RECONSTRUCTION OF GENE NETWORKS USING PHENOMIC ALGORITHMS

Rio G. L. D'Souza[1], K. Chandra Sekaran[2] and A. Kandasamy[3]

[1]Department of Computer Science and Engineering, St Joseph Engineering College, Mangalore, India
rio@ieee.org
[2]Department of Computer Engineering, National Institute of Technology Karnataka - Surathkal, Mangalore, India
kchnitk@ieee.org
[3]Department of Mathematical and Computational Sciences, National Institute of Technology Karnataka - Surathkal, Mangalore, India
kandy@nitk.ac.in

ABSTRACT

The reconstruction of gene networks has become an important activity in Systems Biology. The potential for better methods of drug discovery and disease diagnosis hinges upon our understanding of the interaction networks between the genes. Evolutionary methods are proving to be successful in such problems and a number of such methods have been proposed. However, all these methods are based on processing of genotypic information. We present evolutionary algorithms for reconstructing gene networks from expression data using phenotypic interactions, thereby avoiding the need for an explicit objective function. Specifically, we implement the Phenomic algorithm and validate it for the reconstruction of gene networks. We also extend the basic phenomic algorithm to perform multiobjective optimization for gene network reconstruction. We apply both these algorithms to the yeast sporulation dataset and show that the algorithms can effectively identify gene networks. Both the algorithms are validated for stability and accuracy in the reconstruction of gene networks.

KEYWORDS

Evolutionary computing, Gene expression analysis, Gene networks, Microarray data analysis, Multiobjective optimization, Multiobjective evolutionary algorithms, Phenomic algorithms

1. INTRODUCTION

The advent of high throughput methods such as microarray technology has made it possible for biologists to study hundreds of genes at a time, and to elucidate the relationships between them. The datasets that result from such studies have high dimensionality. Hence several researchers have developed methods of analysis which can determine useful patterns from the datasets without compromising the dimensionality [1]. Gene networks represent relationships between genes, based on observations of how the expression level of each gene affects the expression levels of the others [2]. The determination of these relationships from gene expression measurements is a reverse engineering or reconstruction activity.

Evolutionary methods have been used by others [3] to analyze and capture the relationships between hundreds of genes with varying degrees of success. There is ample scope for better methods and application of better techniques. The Phenomic Algorithm, introduced in [4], presents an approach based on population dynamics. It is based on phenotypic interactions rather than genotypic mechanisms which are used in traditional genetic algorithms. Here the aim is to model gene expression record of each gene as an individual and then to let these

individuals interact in an environment that simulates the survival of the fittest. Thus the need for an explicit objective function is avoided. In this paper, we apply the phenomic algorithm to a number of microarray datasets in order to evaluate its effectiveness when compared to other evolutionary methods.

We also modify the basic phenomic algorithm to handle multiple objectives. It is possible to employ multiobjective optimization to elucidate gene networks which are more biologically plausible [5]. We have chosen to minimize the sparseness of links between genes while simultaneously maximizing the relevance of links in a particular network. We use non-dominated sorting in order to determine the pareto optimal solutions that best represent the balance between the objectives that we have chosen to optimize. We apply the multiobjective phenomic algorithm to the yeast sporulation dataset [6] and results show a marked improvement in the quality of networks discovered.

The rest of this paper is organized as follows: In section 2, we review the related work done by others. We devote section 3 to a discussion about the methodology adopted by the basic phenomic algorithm and its implementation. We discuss the modification of the basic phenomic algorithm and its implementation in section 4. Finally, section 5 presents the results and validation, followed by section 6 which concludes the paper.

2. RELATED WORK

For just about a decade now reconstruction of gene networks has acquired importance due to the dawn of systems biology. One of the first attempts is a simple method that was introduced by Somogyi et al. [7]. Liang et al. [8] developed a general algorithm using mutual information to identify a minimal set of inputs that uniquely define the output for each gene at the next time step. Akutsu et al. [9], [10] and D'haeseleer et al. [11] have also proposed several reverse engineering algorithms.

The S-system proposed by Savageau [12] has been used by some researchers [5], [13] in order to formulate an objective function for the evolutionary algorithm that they use to reverse engineer gene networks. Lubovac and Olsson [14] have suggested bringing in additional information resources into the evolutionary algorithm, so that more relevant relationships between genes can be derived. It is possible to develop better evolutionary algorithms by finding better objective functions since the critical dependency between the genotype and phenotype is characterized by them [15], [16].

