploration into possible applications of hybridizing rough sets with other intelligent systems like neural networks, genetic algorithms, fuzzy logic, etc. to bioinformatics and computational biology could lead to new and interesting avenues of research. Moreover, algorithms like PSO or ACO and their variants involve a large degree of randomness and different runs of the same program may yield different results; so it is necessary to incorporate problem specific domain knowledge in the Swarm Intelligence tools to reduce randomness and computational time and current research should progress in this direction as well.

The main purpose of this chapter was to present to the CI and bioinformatics and computational biology research communities the state of the art in CI applications to bioinformatics and computational biology, and to inspire further research and development on new applications and new concepts in new trend-setting directions and in exploiting computational intelligence.

References

1. Abraham, A.: Intelligent systems: Architectures and perspectives, recent advances in intelligent paradigms and applications. In: Abraham, A., Jain, L., Kacprzyk, J. (eds.) Studies in Fuzziness and Soft Computing, pp. 1–35. Springer, Heidelberg (2002)
2. Abraham, A.: Nature and scope of AI techniques. In: Sydenham, P., Thorn, R. (eds.) Handbook for Measurement Systems Design, pp. 893–900. John Wiley and Sons Ltd., Chichester (2005)
3. Alba, E., Luque, G.: A New Local Search Algorithm for the DNA Fragment Assembly Problem. In: Cotta, C., van Hemert, J. (eds.) EvoCOP 2007. LNCS, vol. 4446, pp. 1–12. Springer, Heidelberg (2007)
4. Alon, U., et al.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc. Natl. Acad. Sci. USA, Cell Biology 96, 6745–6750 (1999)
5. Altman, R.B., Valencia, A., Miyano, S., Ranganathan, S.: Challenges for intelligent systems in biology. IEEE Intelligent Systems 16(6), 14–20 (2001)
6. Angeleri, E., Apolloni, B., de Falco, D., Grandi, L.: DNA Fragment assembly using neural prediction techniques. Intl. J. Neural Systems 9(6), 523–544 (1999)
7. Arima, C., Hanai, T.: Gene expression analysis using Fuzzy k-Means Clustering. Genome Informatics 14, 334–335 (2003)
8. Back, T.: Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic algorithms. Oxford University Press, Oxford (1996)
9. Baker, T.K., et al.: Temporal gene expression analysis of monolayer cultured rat hepatocytes. Chem. Res. Toxicol. 14(9), 1218–1231 (2001)
10. Baldi, P., Brunak, S.: Bioinformatics: The Machine Learning Approach. MIT Press, Cambridge (1998)
11. Baldi, P., Hatfield, G.W.: DNA Microarrays and Gene Expression: From Experiments to Data Analysis and Modeling. Cambridge University Press, Cambridge (2002)
12. Banerjee, M., Mitra, S., Banka, H.: Evolutionary rough feature selection in gene expression data. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 37(4), 622–632 (2007)
13. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
14. Blanco, A., Pelta, D.A., Verdegay, J.L.: Applying a fuzzy sets-based heuristic to the protein structure prediction problem. Intl. J. Intelligent Systems 17(7), 629–643 (2002)
15. Bull, L., Kovacs, T. (eds.): Foundations of Learning Classifier Systems. Studies in Fuzziness and Soft Computing, 183 (2005)
16. Busa-Fekete, R., Kocsor, A., Pongor, S.: Tree-Based Algorithms for Protein Classification. Studies in Computational Intelligence 94, 165–182 (2008)
17. Chen, S.-M., Lin, C.-H., Chen, S.-J.: Multiple DNA sequence alignment based on genetic algorithms and divide-and-conquer techniques. Intl. J. Applied Science and Engineering 3(2), 89–100 (2005)

