

Soft Computing Methodologies in Bioinformatics

Rabindra Ku. Jena

Faculty of IT, Institute of Management Technology, Nagpur, India
E-mail: rk_jena2@rediffmail.com

Musbah M. Aqel

Faculty of IT, Middle East University for Graduate Studies, Ammam, Jordan
E-mail: Aqelm06@yahoo.com

Pankaj Srivastava

*Department of IT, ABV-Indian Institute of Information
Technology & Management, Gwalior, India*
E-mail: pksrivastava@iiitm.ac.in

Prabhat K. Mahanti

Department of CSAS, University of New Brunswick, Saint John, Canada
E-mail: pkmahanti@yahoo.co.in

Abstract

Bioinformatics is a promising and innovative research field in 21st century. Despite of a high number of techniques specifically dedicated to bioinformatics problems as well as many successful applications, we are in the beginning of a process to massively integrate the aspects and experiences in the different core subjects such as biology, medicine, computer science, engineering, chemistry, physics, and mathematics. Recently the use of soft computing tools for solving bioinformatics problems have been gaining the attention of researchers because of their ability to handle imprecision, uncertainty in large and complex search spaces. The paper will focus on soft computing paradigm in bioinformatics with particular emphasis on integrative research.

Keywords: Bioinformatics, Soft computing paradigm, Artificial neural network, Fuzzy logic, Genetic algorithms, Bioinformatics tools

1. Introduction

Advancement in soft computing techniques demonstrates the high standards of technology, algorithms, and tools in bioinformatics for dedicated purposes such as reliable and parallel genome sequencing, fast sequence comparison, search in databases, automated gene identification, efficient modeling and storage of heterogeneous data, etc. The basic problems in bioinformatics like protein structure prediction, multiple alignment, phylogenetic inference etc. are mostly NP-hard in nature. For all these problems, soft computing offers on promising approach to achieve efficient and reliable heuristic solution. On the other side the continuous development of high quality biotechnology, e.g. micro-array techniques and mass spectrometry, which provide complex patterns for the direct characterization of cell processes, offers further promising opportunities for advanced research in bioinformatics. So

bioinformatics must cross the border towards a massive integration of the aspects and experience in the different core subjects like computer science and statistics etc. for an integrated understanding of relevant processes in systems biology. This puts new challenges not only on appropriate data storage, visualization, and retrieval of heterogeneous information, but also on soft computing methods and tools used in this context, which must adequately process and integrate heterogeneous information into a global picture.

The rest of the paper is organized as follows: In section 2 we give a brief introduction to bioinformatics followed by an introduction to soft computing in section 3. In Sub-section 3.2, 3.3 and 3.4 we give some important applications of artificial neural network, fuzzy logic and genetic algorithm in bioinformatics. In section 4 we list some prominent bioinformatics tools and the paper is concluded with conclusions in section 5.

2. Bioinformatics: A Brief Introduction

Bioinformatics is the application of computer technology to the management of biological information. Computers are used to gather, store, analyze and integrate biological and genetic information which can then be applied to gene-based drug discovery and development. It is arguable that the origin of bioinformatics history can be traced back to Mendel's discovery of genetic inheritance in 1865. However, bioinformatics research in a real sense started in late 1960s, symbolized by Dayhoff's atlas of protein sequences and the early modeling analysis of protein and RNA structures. In fact, these early works represented two distinct provenances of bioinformatics: evolution and biochemistry, which still largely define the current bioinformatics research topics. The need for Bioinformatics capabilities has been precipitated by the explosion of publicly available genomic information resulting from the Human Genome Project. According to analysts, "the bioinformatics "industry", though in a fledgling condition at present, could in the next 20 - 30 years actually rival the drug industry in size.

2.1. Tasks of Bioinformatics

Different biological problems considered within the scope of bioinformatics involve the study of genes, proteins, nucleic acid structure prediction, and molecular design with docking. A broad classification of the various bioinformatics tasks is given as follows.

1. Alignment and comparison of DNA, RNA, and protein sequences.
2. Gene mapping on chromosomes.
3. Gene finding and promoter identification from DNA sequences.
4. Interpretation of gene expression and micro-array data.
5. Gene regulatory network identification.
6. Construction of phylogenetic trees for studying evolutionary relationship.
7. DNA structure prediction.
8. RNA structure prediction.
9. Protein structure prediction and classification.
10. Molecular design and molecular docking.

2.2. Applications of Bioinformatics

Bioinformatics has found its applications in many areas. It helps in providing practical tools to explore proteins and DNA in number of other ways. Bio-computing is useful in recognition techniques to detect similarity between sequences and hence to interrelate structures and functions. Another important application of bioinformatics is the direct prediction of protein 3-Dimensional structure from the linear amino acid sequence. It also simplifies the problem of understanding complex genomes by analyzing simple organisms and then applying the same principles to more complicated ones. This would result in identifying potential drug targets by checking homologies of essential microbial proteins. Bioinformatics is useful in designing drugs.

2.3. Aims of Bioinformatics

The aims of Bioinformatics are:

1. To organize data in a way that allows researchers to access existing information and to submit new entries as they are produced
2. To develop tools and resources that aid in the analysis and management of data.
3. To use this data to analyze and interpret the results in a biologically meaningful manner.
4. To help researchers in the pharmaceutical industry in understanding the protein structures to make the drug design easy.

2.4. Algorithms in Bioinformatics

This discussion sheds light on algorithms that are of interest to biologists. The following are some of the most important algorithmic trends in bioinformatics:

1. Finding similarities among strings (such as proteins of different organisms).
2. Detecting certain patterns within strings (such as genes, introns, and α -helices).
3. Finding similarities among parts of spatial structures (such as motifs).
4. Constructing trees (called phylogenetic trees expressing the evolution of organisms whose DNA or proteins are currently known).
5. Classifying new data according to previously clustered sets of annotated data.
6. Reasoning about microarray data and the corresponding behavior of pathways.

The first three trends can be viewed as instances of pattern matching. However, pattern matching in biology differs from its counterpart in computer science. DNA strings contain millions of symbols, and small local differences may be tolerated. The pattern itself may not be exactly known, because it may involve inserted, deleted, or replacement symbols. Regular expressions are useful for specifying a multitude of patterns and are ubiquitous in bioinformatics. However, what biologists really need is to be able to infer these regular expressions from typical sequences and establish the likelihood of the patterns being detected in new sequences.

