Expert Systems with Applications 30 (2006) 50–58

Expert Systems with Applications

www.elsevier.com/locate/eswa

Applying data mining to learn system dynamics in a biological model

Bingchiang Jeng [a,*], Jian-xun Chen [a], Ting-peng Liang [b]

[a] *Department of Information Management, National Sun Yat-sen University, 70 Lien-hai Road, Kaohsiung City 804, Taiwan, ROC*
[b] *Department of Information Management, National Sun Yat-sen University, Taiwan*

Abstract

Data mining consists of a set of powerful methods that have been successfully applied to many different application domains, including business, engineering, and bioinformatics. In this paper, we propose an innovative approach that uses genetic algorithms to mine a set of temporal behavior data output by a biological system in order to determine the kinetic parameters of the system. Analyzing the behavior of a biological network is a complicated task. In our approach, the machine learning method is integrated with the framework of system dynamics so that its findings are expressed in a form of system dynamics model. An application of the method to the cell division cycle model has shown that the method can discover approximate parametric values of the system and reproduce the input behavior.

Keywords: Systems biology; System dynamics; Neural network; Genetic algorithm

1. Introduction

Traditionally, researchers adopt the reductionism to study biological phenomena, i.e. analyzing a system by breaking it into constituents repeatedly until they can be observed directly (Gallagher & Appenzeller, 1999). In order to find the function and role of a component, the researcher has to repeatedly conduct experiments with different system parameters or components. Although this approach works fine in most situations, it often encounters difficulties when we intend to examine the interaction effects within a system or when the system is complicated. It is also well-known that the net behavior of a biological system is usually not the sum of its components' behavior (Csete & Doyle, 2002) because of the existence of the so-called 'emergent property' (Bhalla & Lyengar, 1999; Gardner & Collins, 2000; Yi, Huang, Simon, & Doyle, 2000).

Recently, a system view of biology called *systems biology* has been proposed (Chong & Ray, 2002; Davidson, Rast, Oliveri, Ransick, Calestani and Yuh, 2000; Kitano, 2002a), which aims to the development of a system-level understanding of biological systems (Kitano, 2002a). In other words, one wants to understand not only the molecules but also the cause–effect relationships linking the behavior of molecules as well as the characteristics and functions of a system. Although artificial intelligence has increasingly been used in analyzing biological data for years, this is certainly a more difficult case and needs innovative methods.

We propose an approach that integrates system dynamics and data mining methods to induce the dynamic behavior of a biological system in this paper. System dynamics is a discipline that studies the dynamic behavior of social systems (Forrester, 1961). In particular, it has an advantage in modeling the information-feedback characteristics to see how system structure, amplification (in policies), and time delays (in decisions and actions) may interact to influence the behavior of an organization. Since a social system is a combination of a number of simple entities (or agents) that operate in an environment to generate complex behavior patterns as a collective, it may be suitable for analyzing the information-feedback loops and complicated interactions within a biological system (Becskei & Serrano, 2000; Gardner & Collins, 2000).

A challenge for applying system dynamics to the analysis of biological data is that the base model for analysis is often constructed by human experts who have expertise in the application domain and are able to draw a flow diagram by observing the operation of target system to represent the causal relationships among system entities (variables) (Coyle, 1977; Lyneis & Pugh, 1996; Starr, 1980). This, however, is not the case in biological analysis because in most cases the biological systems under study act like black-boxes and only their input and output behavior can be observed over time. Thus, direct

[*] Corresponding author. Tel.: +886 7 525 2000x4719; fax: +886 7 525 4799.

E-mail addresses: jeng@mail.nsysu.edu.tw (B. Jeng), jamesc02@ms9.hinet.net (J.-x. Chen), tpliang@mail.nsysu.edu.tw (T.-p. Liang).

0957-4174/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2005.09.068

application of system dynamics to the construction of biological models is very difficult, if not impossible. We need a mechanism to bridge the gap.

A possible way to deal with the problem is to use data mining techniques to analyze the observed behavior data to discover the hidden relationships and/or rules behind the system dynamics. In order to do this, a data mining method needs to be augmented; it has to have a conceptual framework beforehand so that the findings from data will be express in the form of a system model. In this paper, we will use a combination of genetic algorithms and artificial neural networks to implement the idea. The artificial network is designed to emulate a system dynamics model and then encode into a genetic form for learning. The proposed approach is applied to experiment on the synthetic cell division cycle model (CDC6, hereafter) created by Tyson (1991). The behavior data generated by CDC6 model is given as an input to the developed method to learn the model's kinetic parameters. The results are then compared with the original data to evaluate the effectiveness of the approach.

The remainder of the paper is organized as follows. Section 2 is a brief review of related literature. Section 3 describes the proposed approach for mining behavior relationships from a set of observed biological data. Section 4 illustrates the result when the approach is applied to the CDC6 model. Section 5 concludes the paper.

2. Review of literature

2.1. Systems biology

One thing that realizes systems biology is the high-throughput measurement devices for DNA, RNA, and proteins. The ascendancy of these high-throughput devices in the past decade has permanently changed the biological landscape of genomics, proteomics, and metabolomics studies (Henry & Washington, 2003). Rising as a new star under this background, systems biology aims at a system-level understanding of biological systems (Kitano, 2002a). Unlike molecular biology, which focuses on the study of molecules such as nucleotide acids or protein sequences, systems biology focuses on dynamics of systems, which cannot be described merely by enumerating the molecular components of the system (Henry & Washington, 2003). Another misleading concept in molecular biology is to believe that only system structure, e.g. network topologies, is important without paying attention to the diversities and functionalities of system components. Both the structure and the components are indispensable in forming the symbiotic state of a system as a whole (Henry & Washington, 2003).

