Bioinformatics With Soft Computing

Sushmita Mitra, *Senior Member, IEEE,* and Yoichi Hayashi, *Senior Member, IEEE*

***Abstract*—Soft computing is gradually opening up several possibilities in bioinformatics, especially by generating low-cost, low-precision (approximate), good solutions. In this paper, we survey the role of different soft computing paradigms, like fuzzy sets (FSs), artificial neural networks (ANNs), evolutionary computation, rough sets (RSes), and support vector machines (SVMs), in this direction. The major pattern-recognition and data-mining tasks considered here are clustering, classification, feature selection, and rule generation. Genomic sequence, protein structure, gene expression microarrays, and gene regulatory networks are some of the application areas described. Since the work entails processing huge amounts of incomplete or ambiguous biological data, we can utilize the learning ability of neural networks for adapting, uncertainty handling capacity of FSs and RSes for modeling ambiguity, searching potential of genetic algorithms for efficiently traversing large search spaces, and the generalization capability of SVMs for minimizing errors.**

***Index Terms*—Artificial neural networks (ANNs), biological data mining, fuzzy sets (FSs), gene expression microarray, genetic algorithms (GAs), proteins, rough sets (RSes), support vector machines (SVMs).**

I. Introduction

BIOINFORMATICS [1], [2] can be defined as the application of computer technology to the management of biological information, encompassing a study of the inherent genetic information, underlying molecular structure, resulting biochemical functions, and the exhibited phenotypic properties. One needs to analyze and interpret the vast amount of data that are available, involving the decoding of around 24 000–30 000 human genes. *Biological data mining* is an emerging field of research and development, posing challenges and providing possibilities in this direction [3].

Proteins constitute an important ingredient of living beings and are made up of a sequence of amino acids. The determination of an optimal three-dimensional (3-D) conformation constitutes protein folding. It is a highly complex process, providing enormous information on the presence of active sites and possible drug interaction. To establish how a newly formed *polypeptide* sequence of amino acids finds its way to its correct fold out of the countless alternatives is one of the greatest challenges in modern structural biology.

Proteins in different organisms that are related to one another by evolution from a common ancestor are called *homologs*. This relationship can be recognized by multiple sequence comparisons. A similar primary structure leads to a similar 3-D structure, resulting in a similar functionality of the proteins. Since the traditional dynamic programming method for local alignment is too slow, the basic local alignment search tool (BLAST) [4] is often found to be more efficient. BLAST is a heuristic method to find the highest locally optimal alignments between a query sequence and a database. BLAST improves the overall speed of search while retaining good sensitivity, by breaking the query and database sequences into fragments (words) and initially seeking matches between these fragments. Although BLAST does not permit the presence of gaps in between, its extension Gapped BLAST [5] allows insertions and deletions to be introduced into alignments. Another efficient extension to BLAST is position-specific iterative BLAST (Psi-BLAST) [5], which includes gaps while searching for distant homologies by building a profile (general characteristics).

Typically, these algorithms compare an unseen protein sequence with existing identified sequences, and return the highest match. However, as the size of the protein sequence databases is very large, it is very time-consuming to perform exhaustive comparison therein. Therefore, one categorizes these sequences into evolutionarily related protein superfamilies that are functionally as well as structurally relevant to each other. This allows molecular analysis to be done within a particular superfamily, instead of handling the entire sequence database. Phylogenetic analysis of sequences, in terms of their taxonomic relationships, is yet another important area of research.

Unlike a genome, which provides only static sequence information, microarray experiments produce gene expression patterns that offer dynamic information about cell function. This information is useful while investigating complex interactions within the cell. Gene expression data being typically high dimensional, it requires appropriate data-mining strategies like feature selection and clustering for further analysis.

Biological networks relate genes, gene products, or their groups (like protein complexes or protein families) to each other in the form of a graph, where nodes and edges correspond to molecules and their existing interrelationships, respectively. Metabolic networks depict a set of chemical reactions, mostly catalyzed by enzymes, and are extremely important for gene expression profiling. This is because the link between the gene regulatory control and the primary causative factors of diseases (like altered protein activities or biochemical composition of cells) is often crucial for application in drug development, medicine, nutrition, and other therapeutic activities. Clustering of gene expression patterns is also being used to generate gene regulatory networks [6].

In addition to the combinatorial approach, there also exists scope for soft computing, especially for generating low-cost, low-precision (approximate), good solutions. Soft computing is a consortium of methodologies that works synergistically and provides flexible information-processing capability for handling

Manuscript received October 6, 2004; revised June 8, 2005 and September 23, 2005. This paper was recommended by Associate Editor Y. Jin.

S. Mitra was with the Department of Computer Science, Meiji University, Kawasaki 214-8571, Japan. She is now with the Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700 108, India (e-mail: sushmita@isical.ac.in).

Y. Hayashi is with the Department of Computer Science, Meiji University, Kawasaki 214-8571, Japan (e-mail: hayashiy@cs.meiji.ac.jp).

Digital Object Identifier 10.1109/TSMCC.2006.879384

real-life ambiguous situations [7]. The main constituents of soft computing, at this juncture, include fuzzy logic, neural networks, genetic algorithms (GAs), rough sets (RSes), and support vector machines (SVMs). Since the work entails processing huge amounts of incomplete or ambiguous data, the learning ability of neural networks, uncertainty handling capacity of FSs and RSes, and the searching potential of GAs can be utilized for this purpose [8]. SVMs have been recently categorized as another component of soft computing [9], mainly due to their learning and generalization capabilities in a data-rich environment.

In this paper, we provide a survey on the role of soft computing in modeling various aspects of bioinformatics involving genomic sequence, protein structure, gene expression microarray, and gene regulatory networks. Major tasks of pattern recognition and data mining, like clustering, classification, feature selection, and rule generation, are considered. While classification pertains to supervised learning, in the presence of known targets, clustering corresponds to unsupervised self-organization into homologous partitions. Feature selection techniques aim at reducing the number of irrelevant and redundant variables in the dataset. Rule generation enables efficient representation of mined knowledge in human-understandable form.

The rest of the paper is organized as follows. Section II introduces the basics from biology and soft computing that are relevant to our subsequent discussion. The major problems of bioinformatics, covered in Sections III–VI, deal with primary genomic sequence, protein structure, microarray, and gene regulatory networks, respectively. The different techniques of soft computing considered include FSs, artificial neural networks (ANNs), GAs, evolutionary programming, RSes, SVMs, and various hybridizations like neuro-fuzzy (NF) models. The categorization is made on the basis of the domain and functions modeled. Finally, Section VII concludes the paper.

II. Preliminaries

Proteins are built up by polypeptide chains of amino acids, which consist of deoxyribonucleic acid (DNA) as the building block. In this section, we provide a basic understanding of the protein structure, folding, DNA microarray data, biological networks, and soft computing that are relevant to this article.

A. DNA

The nucleus of a cell contains chromosomes that are made up of the double helical DNA molecules. DNA consists of two strands, each being a string of four nitrogenous bases, viz., adenine (A), cytosine (C), guanine (G), and thymine (T). DNA in the human genome is arranged into 24 distinct chromosomes. Each chromosome contains many genes, the basic physical and functional units of heredity. However, genes comprise only about 2% of the human genome; the remainder consists of noncoding regions, whose functions may include providing chromosomal structural integrity and regulating where, when, and in what quantity proteins are made.

DNA is *transcribed* to produce messenger (m)-RNA, which is then *translated* to produce protein. The m-RNA is single-stranded and has a ribose sugar molecule. There exist "Promoter" and "Termination" sites in a gene, responsible for the initiation and termination of transcription. Translation consists of mapping from triplets (codons) of four bases to the 20 amino acids building block of proteins. Enzymes and hormones are also proteins.

B. Proteins

An amino acid is an organic molecule consisting of an amine (NH) and a carboxylic (CO) acid group (backbone), together with a side-chain (hydrogen atom and residue R) that differentiates between them. Proteins are polypeptides, formed within cells as a linear chain of amino acids. Chemical properties that distinguish the 20 different amino acids cause the protein chains to fold up into specific 3-D structures that define their particular functions in the cell.

Given the *primary* structure of a protein, in terms of a linear sequence of amino acids, folding attempts to predict its stable 3-D structure. However, considering all interactions governed by the laws of physics and chemistry to predict 3-D positions of different atoms in the protein molecule, a reasonably fast computer would need one day to simulate 1 ns of folding. Protein folding is a thermodynamically determined problem. It is also a reaction involving other interacting amino acids and water molecules.

The two-dimensional (2-D) *secondary* structure can involve an α-helix (with the CO group of the ith residue hydrogen (H)-bonded to the NH group of the $(i+4)$th one) or a β-sheet (corrugated or hairpin structure) formed by the H-bonds between the amino acids. The parts of the protein that are not characterized by any regular H-bonding patterns are called random coils or turns.

The *tertiary* structure refers to the 3-D conformation of the protein. The objective is to determine the minimum energy state for a polypeptide chain folding. The process of protein folding involves minimization of an energy function, which is expressed in terms of several variables like bond lengths, bond angles, and torsional angles. The major factors affecting folding include: 1) hydrogen bonding; 2) hydrophobic effect; 3) electrostatic interactions; 4) Van der Waals' forces; and 5) conformational entropy. One common scheme of classification categorizes tertiary structures into five groups, viz., all α (mainly α-helix secondary structure), all β (mainly β-sheet secondary structure), $\alpha+\beta$ (segment of α-helices followed by segment of β-sheets), α/β (alternating or mixed α-helix and β-sheet segments), and the remaining irregular secondary structural arrangements.

Protein binding sites exhibit highly selective recognition of small organic molecules, utilizing features like complex 3-D *lock* (active sites) into which only specific *keys* (drug molecules or enzymes) will *dock*. Any solution to the docking problem requires a powerful search technique to explore the conformation space available to the protein and *ligand*, along with a good understanding of the process of molecular recognition to devise scoring functions for reliably predicting binding modes.

C. Microarrays

Reverse-transcribed m-RNA or cDNA microarrays (gene arrays or gene chips) [2] usually consist of thin glass or nylon substrates containing specific DNA gene samples spotted in

an array by a robotic printing device. This measures the relative m-RNA abundance between two samples, which are labeled with different fluorescent dyes, viz., red and green. The m-RNA binds (hybridizes) with cDNA[1] probes on the array. The relative abundance of a spot or gene is measured as the logarithmic ratio between the intensities of the dyes, and constitutes the gene expression data.

Gene expression levels can be determined for samples taken: 1) at multiple time instants of a biological process (different phases of cell division) or 2) under various conditions (e.g., tumor samples with different histopathological diagnosis). Each gene corresponds to a high-dimensional vector of its expression profile. The data contain a high level of noise due to experimental procedures. Moreover, the expression values of single genes demonstrate large biological variance within tissue samples from the same class.

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.

D. Biological Networks

Processes that generate mass, energy, information transfer, and cell-fate specification, in a cell or microorganism, are seamlessly integrated through a complex network of cellular constituents and reactions. Such a metabolic network consists of nodes, i.e., substrates (genes or proteins), that are interconnected through links, i.e., metabolic reactions in which enzymes provide the catalytic scaffolds. The degree of interconnectivity of the network may be characterized by its diameter, which is the shortest biochemical pathway averaged over all pairs of substrates. The topology of a network reflects a long evolutionary process molded for a robust response toward internal defects and environmental fluctuations. Despite significant variation of individual constituents and pathways, metabolic networks have the same topological scaling properties and exhibit striking similarities to the inherent organization of complex, robust nonbiological systems [10].

The Kyoto Encyclopedia of Genes and Genomes (KEGG) database [11] provides a public standardized annotation of genes.[2] It is a knowledge base for systematic analysis of gene functions in terms of the networks of genes and molecules. The data objects in KEGG are represented as graphs, and various computational methods are developed to detect graph features that can be related to biological functions. For example, it can: 1) reconstruct biochemical pathways from the complete genome sequence; 2) predict gene regulatory networks from gene expression profiles, obtained by microarray experiments; and 3) determine colinearity of genes between two genomes, for identification of clusters of orthologous genes (which are functionally related/physically coupled/evolutionarily correlated across organisms). The genome is a graph of genes that are one-dimensionally connected, while the pathway is a graph of gene products.

E. Soft Computing

The principal notion in soft computing is that precision and certainty carry a cost, and that computation, reasoning, and decision-making should exploit (wherever possible) the tolerance for imprecision, uncertainty, approximate reasoning, and partial truth for obtaining low-cost solutions.

A fuzzy set A in a space of points $R = \{r\}$ is a class of events with a continuum of grades of membership, and it is characterized by a membership function $\mu_A(r)$ that associates with each element in R a real number in the interval $[0, 1]$ with the value of $\mu_A(r)$ at r representing the grade of membership of r in A. FSs provide a natural framework for the process in dealing with uncertainty or imprecise data.

ANNs [12] are signal-processing systems that try to emulate the behavior of biological nervous systems by providing a mathematical model of combination of numerous neurons connected in a network. The learning capability and robustness of ANNs, typically in data-rich environments, come in handy when discovering regularities from large datasets. This can be unsupervised as in clustering, or supervised as in classification. The connection weights and topology of a trained ANN are often analyzed to generate a mine of meaningful (comprehensible) information about the learned problem in the form of rules. There exist different ANN-based learning and rule-mining strategies, with applications to the biological domain [8]. Some of the major ANN models include perceptron, multilayer perceptron (MLP), radial basis function (RBF) network, Kohonen's self-organizing map (SOM), and adaptive resonance theory (ART).

There has been research in the judicious integration of ANN and FSs, by augmenting each other in order to build more intelligent information systems. The NF computing paradigm [13] often results in better recognition performance than that obtained by individual technologies. This incorporates both the generic and application-specific merits of ANNs and fuzzy logic into hybridization.

The theory of RSes [14] is a major mathematical tool for managing uncertainty that arises from granularity in the domain of discourse—that is, from the indiscernibility between objects in a set. The intention is to approximate a *rough* (imprecise) concept in the domain of discourse by a pair of *exact* concepts, called the lower and upper approximations. The lower approximation is the set of objects definitely belonging to the vague concept, whereas the upper approximation is the set of objects possibly belonging to the same.