18. Chen, Y., Pan, Y., Chen, L., Chen, J.: Partitioned optimization algorithms for multiple sequence alignment. In: Proc. 20th Intl. Conf. on Advanced Information Networking and Applications, pp. 618–622 (2006)
19. Chena, C.-B., Wang, L.-Y.: Rough set-based clustering with refinement using Shannon's entropy theory. Computers and Mathematics with Applications 52(10-11), 1563–1576 (2006)
20. Chu, F., Xie, W., Wang, L.: Gene selection and cancer classification using a fuzzy neural network. In: Proc. IEEE Annual Meeting of Fuzzy Information, pp. 555–559 (2004)
21. Chuang, H.-Y., Lee, E., Liu, Y.-T., Lee, D., Ideker, T.: Network-based classification of breast cancer metastasis. Molecular Systems Biology 3(140) (2007)
22. Cios, K.J., Mamitsuka, H., Nagashima, T., Tadeusiewicz, R.: Computational intelligence in solving bioinformatics problems. Artificial Intelligence in Medicine 35(1-2), 1–8 (2005)
23. Cohen, J.: Bioinformatics: An introduction for computer scientists. ACM Computing Surveys 36(2), 122–158 (2004)
24. Das, S., et al.: Swarm Intelligence Algorithms in Bioinformatics. Studies in Computational Intelligence 94, 113–147 (2008)
25. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and unsupervised discritization of continuous features. In: Proc. XII Intl. Conf. on Machine Learning, pp. 294–301 (1995)
26. Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. PNAS 95(25), 14863–14868 (1998)
27. Ezziane, Z.: Applications of artificial intelligence in bioinformatics: A review. Expert Systems with Applications 30, 2–10 (2006)
28. Feng, D.F., Doolittle, R.F.: Progressive sequence alignment as a prerequisite to correct phylogenetic trees. J. Mol. Evol. 25, 351–360 (1987)
29. Fernando, D., Fdez-Riverola, F., Glez-Pea, D., Corchado, J.M.: Using fuzzy patterns for gene selection and data reduction on microarray data. In: Corchado, E., Yin, H., Botti, V., Fyfe, C. (eds.) IDEAL 2006. LNCS, vol. 4224, pp. 1087–1094. Springer, Heidelberg (2006)
30. Fogel, D.B.: Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. IEEE Press, Los Alamitos (1999)
31. Fogel, G.B.: Gene expression analysis using methods of computational intelligence. Pharmaceutical Discovery 5(8), 12–18 (2005)
32. Fogel, L.J., Owens, A.J., Walsh, M.J.: Artificial Intelligence Through Simulated Evolution. John Wiley & Sons, Chichester (1967)
33. Futschik, M.E., Kasabov, N.K.: Fuzzy clustering of gene expression data. In: Proc. 2002 IEEE Intl. Conf. on Fuzzy Systems, pp. 414–419 (2002)
34. Gentner, D., Markman, A.B.: Structure mapping in analogy and similarity. American Psychologist 52(1), 45–56 (1997)
35. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Publishing, Reading (1989)
36. Golub, T., et al.: Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. Science 286(5439), 531–537 (1999)
37. Gruźdź, A., Ihnatowicz, A., Ślęzak, D.: Interactive Gene Clustering: A Case Study of Breast Cancer Microarray Data. Information Systems Frontiers 8(1), 21–27 (2006)
38. Gusfield, D.: Introduction to the IEEE/ACM transactions on computational biology and bioinformatics. IEEE/ACM Transactions on Computational Biology and Bioinformatics 1(1), 2–3 (2004)

39. Hamam, Y., Al-Ani, T.: Simulated annealing approach for Hidden Markov Models. In: Proc. 4th WG-7.6 Working Conf. on Optimization-Based Computer-Aided Modeling and Design, ESIEE, France (1996)
40. Hassnein, A.-E., Abdelhafez, M., Own, H.: Rough sets data analysis: A case of Kuwaiti diabetic children patients. In: Advances in Fuzzy Systems (in press)
