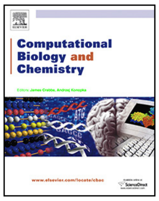
Contents lists available at ScienceDirect

Computational Biology and Chemistry

journal homepage: www.elsevier.com/locate/compbiolchem



Using gene expression programming to infer gene regulatory networks from time-series data

Yongqing Zhang[a,*], Yifei Pu[a], Haisen Zhang[b], Yabo Su[a], Lifang Zhang[c], Jiliu Zhou[a]

[a] College of Computer Science, Sichuan University, Chengdu 610065, PR China
[b] College of Mathematic, Sichuan University, Chengdu 610065, PR China
[c] College of Chemistry, Sichuan University, Chengdu 610064, PR China

ARTICLE INFO

Article history:
Received 4 February 2013
Received in revised form
19 September 2013
Accepted 21 September 2013

Keywords:
Gene regulatory networks
Ordinary differential equation
Gene expression programming
Least mean square

ABSTRACT

Gene regulatory networks inference is currently a topic under heavy research in the systems biology field. In this paper, gene regulatory networks are inferred via evolutionary model based on time-series microarray data. A non-linear differential equation model is adopted. Gene expression programming (GEP) is applied to identify the structure of the model and least mean square (LMS) is used to optimize the parameters in ordinary differential equations (ODEs). The proposed work has been first verified by synthetic data with noise-free and noisy time-series data, respectively, and then its effectiveness is confirmed by three real time-series expression datasets. Finally, a gene regulatory network was constructed with 12 Yeast genes. Experimental results demonstrate that our model can improve the prediction accuracy of microarray time-series data effectively.

1. Introduction

The increasing availability of high-throughput measurements of transcripts has presented a golden opportunity to infer gene regulatory networks. Measuring the levels of gene expression in different conditions is useful in medical diagnosis, treatment, and drug design (Huang et al., 2010; Sun and Hurley, 2009). Many gene expression experiments produce time-series data with only a few time points owing to the high measurement costs. Accurate prediction of the behavior of gene regulatory networks (GRNs) will also speed up biotechnological projects; as such predictions are quicker and cheaper than lab experiments. Therefore, it is highly desired to infer the model of gene regulatory networks using gene expression time-series data. How to predict gene regulatory networks has become an important research area in bioinformatics.

Recently, many dynamic modeling of gene regulatory networks from time-series data has received more and more research interest (De Jong, 2002; Karlebach and Shamir, 2008), such as Boolean network (Akutsu et al., 1999; Bornholdt, 2008), dynamic Bayesian networks (Ghahramani, 1998; Murphy and Mian, 1999; Liu et al., 2006), neural networks (Lee and Yang, 2008), differential equations (De Jong, 2002; Chen et al., 1999; De Hoon et al., 2002; D'haeseleer et al., 1999), state-space model (Wu et al., 2004), stochastic model (Wang et al., 2008, 2010), and so no. A recent review to infer gene regulatory networks based on data integration in dynamical models can be seen in reference Hecker et al. (2009). The system of ordinary differential equations (ODEs) is a powerful and flexible model to describe complex relations, so many methods are proposed to infer genetic regulatory systems using ODEs. For example, Li et al. (2011) have proposed a new hybrid algorithm integrating ordinary differential equation models with local dynamic Bayesian network to infer gene regulatory network. Vilela et al. (2009) identified neutral biochemical network models from time-series data, combining Monte Carlo to optimize the parameters. Zhou et al. (2012) reconstructed GRN from time-series microarray data using stepwise multiple linear regression. Yang et al. (2012) proposed flexible neural tree model which is used for gene regulatory network reconstruction and time-series prediction from gene expression profiling. Unfortunately, most results reported on ODEs have been focused on fix structure of equations which describe the gene regulatory networks and the only one goal was to optimize parameters and coefficients. So it is the motivation of this paper to develop a system biology approach to determine the suitable form of equations which describe the network and to infer reverse engineer gene regulatory network from time-series data with higher accuracy and better scalability.

In our study, we cope with an arbitrary form in the right-hand side of the ODEs models. In order to identify the models, gene expression programming (GEP) is utilized to evolve the right-hand side of the ODEs from the observed time-series gene expression dataset. GEP is a new evolutionary algorithm which has good

* Corresponding author. +86 15208345900.
E-mail address: zhangyongqingscu@hotmail.com (Y. Zhang).

1476-9271/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compbiolchem.2013.09.004

performance to solve time-series prediction problem (Zuo et al., 2004). To decrease the complexity of the genetic network inference problem, the partitioning (Bongard and Lipson, 2007) is used in the process of identification of structure of system. Each ODE can be inferred separately and the research space reduces rapidly. In this paper, two synthetic time-series datasets obtained by E-cell system (Tomita et al., 1999) and three other real microarray datasets from Worm gene expression time-series dataset (Yeung et al., 2001), Human cell time-series dataset (Whitfield et al., 2002) and Yeast time-series dataset (Woolf and Wang, 2000; Schneider and Guarente, 1991) are used to test our method. Finally, a gene regulatory network was constructed with Yeast time-series dataset. Experiment results show that our method is capable of improving the prediction accuracy of microarray time-series data effectively.