Research of systems biology focuses on four key properties of a biological system: (1) system structure, (2) system dynamics, (3) control method, and (4) design method (Kitano, 2002a). System structures include networks of gene (or protein) interactions, biochemical pathways, and the mechanisms to modulate the physical properties of intra- and multicellular structures.

System dynamics concerns a system's dynamics behavior over time under various conditions. There are currently several methods to analyze the dynamics of a system from different perspectives: metabolic analysis, sensitivity analysis, phase portrait analysis, bifurcation analysis, and analysis by identifying essential mechanisms of a specific behavior (Kitano, 2002a). The typical one is bifurcation analysis, which traces the time-varying state changes of the system with a multi-dimensional plot(s) where each dimension represents a particular biochemical concentration involved in the interaction.

Control methods are related to ways to control the states of a biological system, e.g. the methods to transform cells from malfunctioning into healthy ones. It is an application that makes the knowledge obtained from system structure and system dynamics. Design methods move even further. It intends to establish technologies to design biological systems in a way we wish. An example is the attempt to actually design and grow organs from the patient's own tissue (Kitano, 2002a).

Research of systems biology is emerging as an area of potential (Chong & Ray, 2002; Kitano, 2004; Nobel, 2002). For example, Kitano (2004) recently puts forth a new cancer treatment proposal from the viewpoint of systems biology. Cancer disease has a very robust system, which causes many medical treatments fail to control the growth of cancer cells. Thus, it is hard to defeat a cancer by simply alternating some constituent of the system, but it may be possible to put cancer cells into the status of dormancy or apoptosis, if one can change the kinetic parameters (rates) of the system from a system's viewpoint. Recognizing the importance and potential of systems biology, a special issue on this subject has been published in the March 2002 issue of *Science*.

2.2. System dynamics models

As mentioned above, system dynamics concerns a system's dynamics behavior over time under various conditions. Its emergence dates back to the publication *Industrial Dynamics* by Jay W. Forrester early in 1961 (Forrester, 1961). In his book, Forrester defines system dynamics as "the study of the information-feedback characteristics of industrial activity to show how organizational structure, amplification (in policies), and time delays (in decisions and actions) interact to influence the success of the enterprise."

A system dynamics model is one that models the dynamics of such a system. There are some different ways to do this and the most typical one is Forrester's flow diagram (Forrester, 1961). As an example, the elementary inventory control system (see Fig. 1) described in Forrester's book is redrawn here to illustrate the notation and modeling concepts.

For simplicity, assume that the order rate (*OR*) in the model can be either positive or negative, i.e. goods can either be ordered into the inventory or returned back to a supplier. The goal is to maintain the inventory at a fixed level defined by the desired inventory (*DI*) for duration of time. In order to bring the actual inventory toward the desired level, one has to increase the order rate when the inventory falls far below the desired value, and, as it approaches the desired inventory, the order rate should be gradually reduced. If, on the other hand, the inventory becomes

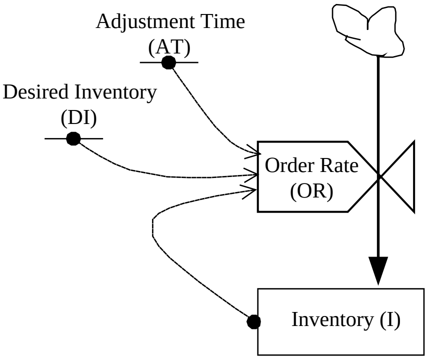

Fig. 1. An inventory model in the flow diagram form.

higher than the desired level, the order rate becomes negative, indicating that goods are being returned to the supplier.

In this diagram, rectangles denote *level* variables, which represent the conditions (or states) of the system at a particular time. Level variables accumulate the results of actions within a system. At the end of each time interval, the value of a level is recalculated, which is determined by its previous value, the rates (or actions) of flows into and off the level, and the length of the time interval.

A *rate* variable is represented by a valve symbol, which denotes a policy statement that describes an action on some level variable in the system. Rate variables determine not the present values of levels but the slopes of them, e.g. the order rate in Fig. 1. The value of a rate itself is dependent on the values of other levels and constants but has nothing to do with its own past value, the time interval length, and other rate variables.

Constants are values that do not change over time during the simulation of a system. They are lines in Fig. 1.

The solid line with an arrow is a *flow*, which represents an amount that is transferred from one-level variable (or boundary) to another in the system. System boundaries are represented by clouds, which are used to define the borders of flows.

The dash lines with an arrow are *wires*, which represent information used from a level or constant to a rate without depleting the source.