This discussion suggests that both optimization and probabilistic approaches are necessary for developing biology-oriented pattern-matching algorithms. In the 1970s, a dynamic programming technique was devised to match two strings, taking into account the costs of insertions, deletions, and substitutions called global pair-wise alignment. This technique was subsequently extended to consider local alignments and today, both methods are often used in bioinformatics. However, dynamic programming is time consuming (it involves quadratic complexity) and therefore cannot be applied in a practical way to strings with hundreds of thousands of symbols. A remarkable bioinformatics development from the 1990s is a pattern-matching approach called BLAST, or the Basic Local Alignment Search Tool, that mimics the behavior of the dynamic programming approach and efficiently yields good results using heuristic based approach. It is fair to say that BLAST is the most frequently used tool for searching sequences in genomic databases.

Another widely used and effective technique is multiple alignments, which helps align several sequences of symbols, so identical symbols are properly lined up vertically, with gaps allowed within symbols. The sequences may represent variants of the same proteins in various species. But the goal is to find conserved parts of the proteins that are unchanged during evolution. Finding conserved parts of proteins also provides hints about a protein's possible function. Methods for multiple alignments are based on dynamic programming techniques developed for pair wise alignment. After aligning multiple genomic or protein sequences, biologists usually depict trees representing the degree of similarity among the sequences being studied. Depicting evolutionary trees is in itself a domain within bioinformatics called phylogenetic trees. The problem of matching spatial structures can be viewed as a combination of computational geometry and computer graphics. Approximate methods are often required to find the longest linkage that is common in two 3D structures.

Bioinformatics involves the pervasive use of searches in genomic databases that often yield very large sets of long sequences. Such searches are often performed automatically by scripting to

download massive amounts of genomic data from a number of Web sites. Script languages (such as Perl and Python) are often used for programming automatic searches in Web databases. An approach commonly used in bioinformatics is: Given a human-annotated list of strings with boundaries specifying meaningful substrings—the learning set— now establish the corresponding likely boundaries for a new string (examples in bioinformatics involve finding genes and identifying the components of proteins). Solutions to these problems are being explored through approaches from machine learning, neural networks, genetic algorithms, and clustering.

Since the early 1990s a clustering technique called support vector machines (SVM) has had considerable success in biology. Classification and machine learning have been studied extensively in artificial intelligence to sort out new data based on a human-annotated set of examples. Perhaps foremost among the machine learning (soft computing) techniques used in biology are the ubiquitous Hidden Markov Models, which are essentially probabilistic finite-state machines that use computed branching probabilities from a learning set and that establish the likelihood that a new string is processed through certain states with pre-established properties.

Two other algorithmic trends relevant to this discussion are related to micro-arrays and biologists' interest in computational linguistics. Recall that the main goal of analyzing micro-array data is to establish relationships among gene behavior, possible protein interactions, and the effects of a cell's environment. From a computer science perspective, that goal is amounted to the generation of parts of a program (flowchart) from data. This was also an early goal of program synthesis. However, it should be stressed that biological data is vast and noisy, spurring development of new heuristic based techniques (such as Bayesian nets, SVM, Fuzzy logic and evolutionary algorithms) is required. Information about the relationships among genes is often buried in countless articles describing the results of biological experiments. In the case of protein interaction, pharmaceutical companies have teams whose task is to search the available literature and “manually” detect phrases of interest. Efforts have been made to computerize these searches. Their implementation requires expertise in biology, computational linguistics and heuristic based methodologies.

Based on the above discussion, it is mandatory to have a machine learning/soft computing based approach for various tasks in bioinformatics. This paper will focus on, how the soft computing techniques suits in bioinformatics. The next section will discuss the concept of soft computing, their constituents and the prominent application to bioinformatics.

3. Soft Computing Paradigm

Soft computing is a consortium of methodologies that work synergistically and provides, in one form or another, flexible information processing capabilities for handling real life ambiguous situations. Its aim, unlike conventional (hard) computing, is to exploit the tolerance for imprecision, uncertainty, approximate reasoning and partial truth in order to achieve tractability, robustness, low solution cost, and close resemblance with human like decision-making. The constituents of soft computing are: Fuzzy Logic (FZ), Artificial Neural Networks (ANN), Evolutionary Algorithms (EAs) (including genetic algorithms (GAs), genetic programming (GP), evolutionary strategies (ES)), Support Vector Machines (SVM), Wavelets, Rough Sets (RS), Simulated Annealing (SA), Swarm Optimization (SO), Memetic Algorithms (MA), Ant Colony Optimization (ACO) and Tabu Search (TS).

In this paper, the application of the main constituent of the soft computing methods like fuzzy set, artificial neural network and genetic algorithm in bioinformatics have been briefly discussed.

3.1. Why Soft Computing Techniques in Bioinformatics

There are a number of reasons why soft computing approaches are widely used in practice, especially in bioinformatics (Narayanan et al., 2003; Baldi and Brunak, 1998; Clark and Westhead, 1996)

1. Traditionally, a human being builds such an expert system by collecting knowledge from specific experts. The experts can always explain what factors they use to assess a situation,

however, it is often difficult for the experts to say what rules they use (for example, for disease analysis and control). This problem can be resolved by soft computing mechanisms. Soft computing mechanism can extract the description of the hidden situation in terms of those factors and then fire rules that match the expert's behavior.

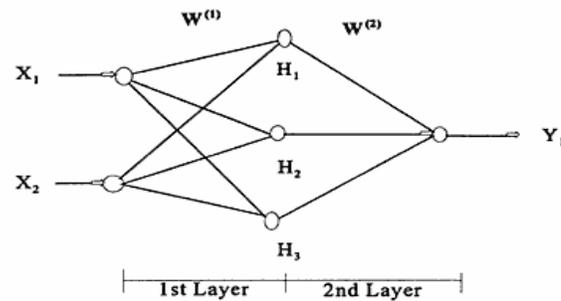
2. Systems often produce results different from the desired ones. This may be caused by unknown properties or functions of inputs during the design of systems. This situation always occurs in the biological world because of the complexities and mysteries of life sciences. However, with its capability of dynamic improvement, soft computing can cope with this problem.
3. In molecular biology research, new data and concepts are generated every day, and those new data and concepts update or replace the old ones. Soft computing can be easily adapted to a changing environment. This benefits system designers, as they do not need to redesign systems whenever the environment changes.
4. Missing and noisy data is one characteristic of biological data. The conventional computer techniques fail to handle this. Soft computing based techniques are able to deal with missing and noisy data.
5. With advances in biotechnology, huge volumes of biological data are generated. In addition, it is possible that important hidden relationships and correlations exist in the data. Soft computing methods are designed to handle very large data sets, and can be used to extract such relationships.