41. He, Y., Tang, Y., Zhang, Y.-Q., Sunderraman, R.: Fuzzy-granular gene selection from microarray expression data. In: Proc. 6th IEEE Intl. Conf. on Data Mining - Workshops, pp. 153–157 (2006)
42. Herrero, J., Valencia, A., Dopazo, J.: A hierarchical unsupervised growing neural network for clustering gene expression patterns. Bioinformatics 17(2), 126–136 (2001)
43. Holland, J.: Adaptation in Natural and Artificial Systems. University of Michigan Press (1975)
44. Hong, J.-H., Cho, S.-B.: The classification of cancer based on DNA microarray data that uses diverse ensemble genetic programming. Artificial Intelligence in Medicine 36, 43–58 (2006)
45. Huang, C.-J., Liao, W.-C.: A comparative study of feature selection methods for probabilistic neural networks in cancer classification. In: Proc. 15th IEEE Intl. Conf. on Tools with Artificial Intelligence, p. 451 (2003)
46. Hunga, C.-M., Huanga, Y.-M., Changb, M.-S.: Alignment using genetic programming with causal trees for identification of protein functions. Nonlinear Analysis 65, 1070–1093 (2006)
47. Hwang, K.B., Cho, D.Y., Wook Park, S.W., Kim, S.D., Zhang, B.Y.: Applying machine learning techniques to analysis of gene expression data: Cancer diagnosis. In: Proc. 1st Conf. on Critical Assessment of Microarray Data Analysis (2000)
48. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. ACM Computing Surveys 31(3), 264–323 (1999)
49. Juliusdottir, T., Keedwell, E., Corne, D., Narayanan, A.: Two-phase EA/k-NN for feature selection and classification in cancer microarray datasets. In: Proc. 2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, pp. 1–8 (2005)
50. Kardia, S.L.R.: Context-dependent genetic effects in hypertension. Curr. Hypertens. Rep. 2, 32–38 (2000)
51. Kelemen, A., Abraham, A., Chen, Y. (eds.): Computational Intelligence in Bioinformatics. Studies in Computational Intelligence. Springer, Heidelberg (2008)
52. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc. IEEE Intl. Conf. on Neural Networks, pp. 1942–1948 (1995)
53. Kennedy, J.: Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. In: Proc. 1999 Congress of Evolutionary Computation, pp. 1931–1938 (1999)
54. Kennedy, J., Eberhart, R., Shi, Y.: Swarm Intelligence. Morgan Kaufmann Academic Press, San Francisco (2001)
55. Khan, J., et al.: Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. Nat. Med. 7(6), 673–679 (2001)
56. Kohonen, T.: Self-organizing maps. Springer, Heidelberg (1995)
57. Koza, J.R.: Genetic Programming. MIT Press, Cambridge (1992)
58. Li, D., Zhang, W.: Gene selection using rough set theory. In: Wang, G.-Y., Peters, J.F., Skowron, A., Yao, Y. (eds.) RSKT 2006. LNCS (LNAI), vol. 4062, pp. 778–785. Springer, Heidelberg (2006)

59. Li, L., Weinberg, C.R., Darden, T.A., Pedersen, L.G.: Gene selection for sample classification based on gene expression data: Study of sensitivity to choice of parameters of the GA/KNN method. Bioinformatics 17, 1131–1142 (2001)
60. Liang, Y., Kelemen, A.: Time course gene expression classification with time lagged recurrent neural network. Studies in Computational Intelligence 94, 149–163 (2008)
61. Lin, T.-C., et al.: Pattern classification in DNA microarray data of multiple tumor types. Pattern Recognition 39(12), 2426–2438 (2006)
62. Lingras, P.: Applications of rough set based k-means, Kohonen SOM, GA Clustering. In: Peters, J.F., Skowron, A., Marek, V.W., Orłowska, E., Słowiński, R., Ziarko, W. (eds.) Transactions on Rough Sets VII. LNCS, vol. 4400, pp. 120–139. Springer, Heidelberg (2007)
