Biotechnology and Bioprocess Engineering **2005**, 10: 400-407

Component-Based Software Architecture for Biosystem Reverse Engineering

Doheon Lee*

Department of BioSystems, KAIST, Daejeon 373-1, Korea

Abstract Reverse engineering is defined as the process where the internal structures and dynamics of a given system are inferred and analyzed from external observations and relevant knowledge. The first part of this paper surveys existing techniques for biosystem reverse engineering. Network structure inference techniques such as Correlation Matrix Construction (CMC), Boolean network and Bayesian network-based methods are explained. After the numeric and logical simulation techniques are briefly described, several representative working software tools were introduced. The second part presents our component-based software architecture for biosystem reverse engineering. After three design principles are established, a loosely coupled federation architecture consisting of 11 autonomous components is proposed along with their respective functions.

Keywords: reverse engineering, biosystem, network inference, simulation, component

INTRODUCTION

Reverse engineering is defined as the process where the internal structure of a given system is inferred from external observations and relevant knowledge [1]. Typical reverse engineering processes also include understanding the structural and dynamic characteristics of a target system, since the primary purpose of reverse engineering is to elicit "actionable" knowledge and experimental plans for the system. Conventionally, reverse engineering techniques have been widely used to derive circuit layouts and underlying software structures by examining external behaviors of electronic products and software systems with the purpose of modifying or integrating them. Since the biology of systems is a discipline used to understand biosystems based on the system theory, reverse engineering can be regarded as a central part of the discipline [2].

Biosystem reverse engineering can be conducted along three steps. The first step is to identify the components of a given biosystem. For example, genes, transcription factors, enzymes, metabolites, ligands, and receptors are components of cell systems. Recent genome sequencing projects and bioinformatics technology have identified a huge number of molecular components [3]. The second step is to unravel the interaction structures among the components. Biological circuits such as genetic regulatory networks, metabolic pathways, and signal transduction pathways are typical types of interaction structures in cell systems. Hybrid biological circuits of heterogeneous types of pathways have recently drawn increasing attention, as single type pathways were insufficient in explaining the complex behaviors of the biosystems. The third step is to understand structural and dynamic characteristics of the system so that we can elicit actionable knowledge. Following that, subsequent actions such as pathway modification, consolidation of artificial functions and a "what-if" analysis can be conducted. For example, novel microbial strains can be engineered by modifying particular metabolic pathways to yield valuable metabolites more productively than the corresponding wild types [4,5]. Understanding of regulatory networks and signal transduction pathways is essential to identify novel drug targets, as well as anticipate the complicated side effects of potential drugs [6].

Depending on the types of biological circuits and target organisms, reverse engineering may require different techniques. For example, inference of metabolic pathways can be achieved effectively by observing and analyzing relative concentration changes of key biomolecules, including the application of deterministic optimization techniques. Meanwhile, the inference of signal transduction pathways requires more sophisticated experimental techniques to measure the tiny amount of signal molecules and stochastic techniques to deal with probabilistic reactions. Yet, they also share a commonality in that they can be represented by networks where the nodes are biomolecules or higher level bio-entities, and the edges are relationships among them, though the biological interpretations vary widely. In most cases, their mutual interactions undergo non-linear fashions primarily due to positive or negative, direct or indirect feedback.

Obviously, any single technique cannot fulfill the requirement of reverse engineering even for a single type of biological circuits since different techniques can shed light on different aspects. Furthermore, the reverse engineering process itself consists of multiple complex sub-

***Corresponding author**
Tel: +82-42-869-4316 Fax: +82-42-869-8680
e-mail: dhlee@biosoft.kaist.ac.kr

tasks. This is similar to business information systems where different business tasks are applied to common data sets, and many subtasks are integrated to meet fundamental business needs. One key state-of-the-art technology that deals with such complexity in business information systems is the component-based development (CBD) [7]. Each subtask is mapped into a component. A component is a program unit with its own data structures, well-defined access methods and self-management capabilities. The components are developed independently while conforming to standard interfaces, and integrated with a larger system in a hierarchical manner. Among several available component infrastructures such as CORBA, .NET, and EJB, the Web Services protocol is widely used for loosely federated software systems [8].

Since such a framework should provide effective and efficient channels for control and data migration (while allowing autonomous executions of participating functions), we decided to adopt the Web Services protocol as a coordination infrastructure standard. It consists of XML-based protocols WSDL (Web Service Definition Language), SOAP (Simple Object Access Protocol), and UDDI (Universal Description, Discovery, and Integration) [8]. After introducing a variety of techniques applicable to biosystem reverse engineering, this paper presents our component-based architecture consisting of 3 layers and 11 components.

Techniques for Biosystem Reverse Engineering

This section examines existing techniques for biosystem reverse engineering. First, structure inference techniques are described, where networks of interactions among bio-entities such as genes, proteins, and metabolites are inferred. Simulation analysis techniques are divided into two distinct approaches - numeric and logical. Numeric simulation aims to obtain quantitative predictions on the system behaviors. Logical simulation focuses on qualitative and coarse-grained analysis. In addition, several representative working software tools are introduced.