3.2. Relevance of Artificial Neural Network in Bioinformatics

An Artificial Neural Network (ANN) is an information processing model that is able to capture and represent complex input-output relationships. The motivation the development of the ANN technique came from a desire for an intelligent artificial system that could process information in the same way the human brain. Its novel structure is represented as multiple layers of simple processing elements, operating in parallel to solve specific problems. ANNs resemble human brain in two respects: learning process and storing experiential knowledge. An artificial neural network learns and classifies a problem through repeated adjustments of the connecting weights between the elements. In other words, an ANN learns from examples and generalizes the learning beyond the examples supplied.

Artificial neural network applications have recently received considerable attention. The methodology of modeling, or estimation, is somewhat comparable to statistical modeling. Neural networks should not, however, be heralded as a substitute for statistical modeling, but rather as a complementary effort (without the restrictive assumption of a particular statistical model) or an alternative approach to fitting non-linear data.

A typical neural network (shown in Figure 1) is composed of input units X_1, X_2, \dots corresponding to independent variables, a hidden layer known as the first layer, and an output layer (second layer) whose output units Y_1, \dots correspond to dependent variables (expected number of accidents per time period).

Figure 1: A simplified Artificial Neural Network

In between are hidden units H_1, H_2, \dots corresponding to intermediate variables. These interact by means of weight matrices $W(1)$ and $W(2)$ with adjustable weights. The values of the hidden units are obtained from the formulas:

$$H_j = f\left(\sum_k W_{jk}^{(1)} X_k\right)$$

$$Y_i = f\left(\sum_j W_{ij}^{(2)} H_j\right).$$

In One multiplies the first weight matrix by the input vector $X = (X_1, X_2, \dots)$ and then applies an activation function f to each component of the result. Likewise the values of the output units are obtained by applying the second weight matrix to the vector $H = (H_1, H_2, \dots)$ of hidden unit values, and then applying the activation function f to each component of the result. In this way one obtains an output vector $Y = (Y_1, Y_2, \dots)$. The activation function f is typically of sigmoid form and may be a logistic function, hyperbolic tangent, etc.:

$$f(u) = \frac{1}{1 + e^{-u}}, \quad f(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}.$$

Usually the activation function is taken to be the same for all components but it need not be. Values of $W(1)$ and $W(2)$ are assumed at the initial iteration. The accuracy of the estimated output is improved by an iterative learning process in which the outputs for various input vectors are compared with targets (observed frequency of accidents) and an average error term E is computed:

$$E = \frac{\sum_{n=1}^N (Y^{(n)} - T^{(n)})^2}{N}.$$

Here

N = Number of highway sites or observations

$Y(n)$ = Estimated number of accidents at site n for $n = 1, 2, \dots, N$

$T(n)$ = Observed number of accidents at site n for $n = 1, 2, \dots, N$.

After one pass through all observations (the training set), a gradient descent method may be used to calculate improved values of the weights $W(1)$ and $W(2)$, values that make E smaller. After reevaluation of the weights with the gradient descent method, successive passes can be made and the weights further adjusted until the error is reduced to a satisfactory level. The computation thus has two modes, the mapping mode, in which outputs are computed, and the learning mode, in which weights are adjusted to minimize E . Although the method may not necessarily converge to a global minimum, it generally gets quite close to one if an adequate number of hidden units are employed.

The most delicate part of neural network modeling is generalization, the development of a model that is reliable in predicting future accidents. Overfitting (i.e., getting weights for which E is so small on the training set that even random variation is accounted for) can be minimized by having two

validation samples in addition to the training sample. According to Smith and Thakar (Smith and Thakar, 1993), the data set should be divided into three subsets: 40% for training, 30% to prevent overfitting, and 30% for testing. Training on the training set should stop at the epoch when the error E computed on the second set begins to rise (the second set is not used for training but merely to decide when to stop training). Then the third set is used to see how well the model performs. The cross-validation helps to optimize the fit in three ways: by limiting/optimizing the number of hidden units, by limiting/optimizing the number of iterations, and by inhibiting network use of large weights.

The major advantages and disadvantages of neural networks in modeling applications are as follows:

Advantages

1. **Adaptive learning:** An ability to learn how to do tasks based on the data given for training or initial experience.
2. **Self-Organisation:** An ANN can create its own organisation or representation of the information it receives during learning time.
3. **Real Time Operation:** ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
4. **Fault Tolerance via Redundant Information Coding:** Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

Applications

Neural networks have been widely used in biology since the early 1980s. They can be used to:

1. Predict the translation initiation sites in DNA sequences (Hatzigeorgiou and Reckzo, 2004).
2. Explain the theory of neural networks using applications in biology (Baldi and Brunak, 1998).
3. Predict immunologically interesting peptides by combining an evolutionary algorithm (Brusic et al., 1998).
4. Study human TAP transporter (Brusic et al., 1999).
5. Carry out pattern classification and signal processing successfully in bioinformatics; in fact, a large number of applications of neural network can be found in this area.
6. Perform protein sequence classification; neural networks are applied to protein sequence classification by extracting features from protein data and using them in combination with the Bayesian neural network (BNN) (Wu and Mclarty, 2000).
7. Predict protein secondary structure prediction (Chenand and Kurgan, 2007); Zhong et al., 2007).
8. Analyze the gene expression patterns as an alternative to hierarchical clusters (Toronen et al., 1999; Ma et al., 2000; Bicciato et al., 2001; Torkkola et al., 2001). Gene expression can even be analyzed using a single layer neural network (Narayanan et al., 2003). Protein fold recognition using ANN and SVM (Ding and Dubchak, 2001).

In summary, a neural network is presented with a pattern on its input nodes, and the network produces an output pattern based on its learning algorithm during the training phase. Once trained, the neural network can be applied to classify new input patterns. This makes neural networks suitable for the analysis of gene expression patterns, prediction of protein structure, and other related processes in bioinformatics.

3.3. Relevance of Fuzzy Logic in Bioinformatics

Fuzzy logic is a relatively new technique (first appeared in 1970s) for solving engineering control problems. This technique can be easily used to implement systems ranging from simple, small or even embedded up to large networked ones. It can be used to be implemented in either software or hardware

The key idea of fuzzy logic is that it uses a simple and easy way in order to get the output(s) from the input(s), actually the outputs are related to the inputs using if-statements and this is the secret behind the easiness of this technique. The most fascinating thing about Fuzzy logic is that it accepts the uncertainties that are inherited in the realistic inputs and it deals with these uncertainties in such a way that their affect is negligible and thus resulting in a precise outputs.