In addition to the components described above, the model uses mathematical equations to define the constraints of the system. For instance, the new value of a level may be defined by a level equation in Forrester's format, as in the following:

$$\mathrm{L.K} = \mathrm{L.J} + \mathrm{DT}(\mathrm{RA.JK} - \mathrm{RS.JK})$$

where

L	level
L.K	level L's new value
L.J	level L's old value
DT	the time period between JK
RA	rate on an inbound flow into the level
RA.JK	the delta value increases between time J and K
RS	rate on an outbound flow off the level
RS.JK	the delta value decreases between time J and K

This equation also shows how *rate* values, e.g. RA.JK, affect the level through controlling the amount of value flowing into or out of it. These values in turn are determined by other level values and constants through a rate equation, which in Forrester's form is:

$$\mathrm{R.JK} = f(\text{all levels and constants})$$

Rate equations are in free format except for three prohibitions: (1) a rate equation should not contain the time interval DT; (2) no rate variable is allowed to appear in the right-hand side of a rate equation; and (3) the left-hand side of the equation can only contain the rate variables being defined.

As a matter of fact, this kind of system dynamics is very common in biological world (e.g. Becskei & Serrano, 2000; Bhalla, Ram, & Iyengar, 2002). Becskei and Serrano (2000), for example, described a simple gene circuit, which consists of a regulator and a transcriptional repressor in *Escherichia coli*. The stability of the model is maintained by a negative-feedback loop, which regulates its own production to reduce noises in gene expression. Therefore, it seems that system dynamics modeling is a good approach to represent the kinetics of biological systems.

3. An integrated approach for mining hidden relationships in biological systems

The modeling process starts with a given set of temporal data from a bio-system and ends with a flow diagram that describes the dynamics of the system. In particular, the approach consists of three steps, which (1) represents a bio-system with a framework of system dynamics model, (2) transforms the model into a form that can be analyzes by a data mining technique, (3) executes a learning algorithm to discover the relationships through a genetic evolutionary process. Fig. 2 shows the major modules of the proposed approach.

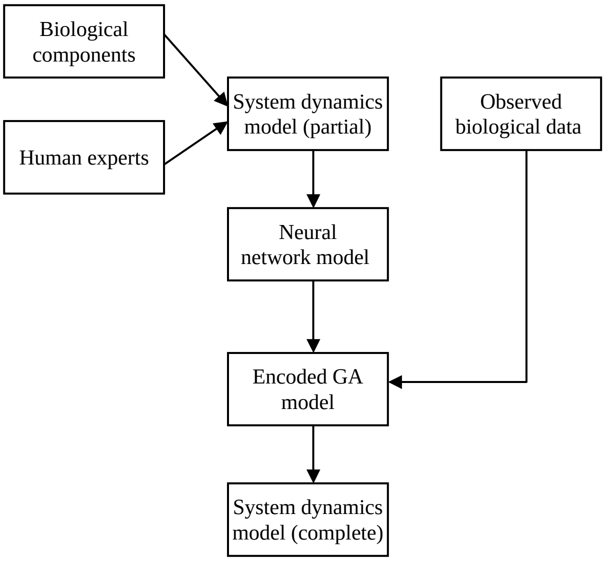

Fig. 2. The process of the approach.

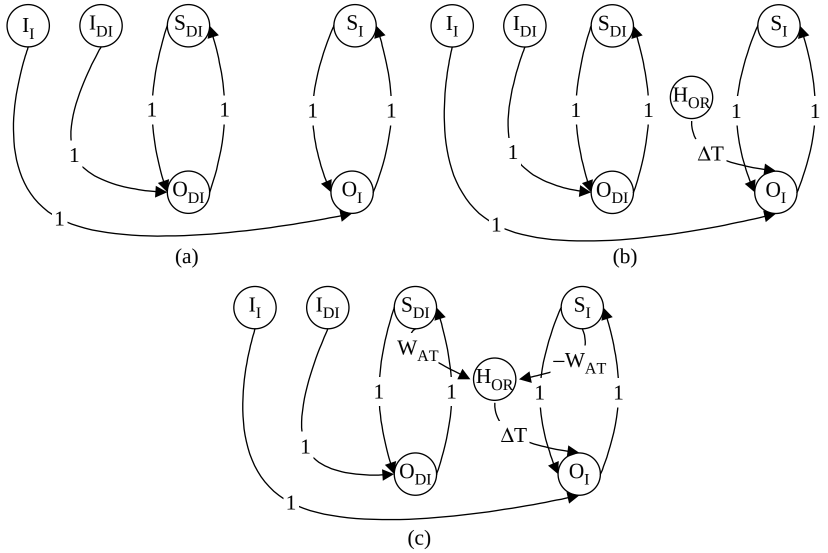

Fig. 3. The inventory model in the partial recurrent neural network form.

3.1. Model representation in neural networks

As we have described above, system dynamics models can provide a good representation of biological systems. However, traditional system dynamics models do not provide induction mechanisms for discovering hidden relationships. Thus, we need to convert a system dynamics model into a form that fits a learning technique, and we find that artificial neural networks are suitable for this task because they two have similar characteristics: system behavior is determined by the whole structure, not any individual elements.

In a previous work, Chen and Jeng (2002) has shown that a specially designed partial recurrent neural network can be made equivalent to a flow diagram for a system dynamics model, and propose an algorithm (FD2PRN) to conduct the transformation. Taking the model in Fig. 1, for example, the flow diagram can be converted into the partially recurrent neural network model shown in Fig. 3(c). The model is composed of three parts: the top left two nodes are input units, which feed data into the network at initialization; the bottom two nodes are output units; and the two nodes on the upper right are state units, which keep the previous values of the output units.

