The Third International Symposium on Optimization and Systems Biology (OSB'09)
Zhangjiajie, China, September 20–22, 2009


# A Parallel Algorithm of Constructing Gene Regulatory Networks

Mei Xiao[1,2,*] Luwen Zhang[1,2] Bing He[1] Jiang Xie[1,2] Wu Zhang[1,2]

[1] School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China
[2] Institute of Systems Biology, Shanghai University, Shanghai 200072, China

**Abstract** One of the grand challenges in post-genomic era should be construction of gene regulatory networks of organisms. As gene expression data increase exponentially, high performance computing is used to guide or even partly replace the biological experiments because traditional biological experiments to study gene regulation networks are very difficult. Gene regulatory networks obtained by parallel stepwise regression method are given in this paper. Numerical results show that the method has good parallel performance.

**Keywords** gene regulatory networks, parallel computing, stepwise regression, parallel performance

## 1 Introduction

Research of systems biology includes traditional molecular biology and computational analysis. One of the important applications is to construct gene regulatory networks(GRNs), which refers to estimate the potential regulatory relationship between the associated genes, proteins or other small molecules (regulatory network nodes), and eventually generate interaction networks, from a group of known gene expression data.

Along with the DNA sequencing of the species, scientists have begun to study the function of genomics. It has been found that each gene has relationships with other 4 to 10 genes on average, and almost all the cell activities are under the control of gene networks [1]. Therefore, to understand the nature of cellular function, it is necessary to study the behavior of genes in a holistic rather than in an individual manner [2].So trying to discover relationships among genes becomes the first step for biomedical researchers [3][4]. On the one hand, as the complexity of gene expression of biological tissue, and the technology of identify the expression of microarray is constantly updated, the scale of the gene expression is increased exponentially [5]. Although some GRNs have been constructed, they are far from our demand [6]; On the other hand, computing power grows significantly. Parallel computing algorithms are being developed for constructing the networks of genetic interactions. It is expectable that the algorithms will make up for the lack of biological experiments.

* Email: xiaomei319@shu.edu.cn

# 2 The algorithm of constructing GRNs

## 2.1 Choose the Model of GRNs

Now there have been a number of models to construct the GRNs, including Boolean models (Kauffman, 1969) [2], Bayesian networks (Murphy and Mian, 1999) [7], linear or nonlinear differential equations (Mestl et al., 1995) [8] [9]. In general, differential equations can accurately predict GRNs. Because linear differential equation model is simple and easy to understand, it is widely used for constructing GRNs.

Gene expression data are generally nonlinear systems. They can be approximated by a linear system describing the rate of accumulation one gene resulting from a transcriptional perturbation [11]:

$$\frac{dx}{dt} = f(x,u) \tag{1}$$

where $\boldsymbol{x}$ is a vector of gene expression data, $\boldsymbol{u}$ is the transcriptional perturbation to $\boldsymbol{x}$, and $dx/dt$ is the rate of accumulation $\boldsymbol{x}$. (1) can be extended for every gene and every gene sample which is obtained by different time:

$$\frac{dx_{il}}{dt} = \sum_{j=1}^{N} a_{ij}x_{jl} + u_{il}, \quad i = 1 \ldots N, \;\; l = 1 \ldots M \tag{2}$$

where $x_{ii}$ is the data of gene $i$ at time $l$, $a_{ij}$ is the coefficient between gene $i$ and $j$. For all $N$ genes and all $M$ times, (2) can be rewritten in more compact form using matrix notation:

$$\frac{dX}{dt} = AX + U \tag{3}$$

where $\boldsymbol{X}$, $\boldsymbol{U}$ are $N \times M$ matrixes. Near a steady-state point (e.g., when gene expression does not change substantially over time) [10], $dX/dt$ is a constant that can be assumed to be 0 and (3) can be transferred to (4):

$$AX = -U \tag{4}$$

where matrix $\boldsymbol{A}$ refers to the relationships between genes. There are $N$ genes and each one has $M$ observation samples, in order to get the relationship between them, each gene will be perturbed respectively.