Fuzzy logic is said to be the control methodology that mimics how a person decides but only much faster. One of the many advantages of fuzzy logic is that it really simplifies complex systems. One may be surprised if told that he is using fuzzy logic statements (descriptions) almost every day. For example when you say: "John is fat" and "Tom is tall", you are giving non-accurate descriptions about those people which turn to be exactly fuzzy logic descriptions, but if you say for example: "John is 80 kg" and "Tom is 190 cm", you are giving a certain and exact numbers which are NOT fuzzy descriptions. The difference between fuzzy logic and other type of logics is in terms of precision and significance. FL is a technique in which the significance is the most important while in other logics the precision is the important aspect.

There are several reasons behind the increasing use of this type of methodology in the world. First of all, Fuzzy Logic reduces the design steps and simplifies complexity that might arise since the first step is to understand and characterize the system behavior by using knowledge and experience.

The concept of Fuzzy Logic (FL) was conceived by Lotfi Zadeh, a professor at the University of California at Berkley, and presented not as a control methodology, but as a way of processing data by allowing partial set membership rather than crisp set membership or non-membership. FL provides a simple way to arrive at a definite conclusion based upon vague, ambiguous, imprecise, noisy, or missing input information. It mimics human control logic.

Applications

Fuzzy systems have been successfully applied to several areas in practice. In bioinformatics, fuzzy systems play an important role for building knowledge-based systems. Most systems involve fuzzy logic-based and fuzzy rule-based models. They can control and analyze processes and diagnose and make decisions in biomedical sciences (Adriaenssens et al., 2004; Lughofer and Guardiola, 2008). There are many application areas in biomedical science and bioinformatics, where fuzzy logic techniques can be applied successfully. Some of the important uses of fuzzy logic are listed below:

1. To increase the flexibility of protein motifs (Anbarasu et al., 1998; Taria et al., 2008).
2. To study differences between polynucleotides (Torres and Nieto, 2003).
3. To analyze experimental expression data (Tomida et al., 2002) using fuzzy adaptive resonance theory.
4. To align sequences based on a fuzzy recast of a dynamic programming algorithm (Schlosshauer and Ohlsson, 2002).
5. DNA sequencing using genetic fuzzy systems (Cordon et al., 2004).
6. To cluster genes from micro-array data (Fickett, 1996; Belacel et al., 2004).
7. To predict proteins sub-cellular locations from their dipeptide composition (Huang and Li, 2004) using fuzzy k- nearest neighbors algorithm.
8. To simulate complex traits influenced by genes with fuzzy-valued effects in pedigreed populations (Carleos et al., 2003).
9. To attribute cluster membership values to genes (Demb'el'e and Kastner, 2003) applying a fuzzy partitioning method, fuzzy C-means.
10. To map specific sequence patterns to putative functional classes since evolutionary comparison leads to efficient functional characterization of hypothetical proteins (Heger and Holm, 2003). The authors used a fuzzy alignment model.
11. To analyze gene expression data (Wolf and Wang, 2000).
12. To unravel functional and ancestral relationships between proteins via fuzzy alignment methods (Blankenbecler et al., 2003), or using a generalized radial basis function neural network architecture that generates fuzzy classification rules (Wang et al., 2003).

13. To analyze the relationships between genes and decipher a genetic network (Ressom et al., 2003).
14. To process complementary deoxyribonucleic acid (cDNA) micro-array images (Lukac et al., 2005). The procedure should be automated due to the large number of spots and it is achieved using a fuzzy vector filtering framework.
15. To classify amino acid sequences into different super families (Bandyopadhyay, 2005).

3.4. Relevance of Genetic Algorithms in Bioinformatics

Genetic algorithms (Goldberg, 1989; Bhandari et al., 1996; Booker et al., 1989; Mitchell et al., 1992), a biologically inspired technology, are randomized search and optimization techniques guided by the principles of evolution and natural genetics. They are efficient, adaptive, and robust search processes, producing near optimal solutions, and have a large degree of implicit parallelism. Therefore, the application of GAs for solving certain problems of bioinformatics, which need optimization of computation requirements, and robust, fast and close approximate solutions, appears to be appropriate and natural (Setubal and Meidanis, 1999). Moreover, the errors generated in experiments with bioinformatics data can be handled with the robust characteristics of GAs. To some extent, such errors may be regarded as contributing to genetic diversity, a desirable property. The problem of integrating GAs and bioinformatics constitutes a new research area.

GAs are executed iteratively on a set of coded solutions, called population, with three basic operators: selection/reproduction, crossover, and mutation. They use only the payoff (objective function) information and probabilistic transition rules for moving to the next iteration. Of all the evolutionarily inspired approaches, Gas seem particularly suited to implementation using DNA, protein, and other bioinformatics tasks (Needleman and Wunsch, 1970). This is because GAs are generally based on manipulating populations of bit-strings using both crossover and point-wise mutation.

Advantages

1. Several tasks in bioinformatics involve optimization of different criteria (such as energy, alignment score, and overlap strength), thereby making the application of Gas more natural and appropriate.
2. Problems of bioinformatics seldom need the exact optimum solution; rather, they require robust, fast, and close approximate solutions, which GAs are known to provide efficiently.
3. GAs can process, in parallel, populations billions times larger than is usual for conventional computation. The usual expectation is that larger populations can sustain larger ranges of genetic variation, and thus can generate high-fitness individuals in fewer generations.
4. Laboratory operations on DNA inherently involve errors. These are more tolerable in executing evolutionary algorithms than in executing deterministic algorithms. (To some extent, errors may be regarded as contributing to genetic diversity—a desirable property.)