GAs [15] are adaptive and robust computational search procedures, modeled on the mechanics of natural genetic systems. They operate on string representation of possible solutions in terms of individuals or chromosomes containing the features. The components of a GA consist of: 1) a population

[1]Single-stranded DNA that is complementary to m-RNA or DNA that has been synthesized from messenger RNA by the enzyme reverse transcriptase.

[2]http://www.genome.ad.jp/kegg/

of individuals; 2) encoding or decoding mechanism of the individuals; 3) objective function and an associated fitness evaluation criterion; 4) selection procedure; 5) genetic operators like recombination or crossover, and mutation; 6) probabilities to perform the genetic operations; 7) replacement technique; and 8) termination conditions. Unlike GAs, evolutionary algorithms [16] rely only on mutation and do not perform crossover.

Another evolutionary strategy, often used in bioinformatics, is *genetic programming* (GP). This invokes exertion of evolutionary pressure on a program to make it evolve, thereby discovering optimal computer programs resulting in innovative solutions to problems [17]. The principle of operation is similar to GAs, with the focus shifting to evolving programs rather than candidate solutions. GP solutions are computer programs represented as tree structures that are probabilistically selected according to their fitness in solving the candidate problem. These are then modified with genetic operators (crossover and mutation) to generate new solutions.

SVMs are a general class of learning architectures, inspired by statistical learning theory, that perform *structural risk minimization* on a nested set structure of separating hyperplanes [18]. Given a training data, the SVM learning algorithm generates the optimal separating hyperplane (between positive and negative examples) in terms of generalization error. As a by-product of learning, it obtains a set of support vectors (SVs) that characterizes a given classification task or compresses a labeled dataset.

In the following sections, we highlight the role of different soft computing paradigms [8], [19]–[22] like FSs, ANNs, GAs, RSes, SVMs, and their hybridizations (including NF), in different areas of bioinformatics.

III. Primary Genomic Sequence

Eukaryotic[3] genes are typically organized as exons (coding regions) and introns (noncoding regions). Hence, the main task of gene identification, from the primary genomic sequence, involves coding region recognition and splice junction[4] detection. Sequence data are typically dynamic and order-dependent. A protein sequence motif is a signature or consensus pattern that is embedded within sequences of the same protein family. Identification of the motif leads to classification of an unknown sequence into a protein family for further biological analysis. Available protein motif databases include PROSITE[5] and PFAM.

Sequence motif discovery algorithms can follow: 1) string alignment; 2) exhaustive enumeration; and 3) heuristic methods. String alignment algorithms detect sequence motifs by minimizing a cost function that is related to the edit distance between the sequences. Multiple alignment of sequences is an NP-hard problem, with its computational complexity increasing exponentially with sequence size. Local search algorithms may lead to local optima instead of the best motif. Exhaustive enumeration algorithms, though guaranteed to find the optimal motif, are computationally too expensive. Here lies the utility of using soft computing techniques for arriving at faster convergence. An overview of their applications in modeling different functions, related to primary genomic sequences, is provided in Table I.

TABLE I
Application of Soft Computing to Primary Genomic Sequences

Biological function modeled	Soft computing paradigm	Reference
Coding region identification	ANN	[23], GRAIL [24], GeneParser [25]
	ANN + GA	[21]
Cleavage site identification	ANN	[26]–[28]
Splice junction detection	ANN	[29]–[33]
Protein family classification	ANN	[34]–[37], ELM [38]
	SVM	[39]
Motif classification	ANN	[40], [41]
	NF	[42]
	GP	[43]
Phylogenetic analysis	ANN	[44], [45]
	GA	[46], [47]
Sequence clustering	FS	[48]
	ANN	[49], [50], SOTA [51]
Multiple sequence alignment	GA	SAGA [52], COFFEE [53]
Sequence reconstruction	GA	[21]
Homology detection	SVM	[39]
Imprecision modeling	FS	[54]
Subcellular localization of proteins	SVM	[55], [56]

A. FSs

Imprecise knowledge of a nucleic acid or a protein sequence of length N has been modeled by a fuzzy biopolymer [54]. This is a fuzzy subset of kN elements, with $k = 4$ bases for nucleic acids and $k = 20$ amino acids for proteins. Profiles, a class of biopolymers generated by multiple alignment of a group of related sequences based on matrices of frequencies, were considered in the study. A sequence is represented as a vector in a unit hypercube (corresponding to an FS) that assigns to each position–monomer pair the possibility with which the monomer (base or amino acid) appears in this position. The midpoint of a pair of fuzzy biopolymers of the same length is interpreted as an average of the knowledge of the sequences represented by them.

A systematic verification and improvement of underlying profiles has been undertaken [48], using fuzzy c-means clustering for contextual analysis. Here, the authors investigate the recognition of potential transcription factor binding sites in genomic sequences.

B. ANNs

The popularity of ANNs in genomic sequence analysis is mainly due to the involvement of high-dimensional space with complex characteristics, which is difficult to model satisfactorily using parameterized approaches. We describe here the role

[3]Organisms (except viruses, bacteria, and algae) having well-developed subcellular compartments, including a discrete nucleus.

[4]Splice junctions are positions at which, after primary transcription of the DNA into RNA, the introns of a gene are excised to form edited m-RNA.

[5]http://www.expasy.ch/sprot/sprot-top.html

of different models, like SOM, MLP, recurrent network, counterpropagation, RBF network, ART, and their combination with other soft computing techniques, in gene identification.

1) MLP: Perceptrons were used to predict coding regions in fixed-length windows [23] with various input encoding methods, including binary encoding of codon and dicodon frequency, and the performance was found to be superior to Bayesian statistical prediction. Perceptrons have also been employed to identify cleavage sites in protein sequences [26], with the physicochemical features (of 12 amino acid residues) like hydrophobicity, hydrophilicity, polarity, and volume serving as the input. However, single-layer perceptrons are limited to linearly separable classification problems.

The MLP has been employed for both classification as well as rule generation.

a) Classification: An MLP, with backpropagation learning, was used to identify exons in DNA sequences in GRAIL [24]. Thirteen input features used include sequence length, exon GC composition, Markov scores, splice site (donor/acceptor) strength, surrounding intron character, etc., calculated within a fixed 99-nucleotide sequence window and scaled to lie between 0 and 1. A single output indicated whether a specific base, central to the said window, was either coding or noncoding.

A three-layered MLP, with binary encoding at input, was employed to predict acceptor and donor site positions in splice junctions of human genomic DNA sequences [29]. A joint assignment, combining coding confidence level with splice site strength, was found to reduce the number of false positives.

Prediction of the exact location of transcription initiation site has been investigated [30] in mammalian promoter regions, using MLP with different window sizes of input sequence. MLPs were also employed to predict the translation initiation sites [31], with better results being generated for bigger windows on the input sequence. Again, some of the limitations of MLPs, like convergence time and local minima, need to be appropriately handled in all these cases.

Protein classification into 137–178 *superfamilies* with a modular architecture involving multiple independent MLPs [34], included 400–1356 input features like counts of amino acid pairs, counts of exchange group pairs and triplets, and other encoded combinations using singular value decomposition. Multiple network modules run in parallel to scale up the system. This sort of divide-and-conquer strategy facilitates convergence.

b) Rule generation: Identification of important binding sites, in a peptide involved in pain and depression, has been attempted [32] using feedforward ANNs. Rules in M-of-N form are extracted by detecting positions in the DNA sequence where changes in the stereochemistry give rise to significant differences in the biological activity. Browne *et al*. also predict splice site junctions in human DNA sequences, which has a crucial impact on the performance of gene finding programs. Donor sites are nearly always located immediately preceding a GT sequence, while acceptor sites immediately follow an AG sequence. Hence, GT and AG pairs within a DNA sequence are markers for potential splice junction sites, and the objective is to identify which of these sites correspond to the real sites followed by prediction of likely genes and gene products. The resulting rules are shown to be reasonably accurate and roughly comparable to those obtained by an equivalent $C5$ decision tree,[6] while being simpler at the same time.

Rules were also generated from a pruned MLP [33], using a penalty function for weight elimination, to distinguish donor and acceptor sites in the splice junctions from the remaining part of the input sequence. The pruned network consisted of only 16 connection weights. A smaller network leads to better generalization capability as well as easier extraction of simpler rules. Ten rules were finally obtained in terms of AG and GT pairs.

2) SOM: Kohonen's SOM has been used for the analysis of protein sequences [35], involving identification of protein families, aligned sequences, and segments of similar secondary structure, with interactive visualization. Other applications of SOM include prediction of cleavage sites in proteins [27], prediction of beta-turns [36], classification of structural motifs [40], and feature extraction [41].

Clustering of human protein sequences into families were investigated [49] with a 15×15 SOM, and the performance was shown to be better than that using statistical nonhierarchical clustering. The study demonstrated that hidden biological information contained in sequence protein databases can be well organized using SOMs.

The self-organizing tree algorithm (SOTA) is a dynamic binary tree that combines the characteristics of SOMs and divisive hierarchical clustering. SOTA has been employed for clustering protein sequences [51] and amino acids [50]. However, if the available training data is too small to be adequately representative of the actual dataset then the performance of the SOM is likely to get affected.

An unsupervised growing self-organizing ANN [44] has been developed for the phylogenetic analysis of a large number of sequences. The network expands itself following the taxonomic relationships existing among the sequences being classified. The binary tree topology of this model enables efficient classification of the sequences. The growing characteristic of this procedure allows termination at the desired taxonomic level, thereby overcoming the necessity of waiting for the generation of a complete phylogenetic tree. The time for convergence is approximately a linear function of the number of sequences being modeled.

3) RBF: A novel extension to the RBF is designed by using the concept of biological similarity between amino acid sequences [28], [57]. Since most amino acid sequences have preserved local motifs for specific biological functions, the numerical RBFs are replaced here by certain such nonnumerical (bio-) basis functions. The neural network leads to reduced computational cost along with improved prediction accuracy. Applications are provided on prediction of cleavage sites as well as the characterization of site activity in the human immunodeficiency virus (HIV) protease. The knowledge of these sites can be used to search for inhibitors (antiviral drugs) that block the cleavage ability of the enzyme. The prediction accuracy is reported to be 93.4%.

[6]http://www.spss.com/spssbi/clementine//

4) ART: Multiple layers of an adaptive resonance theory 2 (ART2) network have been used to categorize DNA fragments [45] at different resolution levels, similar to a phylogenetic (evolutionary) analysis. The ART network trains fast, and incrementally adapts to new data without needing to review old instances. However, the ability to generalize is limited by the lack of a hidden layer.

5) Integration With Other Techniques: Benefits often accrue from using a combination of different learning strategies. A modified counterpropagation network, with supervised learning vector quantization (LVQ) performing nearest-neighbor classification, was used for molecular sequence classification [37].

Dynamic programming has been combined with MLP in GeneParser [25] to predict gene structure. Sequence information is weighted by the MLP to approximate the log-likelihood that each subinterval exactly represents an intron or exon. Dynamic programming is then applied to determine the combination of introns and exons that maximizes the likelihood function. Input to the network consists of the differences for each statistic between the correct and incorrect solutions, and the difference in the number of predicted sequence types. The output maximizes the difference between correct and incorrect solutions.

Evolving ANNs for discriminating between functional elements associated with coding nucleotides (exons) and noncoding sequences of DNA (introns and intragenic spacer) has been reported [21]. The connection weights of a fixed MLP architecture are evolved for classification, using evolutionary computation, with practical application to gene detection. Performance of the evolved network is compared to that of GRAIL [24] and GeneParser [25].

Extreme learning machine (ELM), a new machine learning paradigm with a sigmoidal activation function and Gaussian RBF kernel for the single hidden-layer feedforward neural network, has been used to classify protein sequences from ten classes of superfamilies [38]. The classification accuracy is reported to be better, along with a shorter training time, as compared to that of an MLP of similar size using backpropagation. Since the ELM does not involve any control parameters like learning rate, learning epochs, stopping criteria, that require to be tuned as in MLP, this promises an added advantage.

C. NF

Extraction of motif from a group of related protein sequences has been investigated in an NF framework [42], using data from PROSITE. A statistical method is first used to detect short patterns occurring with high frequency. Fuzzy logic enables the design of approximate membership functions and rules about protein motifs, as obtained from domain experts. An RBF neural network is employed to optimize the classification by tuning the membership functions.

D. GAs

GAs and GP have been primarily applied to primary genomic sequences for functions involving their alignment, reconstruction, and detection. This is described later.

1) Alignment: The simultaneous alignment of many amino acid sequences is one of the major research areas of bioinformatics. Given a set of homologous sequences, multiple alignments can help predict secondary or tertiary structures of new sequences. GAs have been used for this purpose [52]. Fitness is measured by globally scoring each alignment according to a chosen objective function, with better alignments generating a higher fitness. The cost of multiple alignment A_c is expressed as

$$A_c = \sum_{i=1}^{N-1} \sum_{j=1}^{N} W_{i,j} \ \text{cost}(A_i, A_j) \tag{1}$$

where N is the number of sequences, A_i is the aligned sequence i, $\text{cost}(A_i, A_j)$ is the alignment score between two aligned sequences A_i and A_j, and $W_{i,j}$ is the weight associated with that pair of sequences. The cost function includes the sum of the substitution costs, as given by a substitution matrix, and the cost of insertions/deletions using a model with affine gap (gap-opening and gap-extension) penalties. Roulette wheel selection is carried out among the population of possible alignments, and insertion/deletion events in the sequences are modeled using a *gap insertion* mutation operator.

Given N aligned sequences $A_1, \ldots, A_N$ in a multiple alignment, with $A_{i,j}$ being the pairwise projection of sequences A_i and A_j, $\text{length}(A_{i,j})$ the number of ungapped columns in this alignment, $\text{score}(A_{i,j})$ the overall consistency between $A_{i,j}$ and the corresponding pairwise alignment in the library, and $W'_{i,j}$ the weight associated with this pairwise alignment, the fitness function was modified [53] to

$$F = \frac{\sum_{i=1}^{N-1} \sum_{j=1}^{N} W'_{i,j} \times \text{score}(A_{i,j})}{\sum_{i=1}^{N-1} \sum_{j=1}^{N} W'_{i,j} \times \text{length}(A_{i,j})}. \tag{2}$$

The main difference with (1) is the library, which replaces the substitution matrix and provides position-dependent means of evaluation.