2. Methods

2.1. Modeling gene regulation with ordinary differential equation

Ordinary differential equations (ODEs) is one of the most popular tools to model complex systems, which basic relationships are known between the system components. In the inverse problem, we often use ODEs to analysis the model from the observed time-series data.

To allow the flexibility of the model, we consider the following general form:

$$\frac{dx_i}{dt} = f_i(x_1, x_2, \ldots, x_n) \quad (i = 1, 2, 3, \ldots, n), \tag{1}$$

where x_i the state is variable and n is the number of the observed data time points.

In order to identify the system, GEP is used to evolve the ODEs from the observed time-series data. Although GEP could effectively find the suitable structures, it is sometimes difficult to optimize the parameters. So least mean square (LMS) (Ando et al., 2002) is employed to improve the effect of GEP.

2.2. GEP algorithm

2.2.1. Brief introduction of GEP

The GEP algorithm is described in detail in Ferreira (2001, 2006a). We give a brief introduction below.

The first step, how to define a problem by GEP, the encoding of the candidate solution and the definition of the fitness function. It is a most important and most difficult point in GEP. Each problem has specific the encoding and the fitness function. Adequate choices are the key factor for the success of the algorithm.

The next step, utilize the GEP algorithm itself including several stages. A basic flowchart of the algorithm is shown in Fig. 1.

The process starts with the random creation of an initial population of chromosomes. Then each chromosome is translated into an expression tree and each individual is evaluated by fitness function. The individuals are selected in the light of fitness function to reproduce with modification. If the termination criterion is not met, some of the chromosomes are selected and reproduced, resulting in offspring. The new chromosomes will replace the old ones producing a new generation. The process continues until the termination criterion is met or a certain number of iterations run.

Since GEP offers great potential to solve complex modeling and optimization problems, it has been used in many fields such as data and text mining (Zhou et al., 2003; Karakasis and Stafylopatis, 2006), classification (Duan et al., 2006; Karakasis and Stafylopatis, 2008; Duan et al., 2009), time-series analysis (Zuo et al., 2004), neural network design (Ferreira, 2006b) and various engineering application (Si et al., 2011).

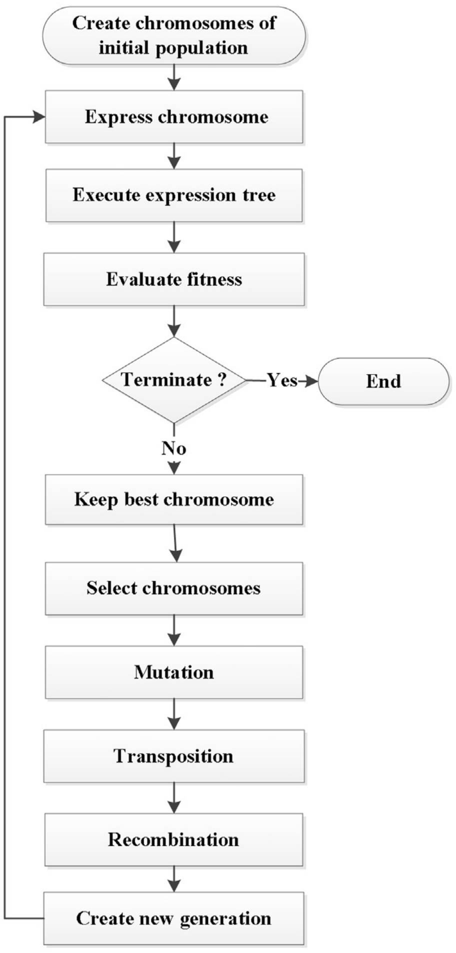


Fig. 1. The flowchart of basic GEP algorithm.

GEP and genetic programming (GP) are evolutionary algorithms. The fundamental difference between them resides in the nature of the individuals: in GEP the individuals are encoded as linear strings of fixed length which are afterwards expressed as nonlinear entities of different sizes and shapes; and in GP the individuals are nonlinear entities of different sizes and shapes. So that GEP provides new and efficient ways to program evolutionary computation (Ferreira, 2001).

2.2.2. Chromosome encoding

In GEP, the basic unit of an individual is called gene. The most distinctive feature of GEP is that each gene has access to a genotype and a corresponding phenotype: the genotype is a symbolic string of some fixed length, and the phenotype is the tree structure for the expression coded by that symbolic string. The symbolic string of a gene is composed of a head and a tail, both having fixed lengths. The head contains both function and term symbols, while the tail contains only term symbols. The function symbol represents a mathematical operator, such as addition, subtraction, multiplication, division, log and sin. The term symbol represents an attribute value. For each problem, the length of the head ($|head|$) and the length of the tail ($|tail|$) satisfy $|tail| = |head| \times (n-1) + 1$, where n is the maximum arity of functions under consideration. The head length ($|head|$) is determined by the user as the maximum number of functions in a gene; the length of a gene ($|head| + |tail|$) remains unchanged in the middle of an execution of a given GEP algorithm. In GEP, an individual problem may involve one or more genes to encode a candidate solution. For multiple genes in an individual