The mappings between the resulting network model and the original system model in Fig. 1 can be described as follows. As illustrated in Fig. 3(a), *level* (inventory I) and *constant* (DI) are corresponded to three units: (*input* I_I, *output* O_I, *state* S_I) and (*input* I_{DI}, *output* O_{DI}, *state* S_{DI}), respectively. Rate (OR) and the flow, as shown in Fig. 3(b), are mapped to a hidden unit H_{OR}, and the link from H_{OR} to O_I, respectively. The other type of constants (e.g. AT) that appear as coefficients in rate equations is converted to the weights of the links from S_I to H_{OR} and from S_{DI} to H_{OR}, respectively, as is shown in Fig. 3(c). The detail mappings of individual components between the two models are listed in Table 1.

3.2. Relationship mining using genetic algorithms

Since data mining discovers knowledge from various types of given data in different problem domains, it includes many different techniques, such as tree induction, clustering, association rules, artificial neural networks and genetic algorithms, etc. (Berry & Linoff, 1997; He et al., 2004; Li & Shue, 2004; Shin & Lee, 2002). In the above, we have shown how to represent our model in the form of neural networks. The next problem is how to use data mining techniques to discover hidden relationships in the model.

A commonly seen approach is to use the gradient descent algorithm to adapt its link weights through back-propagation during the learning process. For large non-linear dynamic biochemical pathways which are known to be frequently ill-conditioned and multi-modal, however, the method usually cannot converge to a good solution (Curran & O'Riordan, 2002; Mendes, 2001). Thus, we use genetic algorithms (Goldberg, 1989; Holland, 1975) to adapt the model. This kind of combination has been seen in Chang, Wang, and Tsai (2005); Versace, Bhatt, Hinds, and Shiffer (2004), and so on.

Table 1
The component mappings between SDM and PRN

Components for SDMs	Components for PRNs
Level variable, constant (not for coefficient)	A triple of input, output, and state units
Rate (or auxiliary) variable	Hidden unit
Wire	Link from a state unit to a hidden unit
Flow	Link from a hidden unit to output unit
Level equation	A weighted sum of the values of hidden and state units connecting to an output unit via links
Rate equation (including constants as coefficients)	Any function of the values of state units connecting to a hidden unit via links
Equation for initial value	Link from an input unit to an output unit
Constant equation	Link from a state unit to an output unit

Genetic algorithms, introduced by John Holland to mimic the mechanisms of natural adaptation (Holland, 1975), are widely adopted evolutionary techniques in science, engineering, and bioinformatics (Fogel & Corne, 2003) for solving combinatorial problems. Unlike other algorithms, it solves problems by 'guess' and 'test', not through an analytical process. Thus, it requires no knowledge of the search space beforehand, which is the case in our problem. Past research has shown that the genetic algorithm and its variations are very successful in various areas and different applications in biology (Fogel & Corne, 2003; Koza, Mydlowec, Lanza, Yu, & Keane, 2003; Ritchie, White, Parker, Hahn, & Moore, 2003). A pseudo code for this algorithm is like the following (Michalewicz, 1994):

```
procedure genetic-algorithm ( )
/* $g$: generation number
  P: population */
begin
  g←0
  initialize P($g$)
  evaluate P($g$)
  while not-termination-condition do
  begin
    g←g+1
    select P($g$) from P($g-1$)
    recombine P($g$)
    evaluate P($g$)
  end
end
```

3.3. Algorithm implementation

3.3.1. Genetic encoding

The first step to use the genetic algorithms is to encode a solution as a chromosome (which is usually represented as a string of symbols). Since we have rephrased a system as a partial recurrent neural network, we can encode the latter instead, and there are existing ways to do it (Gruau & Whitley, 1993; Prusinkiewicz & Lindenmayer, 1992). A direct encoding scheme (Curran & O'Riordan, 2002) is used here that encodes the link weights as genes in a chromosome.

3.3.2. The fitness function

A generated solution during the evolution process is ranked by a fitness function that measures the similarity between the target behavior pattern and the actual one. Since the pattern is a time series of real value points, a simplest function of such a measurement is the reciprocal of the Sum of Squared Errors (SSE). However, in practical experiences, we find that it is better to use the sum of relative square errors instead, when the range of parameter values varies large (e.g. from 0.001 to a large number, e.g. 200). The following equation is our fitness function:

$$\text{fitness} = \frac{1}{\sum_i \sum_t ((y_{it} - \hat{y}_{it})/\hat{y}_{it})^2},$$

where, y_{it} is the desired output; $\hat{y}_{it}$ is the actual output of some system variable y; t represents the tth time point, and i represents the index of a variable.

More complicated functions exist that can measure the similarity/difference of two time series but we found the above one is good enough for the current use.

3.3.3. The evolution process

The last step in genetic algorithms is to set up an evolution process for finding the optimal solution. Parameters to be determined at this stage include population size (i.e. the number of chromosomes) and the number of evolution generations. Trade-offs exist between these two parameters, and the usual way to do it is by trial-and-error. There are ways to select a suitable combination of the two parameters (e.g. Taguchi method), but it is not the focus of this research. Other parameters to be set for executing a genetic algorithm include crossover rate and mutation rate.

Because the genetic algorithm is a kind of stochastic searching processes (Goldberg, 1989; Holland, 1975) with non-deterministic results, it may require a certain number of generations before an expected outcome is obtained.

The result of learning depends on what is known initially as well as the quality of the given data. When applying the approach to a set of biological behavioral data, it shall return with a flow diagram that represents the dynamic behavior of the bio-system.