2) Reconstruction: The generation of accurate DNA sequence is a challenging and time-consuming problem in genomics. A widely used technique in this direction is *hybridization*, which detects all oligonucleotides[7] of a given length k (usually eight to ten bases) that make up the corresponding DNA fragment. The oligonucleotide library is very large, containing 4^k elements, with microarray chip technology being often used in its implementation. However, the hybridization experiment introduces both negative (missing oligonucleotides) and positive (erroneous oligonucleotide) errors in the spectrum of elements. The reconstruction of the DNA sequence, from these errors, is an NP-hard combinatorial problem. GAs have been successfully applied to difficult instances of sequence reconstruction [21], with a fitness function maximizing the number of elements chosen from the spectrum (subject to a restriction on the maximum length n) of the sequence of nucleotides. The representation of a candidate solution is in terms of a permutation of indices of oligonucleotides from the spectrum.

[7] A short sequence of the four nucleotide bases, A, C, T, G.

3) Detection: GP has been combined with finite state automata (FSA) to discover candidate promoter sequences in primary sequence data [43]. FSAs are directed graphs that can represent powerful grammars in the Chomsky hierarchy, and Turing machines. In *GP-Automata*, a GP-tree structure is associated with each state of the FSA. The method is able to take large base pair jumps, thereby being able to handle very long genomic sequences in order to discover gene-specific *cis*-acting sites[8] as well as genes that are regulated together. It is to be noted that an aim of drug discovery is to identify *cis*-acting sites responsible for coregulating different genes.

The training dataset[9] consists of known promoter regions, while nonpromoter examples constitute samples from the coding or intron sequences. The objective of the GP-tree structure, in each state of the GP-Automata, is to find motifs within the promoter and nonpromoter regions. The terminal set includes A, C, T, and G. The method automatically discovers motifs of various lengths in automata states, and combines motif matches using logical functions to arrive at a *cis*-acting region identification decision.

Phylogenetic inference has been attempted using GA [46] and parallel GA [47]. An individual in a population is a hypothesis consisting of the tree, branch lengths, and parameters values for the model of sequence evolution, while the fitness is the likelihood score of the hypothesis. In the parallel version [47], each individual in a population is handled by one processor or node that computes its corresponding likelihood. This operation being extremely time-consuming, the parallelization at this level causes a nearly linear-order search time improvement for large data. The number of processors used is equal to the size of the evolving population, plus an additional processor for the control of operations. Selection is accomplished on the maximum-likelihood score; migration and recombination is permitted between subpopulations; and mutation can be branch-length based or topological. Results are provided on 228 taxa of DNA sequence data.

E. SVMs

Remote homology detection by quantifying the similarity between protein sequences has been attempted using SVMs [39], for the purpose of superfamily recognition in the Structural Classification of Proteins (SCOP) database. The data consist of 4352 sequences extracted from the *Astral* database. Local alignment kernels are adapted from the Smith–Waterman algorithm for strings. These kernels measure the similarity between two sequences, by summing up scores obtained from local alignments with gaps of the sequences.

Proteins can be classified into 12 subcellular locations, viz., chloroplast, cytoplasm, cytoskeleton, endoplasmic reticulum, extracellular, Golgi apparatus, lysosome, mitochondria, nucleus, peroxisome, plasma membrane, and vacuole. Since the subcellular location of a protein strongly influences its functionality, therefore its proper prediction from the sequence is of utmost importance. A novel concept of functional domain composition [55] has been designed to generate the representative vector base of proteins in their high-dimensional space. The SVM is subsequently used to predict the protein subcellular location. Another systematic approach to predicting subcellular localization of human proteins [56] combines SVM with Psi-BLAST. While SVM modules work on amino acid and dipeptide compositions, the Psi-BLAST helps in performing similarity search.

[8]A major *cis*-acting region in both *prokaryotes* and *eukaryotes* is located just upstream of a gene's transcription start site, and is known as the *promoter* region. The promoter attracts a *holoenzyme* that catalyzes production of RNA from the DNA template. At the promoter, the complex attaches to DNA strands to initiate genetic transcription.

[9]http://www.fruitfly.org/

TABLE II
APPLICATION OF SOFT COMPUTING TO PROTEIN STRUCTURE

Biological function modeled	Soft computing paradigm	Reference
Secondary structure prediction	ANN	[59]–[68]
	neuro-GA	[69]
	SVM	[70], [71]
Protein functional prediction	SVM	[72], [73]
Tertiary structure prediction	ANN	[74]–[81]
	GA	[21], [82]
	SVM	[83]
Tertiary structure alignment	FS	[84]
	GA	[85]
Protein folding	GA	[86]–[88]
	EP	[89]
	SVM	[90], [91]
Docking	GA	GOLD [92], AutoDock [93], GEMDOCK [94], [95]

IV. PROTEIN STRUCTURE

Protein structure prediction typically uses experimental information stored in protein structural databases, like the Brookhaven National Laboratory Protein Data Bank (PDB) [58]. A common approach is based on sequence alignment with structurally known proteins. The experimental approach involving X-ray crystallographic analysis and nuclear magnetic resonance (NMR) being expensive and time-consuming, soft computing techniques offer an innovative way to overcome some of these problems. Table II summarizes their application to protein structure prediction.

A. Secondary Structure

A step on the way to a prediction of the full 3-D structure of protein is predicting the local conformation of the polypeptide chain, called the secondary structure. The whole framework was pioneered by Chou and Fasmann [96]. They used a statistical method, with the likelihood of each amino acid being one of the three (alpha, beta, coil) secondary structures estimated from known proteins.

1) ANNs: In this section, we highlight the enhancement in prediction performance of ANNs, with the use of ensembles and the incorporation of alignment profiles.

The data consist of proteins obtained from the PDB. A fixed-size window constitutes the input to the feedforward ANN. The network predicts the secondary structure corresponding to the

TABLE III
COMPARATIVE PERFORMANCE FOR PROTEIN SECONDARY STRUCTURE PREDICTION

Approach	Reported overall per-residue accuracy (%)	Reported MCC
MLP [59]	64.3	0.41, 0.31, 0.41
MLP + multiple sequence alignment [61]	70.8	0.60, 0.52, 0.51
MLP ensemble + Softmax [64]	71.3	0.59, 0.50, 0.41
Recurrent network ensemble + Psi-BLAST [67]	about 75	–
SVM [70]	73.5	0.64, 0.52, 0.51
SVM + Psi-BLAST [71]	75.2	–

centrally located amino acid of the sequence within the window. The contextual information about the rest of the sequence in the window is also considered during network training. A comparative study of performance of different approaches, on this data, is provided in Table III.

Around 1988, the first attempts were made by Qian and Sejnowski [59] to use MLP with backpropagation to predict protein secondary structure. Three output nodes correspond to the three secondary structures. Performance is measured in terms of an overall correct classification Q (64.3%) and Matthews correlation coefficient (MCC). We have

$$Q = \sum_{i=1}^{l} w_i Q_i = \frac{C}{N} \tag{3}$$

for an l-class problem, with Q_i indicating the accuracy for the ith class, w_i being the corresponding normalizing factor, N representing the total number of samples, and C being the total number of correct classifications.

$$\text{MCC} = \frac{(\text{TP} \times \text{TN}) - (\text{FP} \times \text{FN})}{\sqrt{(\text{TP}+\text{FP})(\text{TP}+\text{FN})(\text{TN}+\text{FP})(\text{TN}+\text{FN})}} \tag{4}$$

where TP, TN, FP, and FN correspond to the number of true positive, true negative, false positive, and false negative classifications, respectively. Here, $N = \text{TP} + \text{TN} + \text{FP} + \text{FN}$ and $C = \text{TP} + \text{TN}$, and $-1 \leq \text{MCC} \leq +1$ with $+1(-1)$ corresponding to a perfect (wrong) prediction. The values for MCC for the α-helix, β-strand, and random coil were found to be 0.41, 0.31, and 0.41, respectively.

The performance of this method was improved by Rost and Sander [60], [61], by using a cascaded three-level network with multiple-sequence alignment. The three levels correspond to a sequence-to-structure net, a structure-to-structure net, and a jury (combined output) decision, respectively. Correct classification increased to 70.8%, with the MCC being 0.60, 0.52, and 0.51, respectively, for the three secondary classes. *Supersecondary* structures (folding units), like $\alpha\alpha$- and $\beta\beta$-hairpins, and $\alpha\beta$- and $\beta\alpha$-arches, serve as important building blocks for protein tertiary structure. Prediction of supersecondary structures was made from protein sequences [62] using MLP. The size of the input vector was the same as the length of the sequence window. There were 11 networks, each with one output, for classifying one of the 11 types of frequently occurring motifs. A test sequence was assigned to the motif category of the winning network having the largest output value. Results demonstrated more than 70% accuracy.

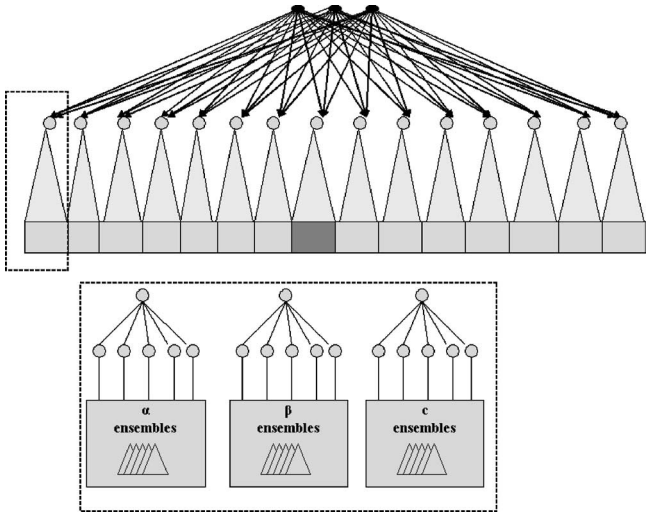

Fig. 1. Secondary protein structure prediction using ensemble of ANNs.

Hybrid approaches to applications related to protein secondary structure also exist in literature. A knowledge-based approach was employed to extract inference rules about a biological problem that were then used to configure ANNs [63]. Integration with GAs was attempted to generate an optimal ANN topology [69], and its performance on secondary structure prediction was found to be comparable to that of Qian and Sejnowski [59].

2) Ensemble Networks: Prediction of protein secondary structure has been further developed by Riis and Krogh [64], with ensembles of combining networks, for greater accuracy in prediction. The *Softmax* method is used to provide simultaneous classification of an input pattern into multiple classes. A normalizing function at the output layer ensures that the three outputs always sum to one. A logarithmic likelihood cost function is minimized, instead of the usual squared error. An adaptive weight encoding of the input amino acid residues reduces the overfitting problem. A window is selected from all the single structure networks in the ensemble. The output is determined for the central residue, with the prediction being chosen as the largest of the three outputs normalized by Softmax.

The use of ensembles of small, customized subnetworks is found to improve predictive accuracy. Customization involves incorporation of domain knowledge into the subnetwork structure for improved performance and faster convergence. For example, the helix-network has a built-in period of three residues in its connections in order to capture the characteristic periodic structure of helices. Fig. 1 provides the schematic network structure. Overall accuracy increased to 71.3%, with the MCC becoming 0.59, 0.50, and 0.41, respectively, for the three secondary classes.

3) Use of Alignment Profile: The alignment profile generated by Psi-BLAST has been incorporated by Jones [65] to design a set of cascaded ANNs. These profiles enable finding more distant sequences, use a more rigorous statistical approach

for computing the probability of each residue at a specific position, and properly weigh each sequence with respect to the amount of information it carries.

Prediction of segments in protein sequences containing aromatic–backbone NH interactions[10] has been attempted [66]. Such interactions help in the stabilization of protein secondary and tertiary structures as well as folding, on the basis of their spatial distribution. Incorporation of evolutionary information in the form of multiple alignment, by Psi-BLAST, enhances the performance in terms of MCC. Two consecutive three-layered feedforward sequence-to-structure and structure-to-structure networks, trained by backpropagation, are employed. It is observed that a segment (window) of seven residues provides sufficient input information for prediction of these aromatic–NH interactions. The actual position of donor aromatic residue within the *potential* predicted fragment is also identified, using a separate sequence-to-structure neural network. The implementation was made on a nonredundant dataset of 2298 protein chains extracted from the Protein Data Bank (PDB).

Ensembles of bidirectional recurrent neural network architectures are used in conjunction with profiles generated by Psi-BLAST to predict protein secondary structure for a given amino acid sequence [67]. The classification decision is determined by three component networks. In addition to the standard central component associated with a local window at location t of the current prediction (as in feedforward ANNs), there exist contribution by two similar recurrent networks corresponding to the left and right contexts (like wheels rolling from the N[11]- and C[12]-terminals along the protein chain). An ensemble of 11 networks are trained, using backpropagation. Two output categorizations are followed, viz., 1) three classes (α-helix, β-strand, random coil), as in SSpro and 2) eight classes as in DSSP[13] programs. The output error is the relative entropy between the output and target probability distributions. At the alignment level, the use of Psi-BLAST, with the ability to produce profiles that include increasingly remote homologs, enhances performance as compared to that employing only BLAST [68]. The system was implemented on proteins from the PDB, which are at least 30 amino acids long, have no chain breaks, produce a DSSP output, and are obtained by X-ray diffraction methods with high resolution. The accuracy of secondary structure prediction is thereby enhanced to about 75%.

4) SVMs: Hua and Sun [70] reported the first use of SVMs to protein secondary structure prediction. A segment overlap measure provides a more realistic assessment of the quality of a prediction, and a useful *reliability index* has been developed. Results are provided on a database of 513 nonhomologous protein chains with multiple sequence alignment. The performance is comparable to that of ANN-based approaches [61], with overall per-residue accuracy being 73.5% and the MCC computed as 0.64, 0.52, 0.51, respectively, for α-helices, β-strands, and random coils. Whereas for ANNs one needs to choose an appropriate topology, the SVM requires the selection of a kernel function. In this case, the RBF has been used. An optimal window length is found to be proportional to the average length of the secondary structure segments. This was extended in [71] by combining a dual-layer SVM with Psi-BLAST. The outputs represented the probability of a residue belonging to that class. Here, the overall accuracy increased to 75.2%.