The field of multiobjective optimization has developed considerably during the last decade and a number of multiobjective evolutionary algorithms (MOEAs) have been applied to the problem of reconstructing gene networks from expression data [17], [18]. Other notable evolutionary algorithms include the non-dominated sorting genetic algorithm (NSGA) and its variations which have been applied to the problem of classification of cancer based on gene expression data [19], [20], [21]. The application of MOEAs to the elucidation of gene networks is an area which is receiving a large amount of attention from researchers due to the perceived benefits in applications such as drug discovery and the diagnosis of chronic diseases.

3. THE BASIC PHENOMIC ALGORITHM

The phenomic algorithm is an algorithm which utilizes phenotypic information to simulate an environment that allows the survival of the fittest individual. It was introduced in [4] and we present a brief description of the algorithm here so that our work, which is an extension of the basic algorithm, is better understood. Like most evolutionary algorithms, the phenomic algorithm begins with of a population of individuals.

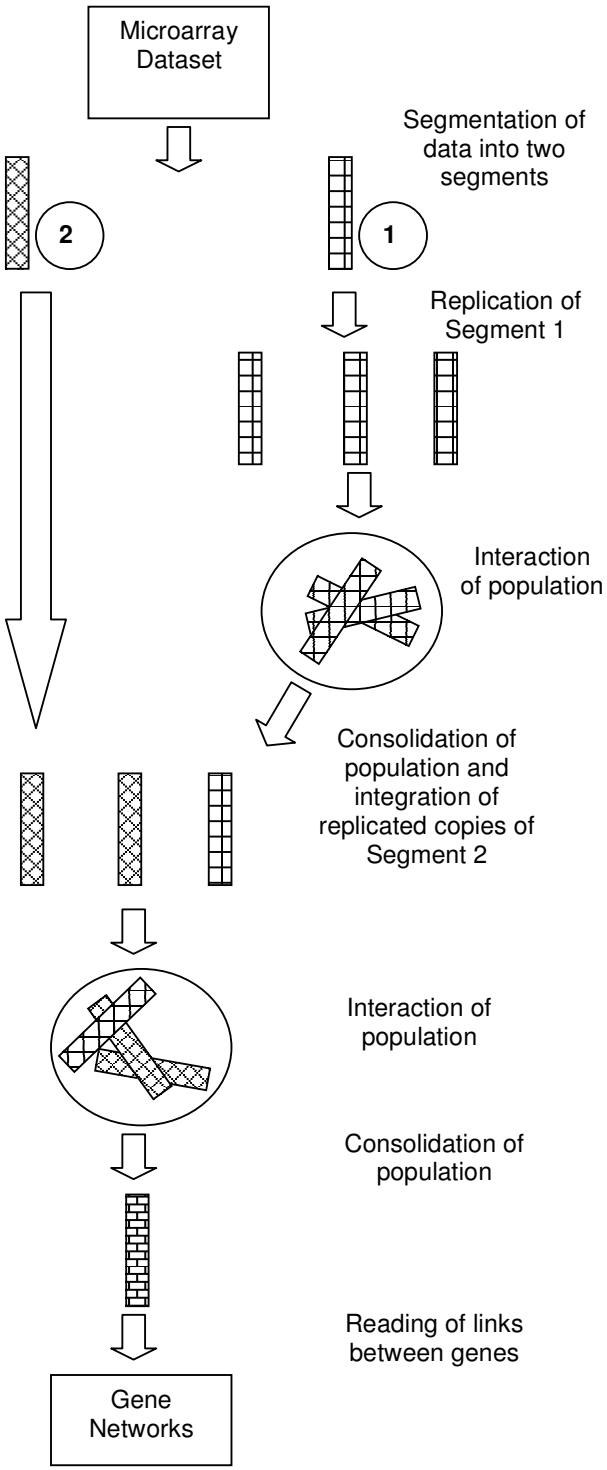

Figure 1. Sequence of processing in the basic phenomic algorithm showing, for simplicity, processing of only two segments

Each individual has genetic information embedded within it. The genotype manifests as the phenotype in the environment and an objective function is generally used in evolutionary

algorithms to characterize this dependence. In the phenomic algorithm, the expression of a gene (taken from the microarray data) is embedded within the individual. Thus each individual has a ready reference for determining its own fitness and does not need an objective function. The presence of a strong correlation between expression patterns of two genes suggests co-regulation of these genes. Co-expressed genes in the same cluster are very likely to be involved in the same cellular processes. This is the basis for elucidation of the regulatory networks.