According to the robust biological mechanism, artificial expression data related topology to the dynamic stability is constructed. Artificial gene expression data including $N$=1000 genes and $M$=1000 observation samples are generated from simulated networks. The expression level of genes is changed by perturbing each gene respectively. According to the sets of expression data, regulatory relationships between genes can be obtained by stepwise regression. Firstly, when the gene $i$ is perturbed, the residual sum of squares (RSS) of the genes one by one is calculated and the genes which has statistically significant relationship with gene $i$ will be chosen (determined by *F-test*). Secondly, when the second regulated gene is selected, the first selected one will be test again. If it does not satisfy the condition, it will be deleted. The above two steps are repeated until no genes can be selected and no ones can be deleted.

### 2.2 Parallel algorithm of constructing GRNs

The computational complexity of the sequential algorithm achieves $N^3$. Table.1 presents the elapsed time of each part in the sequential algorithm. The time proportion of constructing GRNs is 98.69%. And the larger scale of gene expression data, the greater proportion of this part. In order to reduce the computational complexity, the sequential process of constructing GRNs can be paralleled.

Table 1: The elapsed time of each part of the sequential algorithm

| Each part | Run time | Proportion |
|---|---|---|
| Initialize data | 84.48 | 1.3% |
| Reading files | 0.65 | 0.01% |
| Constructing GRNs | 6413.52 | 98.69% |
| Total | 6498.65 | 100% |

In order to parallel the part of constructing GRNs, $N_p$ processors, which are numbered from 0 to $N_p$ are used. The gene expression data and the perturbation data are initialized on processor 0, and then are broadcasted to all the other processors. Each processor completes different jobs which are gathered to processor 0 in the end. And the steps of the parallel algorithm are:

1. Initialize gene expression data and the perturbation data, and broadcast the data to all the other processors;
2. Determine the perturbation $i$ runs on processor ($i$ mod $N_p$);
3. Obtain the relationship between gene $i$ and others by stepwise regression
4. Go to step 2 until all genes are perturbed;
5. Gather the results on each processor, and then the GRN is obtained.

## 3 Performance Analysis

The parallel algorithm of constructing GRNs is based on the platform of MPI (message passing interface) [12]. And the parallel computing is performed on IBM workstation cluster. The cluster has 16.nodes, each of which has two Intel Xeon 5150 dual-core processors (Processor Speed 2.66GHz), 73G hard disk capacity, 4G memory, and 4M cache which is connected by Infiniband [13]. The computing power is more than 450 billion times per second.

### 3.1 Parallel Speedup

Speedup which is an important performance metric to parallel computing refers to how much a parallel algorithm is faster than a corresponding sequential algorithm.It is defined by the following formula:

$$S_p = T_s/T_p \quad (5)$$

where $p$ is the number of processors, $T_s$ is the execution time of the sequential algorithm, and $T_p$ is the execution time of the parallel algorithm with $p$ processors.

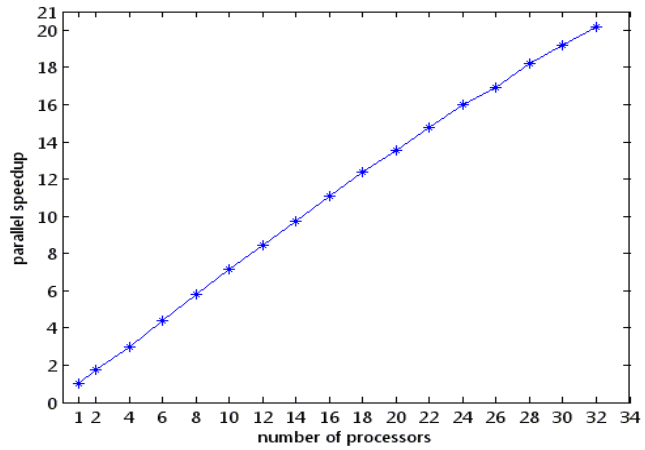


Figure 1: Parallel speedup

In this study, the test experimental data has $N$=1000 genes. And the number of processors is from 1 to 32. Figure.1 describes the speedup form a series of experiments. It shows that the parallel algorithm of constructing GRNs perform well on scalability.