Proteins of a specific functional family share common structural and chemical features and, given sufficient samples, an SVM can be trained to recognize proteins possessing the characteristics of a particular function. Enzymes represent the largest and most diverse group of all proteins, catalyzing chemical reactions in the metabolism of all organisms. SVM has been used to classify enzymes into functional families [72], as defined by the *Enzyme Nomenclature Committee* of IUBMB. While positive samples correspond to enzymes belonging to a particular family, the negative samples constitute representative enzymes from all the other enzyme families as well as nonenzyme proteins. The SVM is also evaluated for its capability in classifying distantly related enzymes as well as homologous enzymes of different functions.

Every enzyme sequence is represented by specific feature vectors, assembled from encoded representations of tabulated residue properties like amino acid composition, hydrophobicity, normalized Van der Waals' volume, polarity, polarizability, charge, surface tension, secondary structure, solvent accessibility, etc., for each residue in the sequence. The performance of the two-class SVM classification is measured in terms of the accuracies for positive $Q_{\rm p} = {\rm TP}/({\rm TP} + {\rm FN})$ and negative $Q_{\rm n} = {\rm TN}/({\rm TN} + {\rm FP})$ prediction, and MCC. The results, implemented on enzymes from 46 families (Swiss-Prot[14] database), suggest its potential for protein functional prediction.

Interaction between mutually binding protein pairs gives rise to specific biological functions. Using a diverse database of known protein interactions (DIP), an SVM was trained to recognize and predict possible interactions solely based on primary structure and associated physicochemical properties [73]. Feature vectors like sequential charge, hydrophobicity, and surface tension were selected as input corresponding to each residue in the amino acid sequences of a protein–protein complex. Binary decisions were generated regarding potential interactions.

B. Tertiary Structure and Folding

Protein structure comparison is often used to identify set of residue equivalencies between proteins based on their 3-D coordinates, and has a wide impact on the understanding of protein sequence, structure, function, and evolution. This is because it can identify more distantly related proteins, as compared to sequence comparison, since protein structures are more conserved than amino acid sequences over evolution.

The determination of an optimal 3-D conformation of a protein corresponds to folding, and has manifold implications to

[10] A nonconventional hydrogen bonding interaction involving side-chain aromatic ring and backbone NH group.

[11] The amino acid residue connected to an end of a polypeptide sequence by its CO group, leaving it with a free NH group.

[12] The amino acid residue connected to an end of a polypeptide sequence by its NH group, leaving it with a free CO group.

[13] http://www.cmbi.kun.nl/gv/dssp/

[14] http://www.expasy.ch/sprot/.

drug design. An active site structure determines the functionality of a protein. A ligand (enzyme or drug) docks into an active site of a protein. Many automated docking approaches have been developed, and can be categorized as: 1) rigid docking: both ligand and protein are rigid; 2) flexible-ligand docking: ligand flexible and protein rigid; and 3) flexible-protein docking: both ligand and protein are flexible (only a limited model of protein variation allowed, such as side-chain flexibility or small motions of loops in the binding site).

1) FSs: A *contact map* is a concise representation of a protein's native 3-D structure. It is expressed as a binary matrix, where each entry is a "1" if the corresponding protein residue pair are in "contact" (with Euclidean distance being within a threshold). When represented graphically, each contact between two residues corresponds to an edge. An alignment between two contact maps is an assignment of residues in one to those of the equivalent other. A pair of contacts is equivalent when the pairs of residues that define their endpoints are also equivalent. The number of such equivalent contacts determine the overlap of the contact maps for a pair of proteins, with a higher overlap indicating increased similarity between them. A generalization of the maximum contact map overlap has been developed [84] using one or more fuzzy thresholds and membership functions. This enables a more biological formulation of the optimization problem. Investigations are reported on three datasets from the PDB. Clustering of protein structures is done to validate the results.

2) ANNs: One of the earliest ANN-based protein tertiary structure prediction in the backbone [74] used MLP, with binary encoding for a 61-amino acid window at the input. There were 33 output nodes corresponding to the three secondary structures, along with distance constraints between the central amino acid and its 30 preceding residues. A large-scale ANN was employed to learn protein tertiary structures from the PDB [75]. The sequence-structure mapping encoded the entire protein sequence (66–129 residues) into 140 input units. The amino acid residue was represented by its hydrophobicity scale, normalized between -1 and $+1$. The network produced good prediction of distance matrices from homologous sequences, but suffered from a limited generalization capability due to the relatively small size of the training set.

Interatomic C^{α} distances between amino acid pairs, at a given sequence separation, were predicted [76] to be above (or below) a given threshold corresponding to contact (or noncontact). The input consisted of two sequence windows, each with 9 or 15 amino acids separated by different lengths of sequence, and a single output indicated the contact (or noncontact) between the central amino acids of the two sequence windows.

Instead of using protein sequence at input, a protein structure represented by a side-chain–side-chain contact map was employed at the input of an ANN to evaluate side-chain packing [77]. Contact maps of globular protein structures in the PDB were scanned using 7×7 windows, and converted to 49 binary numbers for the input. One output unit was used to determine whether the contact pattern is prevalent in the structure database.

Information obtained from secondary structure prediction is incorporated to improve structural class prediction using MLP [78]. The 26 input nodes include the 20-amino acid composition, sequence length, and five secondary structure characteristics of the protein. Four outputs correspond to four tertiary super classes. Prediction of 83 folding classes in proteins has been attempted [79] using multiple two-class MLPs. The input was represented in terms of major physicochemical amino acid attributes, like relative hydrophobicity (hydrophobic, neutral, or polar), predicted secondary structure, predicted solvent accessibility (buried or exposed), along with certain global descriptors like composition, transition, and distribution of different amino acid properties along the protein sequence.

A single-layer feedforward ANN, trained with scaled conjugate gradient algorithm, is used to identify catalytic residues found in enzymes [80] based on an analysis of the structure and sequence. Structural parameters like the solvent accessibility, type of secondary structure, depth, and cleft that the residue lies in, along with the conservation score and residue type are used as inputs for the ANN. Performance is measured in terms of the MCC. The network output is spatially clustered to determine the highly scoring residues, and thereby predict the location of most likely active sites.

Radial basis function (RBF) network, a supervised feedforward ANN, has been employed [81] to optimally predict the free energy contributions of proteins due to hydrogen bonds, hydrophobic interactions, and the unfolded state, with simple input measures.

3) GAs: GAs have been mainly applied to tertiary protein structure prediction, folding, docking, and side-chain packing problems.

a) Structure and folding: Structure alignment has been attempted in proteins using GAs [85], by first aligning equivalent secondary structure element (SSE) vectors while optimizing an elastic similarity score S. This is expressed as

$$S = \begin{cases} \sum_{i=1}^{L} \sum_{j=1}^{L} \left(\theta - \frac{d_{ij}^{A} - d_{ij}^{B}}{\bar{d}_{ij}}\right) e^{-(\bar{d}_{ij}/a)^2}, & i \neq j \\ \theta, & i = j \end{cases} \quad (5)$$

where d_{ij}^{A} and d_{ij}^{B} are the distances between equivalent positions i and j in proteins A and B, respectively, $\bar{d}_{ij}$ is the average of d_{ij}^{A} and d_{ij}^{B}, and θ and a are constant parameters, with the logic implying that equivalent positions in two proteins should have similar distances to other equivalent positions. Second, amino acid positions are optimally aligned within the SSEs. This is followed by superposition of protein backbones, based on the position equivalencies already determined. Finally, additional equivalent positions are searched in the non-SSE regions.

Tertiary protein structure prediction and folding, using GAs, has been reported in [21], [82], [86], and [87]. The objective is to generate a set of *native-like* conformations of a protein based on a force field, while minimizing a fitness function depending on its potential energy. Proteins can be represented in terms of: 1) 3-D Cartesian coordinates of its atoms; and 2) the torsional angle Rotamers, which are encoded as bit strings for the GA. The Cartesian coordinates representation has the advantage of being easily convertible to and from the 3-D conformation of a protein. Bond lengths b are specified in these terms. In the torsional angles representation, the protein is described by a set of angles under the assumption of constant standard binding

geometries. The different angles involved are the: 1) bond angle θ; 2) torsional angle ϕ, between N (amine group) and C_α; 3) angle ψ, between C_α and C' (carboxyl group); 4) peptide bond angle ω, between C' and N; and 5) side-chain dihedral angle χ.

The potential energy $U(r_1, \ldots, r_N)$ between N atoms is minimized, being expressed as

$$U(r_1,\ldots,r_N) = \sum_i K_b(b_i - b_0^i)^2 + \sum_i K_\theta(\theta_i - \theta_0^i)^2 + \sum_i K_\phi[1-\cos(n\phi_i - \delta)] + \sum_{i,j} \frac{q_i q_j}{4\pi\varepsilon_0\varepsilon_r r_{ij}} + \sum_{i,j} \varepsilon \left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - 2\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6} \right].$$

Here, the first three harmonic terms on the right-hand side involve the bond length, bond angle, and torsional angle of covalent connectivity, with b_0^i and θ_0^i indicating the down-state (low energy) bond length and bond angle, respectively, for the ith atom. The effects of hydrogen bonding and that of solvents (for nonbonded atom pairs i, j, separated by at least four atoms) is taken care of by the electrostatic Coulomb interaction and Van der Waals' interaction, modeled by the last two terms of the expression. Here, K_b, K_θ, K_ϕ, σ_{ij}, and δ are constants, q_i and q_j are the charges of atoms i and j, separated by distance r_{ij}, and ε indicates the dielectric constant. Two commercially available software packages, containing variations of the potential energy function, are Chemistry at HARvard Molecular Mechanics (CHARMm) and Assisted Model Building with Energy Refinement (AMBER).

Additionally, a protein acquires a folded conformation favorable to the solvent present. The calculation of the entropy difference between a folded and unfolded state is based on the interactions between a protein and solvent pair. Since it is not yet possible to routinely calculate an accurate model of these interactions, an *ad hoc* pseudo-entropic term $E_{\rm pe}$ is added to drive the protein to a globular state. $E_{\rm pe}$ is a function of its actual diameter, which is defined to be the largest distance between a pair of C_α carbon atoms in a conformation. We have

$$E_{\rm pe} = 4^{(\text{actual\_diameter} - \text{expected\_diameter})} \quad \text{[kcal/mol]} \quad (6)$$

where *expected_diameter*$/m = 8 * \sqrt[3]{len/m}$ is the diameter in its native conformation and len indicates the number of residues. This penalty term ensures that extended conformations have larger energy (or lower fitness) values than globular conformations. It constitutes the conformational entropy constituent of potential energy, in addition to the factors involved in the expression for U.

b) Docking: Genetic optimization for ligand docking (GOLD) [92] is an automated flexible-ligand docking program, employing steady-state GA involving the island model.[15] It evaluates nonmatching bonds while minimizing the potential energy (fitness function), defined in terms of Van der Waals' internal and external (or ligand-site) energy, torsional (or dihedral) energy, and hydrogen bonds. However, 1) an enforced requirement that the ligand must be hydrogen-bonded to the binding site; and 2) an underestimation of the hydrophobic contribution to binding, sometimes lead to failures in docking in certain cases over here.

Each chromosome in GOLD encodes the internal coordinates of both the ligand and active protein site, and a mapping between the hydrogen-bonding sites. Reproduction operators include crossover, mutation, and a migration operator to share genetic material between populations. The output is the ligand and protein conformations associated with the fittest chromosome in the population, when the GA terminates. The files handled are the Cambridge Crystallographic Database, Brookhaven PDB, and the Rotamer library.[16]

AutoDock [93] works on a genome composed of a string of real-valued genes encoding the 3-D coordinates and different angles. Mutation of the real-valued parameters is accomplished through the addition of a Cauchy-distributed random variable. Both conventional as well as Lamarckian[17] GAs are used, along with elitism.

A Generic Evolutionary Method for Molecular Docking (GEMDOCK) [94] has been developed for flexible-ligand docking. The potential energy function, involving numerous atomic interactions, is often computationally too expensive to implement using evolutionary strategies. Hence, rapid recognition of potential ligands is emphasized using a robust, simpler scoring function, encountering fewer local minima. Discrete and continuous search techniques are combined with local search to speed up convergence. The energy function encompasses electrostatic, steric, and hydrogen-bonding potentials of the molecules. A new rotamer-based mutation operator helps reduce the search space of ligand structure conformations. GEMDOCK is an automatic system that generates all related docking variables, like atom formal charge, atom type, and the ligand binding site of a protein. A major problem in GOLD, viz., its sensitiveness to docking hydrophobic ligands, is reduced in GEMDOCK [94]. However, its empirical scoring function is yet to incorporate important functional group interactions between ligands and proteins as in GOLD.

In a slightly different approach, the prediction of the conserved or displaced status of water molecules in the binding site, upon ligand binding, was made [95] by using a k-nearest-neighbors classifier. GAs determine the optimal feature-weight values for the classifier. Fitness is based on the percentage of correct predictions made.

c) Side-chain packing: The side-chain packing problem deals with the prediction of side-chain conformations. This is a crucial aspect of protein folding, since it determines feasible backbone conformations. GAs have been used in the prediction of side-chain packing [88] to search for low-energy hydrophobic core sequences and structures, using a custom rotamer library as input. Each core position is allocated a set of bits in the

[15] Evolves several small, distinct populations, instead of one large population.

[16] Provides the relationship between side-chain dihedral angles and backbone conformation

[17] Provides a local search, with replacement on a small fraction of the population within each generation. In Baldwinian approach, unlike in Lamarckian, the original population is not updated by the solution found in the local search.

chromosome, to encode a specific residue type and a set of torsional angles as specified in the library.

d) Use of evolutionary programming (EP): Evolutionary programming has been employed for *faster* finding of deep minima in the energy landscape of protein folding [89]. One folding step of the protein molecule involves: 1) calculation of molecular motion of the structure, i.e., rotation around one bond and 2) computation of free energy of the new conformation, which is discarded if it increases after the molecular motion. This process is simultaneously repeated to simulate a large set of folding operations, each using a different expanded starting structure for the protein. The program then determines those simulations yielding structures with the lowest free energies. It uses a lattice model of proteins to speed up the simulation, allowing only bond angle changes $(0^\circ, \pm 45^\circ, \pm 90^\circ)$ between adjacent amino acid residues along one or two of the three planes.