When constructing gene networks, we study the relationship between genes. If gi and gj are objects representing two such genes, their expression patterns across m samples may be written as $g_i = \{w_{ik} \mid 1 \le k \le m\}$ and $g_j = \{w_{jk} \mid 1 \le k \le m\}$. The similarity (or proximity) between gene expression patterns can be expressed in terms of a correlation coefficient, where w_{ij} is the expression level of the i^{th} gene in the j^{th} sample and μ_{gi} is the average of expression levels of the i^{th} gene over all the samples.

One such proximity measure, shown in Eqn. (1), is called Pearson correlation coefficient [22], [23]. This proximity measure would be useful only if we want to deduce steady-state relationships between genes.

$$Pear1(g_i, g_j) = \frac{\sum_{k=1}^{m} \left(w_{ik} - \mu_{g_i}\right)\left(w_{jk} - \mu_{g_j}\right)}{\sqrt{\sum_{k=1}^{m} \left(w_{ik} - \mu_{g_i}\right)^2} \sqrt{\sum_{k=1}^{m} \left(w_{jk} - \mu_{g_j}\right)^2}} \quad (1)$$

When the microarray dataset contains records which represent the expression of each gene at m time-steps (instead of m samples) of an experiment, it is possible to verify whether the expression pattern of a gene g_i at a time-step *(t-1)* has any correlation with the expression pattern of a gene g_j at time *t*. For this, we define the Pearson correlation coefficient across time-steps (from gene g_i at time-step t = *(k-1)* to gene g_j at time-step *t* = *k*), as given in Eqn. (2). With a small loss in accuracy, the expression record of each gene is assumed to wrap-around over the time-steps, for the purpose of this calculation.

$$Pear2(g_i, g_j) = \frac{\sum_{k=1}^{m} \left(w_{i(k-1)} - \mu_{g_i}\right)\left(w_{jk} - \mu_{g_j}\right)}{\sqrt{\sum_{k=1}^{m} \left(w_{i(k-1)} - \mu_{g_i}\right)^2} \sqrt{\sum_{k=1}^{m} \left(w_{jk} - \mu_{g_j}\right)^2}} \quad (2)$$

Once the proximity measure for the genes is defined, the gene interactions such as “meet”, “know”, “like”, “dislike” can be defined as operations on genes gi and gj, as shown in Eqns. (3) to (6), where *D* represents a preset threshold distance.

$$meet(g_i, g_j)\ returns\ TRUE\ \ iff\ \ g_i\ and\ g_j\ were\ partners, at\ least\ once \quad (3)$$

$$know(g_i, g_j)\ returns TRUE\ iff\ the\ proximity measure\ for\ g_i\ and\ g_j\ is known. \quad (4)$$

$$like(g_i, g_j)\ returns\ TRUE\ \ iff\ \ proximity\ \ measure\ \ for\ g_i\ and\ g_j\ is \le D \quad (5)$$

$$dislike(g_i, g_j)\ returns TRUE\ iff\ \ proximity measure\ for\ g_i\ and\ g_j\ is > D \quad (6)$$

These operations determine the character of the phenotypic interactions that take place between gene objects. By storing links between genes that “like” each other it is possible to elucidate the

relationships that are required for reconstructing the gene network. We show the sequence of processing in the basic phenomic algorithm in Figure 1. We present the algorithm itself in Figure 2. The structure is very similar to a genetic algorithm since phenotypic processing is encountered in every generation, just like in a genetic algorithm.

```
basic_phenomic_algorithm( )
{
divide gene expression data into segments;
initialize population with first segment replicated;
set segment count to 0;

while population has not reduced to size of single segment and
there are more segments to process
        {
        interact_population;
        consolidate_population;

        replicate and add next segment;
        increment segment count;
        }

read gene-links stored in the final population;
display gene networks constructed from links;
}

interact_population( )
{
for a preset number of iterations
        {
        randomly select two individuals from current population;

        apply interaction criteria in Eqns. (3) to (6);
        update gene-links of both individuals;
        }
}

consolidate_population( )
{
for a preset number of iterations
        {
        randomly select two individuals from current population;

        if the indices of both individuals are same
                eliminate one of them after copying its links;
        }
}
```

