Inference of Gene Networks from Microarray Data through a Phenomic Approach

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

[1] St Joseph Engineering College, Mangalore, India
[2] National Institute of Technology Karnataka, Surathkal, Mangalore, India
rio@ieee.org, kchnitk@ieee.org, kandy@nitk.ac.in

Abstract. The reconstruction of gene networks is crucial to the understanding of cellular processes which are studied in Systems Biology. The success of computational methods of drug discovery and disease diagnosis is dependent upon our understanding of the biological basis of the interaction networks between the genes. Better modelling of biological processes and powerful evolutionary methods are proving to be a key factor in the solution of such problems. However, most of these methods are based on processing of genotypic information. We present an evolutionary algorithm for inferring gene networks from expression data using phenotypic interactions. The benefit of this is that we avoid the need for an explicit objective function in the optimization process. In order to realize this, we have implemented a method called as the Phenomic algorithm and validated it for stability and accuracy in the reconstruction of gene networks.

Keywords: Gene networks, Microarray data analysis, Genetic algorithms, Fitness function, 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. 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 [5]. Here the aim is to model gene expression record of each gene as an individual and then to let these individuals

V. V Das, R. Vijaykumar et al. (Eds.): ICT 2010, CCIS 101, pp. 83–89, 2010.

interact in an environment that simulates the survival of the fittest. Thus the need for an explicit objective function is avoided. We apply the phenomic algorithm to the yeast sporulation dataset [6] and results show a marked improvement in the quality of networks discovered.

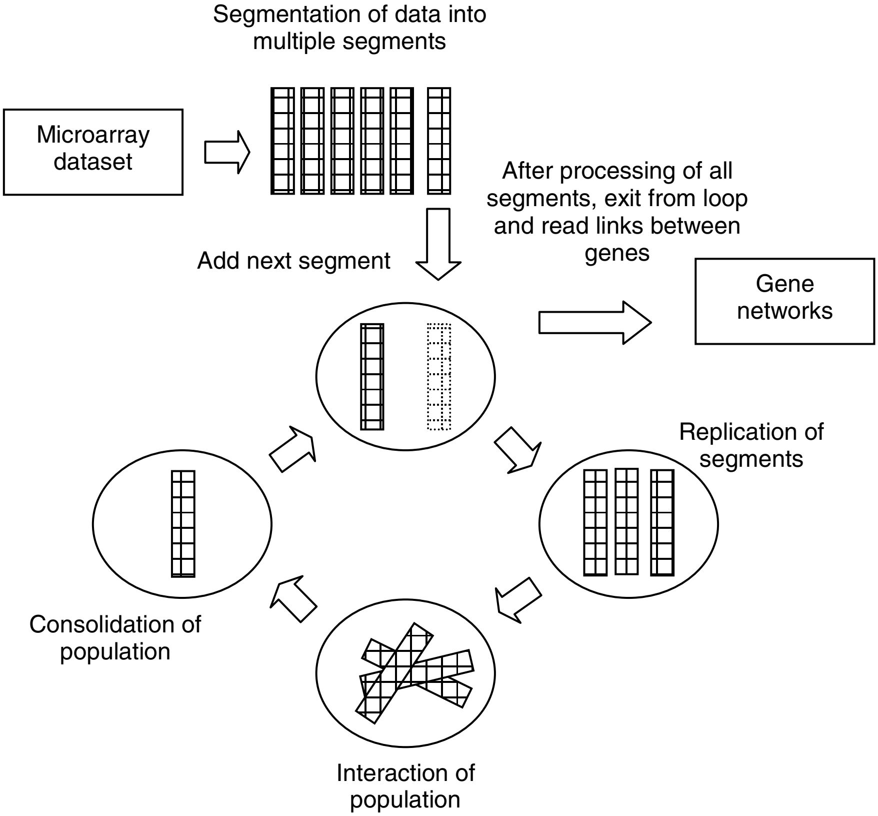
Segmentation of data into multiple segments
Microarray dataset
After processing of all segments, exit from loop and read links between genes
Add next segment
Gene networks
Replication of segments
Consolidation of population
Interaction of population

Fig. 1. Sequence of processing in the basic phenomic algorithm

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. Finally, section 4 presents the results and validation, followed by section 5 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].