Mutations of the program were created using different types or magnitudes of molecular motions, or different positions of bonds around which rotations are performed, or different sequences of these motions. Positive mutants, i.e., those which performed better than the original program, were used for further mutations. Negative program mutants, i.e., those which did not find a deeper energy minimum within a certain period of time, were discarded. It is observed that only 20 evolution steps yielded a more than ten-fold increase in speed of finding deep minima in the energy landscape of two 64-residue proteins.

4) SVMs: Protein fold class prediction, from sequence, has been attempted using SVMs [90], [91], and the performance compared to that of ANNs and other standard statistical classification methods. SVMs were found to converge fast and result in high accuracy. In [90], scores of multiple parameter datasets are combined using majority voting. An l-class problem is modeled by l two-class classifiers, and a polynomial Gaussian kernel used. The 27 most populated folds, from the PDB, are used as output classes. Feature vectors extracted from the primary sequence are based on three descriptors, viz., 1) percent composition of the three constituents (polar, neutral, hydrophobic residues); 2) transition frequencies (polar-to-neutral, neutral-to-hydrophobic, etc.); and 3) distribution pattern of constituents (where the first residue of a given constituent is located, and where 25%, 50%, 75%, and 100% of that constituent are contained).

In [91], a protein is represented as a sequence $(s_1, \ldots, s_q)$, where each s_i stands for one of the 20 amino acids. This is embedded in terms of the relative frequencies of k-tuples of amino acids, resulting in a 20^k-dimensional feature space. The output consists of 42 categories of tertiary structures.

Detection of the active site of an enzyme as well as its microenvironment helps reveal its structural and functional mechanism, and enables conducting of structure-based drug design by regulating the enzyme function. Given the 3-D atomic coordinates of an enzyme, SVM has been employed to identify active sites based on distance [83]. Gaussian RBF has been used as the kernel, with a width selected to minimize an estimate of the VC-dimension.[18]

[18]Vapnik Chervonenkis dimension.

TABLE IV
APPLICATION OF SOFT COMPUTING TO MICROARRAY

Biological function modeled	Soft computing paradigm	Reference
Clustering	FS	[99]–[101]
	ANN	[102]–[105], SOTA [106]
	NF	[107]
	GA	[21]
	SA	[108]
	RS + GA	[109]
Classification	ANN	[110]–[112]
	NF	[113]
	GA	[114], [115]
	SVM	[116]–[119]
	RS	[120]
	RS + GA	[121]
Feature selection/ Rule generation	NF	[122]
	SVM	[123], RSA [124], RFE [125], [126]
	RS + GA	[127]
Biclustering	GA	[128]
Segmentation	GA	[129]
Image filtering	FS	[130]

V. MICROARRAY

Each DNA array contains the measures of the level of expression of many genes. Various distances and/or correlations can be computed from pairwise comparison of these patterns. Let $\text{gene}_j(e_{j1}, \ldots, e_{jn})$ denote the expression pattern for the jth gene for $i = 1, \ldots, n$ samples. The *Euclidean distance* between the jth and the kth genes, computed as

$$d_{j,k} = \sqrt{\sum_i (e_{ji} - e_{ki})^2} \tag{7}$$

is suitable when the objective is to cluster genes displaying similar levels of expression. Cluster validation can be done using either external and internal criterion analyses [97]. External criterion analysis validates a clustering outcome by comparing it to a given *gold standard*, which is another partition of the objects generated by an independent process based on information other than the given dataset. Internal criterion analysis, on the other hand, uses information from the given dataset, like (say) compactness and isolation, to determine the goodness of fit of the clustering. A quantitative data-driven framework has been developed [98] to evaluate different clustering algorithms, without using additional biological knowledge about the gene expression data. The *Pearson correlation coefficient* $-1 \leq r \leq 1$ measures the similarity in trend between two profiles (genes). The distance is given as

$$d_{j,k} = (1 - r) = 1 - \frac{\sum_i \{(e_{ji} - \hat{e}_j)(e_{ki} - \hat{e}_k)\}/n}{\sigma_{e_j} \times \sigma_{e_k}} \tag{8}$$

where $\hat{e}_j$ and σ_{e_j} indicate the mean and standard deviation, respectively, of all points of the jth profile.

There exists considerable literature on the applications of different soft computing paradigms in the area of gene expression data. An overview is provided in Tables IV and V.

TABLE V
USAGE DETAILS FOR MICROARRAY ANALYSIS

Data used	Function	Soft computing paradigm	Reference
Leukemia	Classification	ANN	[110], [111]
		GA	[114]
	Feature selection/ Rule generation	NF	[122]
		RS + GA	[127]
		SVM	[123]–[126]
Colon	Classification	ANN	[111]
		GA	[114]
	Clustering	RS + GA	[109]
	Feature selection/ Rule generation	NF	[122]
		RS + GA	[127]
		SVM	[124]
Lymphoma	Classification	NF	[113]
		GA	[114]
	Feature selection/ Rule generation	RS + GA	[127]
		SVM	[123]
Small round blue cell tumor	Classification	NF	[113]
		GA	[115]
		SVM	[119]
	Feature selection/ Rule generation	SVM	[123], [126]
Liver cancer	Classification	NF	[113]
Gastric tumor	Classification	RS + GA	[121]
Yeast	Classification	ANN	[112]
		SVM	[118]
	Clustering	FS	[100], [101]
		NF	[107]
	Biclustering	GA	[128]
	Segmentation	GA	[129]

A. FSs

Fuzzy c-means [131] is a well-known fuzzy partitive algorithm employed for clustering overlapping data. Use of fuzzy clustering enables genes to simultaneously belong to multiple groups, thereby revealing distinctive features of their function and regulation. Fuzzy c-means algorithm has been applied to cluster microarray data [99]. The value of the fuzzifier m is appropriately tuned for gene selection, based on resultant distribution of distances between genes. The selected genes exhibit tight association to the clusters.

Many proteins serve different functions depending on the demands of the organism, such that a corresponding set of genes is often coexpressed with multiple, distinct groups of genes under different conditions. This type of conditional coregulation of genes is modeled using a heuristically modified version of fuzzy c-means clustering [100], to identify overlapping partitions of genes based on the response of yeast cells to environmental changes.

The temporal order and varying length of sampling intervals are some of the important factors for clustering time-series microarray data into biologically meaningful partitions. However, the shortness and unequal sampling of gene expression time-series data limits the use of conventional modeling in these cases. The fuzzy short time-series algorithm [101] clusters profiles based on the similarity of their relative change in expression level and the corresponding temporal information. Here, the short time-series distance measure is incorporated in the fuzzy c-means framework. The performance, on the transcriptional data of sporulation in budding yeast, is evaluated in terms of Dunn's clustering validity index [97].

An interesting image-processing application to fuzzy filtering of cDNA microarray color images in the two-channel Red–Green space has been developed [130]. The two-component adaptive vector filter integrates concepts from FSs, nonlinear filtering, multidimensional scaling, and robust order statistics. Robust noise removal is achieved by tuning a membership function, which utilizes distance criteria based on a novel color-ratio model, on cDNA vectorial inputs at each image location. This sort of reduction in noise impairment facilitates subsequent analysis of the cDNA images.

B. ANNs

The two major mining tasks, modeled here, are clustering and classification. While unsupervised learning is self-organized, supervised learning helps incorporate known biological functions of genes into the knowledge discovery process of gene expression pattern analysis for gene discovery and prediction.

1) Clustering: Kohonen's SOM has been applied to the clustering of gene expression data [102]–[104]. It generates a robust and accurate clustering of large and noisy data, while providing effective visualization. SOMs require a selected node in the gene expression space (along with its neighbors) to be rotated in the direction of a selected gene expression profile (pattern). However, the predefinition of a 2-D topology of nodes can often be a problem considering its *biological relevance*.

SOTA has also been applied to gene expression clustering [106]. As in SOMs the gene expression profiles are sequentially and iteratively presented at the terminal nodes, and the mapping of the node that is closest (along with its neighboring nodes) is appropriately updated. Upon convergence, the node containing the most variable (measured in terms of distance) population of expression profiles is split into sister nodes, causing a growth of the binary tree. Unlike conventional hierarchical clustering, SOTA is linear in complexity to the number of profiles. The number of clusters need not be known in advance as in c-means clustering. The algorithm starts from the node having the most heterogeneous population of associated input gene profiles. A statistical procedure is followed for terminating the growing of the tree, thereby eliminating the need for an arbitrary choice of cutting level as in hierarchical models. However, no validation is provided to establish the biological relevance.

A binary tree-structured vector quantization [105] uses: 1) SOM for visualization and 2) partitive c-means clustering for grouping the similar component planes of SOMs and organizing them. Results are provided on cDNA microarray lung cancer data.

2) Classification: Classification of acute leukemia, having highly similar appearance in gene expression data, has been made by combining a pair of classifiers trained with mutually exclusive features [110]. Gene expression profiles were constructed from 72 patients having acute lymphoblastic leukemia (ALL) or acute myeloid leukemia (AML), each constituting one sample of the DNA microarray.[19] Each pattern consists of 7129 gene expressions. A neural network combines the outputs of the

[19]http://www.genome.wi.mit.edu/MPR

multiple classifiers. Feature selection with nonoverlapping correlation (such as Pearson and Spearman correlation coefficients) encourages the classifier ensemble to learn different aspects of the training data in a wide solution space. The recognition accuracy and generalization capacity are reported to be higher than those involving SVM [116], [117], SOM, decision tree, and k-nearest neighbors classifier.

An autoassociative neural network has been used for *simultaneous* pattern identification, feature extraction, and classification of gene expression data [111]. The network output approximates a reconstructed version of the input vector. Backpropagation is used to adjust the connection weights. The analysis of the network structure and strength of connections allows the: 1) identification of specific phenotype markers; 2) extraction of peculiar associations among genes and physiological states; and 3) assignment to multiple classes, like different pathological conditions or tissue samples. Results are demonstrated on leukemia and colon cancer[20] datasets.

Bayesian regularized neural network has been employed [112] to classify multiple gene expression temporal patterns, with sequential time points under different experimental conditions. The Bayesian setting, along with the regularization, help overcome experimental as well as biological noise or uncertainty. A feedforward architecture is used, with the input neurons corresponding to the number of time points or experimental conditions in the microarray experiment. Results are provided on the yeast data.

C. NF

Fuzzy ART network [132] has been employed for clustering the time-series expression data related to the sporulation of budding yeast [107].

An evolving modular fuzzy neural network, involving dynamic structure growing (and shrinking), adaptive online learning, and knowledge discovery in rule form, has been applied to the leukemia and colon cancer gene expression data [122]. Feature selection improves classification by reducing irrelevant attributes that do not change their expression between classes. The Pearson correlation coefficient is used to select genes that are highly correlated with the tissue classes. Rule generation provides physicians, on whom the final responsibility for any decision in the course of treatment rests, with a justification regarding how a classifier arrived at a judgement. Fuzzy logic rules, extracted from the trained network, handle the inherent noise in microarray data while offering the knowledge in a human-understandable linguistic form. These rules point to genes (or their combinations) that are strongly associated with specific types of cancer, and may be used for the development of new tests and treatment discoveries.

A dynamic fuzzy neural network, involving self-generation, parameter optimization, and rulebase simplification, is used [113] for the classification of cancer data such as lymphoma,[21] small round blue cell tumor (SRBCT),[22] and liver cancer.[23] Initial feature selection is done in terms of t-tests. It is observed that a small number of important genes (five out of 4026, eight out of 2308, 24 out of 1648 features, in the three datasets, respectively) succeed in attaining 100% classification.

D. GAs

The identification of gene subsets for classifying two-class disease samples has been modeled as a multiobjective evolutionary optimization problem [114], involving minimization of gene subset size to achieve reliable and accurate classification based on their expression levels. The nondominated sorting GA (NSGA-II) [133], a multiobjective GA, is used for the purpose. This employs elitist selection and an explicit diversity-preserving mechanism, and emphasizes the nondominated solutions. It has been shown that this algorithm can converge to the global Pareto front, while simultaneously maintaining the diversity of population.

Results are provided on three cancer samples, viz., leukemia, lymphoma, and colon. An l-bit binary string, where l is the number of selected (filtered) genes in the disease samples, represents a solution. The major difficulties faced in solving the optimization problem include the availability of only a few samples as compared to the number of genes in each sample, and the resultant huge search space of solutions. Moreover, many of the genes are redundant to the classification decision, and hence need to be eliminated. The three objectives simultaneously minimized are: 1) the gene subset size; 2) number of misclassifications in training; and 3) number of misclassifications in test samples.

The grouping GA (GGA) [134] is a modified GA, developed to suit the particular structure of grouping problems like clustering. GGA has also been applied to the clustering of microarray data [21]. The clusters of expression profiles are directly encoded in the chromosomes, based on their ordinal numbers, and the fitness function is defined on this set of groupings. The composition of the groups controls the value of the objective function.

GAs have also been used to correctly classify the SRBCT dataset with a selection of 12 genes [115]. There are four classes of tumors, from 88 samples described by 2308 genes. Simulated annealing (SA) [135] is employed to generate a robust clustering of temporal gene expression profiles [108]. An iterative scheme quantitatively evaluates the optimal number of clusters, while simultaneously optimizing the distribution of genes within them. The ith profile is represented by a vector $\{e_{i1}, \ldots, e_{in}\}$, with expression component e_{it} corresponding to the normalized expression level of gene i at time t in the range $[0, 1]$. The distribution of profiles is optimized for c clusters by minimizing the within-cluster distance between them, using

$$E(c) = (c) \sum_{k=1}^{c} \left[\sum_{i \in U_k} \sum_{j \in U_k} d_{i,j} \right] \tag{9}$$

[20] http://microarray.princeton.edu/oncology
[21] http://llmpp.nih.gov/lymphoma/data/figure1/figure1.cdt
[22] http://research.nhgri.nih.gov/microarray/Supplement/
[23] http://genome-www.stanford.edu/hcc/

where $d_{i,j}$ is the Euclidean distance (7) between profiles belonging to cluster U_k.