Network Structure Inference

One of the frontier techniques for network structure inference is the Correlation Matrix Construction (CMC) method developed by Arkin and Ross [9]. Given a set of time course vectors, this method infers a metabolic pathway structure each representing temporal fluctuations of an abundant level of a specific metabolite. They adopted a time-lagged correlation measure to quantify the relationship between each pair of metabolites. This correlation matrix is fed into a multidimensional scaling technique to draw a network of interactions among a handful of metabolites. Though this technique has succeeded in showing that computational techniques on observational data could uncover internal structures of certain types of biological circuits, it is constrained by its single-link restriction. Combinational effects of more than one metabolite cannot be incorporated as the author have mentioned.

When combinational effects are taken into account, the search space of network structures expands exponentially Roughly speaking, given n nodes, the number of network structures to probe becomes $\Omega(2^n)$, whereas only n^2 comparisons are sufficient when we restrict only single links. There have been many efforts to restrict the search space by premising different assumptions to work around the computational intractability. The most commonly postulated assumption is the linearity where the system behavior can be regarded as a sum of individual effects. In [10], they have modeled the reverse engineering of a genetic regulatory system with a matrix inversion problem. Once the regulatory system behavior is formulated as a combination of linear matrix equations and a sigmoid function, several matrix inversion techniques such as singular value decomposition (SVD) and the Moore-Penrose's pseudo inverse method are applied to determine the coefficients in the matrices. In [11], they adopted a differential equation model where the abundance change rate of each gene is represented as a linear function of the abundance of the other genes. The authors postulated that it is relatively easy to know whether the rates of change are positive or negative. In other words, increasing or decreasing though the specific rate values are unknown. Based on this assumption, they proposed to apply linear programming techniques to fit the coefficients of the linear functions. In [12], they applied the pseudo-steady state assumption to metabolic flux analysis so that the rates of change of metabolites can be assumed to be nearly zero. Thus, the reaction model is reduced into a linear equation system, where Lee can apply linear programming-based optimization techniques to determine model parameters.

Besides such algebraic approaches, several machine learning-based techniques have been proposed. One of the significant works is the REVEAL (REVerse Engineering ALgorithm) program for inferring genetic regulatory networks in the form of Boolean networks [13-15]. It maps the activity level of a gene into a binary state, on or off. The program utilizes the mutual information among gene activity levels to identify which combination of genes determines the next state. Once such combinations are identified, the genetic regulatory network can be drawn, where each node represents a gene, and Boolean connectives such as AND, OR, and NOT, associate with the nodes. Though its two assumptions, which state that each gene has binary states and that state transitions are synchronous, have been regarded as drawbacks by other researchers, it can be regarded as among the first explicit formalisms for automatic inference of genetic regulatory structures.

Bayesian networks have been used to model probabilistic characteristics of relations among events. Due to its proven benefits in solid statistical analysis and noise handling capability, there have been extensive efforts to infer Bayesian networks from observations in the machine learning and statistics communities. In [16], this technique was utilized to infer genetic regulatory networks from microarray gene expression data. Unlikely to the conventional applications of Bayesian networks, the problem of regulatory network inference suffers from a higher

degree of dimensionality. In other words, there were too few observations to infer structures among too many genes. They proposed to adopt several techniques such as sparse candidates and model averaging to alleviate the dimensionality problem. Aside from this statistical heuristics, the others have proposed incorporating additional biological knowledge such as protein-DNA bindings using chromatin immuno-precipitaion (ChIP) assays. This includes a promoter sequence consensus in order to filter biologically plausible links between genes [17,18]. When gene-specific perturbation experiments are available, more reliable inference of structures are possible [19,20]. MONET (MOdularized NETwork Learning) has adopted a divide-and-conquer approach to alleviate the dimensionality problem [21]. First, it divides a whole gene set to overlapped modules considering two complementary sources of information: biological annotations and expression data. Second, it infers a Bayesian network for each module, and integrates the learned subnetworks to a global network. The postulated assumption is that a cellular system is composed of locally interacting biological modules; most of the genes are likely to be related to the genes in the same biological modules rather than the genes in different modules. It can draw a global picture of inter-module relationships as well as a detailed look of 6 intra-module interactions. Lee demonstrated that MONET could lead to at least a two-fold improvement in accuracy on the inference compared to whole-set-based and expression-based clustering approaches.

As in other domains, regression trees turned out to be a useful formalism for inferring network structures. In [22], they have clustered genes based on their expression similarities. A regression tree was built for each cluster where decision nodes are formed in terms of known transcription factors. By modifying cluster compositions repeatedly, they continue to refine the regression trees to fit the entire expression profiles. Though it should be given a set of transcription factors in advance, and despite the inferred networks being bi-partitioned between transcription factors and genes, it has succeeded in finding useful novel relationships. Some of them are even validated through actual wet-lab experiments. In [23], they have built regression trees to find combinational effects of transcription binding motifs, and distinguish more dominant ones with respect to given expression profiles.