3 The Phenomic Approach

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.

When constructing gene networks, we study the relationship between genes. If g_i and g_j 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 ith gene in the jth sample and μ_{gi} is the average of expression levels of the ith gene over all the samples.

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 $(t\text{-}1)$ has any correlation with the expression pattern of a gene g_j at time t. For this, we define the Pearson correlation coefficient [17] across time-steps (from gene gi at time-step $t = (k\text{-}1)$, to gene g_j at time-step $t = k$), as given in Eqn. (1).

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

In the algorithm that follows, random pairs of genes are considered at a time and the proximity measure for each pair of genes is calculated. Only those gene pairs that have a proximity measure less than a preset threshold distance D are assumed to have an interaction between them. This condition is shown in Eqn. (2).

$$Pear\ (g_i, g_j) < D\ . \tag{2}$$

In the next step of processing, these closely related pairs are analyzed further to verify whether causal relationships exist between them. We show the sequence of processing in the basic phenomic algorithm in Fig. 1.

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. (1) and (2) define the nature of these interactions between partners that meet. If a pair of genes has a close proximity as defined by the time-shifted Pearson correlation coefficient, a link is stored between them.

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.

We present the phenomic algorithm below. The structure is very similar to a genetic algorithm since phenotypic processing is encountered in every generation, just like in a genetic algorithm.

The basic phenomic algorithm and its main functions.

```
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;
    {
    select two individuals from current population;
```

```
        apply interaction criteria in Eqn. (2)
        update gene-links of both individuals;
        }
}

consolidate_population( )
{
for a preset number of iterations;
        {
        select two individuals from current population;
        if the indices of both individuals are same
                eliminate one of them after copying its links;
        }
}
```

As seen from experimental results in Fig. 2, the algorithm is able to discover links between genes when applied to gene expression data.

4 Results and Discussion

In this study, we used expression data from a study by Chu et al. [6]. 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. [18].

A typical gene network obtained from this yeast dataset is shown in Fig. 2, when applying the basic phenomic algorithm. We validated the results by performing 10-fold leave-one-out-crossover validation (LOOCV). We made ten runs of the algorithm and compared the gene networks from of each run taken separately against the consensus gene networks of the other nine runs.

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. We formally define these metrics in Eqn. (3) and Eqn. (4).

$$SF = \frac{1}{n}\sum_{i=1}^{n}\frac{CE_i}{E_i}. \tag{3}$$

$$AF = 1 - \frac{1}{n}\sum_{i=1}^{n}\frac{IE_i}{E_i}. \tag{4}$$

Where CE_i is the number of correctly identified edges in the ith comparison, IE_i is the number of incorrectly identified edges in the ith 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 ith consensus network.

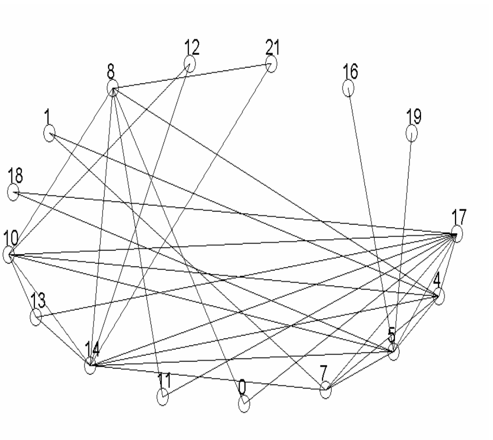

Fig. 2. A typical gene network generated by the basic phenomic algorithm, when D = 0.2

We compared our algorithm to two other algorithms that use evolutionary methods to infer gene networks from microarray data. We provide brief descriptions of the two algorithms here. Interested readers are referred to the original papers. In the first algorithm by Akutsu et al. [10], linear programming (LP) is used to solve the set of differential equations derived from a S-system [12] model of the gene network. In the second algorithm by Noman and Iba [13], trigonometric differential evolution (TDE) is used to derive the set of S-system parameters that best explain the observed microarray data.