63. Lipman, D.J., Altschul, S.F., Kececioglu, J.D.: A tool for multiple sequence alignment. Proc. Natl. Acad. Sci. USA 86, 4412–4415 (1989)
64. Luscombe, N.M., Greenbaum, D., Gerstein, M.: What is Bioinformatics? A proposed definition and overview of the field. Yearbook of Medical Informatics, 83–100 (2001)
65. Mahonya, S., Benosa, P.V., Smithd, T.J., Goldend, A.: Self-organizing neural networks to support the discovery of DNA-binding motifs. Neural Networks 19, 950–962 (2006)
66. Mamitsuka, H.: Finding the biologically optimal alignment of multiple sequences. Artificial Intelligence in Medicine 35(1-2), 9–18 (2005)
67. Meksangsouy, P., Chaiyaratana, N.: DNA fragment assembly using an ant colony system algorithm. In: Proc. Congress on Evolutionary Computation (2003)
68. Midelfart, H., Komorowski, J., Nørsett, K., Yadetie, F., Sandvik, A.K., Lægreid, A.: Learning rough set classifiers from gene expressions and clinical data. Fundamenta Informaticae 53, 155–183 (2002)
69. Mitra, S.: An evolutionary rough partitive clustering. Pattern Recognition Letters 25, 1439–1449 (2004)
70. Mitra, S., Hayashi, Y.: Bioinformatics with soft computing. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 36, 616–635 (2006)
71. Mitra, S., Banka, H., Paik, J.H.: Evolutionary fuzzy biclustering of gene expression data. In: Yao, J., Lingras, P., Wu, W.-Z., Szczuka, M.S., Cercone, N.J., Ślęzak, D. (eds.) RSKT 2007. LNCS (LNAI), vol. 4481, pp. 284–291. Springer, Heidelberg (2007)
72. Mohamed, S., Rubin, D., Marwala, T.: Multi-class Protein Sequence Classification Using Fuzzy ARTMAP. In: Proc. IEEE Intl. Conf. on Systems, Man, and Cybernetics, pp. 1676–1681 (2006)
73. Moore, J.H., Williams, S.M.: New strategies for identifying gene-gene interactions in hypertension. Ann. Med. 34, 88–95 (2002)
74. Motsinger, A.A., Dudek, S.M., Hahn, L.W., Ritchie, M.D.: Comparison of Neural Network Optimization Approaches for Studies of Human Genetics. In: Rothlauf, F., Branke, J., Cagnoni, S., Costa, E., Cotta, C., Drechsler, R., Lutton, E., Machado, P., Moore, J.H., Romero, J., Smith, G.D., Squillero, G., Takagi, H. (eds.) EvoWorkshops 2006. LNCS, vol. 3907, pp. 103–114. Springer, Heidelberg (2006)
75. Nasser, S., Vert, G.L., Nicolescu, M., Murray, A.: Multiple Sequence Alignment using Fuzzy Logic. In: Proc. IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology, pp. 304–311 (2007)

76. Nguyen, H.S.: Approximate Boolean reasoning: Foundations and applications in data mining. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets V. LNCS, vol. 4100, pp. 334–506. Springer, Heidelberg (2006)
77. Ning, S., Ziarko, W., Hamilton, J., Cercone, N.: Using rough sets as tools for knowledge discovery. In: Proc. 1st Intl. Conf. on Knowledge Discovery and Data Mining, pp. 263–268 (1995)
78. Notredame, C., Higgins, D.G.: SAGA: sequence alignment by genetic algorithm. Nucleic Acids Research 24(8), 1515–1524 (1996)
79. Okada, Y., et al.: Knowledge-assisted recognition of cluster boundaries in gene expression data. Artificial Intelligence in Medicine 35(1-2), 171–183 (2005)
80. Pan, Y.: Protein structure prediction and understanding using machine learning methods. In: Proc. IEEE Intl. Conf. on Granular Computing, pp. 13–20 (2005)
81. Paul, T.K.: Gene expression based cancer classification using evolutionary and non-evolutionary methods. Technical Report No. 041105A1, Dept. of Frontier Informatics, University of Tokyo, Japan (2004)
82. Pawlak, Z.: Rough sets. Intl. J. Comp. Inform. Science 11, 341–356 (1982)
83. Pawlak, Z.: Rough Sets – Theoretical Aspects of Reasoning About Data. Kluwer, Dordrecht (1991)