Some recent applications of GAs, in microarray, deal with biclustering and segmentation. Biclustering aims at determining subsets of genes that are similarly expressed over an optimal subset of conditions (or samples), thereby better reflecting the biological reality. Existing greedy algorithms for biclustering often yield suboptimal solutions. GAs are employed [128], by integrating a greedy algorithm as a local search in order to improve the quality of biclustering. Optimization is done with respect to the conflicting goals of homogeneity and size. Results are provided on 2884 genes of yeast data, involving 17 conditions. Evolutionary segmentation of the yeast genome has also been attempted in literature [129].

E. RSes

A basic issue related to many practical applications of knowledge databases is whether the whole set of attributes in a given information system is always necessary to define a given partition of the universe. Many of the attributes are superfluous, i.e., we can have *optimal* subsets of attributes that define the same partition as the whole set of attributes. These subsets are called the *reducts* in rough set theory [14], and correspond to the minimal feature set that are sufficient to represent a decision. These have considerable impact on subsequent decision-making.

RSes have been applied mainly to microarray gene expression data, in mining tasks like classification [120], [121], clustering [109], and feature selection [127].

1) Classification: Classification rules (in *if–then* form) have been extracted from microarray data [120], using RSes with supervised learning. The underlying assumption is that the associated genes are organized in an ontology, involving super- and subclasses. This biological knowledge is utilized while generating rules in terms of the minimal characteristic features (reducts) of temporal gene expression profiles. A rule is said to *cover* a gene if the gene satisfies the conditional part, expressed as a conjunction of attribute–value pairs. The rules do not discriminate between the super- and subclasses of the ontology, while retaining as much detail about the predictions without losing precision.

Gastric tumor classification in microarray data is made using rough set-based learning [121], implemented with ROSETTA involving GAs and dynamic reducts [136]. The fitness function incorporates measures involving the classification performance (discernibility) along with the size of the reduct. Thereby precedence is provided to solutions having less number of attributes. A major problem with microarray data being the smaller number of objects with a comparatively larger number of attributes, a preprocessing stage of feature selection based on bootstrapping is made. The dataset consists of 2504 human genes corresponding to the conditional attributes, while the 17 tumor types are clubbed as six different clinical parameters or the decision attributes.

2) Clustering: In the rough c-means clustering algorithm, the concept of c-means is extended by viewing each cluster as an interval or rough set [137]. A rough set Y is characterized by its lower and upper approximations $\underline{B}Y$ and $\bar{B}Y$, respectively. This permits overlaps between clusters. Here, an object $\mathbf{X}_k$ can be part of at most *one* lower approximation. If $\mathbf{X}_k \in \underline{B}Y$ of cluster Y, then simultaneously $\mathbf{X}_k \in \bar{B}Y$. If $\mathbf{X}_k$ is not a part of any lower approximation, then it belongs to two or more upper approximations.

An evolutionary rough c-means clustering algorithm has been applied to microarray gene expression data [109]. RSes are used to model the clusters in terms of upper and lower approximations. GAs are used to tune the threshold, and relative importance of upper and lower approximation parameters of the sets. The Davies–Bouldin clustering validity index [97] is used as the fitness function of the GA, which is minimized while arriving at an optimal partitioning. It was found that the algorithm performed particularly well over the colon cancer gene expression data, involving a collection of 62 measurements from colon biopsy samples with 2000 genes (features).

3) Feature Selection: An evolutionary rough feature selection algorithm [127] has been used for classifying microarray gene expression patterns. Since the data typically consist of a large number of redundant features, an initial redundancy reduction of the attributes is done to enable faster convergence. Thereafter 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, viz., colon, lymphoma, and leukemia.

While Chu *et al*. [113] generated a five-genes set for 100% correct classification on the lymphoma data in the NF framework, Banerjee *et al*. [127] obtained a misclassification for just two samples from the test data using a two-genes set. In case of the leukemia data, a two-genes set is selected, whereas the colon data results in an eight-genes reduct size.

F. SVMs

SVMs are particularly suited to handling large feature spaces and identifying outliers. This characteristic makes them capable of efficiently modeling high-dimensional microarray data. Use of SVMs has been reported [118] for functionally classifying gene expression data from budding yeast.

Classification of the SRBCT dataset was performed with SVM [119], providing 100% accuracy for a selection of 20 important genes. Extraction of three principal components also resulted in similar classification performance. Use of RBF kernels in SVM [123] resulted in 100% training and testing accuracy with a reduced set of important genes (7 for SRBCT, 5 for Lymphoma, 20 for Leukemia). Some other applications of SVMs include [116], and [117].

Cao *et al*. [124] apply saliency analysis to SVMs for gene selection in tissue classification. The importance of genes is ranked by evaluating the sensitivity of the output to the inputs, in terms of the partial derivative. The recursive saliency analysis (RSA) algorithm is developed to remove irrelevant genes in case of the leukemia and colon data.

Recursive feature elimination (RFE) [125] selects a set of genes by continuously eliminating those that make a relatively

small contribution to classification, as measured by the accuracy of the SVM on the whole gene set. This is a *greedy, wrapper* feature selection approach that iteratively trains new SVMs while eliminating genes with the smallest weights from the set. A four-genes set, with perfect accuracy, is selected over the leukemia data. This multivariate approach is, however, sensitive to the presence of irrelevant genes as well as outliers.

In order to overcome this problem, a hybrid of univariate (maximum likelihood) and multivariate feature selection (RFE) has been designed [126] to generate a good selection of fewer genes providing a high prediction accuracy. While a univariate approach considers the contribution of each gene (or feature) in isolation from the others, a multivariate approach focuses on the selection of a fewer genes on the whole. At first the maximum likelihood method identifies and removes genes that are expected to have low discrimination ability. This is followed by the application of RFE to further reduce the size of the feature set. Authors claim a resultant significant gain in run time. Applications of this integrated approach are reported on the leukemia and SRBCT datasets. Perfect accuracy was obtained with a three-genes set for leukemia, while a set of 15 genes were found to be sufficient to differentiate the samples of SRBCT.

VI. Gene Regulatory Network

Understanding of regulatory networks is crucial to the understanding of fundamental cellular processes involving growth, development, hormone secretion, and cellular communication. Determination of transcriptional factors that control gene expression can offer further insight into the misregulated expressions common in many human diseases.

In this section, we outline some of the recent literature on the use of ANNs and SVMs in the area of gene regulatory networks.

A. ANNs and Hybridizations

Recurrent neural network has been used to model the dynamics of gene expression [138]. The significance of the regulatory effect of one gene product on the expression of other genes of the system is defined by a weight matrix. Multigenic regulation, involving positive and/or negative feedback, is considered. The process of gene expression is described by a single network, along with a pair of linked networks independently modeling the transcription and translation schemes.

Adaptive double self-organizing map (ADSOM) [139] provides a clustering strategy for identifying gene regulatory networks. It has a flexible topology and allows simultaneous visualization of clusters. DSOM combines features of SOM with 2-D position vectors, to provide a visualization tool for deciding on the required number of clusters. However, its free parameters are difficult to control to guarantee proper convergence. ADSOM updates these free parameters during training, and allows convergence of its position vectors to a fairly consistent number of clusters (provided its initial number of nodes is greater than the expected number of clusters). The effectiveness of ADSOM in identifying the number of clusters is proven by applying it to publicly available gene expression data from multiple biological systems such as yeast, human, and mouse.

Ritchie *et al.* [140] optimized the backpropagation neural network architecture, using GP, in order to improve upon the ability of ANNs to model, identify, characterize, and detect nonlinear gene–gene interactions in studies of common human diseases. The performance is reported to be superior in terms of predictive ability and power to detect gene–gene interactions when nonfunctional polymorphisms are present.

Bayesian networks with Bayesian learning were employed [141], in a reverse engineering approach, to infer gene regulatory interactions from simulated gene expression data. Use of GAs for reconstructing genetic networks has been reported in literature [142], [143]. Typically the GA searches for the most likely genetic networks that best fit the data, considering the set of genes to be included in the network along with the strength of their interactions.

Identification of protein–DNA interactions in the promoter region, in terms of DNA motifs that characterize the regulatory factors operating in the transcription of a gene, is important for recognizing genes that participate in a regulation process. This enables determination of their interconnection in a gene regulatory network. A hybrid methodology for this purpose has been developed [144] by combining ANN, fuzzy sets, and multiobjective GAs. A time-delayed neural network (TDNN) learns compound binding site motifs from nonspecific DNA sequences by decomposing it into modules corresponding to submotifs. The MCC of (4) is used to discriminate between promoters and nonpromoters. The system can handle multiplicity of RNA polymerase targets and multiple functional binding sites in closely located regulatory regions, along with the associated uncertainty of the motifs.

B. SVMs

Regulatory network is predicted by SVMs [145] for the budding yeast genome by mining the gene expression data from different physiological conditions. The relationship between the expression time-course of a transcription factor (TF) and its target factor not being a simple correlation, SVMs are found to fare better than conventional hierarchical clustering. SVMs are trained using both positive and negative examples from the dataset. A negative example is a gene pair that definitely has no regulatory relationship. The training set consists of a pair of genes, with the first being the known TF (R) and the second being the target gene (T) that is potentially regulated by R. After training, the system determines the probabilities of each $R \Rightarrow T$ pairing in order to construct parts of a regulatory network. The dataset consists of 209 TFs $\times$ 6128 genes, resulting in mining among 1 280 752 combinations to determine which of these pairs represent a true regulatory relationship. The accuracy of the prediction is reported to be 93%, involving both positive and negative examples.

VII. Conclusion and Discussion

Bioinformatics is a new area of science where a combination of statistics, molecular biology, and computational methods is used for analyzing and processing biological information like gene, DNA, RNA, and proteins. Improper folding of protein

structure is responsible for causing many diseases. Therefore, accurate structure prediction of proteins is a major goal of study. With the availability of huge volume of high-dimensional data, there exists a lot of possibilities for the emergent field of biological data mining. Hybrid approaches, combining powerful algorithms and interactive visualization tools with the strengths of fast processors, hold promise for enhanced performance in the near future.

Soft computing paradigms, like fuzzy sets, ANNs, GAs, RSes, and SVMs, have been used for analyzing the different protein sequences, structures and folds, microarrays, as well as regulatory networks. Since soft computing permits approximate, good solutions, instead of the high-precision, globally optimum solution, it allows one to arrive at a low-cost goal faster.

We have provided, in this paper, a detailed review on the role of soft computing techniques in different aspects of bioinformatics, mainly involving data-mining tasks. It is categorized based on the domain of operation, the function modeled, and the tool used. The major tasks covered include classification, clustering, feature selection, and rule mining. Gene regulatory networks, a relatively new area of study, have also been surveyed.

The characteristics of adaptivity and learning help ANNs to minimize error and self-organize in data-rich environments. The low-precision, approximate reasoning of fuzzy sets allows faster convergence. Different types of hybridizations incorporate the generic and application-specific merits of the constituent paradigms. Exhaustive enumeration and evaluation of all gene combinations being NP-hard, the GAs use intelligent, goal-directed search while optimizing a fitness function determined by the knowledge about the environment. RSes allow dimensionality reduction for high-dimensional data, and are found suitable in mining microarray gene expressions.

Knowledge about the domain is often found to be useful in improving performance of a system. For example, the incorporation of the alignment profile generated by Psi-BLAST is found to be advantageous in protein secondary structure determination by both ANNs and SVMs. This is evident from the comparative study projected in Table III. Similarly, the use of prior knowledge about the secondary structure at the input enhances the performance for tertiary structure determination. The role of soft computing in exploring protein sequence and structure data has been summarized in Tables I and II. An overview of soft computing applications to microarray analysis has been provided in Tables IV and V.

Metabolism is the chemical engine that drives a living process. By means of utilization of a vast repertoire of enzymatic reactions and transport processes, organisms process and convert thousands of organic compounds into various biomolecules necessary to support their existence. The cells as well as the organisms direct the distribution and processing of metabolites throughout an extensive map of pathways. While we seek to develop strategies to effectively eliminate metabolic pathways due to microorganisms through antibiotics in order to curb bacterial infection, we also strive to enhance the performance of certain other pathways or introduce novel routes for the production of biochemicals of commercial interest. The domain of metabolic pathways and gene regulatory networks open up significant challenges for research involving application of soft computing techniques.

REFERENCES

[1] P. Baldi and S. Brunak, *Bioinformatics: The Machine Learning Approach. Adaptive Computation and Machine Learning*. Cambridge, MA: MIT, 2001.

[2] Special Issue on Bioinformatics, *IEEE Comput.*, vol. 35, no. 7, Jul. 2002.

[3] Special Issue on Bioinformatics, Part I: Advances and Challenges, *Proc. IEEE*, vol. 90, no. 11, Nov. 2002.

[4] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *J. Mol. Biol.*, vol. 215, pp. 403–410, 1990.

[5] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped BLAST and PSI-BLAST: A new generation of protein database search programs," *Nucleic Acids Res.*, vol. 25, pp. 3389–3402, 1997.

[6] S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church, "Systematic determination of genetic network architecture," *Nat. Genet.*, vol. 22, pp. 281–285, 1999.

[7] L. A. Zadeh, "Fuzzy logic, neural networks, and soft computing," *Commun. ACM*, vol. 37, pp. 77–84, 1994.

[8] S. Mitra and T. Acharya, *Data Mining: Multimedia, Soft Computing, and Bioinformatics*. New York: Wiley, 2003.

[9] V. Kecman, *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models. Complex Adaptive Systems*. Cambridge, MA: MIT Press, 2001.

[10] H. Jeong, B. Tombor, R. Albert, Z. N. Oltval, and A. L. Barabasl, "The large-scale organization of metabolic networks," *Nature*, vol. 407, pp. 651–654, 2000.

[11] H. Ogata, S. Goto, K. Sato, W. Fujibuchi, H. Bono, and M. Kanehisa, "KEGG: Kyoto Encyclopedia of Genes and Genomes," *Nucleic Acids Res.*, vol. 27, pp. 29–34, 1999.

[12] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: Macmillan, 1994.

[13] S. K. Pal and S. Mitra, *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing*. New York: Wiley, 1999.

[14] Z. Pawlak, *Rough Sets, Theoretical Aspects of Reasoning about Data*. Dordrecht, The Netherlands: Kluwer, 1991.