Applications

The most suitable applications of GAs in bioinformatics are:

1. Alignment and comparison of DNA, RNA, and protein sequences (Chen et al., 1999; Smith and Waterman, 2001; Murata and Ishibuchi, 1996; Zhang, 1994; Szustakowski and Weng, 2000; Hanada et al., 2002; Anbarasu et al., 1998; Nguyen et al., 2002; Gaspin and Schiex, 1997),
2. Gene mappings in chromosomes (Chen et al., 1999; Hurao et al., 2002; Fickett and Cinkosky, 1993; Fickett, 1996),
3. Gene finding and promoter identification from DNA sequences (Kel et al., 1998; Levitsky and Katokhin, 2003; Knudsen, 1999; Luscombe et al., 2000),
4. Interpretation of gene expression and micro array data (Quackenbush, 2001; Tsai et al., 2002; Tsai et al., 2004; Wu and Garibay, 2002; Akutsu et al., 1999),

5. Gene regulatory network identification (Chen et al., 1999; Ando and Iba, 2001; Behera and Nanjundiah, 1997; Ando and Iba, 2000; Leping et al., 2007; Tominaga et al., 1999; Lewis, 1998),
6. Construction of phylogenetic tree for studying evolutionary relationship (Lemmon and Milinkovitch, 2002; Katoh et al., 2001; Matsuda, 1996; Skourikhine, 2000)
7. DNA structure prediction (Baldi and Baisnee, 2000; Anselmi et al., 2000; Landau and Lifshitz, 1970; Becker and Buydens, 1997; Parbhane et al., 2000),
8. RNA structure prediction (Adrahams and Breg, 1990; Waterman, 1988; Zuker and Stiegler, 1981; Batenburg et al., 1995; Gulyaev et al., 1995; Wiese and Glen, 2003; Shapiro and Navetta, 1994; Shapiro and Wu, 1996; Shapiro et al., 2001),
9. Protein structure prediction and clustering (Ghou and Fasmann, 1978; Riis and Krogh, 1996; Qian and Sejnowski, 1988; Salamov and Solovyev, 1995; Salzberg and Cost, 1992; Garnier et al., 1996; Unger and Moul, 1993; Schulze-Kremer, 2000; Morris et al., 1998; Chen et al., 1998),
10. Molecular design and molecular docking (Rosin et al., 1997; Yang and Kao, 2000; Oshiro et al., 1995; Clark and Westhead, 1996; Venkatasubramanian et al., 1994; Deaven and Ho, 1995; Jones et al., 1995; Jones et al., 1999; McGarrah and Judson, 1993; Hou et al., 1999; Hatzigeorgiou and Reckzo, 2004) etc.

4. Bioinformatics Tools

Some of the important soft computing based tools are listed below in table-1.

Table 1: Prominent research area wise tools in bioinformatics

Research Area	Application	Reference
Sequence alignment	BLAST	http://www.ncbi.nlm.nih.gov/BLAST/
	FASTA	http://www.ebi.ac.uk/fasta33/
Multiple sequence alignment	ClustalW	http://www.ebi.ac.uk/clustalw/
	MultAlin	http://prodes.toulouse.inra.fr/multalin/multalin.html
	DiAlign	http://www.genomatix.de/cgi-bin/dialign/dialign.pl
Gene finding	Genscan	http://genes.mit.edu/GENSCAN.html
	GenomeScan	http://genes.mit.edu/genomescan/
	GeneMark	http://www.ebi.ac.uk/genemark/
Protein domain analysis and identification	Pfam	http://www.sanger.ac.uk/Software/Pfam/
	BLOCKS	http://www.blocks.fhcrc.org/
	ProDom	http://prodes.toulouse.inra.fr/prodom/current/html/home.php
Pattern identification	Gibbs Sampler	http://hayesweb.wadsworth.org/gibbs/gibbs.html
	AlignACE	http://atlas.med.harvard.edu/cgi-bin/alignace.pl
Protein folding prediction	MEME	http://meme.sdsc.edu/meme/website/intro.html
	PredictProtein	http://www.embl-heidelberg.de/predictprotein/predictprotein.html
	SwissModle	http://www.expasy.org/swissmod/SWISS-MODEL.html

5. Conclusions

Bioinformatics is a developing interdisciplinary science. The involvement of other sciences (such as computer science) holds great promise; this century's major research and development efforts will likely be in the biological and health sciences. Computer science departments planning to diversify their offerings can thus only gain through early entry into bioinformatics. Even using minimal resources, such efforts are wise, as computer science graduates will enhance their employment qualifications. Still unclear is whether bioinformatics will eventually become an integral part of computer science (in the same way as, say, computer graphics and databases) or will develop into an independent application. Regardless of the outcome, computer scientists are sure to benefit from being active and assertive partners with biologists.

References

- [1] A. Kel, A. Ptitsyn, V. Babenko, S. Meier-Ewert and H. Lehrach (1998), "A genetic algorithm for designing gene family-specific oligonucleotide sets used for hybridization: The G protein-coupled receptor protein superfamily", *Bioinformatics*, Vol. 14, No. 3, pp. 259–270.
- [2] A. Narayanan, E. Keedwell, and B. Olsson (2003), "Artificial Intelligence Techniques for Bioinformatics", *Applied Bioinformatics*, Vol.1, No. 4, pp. 191-222.
- [3] A. P. Gulyaev, V. Batenburg and C. W. A. Pleij (1995), "The computer simulation of RNA folding pathways using a genetic algorithm", *J. Mol. Biol.*, Vol. 250, pp. 37–51.
- [4] A. R. Lemmon and M. C. Milinkovitch (2002), "The metapopulation genetic algorithm: An efficient solution for the problem of large phylogeny estimation", *Proc. Nat. Acad. Sci.*, Vol. 99, No. 16, pp. 10516–10521.
- [5] A. S. Wu and I. Garibay (2002), "The proportional genetic algorithm: Gene expression in a genetic algorithm", *Genetic Programm. Evol. Hardware*, Vol.3, No. 2, pp. 157–192.
- [6] A. Salamov and V. Solovyev (1995), "Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments", *J. Mol. Biol.*, Vol. 247, pp. 11–15.
- [7] A. Skourikhine (2000), "Phylogenetic tree reconstruction using self-adaptive genetic algorithm", In *IEEE Int. Symp. Bio-Informatics and Biomedical Engineering*, pp.129–134.
- [8] Adriaenssens V., Baets B. D., Goethals P. L. M. and Pauwa N. D. (2004), "Fuzzy rule-based models for decision support in ecosystem management", *Science of The Total Environment*, Vol. 319, pp. 1-12.
- [9] B. A. Shapiro and J. C. Wu (1996), "An annealing mutation operator in the genetic algorithms for RNA folding", *Comput. Appl. Biosci.*, Vol. 12, pp.171–180.
- [10] B. A. Shapiro and J. Navetta (1994), "A massively parallel genetic algorithm for RNA secondary structure prediction", *J. Supercomput.*, Vol. 8, pp. 195–207.
- [11] B. A. Shapiro, J. C. Wu, D. Bengali and M. J. Potts (2001), "The massively parallel genetic algorithm for RNA folding: MIMD implementation and population variation", *Bioinformatics*, vol. 17, no. 2, pp. 137–148.
- [12] Baldi P. and Brunak S. (1998), "Bioinformatics: the Machine Learning Approach", MIT Press.
- [13] Bandyopadhyay S. (2005), "An efficient technique for super family classification of amino acid sequences: feature extraction, fuzzy clustering and prototype selection", *Fuzzy Sets and Systems*, 152(1): 5–16.
- [14] Belacel N., Cuperlović-Culf M., Laflamme M., Ouellette R. Fuzzy, "J-Means and VNS methods for clustering genes from microarray data", *Bioinformatics*, 20(11): 1690–1701.
- [15] Bicciato S., Pandin M., Didone G., and Bello, C. (2001), "Analysis of an Associative Memory neural Network for Pattern Identification in Gene expression data", *BIOKDD01, Workshop on Data mining in Bioinformatics*.