4. Empirical evaluation

In order to evaluate the feasibility and performance of the proposed approach, an empirical study has been conducted on a biological system known as CDC6. We selected the reasonable comprehensive biological network whose biochemical reactions are known from literature and then generated its temporal behavior by a computer model. The simulated data is later used as an input to our method, and the mining resulting are compared with the original ones for evaluation.

The selected biological system is one that models the cell division cycle as shown in Fig. 4 (modified from Tyson, 1991). A square bracket [ ] in the diagram stands for concentration of a protein with its abbreviation name inside. The division cycle can be illustrated as in the following steps. Assume cyclin is synthesized de novo (step 1) initially. Because this chemical is unstable (step 2), it combines with cdc2-P (step 3) to form 'preMPF'. After heterodimer formation, the cyclin subunit is phosphorylated. Assuming phosphorylation is faster than dimerization. Thus, the cdc2 subunit is dephosphorylated (step 4) at some point to form 'active MPF'. The activation of MPF, in principle, is opposed by a protein kinase (step 5). Since active MPF enhances the catalytic activity of the phosphatase (as indicated by the dashed arrow), the MPF activation is switched on in an autocatalytic fashion. When a sufficient amount of MPF is activated, nuclear division is triggered and the active MPF is destroyed concurrently (step 6) . The destruction breaks the MPF complex and releases phosphorylated cyclin, which is subject to rapid proteolysis

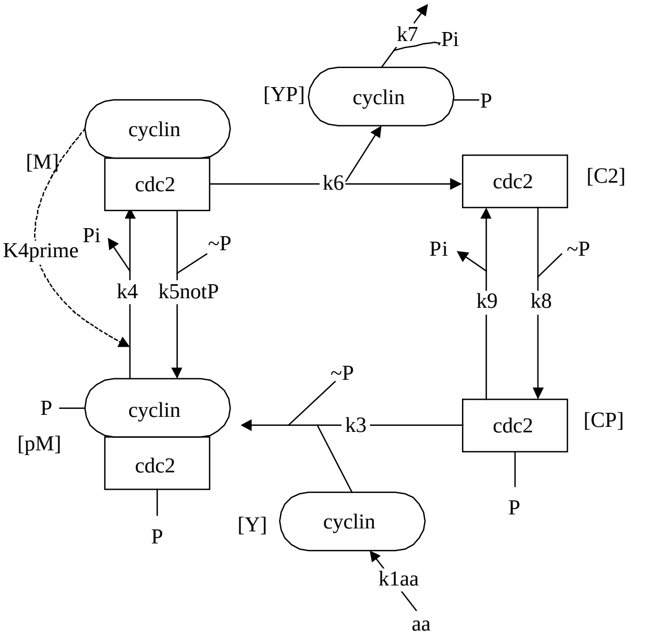

Fig. 4. The relationship between cyclin and cdc2 in the cell cycle. aa, amino acids; ∼P, ATP; Pi, inorganic phosphate.

(step 7). Finally, the cdc2 subunit is phosphorylated again (step 8, possibly reversed by step 9), and the cycle repeats (Tyson, 1991). The implementation of the proposal approach to this biological system is illustrated below.

4.1. Initial model

The above is a typical model that describes how chemical substances interact within a biological system. We convert it from the perspective of system dynamics with the notations of flow diagrams seen before. The present concentration of a protein is determined by its previous value plus the increment/decrement amount in the current time step. So, each protein in Fig. 4 will be represented with a 'level' (represented by a 'box') associated with a 'rate' (represented by a 'valve') in the Fig. 5. A wire from a 'level' to a 'rate' denotes an information-feedback effect from a protein to the 'rate'.

Let us take protein C2 in Fig. 4 as an example. It is associated with two input links: M to C2 and CP to C2, indicating that the current values of M and CP determine the increment of C2 in each time unit. So, two wires from M and CP, respectively, to the 'rate' for protein C2 are connected. In addition, C2 has an output link to CP, which indicates a decrement of C2 in a time unit. So, a wire from 'level' C2 to the 'rate' for CP exists.

One can repeat the procedure for the rest of proteins in Fig. 4 except for the bottom part of the diagram, where there is an input link from aa to Y, indicating a constant increment (k1aa). So, we need to add a 'constant' variable k1aa (represented by diamond) and connect a wire from it to the 'rate' for protein Y. Also, according to literature, the 'rate' of pM to M or the 'rate' of M to pM is related to the kinetic parameter (k4prime) plus the square of the M's concentration. Thus, two 'constant' k4p and M2 are linked to the 'rate' of M and pM, respectively. The finished graph after the conversion is shown in Fig. 5, which is further transformed into a partial recurrent neural network as shown in Fig. 6.

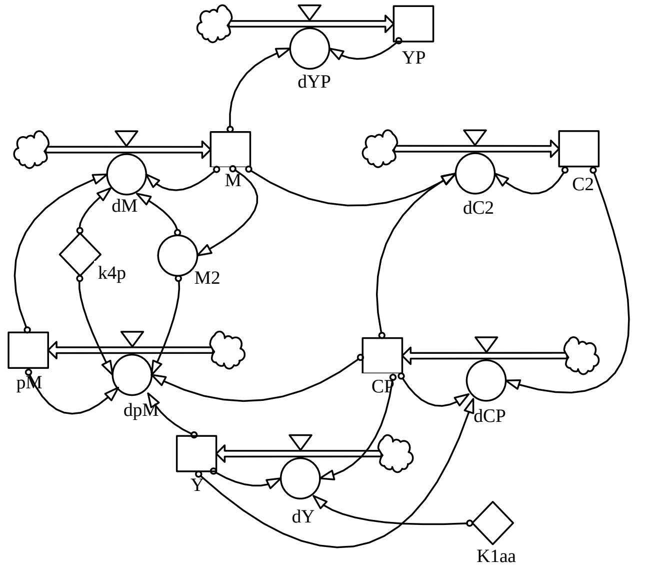

Fig. 5. System dynamic model of CDC6.