84. Pawlak, Z., Grzymala-Busse, J., Slowinski, R., Ziarko, W.: Rough sets. Communications of the ACM 38(11), 88–95 (1995)
85. Peterson, D.A., Thaut, M.H.: Model and feature selection in microarray classification Peterson. In: Proc. IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, pp. 56–60 (2004)
86. Polkowski, L.: Rough Sets: Mathematical Foundations. Physica-Verlag, Heidelberg (2003)
87. Quackenbush, J.: Computational analysis of microarray data. National Review of Genetics 2, 418–427 (2001)
88. Rasmussen, T.K., Krink, T.: Improved Hidden Markov Model training for multiple sequence alignment by a particle swarm optimization-evolutionary algorithm hybrid. BioSystems 72, 5–17 (2003)
89. Raychaudhuri, S., Stuart, J.M., Altman, R.B.: Principal components analysis to summarize microarray experiments: Application to sporulation rime series. In: Proc. Pacific Symposium on Biocomputing, pp. 452–463 (2000)
90. Ritchie, M.D., et al.: Optimization of neural network architecture using genetic programming improves detection of gene-gene interactions in studies of human diseases. BMC Bioinformatics 4(28) (2003)
91. Ritchie, M.D., et al.: Genetic programming neural networks: A powerful bioinformatics tool for human genetics. Applied Soft Computing 7, 471–479 (2007)
92. Ruffino, F., Costacurta, M., Muselli, M.: Evaluating switching neural networks for gene selection. In: Masulli, F., Mitra, S., Pasi, G. (eds.) WILF 2007. LNCS (LNAI), vol. 4578, pp. 557–562. Springer, Heidelberg (2007)
93. Segal, E., et al.: Decoding global gene expression programs in liver cancer by noninvasive imaging. Nature Biotechnology 25, 675–680 (2007)
94. Setubal, J., Meidanis, J.: Introduction to Computational Molecular Biology. Intl Thomson Publishing (1999)
95. Ślęzak, D., Wróblewski, J.: Rough Discretization of Gene Expression Data. In: Proc. 2006 Intl. Conf. on Hybrid Information Technology, pp. 265–267 (2006)
96. Ślęzak, D., Wróblewski, J.: Roughfication of numeric decision tables: The case study of gene expression data. In: Yao, J., Lingras, P., Wu, W.-Z., Szczuka, M.S., Cercone, N.J., Ślęzak, D. (eds.) RSKT 2007. LNCS (LNAI), vol. 4481, pp. 316–323. Springer, Heidelberg (2007)

97. Spellman, E.M., Brown, P.L., Brown, D.: Cluster analysis and display of genome-wide expression patterns. Proc. Natl. Acad. Sci. USA 95, 14863–14868 (1998)
98. Stolcke, A., Omohundro, S.: Hidden Markov Model induction by Bayesian model merging. NIPS 5, 11–18 (1993)
99. Sun, L., Miao, D., Zhang, H.: Gene selection with rough sets for cancer classification. In: Proc. 4th Intl. Conf. on Fuzzy Systems and Knowledge Discovery, pp. 167–172 (2007)
100. Sushmita, M.: An evolutionary rough partitive clustering. Pattern Recognition Letters 25, 1439–1449 (2004)
101. Tamayo, P., et al.: Interpreting patterns of gene expression with self organizing maps: Methods and applications to hematopoietic differentiation. PNAS 96, 2907–2912 (1999)
102. Tang, Y., Jin, B., Zhang, Y.-Q.: Granular support vector machines with association rules mining for protein homology prediction. Artificial Intelligence in Medicine 35(1-2), 121–134 (2005)
103. Tantar, A.A., Melab, N., Talbi, E.G., Parent, B., Horvath, D.: A parallel hybrid genetic algorithm for protein structure prediction on the computational grid. Future Generation Computer Systems 23(3), 398–409 (2007)
104. Tasoulis, D.K., Plagianakos, V.P., Vrahatis, M.N.: Computational intelligence algorithms and DNA microarrays. Studies in Computational Intelligence 94, 1–31 (2008)