- [16] Blankenbecler R., Ohlsson M., Peterson C., Ringnér M. (2003), "Matching protein structures with fuzzy alignments", *Proceedings of the National Academy of Sciences of the United States of America*, 100(21):11936–11940.
- [17] Brusica V., van Endert P., Zeleznikow J., Daniel S., Hammer J., and Petrovsky N. I. (1999), "A neural network model approach to the study of human TAP transporter", *In Silico Biol.* 1: pp. 9–21.
- [18] Brusica V., Rudy G., Honeyman G., Hammer J. and Harrison L. (1998), "Prediction of MHC class II-binding peptides using an evolutionary Algorithm and artificial neural network", *Bioinformatics*, 14, pp. 121-130.
- [19] C. Anselmi, G. Bocchinfuso, P. De Santis, M. Savino and A. Scipioni (2000), "A theoretical model for the prediction of sequence-dependent nucleosome thermodynamic stability", *J. Biophys.*, Vol. 79, No. 2, pp. 601–613.
- [20] C. Chen, L. H. Wang, C. Kao, M. Ouhyoung and W. Chen (1998), "Molecular binding in structure-based drug design: A case study of the population based annealing genetic algorithms", *In Proc. IEEE Int. Conf. Tools with Artificial Intelligence*, pp. 328–335.
- [21] C. D. Rosin, R. S. Halliday, W. E. Hart and R. K. Belew (1997), "A comparison of global and local search methods in drug docking", *In Proc. Int. Conf. Genetic Algorithms*, pp. 221–228.
- [22] C. Gaspin and T. Schiex (1997), "Genetic algorithms for genetic mapping", *In Proc. 3rd Eur. Conf. Artificial Evolution*, pp. 145–156.
- [23] C. H. Wu and J. McLarty (2000), "Neural Networks and Genome Informatics", Elsevier Science.
- [24] C. M. Oshiro, I. D. Kuntz and J. S. Dixon (1995), "Flexible ligand docking using a genetic algorithm", *J. Comput.-Aided Mol. Design*, Vol. 9, No. 2, pp. 113–130.
- [25] C. Zhang (1994), "A genetic algorithm for molecular sequence comparison", *In Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, Vol. 2, pp. 1926–1931.
- [26] Carleos C., Rodriguez F., Lamelas H., Baro J. A. (2003), "Simulating complex traits influenced by genes with fuzzy-valued effects in pedigreed populations", *informatics*, 19(1): 144–148.
- [27] Cordón O., Gomide F., Herrera F., Hoemann F., Magdalena L. (2004), "Ten years of genetic fuzzy systems: current framework and new trends", *Fuzzy Sets and Systems*, 141(1): 5–31.
- [28] D. B. McGarrah and R. S. Judson (1993), "Analysis of the genetic algorithm method of molecular conformation determination", *J. Comput. Chem.*, Vol. 14, No. 11, pp. 1385–1395.
- [29] D. Bhandari, C. A. Murthy, and S. K. Pal (1996), "Genetic algorithm with elitist model and its convergence", *Int. J. Pattern Recognit. Artif. Intell.*, Vol. 10, No. 6, pp. 731–747.
- [30] D. E. Clark and D. R. Westhead (1996), "Evolutionary algorithms in computer aided molecular design", *J. Comput.-Aided Mol. Design*, Vol. 10, No. 4, pp. 337–358.
- [31] D. Goldberg (1989), "Genetic Algorithms in Optimization, Search, and Machine Learning", Reading, MA: Addison-Wesley.
- [32] D. M. Deaven and K. O. Ho (1995), "Molecular-geometry optimization with a genetic algorithm", *Phys. Rev. Lett.*, Vol. 75, No. 2, pp. 288–291.
- [33] D. Tominaga, M. Okamoto, Y. Maki, S. Watanabe and Y. Eguchi (1999), "Nonlinear numerical optimization technique based on a genetic algorithm for inverse problems: Towards the inference of genetic networks", *Comput. Science and Biology (Proc. German Conf. Bioinformatics)*, pp. 127–140.
- [34] Demb'el'e D., Kastner P. (2003), "Fuzzy C-means method for clustering microarray data", *Bioinformatics*, 19(8): 973–980.
- [35] Ding C. H. Q. and Dubchak I. (2001), "Multi-class protein fold recognition using support vector machines and neural networks", *Bioinformatics*, 17: 349–358.
- [36] Edwin Lughofer and Carlos Guardiola (2008), "Applying Evolving Fuzzy Models with Adaptive Local Error Bars to On-Line Fault Detection", *International Workshop on Genetic and Evolving Fuzzy Systems*.