[15] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[16] D. B. Fogel, L. J. Fogel, and V. W. Porto, "Evolving neural networks," *Biol. Cybern.*, vol. 63, pp. 487–493, 1990.

[17] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press, 1992.

[18] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[19] C. H. Wu and J. W. McLarty, Neural Networks and Genome Informatics, in *Methods in Computational Biology and Biochemistry*. Amsterdam: Elsevier, 2000, vol. 1.

[20] L. P. Wang and X. Fu, *Data Mining with Computational Intelligence*. Berlin, Germany: Springer, 2005.

[21] G. Fogel and D. Corne, Eds., *Evolutionary Computation in Bioinformatics*. San Francisco, CA: Morgan Kaufmann, 2002.

[22] L. Wang, Ed., *Support Vector Machines: Theory and Applications*. Berlin, Germany: Springer, 2005.

[23] R. Farber, A. Lapedes, and K. Sirotkin, "Determination of eukaryotic protein coding regions using neural networks and information theory," *J. Mol. Biol.*, vol. 226, pp. 471–479, 1992.

[24] E. C. Uberbacher, Y. Xu, and R. J. Mural, "Discovering and understanding genes in human DNA sequence using GRAIL," *Methods Enzymol.*, vol. 266, pp. 259–281, 1996.

[25] E. E. Snyder and G. D. Stormo, "Identification of protein coding regions in genomic DNA," *J. Mol. Biol.*, vol. 248, pp. 1–18, 1995.

[26] G. Schneider, S. Rohlk, and P. Wrede, "Analysis of cleavage-site patterns in protein precursor sequences with a perceptron-type neural network," *Biochem. Biophys. Res. Commun.*, vol. 194, pp. 951–959, 1993.

[27] Y. D. Cai, H. Yu, and K. C. Chou, "Artificial neural network method for predicting HIV protease cleavage sites in protein," *J. Protein Chem.*, vol. 17, pp. 607–615, 1998.

[28] R. Thomson, T. C. Hodgman, Z. R. Yang, and A. K. Doyle, "Characterizing proteolytic cleavage site activity using bio-basis function neural networks," *Bioinformatics*, vol. 19, pp. 1741–1747, 2003.

[29] S. Brunak, J. Engelbrecht, and S. Knudsen, "Prediction of human mRNA donor and acceptor sites from the DNA sequence," *J. Mol. Biol.*, vol. 220, pp. 49–65, 1991.
[30] N. I. Larsen, J. Engelbrecht, and S. Brunak, "Analysis of eukaryotic promoter sequences reveals a systematically occurring CT-signal," *Nucleic Acids Res.*, vol. 23, pp. 1223–1230, 1995.
[31] A. G. Pedersen and H. Nielsen, "Neural network prediction of translation initiation sites in eukaryotes: Perspectives for EST and genome analysis," in *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 1997, vol. 5, pp. 226–233.
[32] A. Browne, B. D. Hudson, D. C. Whitley, M. G. Ford, and P. Picton, "Biological data mining with neural networks: Implementation and application of a flexible decision tree extraction algorithm to genomic problem domains," *Neurocomputing*, vol. 57, pp. 275–293, 2004.
[33] R. Setiono, "Extracting rules from neural networks by pruning and hidden-unit splitting," *Neural Comput.*, vol. 9, pp. 205–225, 1997.
[34] C. H. Wu, M. Berry, S. Shivakumar, and J. McLarty, "Neural networks for full-scale protein sequence classification: Sequence encoding with singular value decomposition," *Machine Learn.*, vol. 21, pp. 177–193, 1995.
[35] J. Hanke and J. G. Reich, "Kohonen map as a visualization tool for the analysis of protein sequences: Multiple alignments, domains and segments of secondary structures," *Comput. Appl. Biosci.*, vol. 6, pp. 447–454, 1996.
[36] Y. D. Cai, H. Yu, and K. C. Chou, "Prediction of beta-turns," *J. Protein Chem.*, vol. 17, pp. 363–376, 1998.
[37] C. H. Wu, H. L. Chen, and S. C. Chen, "Counter-propagation neural networks for molecular sequence classification: Supervised LVQ and dynamic node allocation," *Appl. Intell.*, vol. 7, pp. 27–38, 1997.
[38] D. Wang and G. B. Huang, "Protein sequence classification using extreme learning machine," in *Proc. Int. Joint Conf. Neural Networks (IJCNN'05)*, Montreal, QC, Canada, Aug. 2005, pp. 1406–1411.
[39] H. Saigo, J. P. Vert, N. Ueda, and T. Akutsu, "Protein homology detection using string alignment kernels," *Bioinformatics*, vol. 20, pp. 1682–1689, 2004.
[40] J. Schuchhardt, G. Schneider, J. Reichelt, D. Schomberg, and P. Wrede, "Local structural motifs of protein backbones are classified by self-organizing neural networks," *Protein Eng.*, vol. 9, pp. 833–842, 1996.
[41] P. Arrigo, F. Giuliano, F. Scalia, A. Rapallo, and G. Damiani, "Identification of a new motif on nucleic acid sequence data using Kohonen's self organizing map," *Comput. Appl. Biosci.*, vol. 7, pp. 353–357, 1991.
[42] B. C. H. Chang and S. K. Halgamuge, "Protein motif extraction with neuro-fuzzy optimization," *Bioinformatics*, vol. 18, pp. 1084–1090, 2002.
[43] D. Howard and K. Benson, "Evolutionary computation method for pattern recognition of cis-acting sites," *BioSystems*, vol. 72, pp. 19–27, 2003.
[44] J. Dopazo and J. M. Carazo, "Phylogenetic reconstruction using an unsupervised growing neural network that adopts the topology of a phylogenetic tree," *J. Mol. Evol.*, vol. 44, pp. 226–233, 1997.
[45] C. LeBlanc, C. R. Katholi, T. R. Unnasch, and S. I. Hruska, "DNA sequence analysis using hierarchical ART-based classification networks," in *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 1994, vol. 2, pp. 253–260.
[46] P. O. Lewis, "A genetic algorithm for maximum-likelihood phylogeny inference using nucleotide sequence data," *Mol. Biol. Evol.*, vol. 15, pp. 277–283, 1998.
[47] M. J. Brauer, M. T. Holder, L. A. Dries, D. J. Zwickl, P. O. Lewis, and D. M. Hillis, "Genetic algorithms and parallel processing in maximum-likelihood phylogeny inference," *Mol. Biol. Evol.*, vol. 19, pp. 1717–1726, 2002.
[48] L. Pickert, I. Reuter, F. Klawonn, and E. Wingender, "Transcription regulatory region analysis using signal detection and fuzzy clustering," *Bioinformatics*, vol. 14, pp. 244–251, 1998.
[49] E. A. Ferran, B. Pflugfelder, and P. Ferrara, "Self-organized neural maps of human protein sequences," *Protein Sci.*, vol. 3, pp. 507–521, 1994.
[50] H. C. Wang, J. Dopazo, and J. M. Carazo, "Self-organizing tree-growing network for classifying amino acids," *Bioinformatics*, vol. 14, pp. 376–377, 1998.
[51] H. C. Wang, J. Dopazo, L. G. de la Fraga, Y. P. Zhu, and J. M. Carazo, "Self-organizing tree-growing network for the classification of protein sequences," *Protein Sci.*, vol. 7, pp. 2613–2622, 1998.
[52] C. Notredame and D. G. Higgins, "SAGA: Sequence alignment by genetic algorithm," *Nucleic Acids Res.*, vol. 24, pp. 1515–1524, 1996.
[53] C. Notredame, L. Holm, and D. G. Higgins, "COFFEE: An objective function for multiple sequence alignments," *Bioinformatics*, vol. 14, pp. 407–422, 1998.
[54] J. Casasnovas and F. Rosselló, "Averaging fuzzy biopolymers," *Fuzzy Sets Syst.*, vol. 152, pp. 139–158, 2005.
[55] K. C. Chou and Y. D. Cai, "Using functional-domain composition and support vector machines for prediction of protein subcellular location," *J. Biol. Chem.*, vol. 29, pp. 45765–45769, 2002.
[56] A. Garg, M. Bhasin, and G. P. S. Raghava, "Support vector machine-based method for subcellular localization of human proteins using amino acid compositions, their order, and similarity search," *J. Biol. Chem.*, vol. 280, pp. 14427–14432, 2005.
[57] R. Thomson and Z. R. Yang, "A novel basis function neural network," in *Proc. 9th Int. Conf. Neural Information Processing (ICONIP'02)*, Nov. 2002, pp. 1:441–1:446.
[58] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissing, I. Shindyalov, and P. Bourne, "The protein data bank," *Nucleic Acids Res.*, vol. 28, pp. 235–242, 2000.
[59] N. Qian and T. Sejnowski, "Predicting the secondary structure of globular proteins using neural network models," *J. Mol. Biol.*, vol. 202, pp. 865–884, 1988.
[60] B. Rost and C. Sander, "Prediction of protein secondary structure at better than 70% accuracy," *J. Mol. Biol.*, vol. 232, pp. 584–599, 1993.
[61] B. Rost, "PHD: Predicting one-dimensional protein structure by profile-based neural networks," *Methods Enzymol.*, vol. 266, pp. 525–539, 1996.
[62] Z. Sun, X. Rao, L. Peng, and D. Xu, "Prediction of protein supersecondary structures based on the artificial neural network method," *Protein Eng.*, vol. 10, pp. 763–769, 1997.
[63] R. Maclin and J. W. Shavlik, "Using knowledge-based neural network to improve algorithms: Refining Chou–Fasman algorithm for protein folding," *Machine Learn.*, vol. 11, pp. 195–215, 1993.
[64] S. K. Riis and A. Krogh, "Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignments," *J. Comput. Biol.*, vol. 3, pp. 163–183, 1996.
[65] D. T. Jones, "Protein secondary structure prediction based on position-specific scoring matrices," *J. Mol. Biol.*, vol. 292, pp. 195–202, 1999.
[66] H. Kaur and G. P. S. Raghava, "Role of evolutionary information in prediction of aromatic-backbone NH interactions in proteins," *FEBS Lett.*, vol. 564, pp. 47–57, 2004.
[67] G. Pollastri, D. Przybylski, B. Rost, and P. Baldi, "Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles," *Proteins Struct. Funct. Genet.*, vol. 47, pp. 228–235, 2002.
[68] P. Baldi, S. Brunak, P. Frasconi, G. Pollastri, and G. Soda, "Exploiting the past and the future in protein secondary structure prediction," *Bioinformatics*, vol. 15, pp. 937–946, 1999.
[69] F. Vivarelli, G. Giusti, M. Villani, R. Campanini, P. Fariselli, M. Compiani, and R. Casadio, "LGANN: A parallel system combining a local genetic algorithm and neural networks for the prediction of secondary structure of proteins," *Comput. Appl. Biosci.*, vol. 11, pp. 253–260, 1995.
[70] S. Hua and Z. Sun, "A novel method of protein secondary structure prediction with high segment overlap measure: Support vector machine approach," *J. Mol. Biol.*, vol. 308, pp. 397–407, 2001.
[71] J. Guo, H. Chen, Z. Sun, and Y. Lin, "A novel method for protein secondary structure prediction using dual-layer SVM and profiles," *Proteins Struct. Funct. Bioinformatics*, vol. 54, pp. 738–743, 2004.
[72] C. Z. Cai, L. Y. Han, Z. L. Ji, and Y. Z. Chen, "Enzyme family classification by support vector machines," *Proteins Struct. Funct. Bioinformatics*, vol. 55, pp. 66–76, 2004.
[73] J. R. Bock and D. A. Gough, "Predicting protein–protein interactions from primary structure," *Bioinformatics*, vol. 17, pp. 455–460, 2001.
[74] H. Bohr, J. Bohr, S. Brunak, R. M. J. Cotterill, and H. Fredholm, "A novel approach to prediction of the 3-dimensional structures of protein backbones by neural networks," *FEBS Lett.*, vol. 261, pp. 43–46, 1990.
[75] G. L. Wilcox, M. O. Poliac, and M. N. Liebman, "Neural network analysis of protein tertiary structure," *Tetrahedron Comput. Methodol.*, vol. 3, pp. 191–211, 1991.
[76] O. Lund, K. Frimand, J. Gorodkin, H. Bohr, J. Bohr, J. Hansen, and S. Brunak, "Protein distance constraints predicted by neural networks and probability distance functions," *Protein Eng.*, vol. 10, pp. 1241–1248, 1997.
[77] M. Milik, A. Kolinski, and J. Skolnick, "Neural network system for the evaluation of side-chain packing in protein structures," *Protein Eng.*, vol. 8, pp. 225–236, 1995.
[78] J. M. Chandonia and M. Karplus, "Neural networks for secondary structure and structural class predictions," *Protein Sci.*, vol. 4, pp. 275–285, 1995.

[79] I. Dubchak, I. Muchnik, S. R. Holbrook, and S. H. Kim, "Prediction of protein folding class using global description of amino acid sequence," *Proc. Nat. Acad. Sci. USA*, vol. 92, pp. 8700–8704, 1995.

[80] A. Gutteridge, G. J. Bartlett, and J. M. Thornton, "Using a neural network and spatial clustering to predict the location of active sites in enzymes," *J. Mol. Biol.*, vol. 330, pp. 719–734, 2003.

[81] R. Casadio, M. Compiani, P. Fariselli, and F. Vivarelli, "Predicting free energy contributions to the conformational stability of folded proteins," *Intell. Syst. Mol. Biol.*, vol. 3, pp. 81–88, 1995.

[82] S. Schulze-Kremer, "Genetic algorithms for protein tertiary structure prediction," in *Parallel Problem Solving from Nature II*, R. Männer and B. Manderick, Eds. Amsterdam, The Netherlands: North Holland, 1992, pp. 391–400.

[83] Y. D. Cai, G. P. Zhou, C. H. Jen, S. L. Lin, and K. C. Chou, "Identify catalytic triads of serine hydrolases by support vector machines," *J. Theor. Biol.*, vol. 228, pp. 551–557, 2004.

[84] D. Pelta, N. Krasnogor, C. Bousono-Calzon, J. L. Verdegay, J. Hirst, and E. Burke, "A fuzzy sets based generalization of contact maps for the overlap of protein structures," *Fuzzy Sets Syst.*, vol. 152, pp. 103–123, 2005.