105. Thompson, J.D., Higgins, D.G., Gibson, T.J.: CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice. Nucleic Acids Research 22(22), 4673–4680 (1994)
106. Tomida, S., Hanai, T., Honda, H., Kobayashi, T.: Gene expression analysis using Fuzzy ART. Genome Informatics 12, 245–246 (2001)
107. Toronen, P., Kolehmainen, M., Wong, G., Castren, E.: Analysis of gene expression data using self-organizing maps. FEBS letters 451, 142–146 (1999)
108. Unger, R.: The genetic algorithm approach to protein structure prediction. Structure and Bonding 110, 153–175 (2004)
109. Valdes, J.J., Barton, A.J.: Relevant attribute discovery in high dimensional data: Application to breast cancer gene expressions. In: Wang, G.-Y., Peters, J.F., Skowron, A., Yao, Y. (eds.) RSKT 2006. LNCS (LNAI), vol. 4062, pp. 482–489. Springer, Heidelberg (2006)
110. van de Vijver, M.J., et al.: A gene-expression signature as a predictor of survival in breast cancer. N. Engl. J. Med. 347, 1999–2009 (2002)
111. Wang, D., Lee, N.K., Dillon, T.S.: Extraction and optimization of fuzzy protein sequences classification rules using GRBF neural networks. Neural Information Processing - Letters and Reviews 1(1), 53–57 (2003)
112. Wang, Y., et al.: Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. Lancet 365, 671–679 (2005)
113. Wen, X., et al.: Large scale temporal gene expression mapping of cns development. Proc. Natl. Acad. Sci. USA, Neurobiology 95, 334–339 (1998)
114. Wetcharaporn, W., Chaiyaratana, N., Tongsima, S.: DNA fragment assembly by ant colony and nearest neighbour heuristics. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 1008–1017. Springer, Heidelberg (2006)
115. Weyde, T., Dalinghaus, K.: A neuro-fuzzy system for sequence alignment on two levels. Mathware and Soft Computing XI(2-3), 197–210 (2004)

116. Xiao, X., Dow, E.R., Eberhart, R.C., Miled, Z.B., Oppelt, R.J.: Gene clustering using self-organizing maps and particle swarm optimization. In: Proc. 17th Intl. Symposium on Parallel and Distributed Processing (2003)
117. Xie, W., Chu, F., Wang, L.: Fuzzy neural network applications for gene selection and cancer classification. In: Proc. Artificial Intelligence and Soft Computing (2004)
118. Yang, Q., Wu, X.: Challenging problems in data mining research. Intl. J. Information Technology and Decision Making 5(4), 597–604 (2006)
119. Yeung, K.Y., Ruzzo, W.L.: Principal component analysis for clustering gene expression data. Bioinformatics 17, 763–774 (2001)
120. Yuhui, Y., Lihui, C., Goh, A., Wong, A.: Clustering gene data via associative clustering neural network. In: Proc. 9th Intl. Conf. on Information Processing, pp. 2228–2232 (2002)
121. Zadeh, L.A.: Fuzzy sets. Information and Control 8, 338–353 (1965)
122. Zhang, G.-Z., Huang, D.-S.: Aligning multiple protein sequence by an improved genetic algorithm. In: Proc. IEEE Intl. Joint Conf. on Neural Networks, pp. 1179–1183 (2004)
123. Zhang, J., Lee, R., Wang, Y.J.: Support vector machine classifications for microarray expression data set. In: Proc. 5th Intl. Conf. on Computational Intelligence and Multimedia Applications, pp. 67–71 (2003)
124. Zhang, Q.: An approach to rough set decomposition of incomplete information systems. In: Proc. 2nd IEEE Conf. on Industrial Electronics and Applications, pp. 2455–2460 (2007)
125. Ziarko, W.: Variable precision rough sets model. J. Computer and Systems 46(1), 39–59 (1993)
126. NIH: `http://www.bisti.nih.gov` (last accessed December 2007)
127. Special Issue on Bioinformatics. IEEE Computer 35 (July 2002)
128. `http://en.wikipedia.org/wiki/DNA_microarray` (last accessed December 2007)