- [37] G. Jones, P. Willett and R. C. Glen (1995), "Molecular recognition of receptor sites using a genetic algorithm with a description of desolvation", *J. Mol. Biol.*, Vol. 245, pp. 43–53.
- [38] G. Jones, P. Willett, R. C. Glen, A. R. Leach and R. Taylor (1999), "Further development of a genetic algorithm for ligand docking and its application to screening combinatorial libraries", *American Chemical Society Symposium Series*, Vol. 719, Washington, DC: ACS, pp. 271–291.
- [39] G. M. Morris, D. S. Goodsell, R. S. Halliday, R. Huey, W. E. Hart, R. K. Belew and A. J. Olson (1998), "Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function", *J. Comput. Chem.*, Vol. 19, No. 14, pp. 1639–1662.
- [40] H. D. Nguyen, I. Yoshihara, K. Yamamori and M. Yasunaga (2002), "A parallel hybrid genetic algorithm for multiple protein sequence alignment", In *Proc. Congress Evolutionary Computation*, Vol., pp. 309–314.
- [41] H. K. Tsai, J. M. Yang and C. Y. Kao (2002), "Applying genetic algorithms to finding the optimal order in displaying the microarray data", In *Proc. GECCO*, pp. 610–617.
- [42] H. K. Tsai, J. M. Yang, Y. F. Tsai and C. Y. Kao (2004), "An evolutionary approach for gene expression patterns", *IEEE Trans. Inf. Technol. Biomed.*, Vol. 8, No. 2, pp. 69–78.
- [43] H. Matsuda (1996), "Protein phylogenetic inference using maximum likelihood with a genetic algorithm", *Pacific Symp. Biocomputing*, pp. 512–523.
- [44] H. Murao, H. Tamaki, and S. Kitamura (2002), "A coevolutionary approach to adapt the genotype-phenotype map in genetic algorithms", In *Proc. Congress Evolutionary Computation*, Vol. 2, pp. 1612–1617.
- [45] Hatzigeorgiou, A. G. Reckzo, M. (2004), "Signal peptide prediction on DNA sequences with artificial neural networks", *Biomedical Circuits and Systems*.
- [46] Heger A, Holm L. (2003), "Sensitive pattern discovery with 'fuzzy' alignments of distantly related proteins", *Bioinformatics*, 19(suppl1): i130–i137.
- [47] Huang Y., Li Y. (2004), "Prediction of protein subcellular locations using fuzzy k-NN method", *Bioinformatics*, 20(1): 21–28.
- [48] J. Setubal and J. Meidanis (1999), "Introduction to Computational Molecular Biology", Boston, MA: Thomson.
- [49] J. Chen, E. Antipov, B. Lemieux, W. Cedeno, and D. H. Wood (1999), "DNA computing implementing genetic algorithms", In *Evolution as Computation*, New York: Springer-Verlag, pp. 39-49.
- [50] J. D. Szustakowski and Z. Weng (2000), "Protein structure alignment using a genetic algorithm", *Proteins*, Vol. 38, No. 4, pp. 428–440.
- [51] J. Fickett and M. Cinkosky (1993), "A genetic algorithm for assembling chromosome physical maps", In *Proc. 2nd Int. Conf. Bioinformatics, Supercomputing, and Complex Genome Analysis*, pp. 272–285.
- [52] J. Garnier, J. F. Gibrat, and B. Robson (1996), "GOR method for predicting protein secondary structure from amino acid sequence", In *Methods in Enzymology*, Vol. 266, pp. 540–553.
- [53] J. M. Yang and C. Y. Kao (2000), "A family competition evolutionary Algorithm for automated docking of flexible ligands to proteins", *IEEE Trans. Inf. Technol. Biomed.*, Vol. 4, No. 3, pp. 225–237.
- [54] J. P. Adrahams and M. Breg (1990), "Prediction of RNA secondary structure including pseudoknotting by computer simulation", *Nucleic Acids Res.*, Vol. 18, pp. 3035–3044.
- [55] J. Quackenbush (2001), "Computational analysis of microarray data", *Nat. Rev. Genetics*, Vol. 2, pp. 418–427.
- [56] J. W. Fickett (1996), "Finding genes by computer: The state of the art", *Trends Genetics*, Vol. 12, No. 8, pp. 316–320.
- [57] K. C. Wiese and E. Glen (2003), "A permutation-based genetic algorithm for the RNA folding problem: A critical look at selection strategies, crossover operators, and representation issues", *Biosystems*, Vol. 72, No. 1–2, pp. 29–41.

- [58] K. Chenand, L. Kurgan (2007), "PFRES: protein fold classification by using evolutionary information and predicted secondary structure", *Bioinformatics*, 23(21): 2843 – 2850.
- [59] K. Hanada, T. Yokoyama and T. Shimizu (2000), "Multiple sequence alignment by genetic algorithm", *Genome Inf.*, Vol. 11, pp. 317–318.
- [60] K. Katoh, K. Kuma and T. Miyata (2001), "Genetic algorithm-based maximum likelihood analysis for molecular phylogeny", *J. Mol. Evol.*, Vol. 53, No. 4-5, pp. 477–484.
- [61] Kari Torkkola, Robert Mike Gardner, Tamma Kaysser-Kranich and Calvin Ma (2001), "Self-organizing maps in mining gene expression data", *Information Sciences*, 139 (1-2): 79–96.
- [62] L. D. Landau and E. M. Lifshitz (1970), "Theory of Elasticity", New York: Pergamon.
- [63] L. A. Anbarasu, P. Narayanasamy and V. Sundararajan (2000), "Multiple molecular sequence alignment by island parallel genetic algorithm", *Current Sci.*, Vol. 78, No. 7, pp. 858–863.
- [64] L. B. Booker, D. E. Goldberg and J. H. Holland (1989), "Classifier systems and genetic algorithms", *Artif. Intell.*, Vol. 40, No. 1–3, pp. 235–282.
- [65] Leping Li 1,*, Yu Liang 1 and Robert L. Bass (2007), "GAPWM: a genetic algorithm method for optimizing a position weight matrix", *Bioinformatics*, 23(10): 1188-1194.
- [66] Luis Taria, Chitta Barala, and Seungchan Kim (2008), "Fuzzy c-means clustering with prior biological knowledge", *Journal of Biomedical Informatics*.
- [67] Lukac R., Plataniotis K. N., Smolka B., Venetsanopoulos A. N. (2005), "cDNA microarray image processing using fuzzy vector filtering framework", *Fuzzy Sets and Systems*, 152(1): 17–35.
- [68] M. L. Beckers, L. M. Buydens, J. A. Pikkemaat and C. Altona (1997), "Application of a genetic algorithm in the conformational analysis of methyleneacetal-linked thymine dimers in DNA: Comparison with distance geometry calculations", *J. Biomol. NMR*, Vol. 9, No. 1, pp. 25–34.
- [69] M. Mitchell, S. Forrest, and J. H. Holland (1992), "The royal road for genetic algorithms: Fitness landscapes and GA performance", In *Proc. 1st Eur. Conf. Artificial Life*.
- [70] M. Waterman (1988), "RNA structure prediction", In *Methods in Enzymology*. San Diego, CA: Academic, Vol. 164.
- [71] M. Zuker and P. Stiegler (1981), "Optimal computer folding of large RNA sequences using thermo-dynamics and auxiliary information", *Nucleic Acids Res.*, Vol. 9, pp. 133–148.
- [72] Ma Q., Wang J. T. L. and C. H. Wu. (2000), "Application of neural networks to biological data mining: A case study in DNA sequence classification", *Proceedings of SEKE-2000*, pp. 23-30.
- [73] N. Behera and V. Nanjundiah (1997), "Trans gene regulation in adaptive evolution: A genetic algorithm model", *J. Theore. Biol.*, Vol. 188, pp.153–162.
- [74] N. M. Luscombe, D. Greenbaum and M. Gerstein (2000), "What is bioinformatics? A proposed definition and overview of the field", In *Yearbook Medical Informatics: Edmonton, AB, Canada: IMIA*, pp. 83–100.
- [75] N. Qian and T. J. Sejnowski (1988), "Predicting the secondary structure of globular proteins using neural network models", *J. Mol. Biol.*, Vol. 202, No. 4, pp. 865–884.
- [76] P. Chou and G. Fasman (1978), "Prediction of the secondary structure of proteins from their amino acid sequence", *Adv. Enzymol.*, Vol. 47, pp. 145–148.
- [77] P. Baldi and P. F. Baisnee (2000), "Sequence analysis by additive scales: DNA structure for sequences and repeats of all lengths", *Bioinformatics*, Vol. 16, pp. 865–889.
- [78] P. O. Lewis (1998), "A genetic algorithm for maximum likelihood phylogeny inference using nucleotide sequence data", *Mol. Biol. Evol.*, Vol. 15, No. 3, pp. 277–283.
- [79] R. Unger and J. Moult (1993), "On the applicability of genetic algorithms to protein folding", In *Proc. Hawaii Int. Conf. System Sciences*, Vol. 1, pp. 715–725.
- [80] R. V. Parbhane, S. Unniraman, S. S. Tambe, V. Nagaraja and B. D. Kulkarni (2000), "Optimum DNA curvature using a hybrid approach involving an artificial neural network and genetic algorithm", *J. Biomol. Struct. Dyn.*, Vol. 17, No. 4, pp. 665–672.
- [81] Resson H., Reynolds R., Varghese R. S. (2003), "Increasing the efficiency of fuzzy logic-based gene expression data analysis", *Physiological Genomics*, 13(2): 107–117.