4.2. Discovering kinetic parameter values

Note that, although the above diagrams draw the initial models of the system, they are just the skeleton with no parametric information and hence cannot be executed. So, the simulated behavior data generated from the original CDC6 system is used as an input to train the model. We allow the genetic evolutionary process to run for 500 generations and each generation has 500 chromosomes. The initial value of each parameter is guessed randomly in a range of [0,500]. This process is repeated for 40 times and each takes about a half-minute on an Intel Pentium 4, 1.8 GHz computer. The best parameter solution of each process is compared with each other and the best one of them is the final solution, which is shown in Table 2a and b. In these two tables, there also show the initial parameter values and its evolution history. (For reference, the target parametric values are also given in the last row of the tables.)

In Table 2, one can see that the parameter values discovered in the process are quite close although some of them (k1aa, k3, k9) have a little larger difference. These reflect the sensitivity of different parameters with respect to the system behavior changes. However, the behavior of the system is not affected by these minor parameter variations. When we compare the learned behaviors with the original ones in Fig. 7, one can see that they match quite well in all proteins.

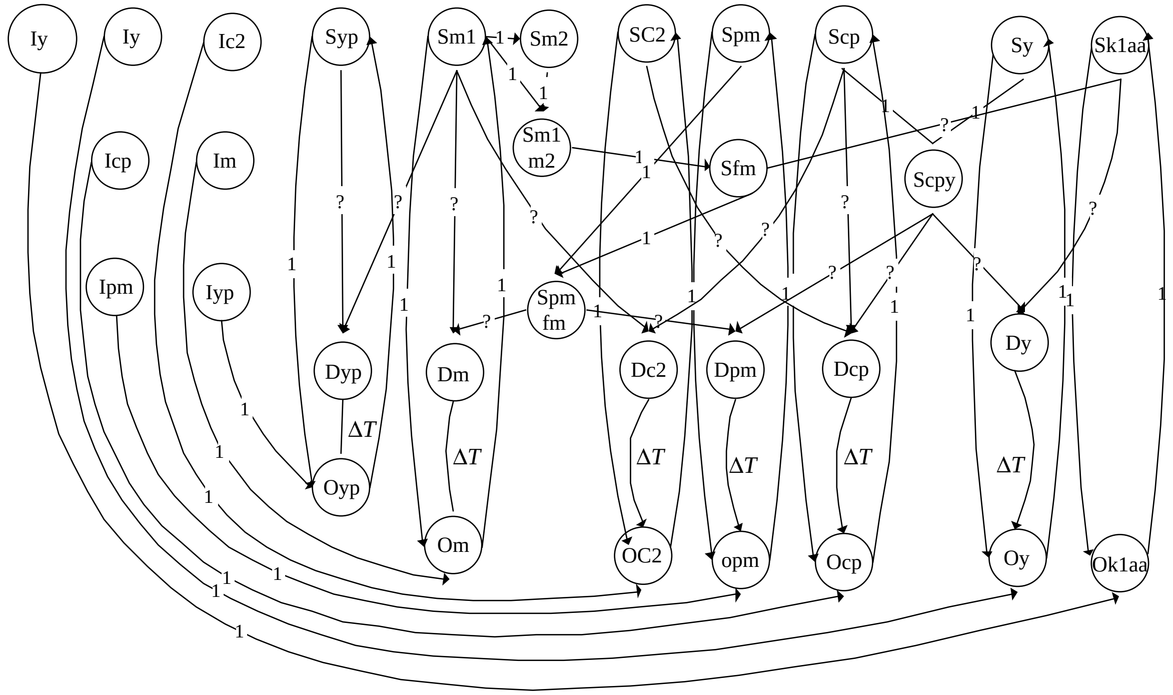

Fig. 6. Partial recurrent neural network of CDC6.

5. Discussion and conclusions

In this paper, we present an innovative approach that represents a biological system with a specially designed artificial neural network and then uses genetic algorithms to modify the link weights of the network to discover the system's kinetic parameters. Although there are tools in some systems biology's websites (e.g. http://sbml.org/index.psp) in which

Table 2
The initial and learning history of the kinetic parameters

G. No.	Para				
	Fitness	k1aa	k3	k4	K4prime
0	84905.441	46.64439	278.14126	169.75511	219.85746
50	61.816760	0.016121	215.24046	133.42039	0.018005
100	61.770046	0.016121	215.13488	190.15931	0.018005
150	25.922052	0.016121	214.83078	185.96394	0.018005
200	25.917173	0.016105	214.83078	182.62815	0.018005
250	13.830711	0.016105	214.80424	182.27636	0.018005
300	13.826469	0.016111	214.80424	181.35047	0.018005
350	0.9420064	0.016111	214.80424	180.56614	0.018005
400	0.9408610	0.016111	214.80424	180.91209	0.018005
450	0.9408466	0.016111	214.80424	180.69822	0.018005
500	0.9406935	0.016111	214.80424	180.69822	0.017998
Target	0	0.015	200	180	0.018