## 3.2 Parallel Efficiency

Efficiency measures how much the available processing power is being used. The simplest evaluation method is to calculate the speedup per processor [14].

$$E_p = S_p/p = T_S/(pT_p) \quad (6)$$

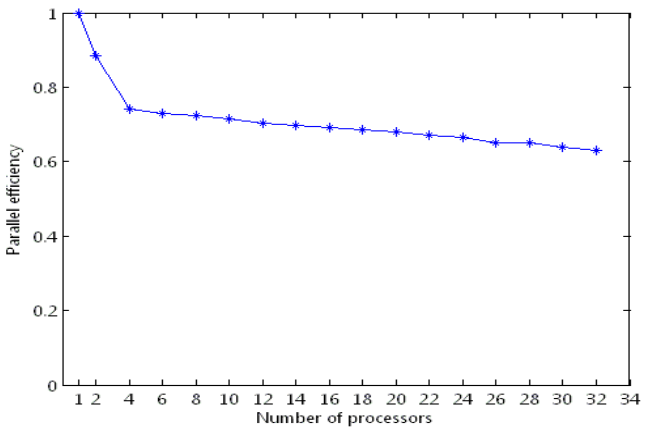


Figure 2: Parallel efficiency

Figure.2 illustrates the parallel efficiency of the algorithm which scale is $N$=1000 genes. When the number of processors increases from 4 to 32, the parallel efficient has little change, which says the additional expenses of parallel computing (communications, etc.) have not much significant increase when the number of processors increases.

# 4 Conclusion

Based on the gene expression data, a parallel stepwise regression algorithm to construct GRNs have been proposed in this paper. Assist with high performance computing, the time complexity of constructing lager scale GRNs has been significantly reduced. This algorithm can be also extended to other bioinformatics problems including protein-protein interactions detection.

# References

[1] Kanehisa M. Post-Genome Informatics. Oxford University Press, pages 1-53, 2001
[2] Ilya Shmulevich, Edward R. Dougherty. Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. Bioinformatics, 18(2):261-274, 2002
[3] Winston Patrick Kuo, Eun-Young Kim. A primer on gene expression and microarrays for machine. Journal of Biomedical Informatics, 37:293-303, 2004
[4] Hongyi Zhang, Jiexin Pu. Construction of Gene Regulatory Networks based on Gene Ontology and Multi-variable Regression. International Conference on Mechatronics and utomation, June 25-28, 2006
[5] LANG Xian-Yu, LU Zhong-Hua. A Parallel Clustering Algorithm of Gene Expression Patterns. Chinese Journal of Computers, 30(2):311-316, 2007
[6] Gong-Hong Wei, De-Pei Liu and Chih-Chuan Liang. Charting gene regulatory networks- strategies, challenges and perspectives. Biochem J, 381(1):1-12, 2004
[7] E.J. Moler, D.C. Radisky, I.S. Mian. Integrating naive Bayes models and external knowledge to examine copper and iron homeostasis in S. cerevisiae, Physiol. Genomics, 4: 127-135, 2000
[8] T. Mestl, E. Plahte, S.W. Omholt. A mathematical framework for describing and analysing gene regulatory networks. J. Theory. Biol. 176(2): 291-300, 1995
[9] Shuhei Kimura, Kaori Ide and Aiko Kashihara. Inference of S-system models of genetic networks using a cooperative coevolutionary algorithm. Bioinformatics, 21(7):1154-1163, 2005
[10] Fengli R, Jinde C, Asymptotic and robust stability of genetic regulatory networks with time-varying delays. Nerocomputing, 71:834-842, 2008
[11] M.K. Yeung, J. Tegner and J.J. Collins, Reverse engineering gene networks using singular value decomposition and robust regression. PNAS 99:6163–6168, 2002.
[12] Du ZhiHui. Parallel Programming Technology for High Performance Computing: Parallel Programs Design with MPI. Tsinghua University Publication, pages 13-15, 2001
[13] Noronha, R., Panda, D.K. Designing high performance DSM systems using InfiniBand features. International Symposium on Cluster Computing and the Grid, Pages: 467-474, April 2004
[14] http://www.shodor.org/refdesk/Resources/Tutorials/Speedup/