- [82] S. Ando and H. Iba (2000), "Quantitative modeling of gene regulatory network identifying the network by means of genetic algorithms", Presented at the 11th Genome Informatics Workshop.
- [83] S. Ando and H. Iba (2001), "Inference of gene regulatory model by genetic algorithms", In Proc. Congress Evolutionary Computation, Vol. 1, pp. 712–719.
- [84] S. B. Needleman and C. D. Wunsch (1970), "A general method applicable to the search for similarities in the amino acid sequence of two proteins", *J. Mol. Biol.*, Vol. 48, pp. 443–453.
- [85] S. K. Riis and A. Krogh (1996), "Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignments", *J. Comput. Biol.*, Vol. 3, pp. 163–183.
- [86] S. Knudsen (1999), "Promoter2.0: For the recognition of PolII promoter sequences", *Bioinformatics*, Vol. 15, pp. 356–361.
- [87] S. Salzberg and S. Cost (1992), "Predicting protein secondary structure with a nearest-neighbor algorithm", *J. Mol. Biol.*, Vol. 227, pp. 371–374.
- [88] S. Schulze-Kremer (2000), "Genetic algorithms and protein folding. Methods in molecular biology", *Protein Structure Prediction: Methods and Protocols*, Vol. 143, pp. 175–222.
- [89] Schlosshauer M., Ohlsson M. (2002), "A novel approach to local reliability of sequence alignments", *Bioinformatics*, 18(6):847–854.
- [90] Schneider Murray-Smith, R. and Thakar, S. (1993), "Combining case based reasoning with neural networks", In *AAAI Workshop on Case Based Reasoning*, Washington, D. C., USA.
- [91] T. Hou, J. Wang, L. Chen and X. Xu (1999), "Automated docking of peptides and proteins by using a genetic algorithm combined with a tabu search", *Protein Eng.*, Vol. 12, pp. 639–647.
- [92] T. Akutsu, S. Miyano and S. Kuhara (1999), "Identification of genetic networks from a small number of gene expression patterns under the boolean network model", In Proc. Pacific Symp. Biocomputing, Vol. 99, pp. 17– 28.
- [93] T. F. Smith and M. S. Waterman (2001), "Identification of common Informatics: Edmonton", AB, Canada: IMIA, pp. 83– 100.
- [94] T. Murata and H. Ishibuchi (1996), "Positive and negative combination effects of crossover and mutation operators in sequencing problems", *Evol. Comput.*, Vol. 20–22, pp. 170–175.
- [95] Tomida S., Hanai T., Honda H., Kobayashi T. (2002), "Analysis of expression profile using fuzzy adaptive resonance theory", *Bioinformatics*, 18(8):1073–1083.
- [96] Torres A, Nieto J. J. (2003), "The fuzzy polynucleotide space: basic properties", *Bioinformatics*, 19(5): 587–592.
- [97] V. Venkatasubramanian, K. Chan and J. Caruthers (1994), "Computer aided molecular design using genetic algorithms", *Comput. Chem. Eng.*, Vol. 18, No. 9, pp. 833–844.
- [98] V. Batenburg, A. P. Gulyaev and C. W. A. Pleij (1995), "An APL-programmed genetic algorithm for the prediction of RNA secondary structure", *J. Theoret. Biol.*, Vol. 174, No. 3, pp. 269–280.
- [99] V. G. Levitsky and A. V. Katokhin (2003), "Recognition of eukaryotic promoters using a genetic algorithm based on iterative discriminant analysis", In *Silico Biol.*, Vol. 3, No. 1–2, pp. 81–87.
- [100] Wang D. H., Lee N. K., Dillon T. S. (2003), "Extraction and optimization of fuzzy protein sequence classification rules using GRBF neural networks", *Neural Information Processing— Letters and Reviews*, 1(1): 53–59.
- [101] Woolf P. J., Wang Y. (2000), "A fuzzy logic approach to analyzing gene expression data", *Physiological Genomics*, 3(1): 9–15.
- [102] Zhong Wei1, Altun Gulsah, Tian Xinmin, Harrison Robert, Tai Phang, Pan Yi (2007), "Parallel protein secondary structure prediction schemes using Pthread and OpenMP over hyper-threading technology", *The Journal of Supercomputing*, Vol. 41, No. 1, pp. 1-16.