Figure 2. The basic phenomic algorithm and its main functions

We give a brief description of the main functions of the basic phenomic algorithm here:

1. Modelling genes as individuals: While modelling the genes as individuals, we embed the expression profile of the gene within the object itself. Also we store the relationships with other genes, which are discovered during the interaction phase, within the individual itself. We ensure sufficient density of individuals by replicating them as required.

2. Simulating gene interaction: We set the stage for the survival-of-the-fittest by letting individuals to meet randomly. Eqns. (3) to (6) define the nature of these interactions between partners that meet. Partners would meet, know, like, or dislike each other depending upon the closeness of their expression profiles and whether they have met already.

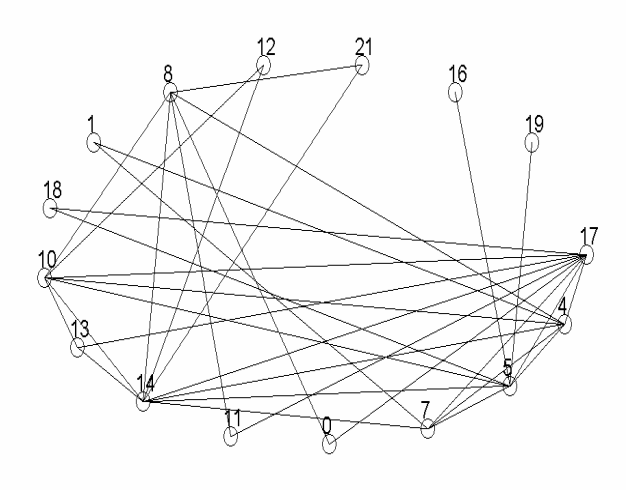

Figure 3. A typical gene network generated by the basic phenomic algorithm, when D = 0.2

3. Enforcing natural processes: From time to time we consolidate the population by eliminating individuals which are replicates and have not acquired any links with other individuals. Thereafter we bring in the remaining segments of the data, one by one, till all segments have been considered. At the end of the process, the links between the genes, which are stored in the individuals, are used to construct the gene networks.

As seen from experimental results in Figure 3, the algorithm is able to discover links between genes when applied to gene expression data. The Pearson correlation coefficient, as given in Eqn. (2), is used to determine the distance between gene profiles across time-steps, in a time-varying dataset. Only those gene relationships which are closer than a preset distance threshold *D* are considered significant.

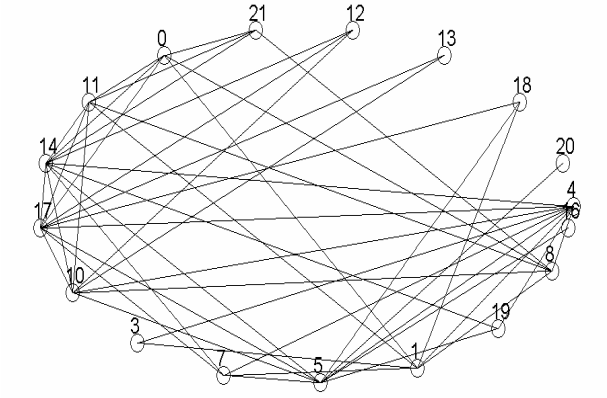

Figure 4. A typical gene network generated by the multiobjective algorithm, when D = 0.2

4. THE MULTIOBJECTIVE PHENOMIC ALGORITHM

Gene networks discovered by the basic phenomic algorithm are based on correlation between expression profiles of the genes. When the correlation distance is set very low (< 0.05) only very few links are recovered. However when the correlation distance is set high (> 0.25) the links are too numerous and obviously implausible. The optimal setting lies in between these two extreme cases. It is well known that most biological networks display the small-world network property that predicates sparseness between key nodes and dense local connections around each key node [5].

Table 1. Yeast Dataset Gene Legend.