G. No.	Para				
	K5notP	k6	k7	k8notP	K9
0	290.16784	497.05465	0.288098	354.15322	3.742141
50	0.042558	0.866736	0.614125	3.226575	0.006264
100	0.042558	0.983811	0.591082	1.814398	0.006264
150	0.042558	0.975722	0.585800	1.305426	0.002682
200	0.009857	0.990632	0.583559	1.305426	0.002682
250	0.009857	0.990632	0.594423	1.327185	0.001474
300	0.009857	0.990632	0.593645	1.169690	0.001474
350	0.009857	0.992455	0.593645	1.044279	0.000126
400	0.009857	0.990845	0.593964	1.002710	0.000126
450	0.009857	0.990845	0.593964	1.002710	0.000126
500	0.009857	0.991116	0.593964	1.002710	0.000126
Target	0	1	0.6	1	0.0001

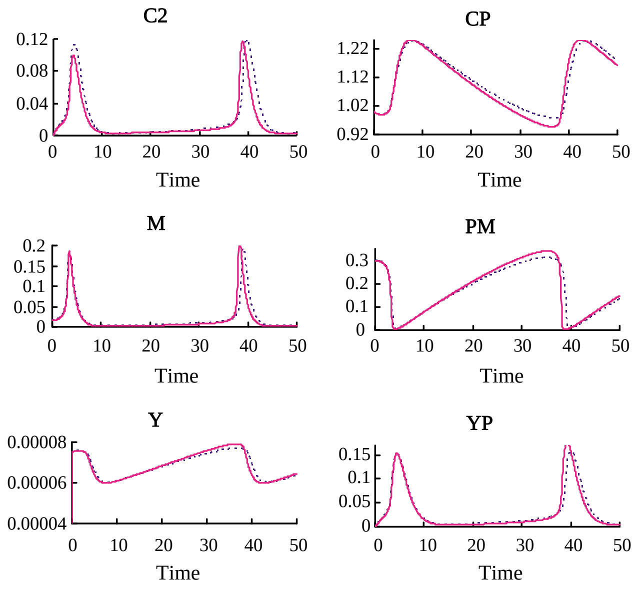

Fig. 7. The comparison of the learned model behavior (dot line) to the real one (solid line).

some of them can perform modeling and simulation, the proposed approach is unique in that it provides a means to bridge two sciences: system dynamics and systems biology so that their knowledge can be shared and transferred for better integration.

Applying data mining to analysis of bio-information is an important area of study. The fast progress of biology development has accumulated a tremendous amount of experimental data, which becomes a big challenge to efficiently extract valuable knowledge hidden behind. Data mining can contribute substantially in this area by generating potential solutions to save the time and effort of a biologist. The example shown in our approach is just an initial step to discover related information from a biological system. The ultimate goal of this line of study can be using data mining techniques to assist model construction and behavior analysis in systems biology.

Although we have shown that the proposed approach is capable of modeling a biological system in systems dynamics to analyze its behavior, there are many issues that need further elaboration or investigation. For instance, a further issue coming after is 'can this method be applied to reveal structure information for a biological model?' This is an area that traditionally can be handled only by human experts and little literature can be found in biology. Koza et al. (2001) used genetic programming to discover the network of chemical reactions from a set of temporal data, but it required thousands of processors to run in parallel for a number of hours. Since our method has been demonstrated to be able to learn kinetic parameters successfully, it is highly possible that our approach can be extended to mine structural information of a network.

Since evolving a network involves adding and removing nodes and links into or off from the network, an extension of this study is to revise the encoding scheme for partial recurrent neural networks. For example, we may use a strategy to encode it indirectly (Curran & O'Riordan, 2002), and describe the network structure by a set of construction instructions (i.e. a script) so that, by modifying them, the network structure changes.

Inferring structural information, however, needs to be more careful because isomorphism may exist among different structures, which makes different models generate the same (or similar) behavior patterns. Over-fitting problems may also occur when we use data mining. This is a concern about whether an automatic method will produce a model that generates 'the right behavior for the wrong reasons', or just tries to 'confirm but not falsify' a hypothesis (Oliva, 2003).

Another issue related to this discussion is the 'robustness' of a biological system, which is currently actively investigated in systems biology (Kitano, 2002b; Morohashi, Winn, Borisuk, Bolouri, Doyle and Kitano, 2002). It is suggested that biochemical networks are conserved across species and are robust to variations in concentrations and kinetic parameters. If this is true, then what data mining discovers may not have reliable biological meanings or at least judgments from domain experts are essential for interpreting and using the resulting models.

Acknowledgements

This research was partially supported by National Science Council of Taiwan, under contracts NSC 92-2213-E-110-037.

References

Becskei, A., & Serrano, L. (2000). Engineering stability in gene networks by autoregulation. *Nature*, *405*(6786), 590–593.

Berry, M., & Linoff, G. (1997). *Data mining techniques*. New York: Wiley.

Bhalla, U. S., & Lyengar, R. (1999). Emergent properties of networks of biological signaling pathways. *Science*, *283*, 381–397.

Bhalla, U. S., Ram, P. T., & Iyengar, R. (2002). Map kinase phosphatase as a locus of flexibility in a mitogen-activated protein kinase signaling network. *Science*, *297*(5583), 1018–1023.