Table 1. Validation results obtained by 10-fold LOOCV using Yeast sporulation dataset

Validation Metric	LP based approach [10]	TDE based approach [13]	Phenomic approach
Stability Factor, SF	0.90	0.91	0.94
Accuracy Factor, AF	0.82	0.86	0.89

The results of the validation tests are given in Table 1. As seen, the algorithm performs well in terms of stability, as well as accuracy. Hence this algorithm could be a viable alternative method for determination of gene networks.

5 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.

Currently we are working on applying these algorithms to other datasets in order to study their effectiveness as optimization tools.

References

1. Schulze, A., Downward, J.: Navigating gene expression using microarrays - a technology review. Nature Cell Biology 3, E190–E195 (2001)
2. Soinov, L.A., Krestyaninova, M.A., Brazma, A.: Towards reconstruction of gene networks from expression data by supervised learning. Genome Biology 4(1), R6 (2003)
3. D'haeseleer, P., Liang, S., Somogyi, R.: Gene expression analysis and genetic network modelling: Tutorial. In: Pacific Symposium on Biocomputing, PSB 1999 (1999)
4. D'Souza, R.G.L., Chandra Sekaran, K., Kandasamy, A.: A phenomic algorithm for reconstruction of gene networks. In: CICI 2007. Venice, WASET, pp. 53–58 (2007)
5. Spieth, C., Streichert, F., Speer, N., Zell, A.: Optimizing topology and parameters of gene regulatory network models from time series experiments. In: Kurumatani, K., Chen, S.-H., Ohuchi, A. (eds.) IJCAI-WS 2003 and MAMUS 2003. LNCS (LNAI), vol. 3012, pp. 461–470. Springer, Heidelberg (2004)
6. Chu, S., DeRisi, J., Eisen, M., et al.: The transcriptional program of sporulation in budding yeast. Science 282, 699–705 (1998)
7. Somogyi, R., Fuhrman, S., Askenazi, M., Wuensche, A.: The gene expression matrix: towards the extraction of genetic network architectures. In: Proc. of Second World Cong. of Nonlinear Analysts (WCNA 1996), vol. 30(3), pp. 1815–1824 (1997)
8. Liang, S., Fuhrman, S., Somogyi, R.: REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. In: PSB 1998, vol. 3, pp. 18–29 (1998)
9. Akutsu, T., Miyano, S., Kuhara, S.: Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. In: Pacific Symp. on Biocomputing, vol. 4, pp. 17–28 (1999)
10. Akutsu, T., Miyano, S., Kuhara, S.: Algorithms for inferring qualitative models of biological networks. In: Pacific Symp. on Biocomputing (2000)
11. D'haeseleer, P., Liang, S., Somogyi, R.: Genetic network inference: from co-expression clustering to reverse engineering. Bioinformatics 16(8), 707–726 (2000)
12. Savageau, M.A.: Power-law formalism: a canonical nonlinear approach to modelling and analysis. In: Proc. of the World Congress of Nonlinear Analysts 1992, pp. 3323–3334 (1995)
13. Noman, N., Iba, H.: Reverse engineering genetic networks using evolutionary computation. Genome Informatics 16(2), 205–214 (2005)
14. Lubovac, Z., Olsson, B.: Towards reverse engineering of genetic regulatory networks, Technical Report No. HS-IDA-TR-03-003, University of Skovde, Sweden (2003)
15. Kampis, G.: A Causal Model of Evolution. In: Proc. of 4th Asia-Pacific Conf. on Simulated Evolution and Learning (SEAL 2002), pp. 836–840 (2002)
16. Dawkins, R.: The blind watchmaker. Penguin Books (1988)
17. Stekel, D.: Microarray bioinformatics. Cambridge University Press, Cambridge (2003)
18. Kupiec, M., Ayers, B., Esposito, R.E., Mitchell, A.P.: The molecular and cellular biology of the yeast Saccharomyces. Cold Spring Harbour, 889–1036 (1997)