Index	Gene Name
1	CDC14
2	CDC16
3	CDC20
4	CDC23
5	CDC5
6	ISC10
7	MEI5
8	MPS1
9	MSH4
10	MSH5
11	MSI1
12	NDT80
13	POL30
14	RAD51
15	RAD54
16	RAP1
17	RFA1
18	SAE3
19	SPO20
20	SPR6
21	SPS18
22	SPS10
23	YPT1

Index	Gene Name
24	ZIP2
25	DMC1
26	HOP1
27	IME2
28	IME4
29	MEI4
30	MEK1
31	REC102
32	REC104
33	REC114
34	RED1
35	SPO11
36	SPO13
37	SPO16
38	ZIP1
39	SPO12
40	SPS1
41	DIT1
42	DIT2
43	SGA1
44	SPR1
45	SPR3

In a conventional multiobjective evolutionary algorithm, we could use the similarity of the target network to small-world networks as an objective in order to determine the network that has the optimal number of links. Also, since we intend to find as many links as possible, we could use the number of links discovered as the other objective. When creating the new population, as required in any evolutionary algorithm, we could select individuals after non-dominated sorting [19] based upon the two objectives: number of links (*NOL*) and small-world similarity factor (*SWSF*) [5], which are shown in Eqn. (7) and Eqn. (8).

$$NOL = \sum_{i=1}^{N} l_{ij} \qquad (7)$$

$$SWSF = \sum_{i=1}^{N} \sum_{j=1}^{N} l_{ij} \qquad (8)$$

Where $l_{ij} = 1$ if gene g_i is linked to gene g_j, else $l_{ij} = 0$ and N is the total number of genes in the target network. We could maximize objective *NOL*, while we minimize objective *SWSF*.

However, the phenomic algorithm does not need to explicitly evaluate these objectives. At the consolidation phase of the phenomic algorithm, when two individuals have the same gene ID, we copy a consensus network comprising of the unique links from both the individuals, to one of the individuals, and the other is discarded. Hence we create the new population without explicitly evaluating or ranking the individuals. The consensus mechanism ensures that all the links discovered are retained in the population. Thus the first objective of maximizing the number of links is always met. It should be noted here that this reduces competition, which is the essential feature of genetic algorithms and is required when large search spaces need to be explored.

The second objective of ensuring that networks discovered are similar to small-world networks is also always met because we do not have random recombination as in genetic algorithms. Random recombination (crossover) brings in the possibility of invalid combinations which might deviate from the natural small-world networks. We explore only valid links and hence the networks discovered are always biologically relevant.

Thus we perform multiobjective optimization within the phenomic algorithm. The experimental results of this multiobjective algorithm are shown in Figure 4. For comparison, we have also implemented the multiobjective algorithm wherein we explicitly evaluate the two objectives as in Eqns. (7) and (8). Thereafter we assign ranks through non-dominated sorting and proceed as in any typical multiobjective evolutionary algorithm. The performance of this implementation is compared with that of our multiobjective phenomic algorithm in the next section.

5. RESULTS AND DISCUSSION

In this study, we used expression data from a study by Chu et al. [6]. Saccharomyces cerevisiae (common yeast) was synchronized by transferring to a sporulation medium at time t=0 to maximize the synchrony of sporulation. RNA was harvested at time t = 0, 0.5, 2, 5, 7, 9 and 11.5 hours after transfer to sporulation medium. Each gene's mRNA expression level just before transfer was used as control.

Expression profiles of about 6100 genes are included in this dataset. Using them, we followed the same method as Chu et al. [6] to extract the genes that showed significant increase of mRNA levels during sporulation. Among them, we finally selected 45 genes, whose functions are biologically characterized by Kupiec et al. [24].

Typical gene networks obtained from this yeast dataset are shown in Figure 3, when applying the basic phenomic algorithm and in Figure 4, when applying the multiobjective variant of the same algorithm. Table 1 gives the legend of the gene names for the indices used in the figures. A visual inspection of the networks shows a marked increase in the connectivity between genes when using the multiobjective phenomic algorithm.

We validated these results by performing 10-fold leave-one-out-crossover validation (LOOCV). We made ten runs of each algorithm and compared the gene networks from of each run taken separately against the consensus gene networks of the other nine runs.

Table 2. Validation results using 10-fold LOOCV with yeast sporulation dataset.