[85] J. D. Szustakowski and Z. Weng, "Protein structure alignment using a genetic algorithm," *Proteins*, vol. 38, pp. 428–440, 2000.

[86] R. König and T. Dandekar, "Improving genetic algorithms for protein folding simulations by systematic crossover," *BioSystems*, vol. 50, pp. 17–25, 1999.

[87] J. Pedersen and J. Moult, "Protein folding simulations with genetic algorithms and a detailed molecular description," *J. Mol. Biol.*, vol. 269, pp. 240–259, 1997.

[88] J. Desjarlais and T. Handel, "De novo design of the hydrophobic cores of proteins," *Protein Sci.*, vol. 4, pp. 2006–2018, 1995.

[89] B. Noelting, D. Juelich, W. Vonau, and K. Andert, "Evolutionary computer programming of protein folding and structure predictions," *J. Theor. Biol.*, vol. 229, pp. 13–18, 2004.

[90] C. H. Q. Ding and I. Dubchak, "Multi-class protein fold recognition using support vector machines and neural networks," *Bioinformatics*, vol. 17, pp. 349–358, 2001.

[91] F. Markowetz, L. Edler, and M. Vingron, "Support vector machines for protein fold class prediction," *Biometrical J.*, vol. 45, pp. 377–389, 2003.

[92] G. Jones, P. Willett, R. C. Glen, A. R. Leach, and R. Taylor, "Development and validation of a genetic algorithm for flexible docking," *J. Mol. Biol.*, vol. 267, pp. 727–748, 1997.

[93] G. Morris, D. Goodsell, R. Halliday, R. Huey, W. Hart, R. Belew, and A. Olson, "Automated docking using Lamarckian genetic algorithm and an empirical binding free energy function," *J. Comput. Chem.*, vol. 19, pp. 1639–1662, 1998.

[94] J. M. Yang and C. C. Chen, "GEMDOCK: A generic evolutionary method for molecular docking," *Proteins Struct. Funct. Bioinf.*, vol. 55, pp. 288–304, 2004.

[95] M. Raymer, P. Sanschagrin, W. Punch, S. Venkataraman, E. Goodman, and L. Kuhn, "Predicting conserved water-mediated and polar ligand interactions in proteins using a k-nearest neighbors genetic algorithm," *J. Mol. Biol.*, vol. 265, pp. 445–464, 1997.

[96] P. Chou and G. Fasmann, "Prediction of the secondary structure of proteins from their amino acid sequence," *Adv. Enzymol.*, vol. 47, pp. 45–148, 1978.

[97] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice Hall, 1988.

[98] K. Y. Yeung, C. Fraley, A. Murua, A. E. Raftery, and W. L. Ruzzo, "Model-based clustering and data transformations for gene expression data," *Bioinformatics*, vol. 17, pp. 977–987, 2001.

[99] D. Dembele and P. Kastner, "Fuzzy c-means method for clustering microarray data," *Bioinformatics*, vol. 19, pp. 973–980, 2003.

[100] A. P. Gasch and M. B. Eisen, "Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering," *Genome Biol.*, vol. 3, pp. 0059.1–0059.22, 2002.

[101] C. S. Möller-Levet, F. Klawonn, K. H. Cho, H. Yin, and O. Wolkenhauer, "Clustering of unevenly sampled gene expression time-series data," *Fuzzy Sets Syst.*, vol. 152, pp. 49–66, 2005.

[102] K. Torkkola, R. M. Gardner, T. Kaysser-Kranich, and C. Ma, "Self-organizing maps in mining gene expression data," *Inf. Sci.*, vol. 139, pp. 79–96, 2001.

[103] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Smitrovsky, E. S. Lander, and T. R. Golub, "Interpreting patterns of gene expression with self-organizing maps: Methods and applications to hematopoietic differentiation," *Proc. Nat. Acad. Sci. USA*, vol. 96, pp. 2907–2912, 1999.

[104] P. Törönen, M. Kolehmainen, G. Wong, and E. Castrén, "Analysis of gene expression data using self-organizing maps," *FEBS Lett.*, vol. 451, pp. 142–146, 1999.

[105] M. Sultan, D. A. Wigle, C. A. Cumbaa, M. Maziarz, J. Glasgow, M. S. Tsao, and I. Jurisica, "Binary tree-structured vector quantization approach to clustering and visualizing microarray data," *Bioinformatics*, vol. 18, Suppl. 1, pp. S111–S119, 2002.

[106] J. Herrero, A. Valencia, and J. Dopazo, "A hierarchical unsupervised growing neural network for clustering gene expression patterns," *Bioinformatics*, vol. 17, pp. 126–136, 2001.

[107] S. Tomida, T. Hanai, H. Honda, and T. Kobayashi, "Analysis of expression profile using fuzzy adaptive resonance theory," *Bioinformatics*, vol. 18, pp. 1073–1083, 2002.

[108] A. V. Lukashin and R. Fuchs, "Analysis of temporal gene expression profiles: Clustering by simulated annealing and determining the optimal number of clusters," *Bioinformatics*, vol. 17, pp. 405–414, 2001.

[109] S. Mitra, "An evolutionary rough partitive clustering," *Pattern Recognit. Lett.*, vol. 25, pp. 1439–1449, 2004.

[110] S. B. Cho and J. Ryu, "Classifying gene expression data of cancer using classifier ensemble with mutually exclusive features," *Proc. IEEE*, vol. 90, no. 11, pp. 1744–1753, Nov. 2002.

[111] S. Bicciato, M. Pandin, G. Didonè, and C. Di Bello, "Pattern identification and classification in gene expression data using an autoassociative neural network model," *Biotechnol. Bioeng.*, vol. 81, pp. 594–606, 2003.

[112] A. Kelemen and Y. Liang, "Bayesian regularized neural network for multiple gene expression pattern classification," in *Proc. Int. Joint Conf. Neural Networks*, Jul. 2003, pp. 1:654–1:659.

[113] F. Chu, W. Xie, and L. Wang, "Gene selection and cancer classification using a fuzzy neural network," in *Proc. 2004 Annu. Meet. North Amer. Fuzzy Information Processing Soc. (NAFIPS)*, vol. 2, pp. 555–559.

[114] K. Deb and A. Raji Reddy, "Reliable classification of two-class cancer data using evolutionary algorithms," *BioSystems*, vol. 72, pp. 111–129, 2003.

[115] J. M. Deutsch, "Evolutionary algorithms for finding optimal gene sets in microarray prediction," *Bioinformatics*, vol. 19, pp. 45–52, 2003.

[116] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, pp. 906–914, 2000.

[117] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and N. Yakhini, "Tissue classification with gene expression profiles," *J. Comput. Biol.*, vol. 7, pp. 559–584, 2000.

[118] M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianni, C. Sugnet, T. S. Furey, M. Ares, and D. Haussler, "Knowledge-based analysis of microarray gene expression data using support vector machines," *Proc. Nat. Acad. Sci. USA*, vol. 97, pp. 262–267, 2000.

[119] Y. Lee and C. K. Lee, "Classification of multiple cancer types by multicategory support vector machines using gene expression data," *Bioinformatics*, vol. 19, pp. 1132–1139, 2003.

[120] H. Midelfart, A. Lægreid, and J. Komorowski, *Classification of Gene Expression Data in an Ontology,* vol. 2199. Lecture Notes in Computer Science, Berlin, Germany: Springer-Verlag, 2001, pp. 186–194.

[121] H. Midelfart, J. Komorowski, K. Nørsett, F. Yadetie, A. K. Sandvik, and A. Lægreid, "Learning rough set classifiers from gene expressions and clinical data," *Fundamenta Inf.*, vol. 53, pp. 155–183, 2002.

[122] M. E. Futschik, A. Reeve, and N. Kasabov, "Evolving connectionist systems for knowledge discovery from gene expression data of cancer tissue," *Artif. Intell. Med.*, vol. 28, pp. 165–189, 2003.

[123] F. Chu and L. P. Wang, "Cancer classification with microarray data using support vector machines," in *Bioinformatics Using Computational Intelligence Paradigms*, U. Seiffert, L. C. Jain, and P. Schweizer, Eds. Berlin, Germany: Springer, 2005, pp. 167–189.

[124] L. Cao, H. P. Lee, C. K. Seng, and Q. Gu, "Saliency analysis of support vector machines for gene selection in tissue classification," *Neural Comput. Appl.*, vol. 11, pp. 244–249, 2003.

[125] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learn.*, vol. 46, pp. 389–422, 2002.

[126] M. Xu and R. Setiono, "Gene selection for cancer classification using a hybrid of univariate and multivariate feature selection methods," *Appl. Genomics Proteomics*, vol. 2, pp. 79–91, 2003.

[127] M. Banerjee, S. Mitra, and H. Banka, "Evolutionary-rough feature selection in gene expression data," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, to be published.

[128] S. Bleuler, A. Prelić, and E. Zitzler, "An EA framework for biclustering of gene expression data," in *Proc. Congr. Evolutionary Computation*, 2004, pp. 166–173.
[129] D. Mateos, J. C. Riquelme, and J. S. Aguilar-Ruiz, "Evolutionary segmentation of yeast genome," in *Proc. ACM Symp. Appl. Comput.*, Mar. 2004, pp. 1026–1027.
[130] R. Lukac, K. N. Plataniotis, B. Smolka, and A. N. Venetsanopoulos, "cDNA microarray image processing fuzzy vector filtering framework," *Fuzzy Sets Syst.*, vol. 152, pp. 17–35, 2005.
[131] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.
[132] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Netw.*, vol. 4, pp. 759–771, 1991.
[133] K. Deb, S. Agarwal, A. Pratap, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[134] E. Falkenauer, *Genetic Algorithms and Grouping Problems*. Chichester, U.K.: Wiley, 1998.
[135] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.
[136] J. Wroblewski, "Finding minimal reducts using genetic algorithms," Warsaw Inst. Technol.-Inst. Comput. Sci., Warsaw, Poland, Tech. Rep. 16/95, 1995.
[137] P. Lingras and C. West, "Interval set clustering of web users with rough k-means," Dept. Math. Comput. Sci., St. Mary's Univ., Halifax, Canada, Tech. Rep. 2002-002, 2002.
[138] J. Vohradsky, "Neural network model of gene expression," *FASEB J.*, vol. 15, pp. 846–854, 2001.
[139] H. Ressom, D. Wang, and P. Natarajan, "Clustering gene expression data using adaptive double self-organizing map," *Physiol. Genomics*, vol. 14, pp. 35–46, 2003.
[140] M. D. Ritchie, B. C. White, J. S. Parker, L. Hahn, and J. H. Moore, "Optimization of neural network architecture using genetic programming improves detection and modeling of gene–gene interactions in studies of human diseases," *BMC Bioinformatics*, vol. 4, pp. 28–36, 2003.
[141] D. Husmeier, "Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic bayesian networks," *Bioinformatics*, vol. 19, pp. 2271–2282, 2003.
[142] S. Kikuchi, D. Tominaga, M. Arita, K. Takahashi, and M. Tomita, "Dynamic modeling of genetic networks using genetic algorithm and S-system," *Bioinformatics*, vol. 19, pp. 643–650, 2003.
[143] M. Xiong, J. Li, and X. Fang, "Identification of genetic networks," *Genetics*, vol. 166, pp. 1037–1052, 2004.
[144] V. Cotik, R. Romero Zaliz, and I. Zwir, "A hybrid promoter analysis methodology for prokaryotic genomes," *Fuzzy Sets Syst.*, vol. 152, pp. 83–102, 2005.
[145] J. Qian, J. Lin, N. M. Luscombe, H. Yu, and M. Gerstein, "Prediction of regulatory networks: Genome-wide identification of transcription factor targets from gene expression data," *Bioinformatics*, vol. 19, pp. 1917–1926, 2003.

Sushmita Mitra (S'91–M'92–SM'00) is currently a Professor at the Machine Intelligence Unit, Indian Statistical Institute, Kolkata. From 1992 to 1994, she was with RWTH, Aachen, Germany as a DAAD Fellow. She was a Visiting Professor in the Computer Science Departments of the University of Alberta, Edmonton, AB, Canada in 2004, Meiji University, Japan in 1999, 2004, 2005, and Aalborg University, Esbjerg, Denmark in 2002 and 2003. She is the author of more than 100 research publications in referred international journals and is the author of *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing* (Wiley, 1999) and *Data Mining: Multimedia, Soft Computing, and Bioinformatics* (Wiley, 2003). According to the Science Citation Index, two of her papers have been ranked 3rd and 15th in the list of top-cited papers in engineering science from India during 1992–2001. Her current research interests include data mining, pattern recognition, soft computing, image processing, and bioinformatics. She has served as a Guest Editor for special issues of journals, and is an Associate Editor of *Neurocomputing*.

Dr. Mitra has been the Program Chair, Tutorial Chair, and a Member of the program committees of many international conferences. She was the recipient of the National Talent Search Scholarship (1978–1983) from the National Council of Educational Research and Training (NCERT), India, the IEEE TNN Outstanding Paper Award in 1994 for her pioneering work in neuro-fuzzy computing, and the CIMPA-INRIA-UNESCO Fellowship in 1996.

Yoichi Hayashi (M'86–SM'00) received the B.E. degree in management science, and the M.E. and Dr.Eng. degrees in systems engineering, all from the Tokyo University of Science, Tokyo, Japan, in 1979, 1981, and 1984, respectively.

In 1986, he joined Ibaraki University, Japan, as an Assistant Professor and was a Visiting Professor at the University of Alabama at Birmingham and the University of Canterbury, respectively for ten months. Currently, he is a Professor and Chairman of Computer Science at Meiji University, Kawasaki, Japan. He is the author of 147 papers published in academic journals and international conference proceedings in the fields of computer and information sciences. His current research interests include bioinformatics, artificial neural networks, fuzzy logic, soft computing, expert systems, computational intelligence, data mining, database management, and medical informatics. He is the Action Editor of *Neural Networks*.

Dr. Hayashi is a member of the ACM, AAAI, IFSA, INNS, NAFIPS, IPSJ, and IEICE. He has been an Associate Editor of IEEE TRANSACTIONS ON FUZZY SYSTEMS.