Chang, P. C., Wang, Y. W., & Tsai, C. Y. (2005). Evolving neural network for printed circuit board sales forecasting. *Expert Systems with Applications*, *29*(1), 83–92.

Chen, Y., & Jeng, B. (2002). Yet another representation for system dynamics models, and its advantages. *The 20th international conference of the system dynamics society*. Italy: Palermo.

Chong, L., & Ray, L. B. (2002). Whole-istic biology. *Science*, *295*(5560), 1661.

Coyle, R. G. (1977). *Management system dynamics*. Chichester: Wiley.

Csete, M. E., & Doyle, J. C. (2002). Reverse engineering of biological complexity. *Science*, *295*(5560), 1664–1669.

Curran, D., & O'Riordan, C. (2002). Applying evolutionary computation to designing neural networks: A study of the state of the art. *Technical report NUIG-IT-111002*. Galway: National University of Ireland.

Davidson, E. H., Rast, J. P., Oliveri, P., Ransick, A., Calestani, C., Yuh, C.-H., et al. (2000). A genomic regulatory network for development. *Science*, *295*(5560), 1669–1678.

Fogel, G. B., & Corne, D. W. (2003). *Evolutionary computation in bioinformatics*. California: Morgan Kaufmann.

Forrester, J. W. (1961). *Industrial dynamics*. Cambridge, MA: MIT Press.

Gallagher, R., & Appenzeller, T. (1999). Beyond reductionism. *Science*, *284*(5411), 79.

Gardner, T. S., & Collins, J. (2000). Neutralizing noise in gene networks. *Nature*, *405*(6786), 520–521.

Goldberg, D. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading, MA: Addison-Wesley.

Gruau, F., & Whitley, D. (1993). Adding learning to the cellular development of neural networks. *Evolutionary Computation*, *1*(3), 213–233.

He, Z., Xu, X., Huang, J. Z., & Deng, S. (2004). Mining class outliers: Concepts, algorithms, and applications in CRM. *Expert Systems with Applications*, *27*(4), 681–697.

Henry, C. M., & Washington, E. (2003). Systems biology. *Chemical and Engineering News*, *81*(20).

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Michigan: University of Michigan Press.

Kitano, H. (2002a). Systems biology: A brief overview. *Science*, *295*(5560), 1662–1664.

Kitano, H. (2002b). Computational systems biology. *Nature*, *420*(6912), 206–210.

Kitano, H. (2004). Cancer as a robust system: Implication for anticancer therapy. *Nature Reviews Cancer*, *4*(3), 227–235.

Koza, J. R., Keane, M. A., & Streeter, M. J. (2003). Evolving inventions. *Scientific American*, *288*(2), 52–59.

Koza, J. R., Mydlowec, W., Lanza, G., Yu, J., & Keane, M. A. (2001). Reverse engineering of metabolic pathways from observed data using genetic programming. *Pacific Symposium on Biocomputing*, *6*, 434–445.

Li, S., & Shue, L. (2004). Data mining to aid policy making in air pollution management. *Expert Systems with Applications*, *27*(3), 331–340.

Lyneis, J. M., & Pugh A. L. (1996). Automated vs. 'hand' calibration of system dynamics models—an experiment with a simple project model. *Proceedings of the 1996 international system dynamics conference, Cambridge*.

Mendes, P. (2001). Modeling large biological systems from functional genomic data: Parameter estimation. In H. Kitano (Ed.), *Foundations of systems biology*. Cambridge, MA: MIT Press.

Michalewicz, Z. (1994). Evolutionary computation techniques for nonlinear programming problems. *International Transactions in Operational Research*, *1*(2), 223–240.

Morohashi, M., Winn, A. E., Borisuk, M. T., Bolouri, H., Doyle, J., & Kitano, H. (2002). Robustness as a measure of plausibility in models of biochemical networks. *Journal of Theoretical Biology*, *216*(1), 19–30.

Nobel, D. (2002). Modeling the heart—From genes to cells to the whole organ. *Science*, *295*(5560), 1678–1682.

Oliva, R. (2003). Model calibration as a testing strategy for system dynamics models. *European Journal of Operational Research*, *151*(3), 552–568.

Prusinkiewicz, P., & Lindenmayer, A. (1992). *The algorithmic beauty of plants*. New York: Springer.

Ritchie, M. D., White, B. C., Parker, J. S., Hahn, L. W., & Moore, J. H. (2003). Optimization of neural network architecture using genetic programming improves detection and modeling of gene–gene interactions in studies of human diseases. *BMC Bioinformatics*, *4*(28).

Shin, K., & Lee, Y. (2002). A genetic algorithm application in bankruptcy prediction modeling. *Expert Systems with Applications*, *23*(3), 321–328.

Starr, P. J. (1980). Modeling issues and decisions in system dynamics. *TIMS Studies in the Management Science*, *14*, 45–59.

Tyson, J. (1991). Modeling the cell division cycle: cdc2 and cyclin interactions. *PNAS*, *88*, 7328–7332.

Versace, M., Bhatt, R., Hinds, O., & Shiffer, M. (2004). Predicting the exchange traded fund DIA with a combination. *Expert Systems with Applications*, *27*(3), 417–425.

Yi, T., Huang, Y., Simon, M. L., & Doyle, J. (2000). Robust perfect adaption in bacterial chemoyaxis through integral feedback control. *PNAS*, *97*(9), 4649–4653.