Validation metric	**Basic phenomic algorithm**	**Multiobjective evolutionary algorithm (NSGA)**	**Multiobjective phenomic algorithm**
Stability Factor, SF	0.68	0.76	0.82
Accuracy Factor, AF	0.74	0.95	0.97

The average number of correctly-identified edges resulting from all the ten comparisons indicates the stability of the algorithm. The complement of the average number of incorrectly-identified edges resulting from all the ten comparisons indicates the accuracy of the algorithm. Note that, for any given consensus network, the sum of the correctly-identified and incorrectly-identified edges is not necessarily equal to the total number of edges in the network. This is because, in some cases, there might be existing edges that are not discovered at all. We formally define these metrics in Eqn. (9) and Eqn. (10).

$$\text{Stability Factor,} \qquad SF = \frac{1}{n}\sum_{i=1}^{n}\frac{CE_i}{E_i} \tag{9}$$

$$\text{Accuracy Factor,} \qquad AF = 1 - \frac{1}{n}\sum_{i=1}^{n}\frac{IE_i}{E_i} \tag{10}$$

Where CE_i is the number of correctly identified edges in the i^{th} comparison, IE_i is the number of incorrectly identified edges in the i^{th} comparison, n is the total number of comparisons, which is ten in our case, and E_i is the total number of edges in the i^{th} consensus network.

The results of the validation tests are given in Table 2. As seen, the algorithms perform well in terms of stability, as well as accuracy. Due to the stochastic nature of the algorithms, the results obtained vary from run to run. However, we have statistically validated the results and found that the gene networks are elucidated both stably and accurately. Hence these algorithms could be viable alternative methods for determination of gene networks in general.

6. CONCLUSION

We have presented the reconstruction of gene networks using the basic phenomic algorithm and also validated it for stability and accuracy. The phenomic nature of the algorithm is manifested in its focus on the phenotypic, rather than genetic, information of an individual. Due to the implicit survival-of-the-fittest mechanisms the need for an explicit objective function was avoided. The algorithm was applied to yeast sporulation data and the resulting gene networks are found to be biologically relevant, when compared to the networks found at the Saccharomyces genome database [25].

The multiobjective variant of the phenomic algorithm performs better on the validation metrics, but this comes at a higher computational cost. Currently we are working on applying these algorithms to other datasets in order to study their effectiveness as optimization tools.

REFERENCES

[1] Schulze, A. and Downward J. (2001), "Navigating gene expression using microarrays - a technology review", *Nature Cell Biology*, Vol. 3, pp. E190-E195, Aug 2001.

[2] Soinov, L.A., Krestyaninova M.A. and Brazma A. (2003), "Towards reconstruction of gene networks from expression data by supervised learning", *Genome Biology*, 4(1), pp. R6.

[3] D'haeseleer, P., Liang, S. and Somogyi R. (1999), "Gene expression analysis and genetic network modelling: Tutorial", Pacific Symposium on Biocomputing (PSB '99).

[4] D'Souza, R.G.L., Chandra Sekaran K. and Kandasamy A. (2007), "A phenomic algorithm for reconstruction of gene networks", CICI 2007, pp. 53-58, IV International Conference on Computational Intelligence and Cognitive Informatics, Venice, WASET, Nov 2007.

[5] Spieth, C., Streichert, F., Speer, N. and Zell, A. (2004) "Optimizing topology and parameters of gene regulatory network models from time series experiments", Genetic and Evolutionary Computation Conference (GECCO 04) in *Lecture Notes in Computer Science 3012* (LNCS 3012) Deb et al. (Eds.) Vol. 2, pp. 461-470, Springer.

[6] Chu, S., DeRisi, J., Eisen, M., et al. (1998), "The transcriptional program of sporulation in budding yeast", *Science*, 282, pp. 699-705.

[7] Somogyi, R., Fuhrman, S., Askenazi, M. and Wuensche, A. (1997), "The gene expression matrix: towards the extraction of genetic network architectures", Proc. of Second World Cong. of Nonlinear Analysts (WCNA96), 30(3), pp. 1815-1824.

[8] Liang, S., Fuhrman, S. and Somogyi, R. (1998), "REVEAL, a general reverse engineering algorithm for inference of genetic network architectures", Pacific Symp. on Biocomputing, 3, pp. 18-29.

[9] Akutsu, T., Miyano, S. and Kuhara, S. (1999), "Identification of genetic networks from a small number of gene expression patterns under the boolean network model", Pacific Symp. on Biocomputing, 4, pp. 17-28.

[10] Akutsu, T., Miyano, S. and Kuhara, S. (2000), "Algorithms for inferring qualitative models of biological networks", Pacific Symp. on Biocomputing.

[11] D'haeseleer, P., Liang, S. and Somogyi, R. (2000), "Genetic network inference: from co-expression clustering to reverse engineering", *Bioinformatics*, 16(8), pp. 707-726.

[12] Savageau, M.A. (1995), "Power-law formalism: a canonical nonlinear approach to modelling and analysis", Proceedings of the World Congress of Nonlinear Analysts '92, pp. 3323-3334.

[13] Noman, N. and Iba, H. (2005), "Reverse engineering genetic networks using evolutionary computation", *Genome Informatics*, 16(2), pp. 205-214.

[14] Lubovac, Z. and Olsson, B. (2003), "Towards reverse engineering of genetic regulatory networks", *Technical Report No. HS-IDA-TR-03-003*, University of Skovde, Sweden.

[15] Kampis, G. (2002), "A Causal Model of Evolution", Proc. of 4th Asia-Pacific Conf. on Simulated Evolution and Learning (SEAL 02), pp. 836-840.

[16] Dawkins, R. (1988), *The blind watchmaker*, Penguin Books.

[17] Van Veldhuizen, D.A. and Lamont, G.B. (2000), "Multiobjective evolutionary algorithms: analyzing the state-of-the-art", *Evolutionary Computation*, Vol. 8, No. 2, pp. 125-147.

[18] Deb, K. (2001), *Multi-objective optimization using evolutionary algorithms*, Wiley, Chichester, UK.

[19] Deb, K. and Reddy, A.R. (2003), "Classification of two-class cancer data reliably using evolutionary algorithms", Publ. of Kanpur Genetic Algorithms Lab., India, *KanGAL Report No. 2003001*.

[20] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002), "A fast and elitist multi-objective genetic algorithm: NSGA-II", *IEEE Trans. Evolutionary Computation*, Vol. 6, No. 2, pp. 182-197.

[21] Kumar, P. K., Sharath, S., D'Souza, R.G. and Chandra Sekaran K. (2007), "Memetic NSGA—A multi-objective genetic algorithm for classification of microarray data", adcom, pp. 75-80, 15th Intl. Conf. on Advanced Computing and Communications (ADCOM 2007), IEEE.

[22] Kuo, W.P., Mendez, E., Chen, C., et al. (2003), "Functional relationships between gene pairs in oral squamous cell carcinoma", Proc. of AMIA Symp. 2003, pp. 371-375.

[23] Stekel, D. (2003), *Microarray bioinformatics*, Cambridge University Press.

[24] Kupiec, M., Ayers, B., Esposito, R.E. and Mitchell, A.P. (1997), "The molecular and cellular biology of the yeast Saccharomyces", Cold Spring Harbour, 889–1036.

[25] SGD project (2007), *Saccharomyces genome database*, http://www.yeastgenome.org/ (15/9/2007).

Authors

Rio G. L. D'Souza is a Research Scholar at the Department of Computer Engineering at National Institute of Technology Karnataka, India. He is currently on sabbatical leave from St Joseph Engineering College, Mangalore. His research interests include Soft Computing, Computer Networks, and Bioinformatics. He is a member of IEEE and the IEEE Computational Intelligence Society.

Dr. K. Chandra Sekaran is a Professor of Computer Engineering at National Institute of Technology Karnataka, India. His research includes Computer Networks, Dependable network/Distributed computing, Autonomic computing and Community Informatics. He has 20 years of teaching and research and one year Industry experience. He has published more than 86 publications in International and National proceedings and authored two books.

Dr. A. Kandasamy is a Professor at the Department of Mathematical and Computational Sciences, National Institute of Technology Karnataka, India. In the recent past, he has been a Visiting Faculty at IES, School of Engineering & Technology, Asian Institute of Technology, Thailand. His research interests include Computational Techniques and Algorithms, Computational Fluid Dynamics, Optimization Techniques, Stochastic Processes, Genetic Algorithms.