Though a large range of techniques have been developing for network structure inference, there is still much room to improve upon. One of the most critical critiques from the potential users of these techniques is the false positive prediction of relationships. Effective incorporation of biological domain biases to the learning process is imperative along with efforts to improve computational sophistication.

Numeric Simulation

Once the network structures are given or inferred, the next step of reverse engineering is to analyze the structural characteristics and dynamic behavior of the system. Thus, we can understand the critical pathways, conduct a "what-if" analysis, and uncover natural principles underlying the biosystems.

Nonlinear differential equations have been widely used to understand time-variant phenomena in many decades [24]. After composing a set of differential equations where the left-hand side represents the rates of change of particular measurements, and the right-hand side is a nonlinear function of other relevant measurements, we can monitor the system state trajectories given specific initial conditions. Depending on the characteristics of equations, we can further analyze the important properties of the target system such as boundaries and plasticity. Recently, bifurcation analysis has drawn increasing attention since attractors and bifurcations can be mapped into phenotypical state transitions of the target biosystems whereas detailed value changes are hard to interpret in many cases. Due to the long-standing mathematical endeavor in dealing with nonlinear differential equations, the system behaviors can be well understood as long as we secure precise equations and the necessary coefficient assignment. However, these two premises are hard to attain in reverse engineering biological circuits, where precise models of biological reactions are not available in many cases, and the proper assignment of relevant coefficients are much harder. To remedy this constraint, several approximation techniques have been proposed. Among them, qualitative transition graphs based on the piecewise linear model has drawn interests [25]. The assumption is that the change rate of a species can be represented as a linear function of the concentration of other species when the concentrations of some genes fall within particular ranges. In other words, in the ranges, the system is assumed to show linear characteristics in terms of the concentration of species. The technique partitions the entire system state space into disjointed domains, within each, the system behaves in a linear manner. It also extracts state transition diagrams between domains, and provides a method to simulate the system behavior in terms of state transitions.

While the differential equation formalism can effectively model deterministic system behaviors in terms of species concentrations, there are many cases where such determinism and concentration-based handling are inappropriate. It is especially hard to deal with the concentration of species in cellular signal transduction where a few molecules can trigger different cellular responses. Furthermore, it has stochastic characteristics since species interactions are probabilistic. To deal with such classes of biological phenomena, several stochastic techniques including stochastic Petri nets have been proposed [26]. When a large part of the biosystem is under consideration, the network itself becomes too complicated to understand even though individual regions are well understood. They have adopted the workflow concepts, originally developed to model complex business operations by the Workflow Management Coalition. They also propose to utilize hierarchical Petri nets to control abstraction levels of system specification [27].

Logic-Based Simulation

Since Kauffman has proposed to utilize the Boolean

network to model and simulate genetic regulatory networks [15], there have been many techniques developed based on the Boolean network and its variations. Unlikely to the numeric simulations, the system states are represented in terms of Boolean states, where each elementary state is either on or off. Furthermore, state transitions are synchronous and Boolean logic-driven. Though this style of simulation cannot identify precise changes in the concentrations of species or continuous state changes, it can probe coarse-grained system behaviors effectively without either detail kinetic equations or precise parameter values. In addition, the Boolean network-based analysis has been extended to study global characteristics of large scaled regulatory systems [28,29]. Analysis attractors, trajectories, and basins of attractors can show the implications of local properties for the global dynamics of the regulatory networks. Two major constraints of Boolean networks (binary states and synchronous state transitions) have been eased by the generalized logical network formalism [30]. It allows each state to have more than two states and asynchronous state transitions. Other researchers have applied probabilistic characteristics to the Boolean network so that it can model and simulate probabilistic state transitions [31].

If-then rules, which have been extensively used to model business knowledge, have also been applied for biosystem modeling and simulation [32,33]. For example, they can compose a set of rules such as 'if the temperature and pH of an experimental condition fall between 0 and 30 and 6 and 8, respectively, then the activity of an enzyme is below 0.3'. Both forward and backward reasoning can be applied to the rules. As in the conventional production systems, forward reasoning can enumerate possible consequences of given facts and rules, while backward reasoning can keep track of necessary preconditions for given situations. This formalism has relative advantages in its capability to encompass a wider range of biological knowledge. Since biological entities, such as protein complexes and DNA-protein binding, have compound structures, and biological processes are regarded as hierarchical processes, a compound graph-based formalism has been proposed [34]. Once a target biological system is represented in the form of a compound graph, a logical inference system such as HiLog can be utilized to answer forward and backward queries.

Working Software Tools

A lot of software tools have been developed and distributed for biosystem modeling and simulation. Several graph visualization tools including Osprey, PIMrider, Graphlet and daVinci can be adopted to visualize biocircuits. Some of them also provide connections to biomolecular interaction databases such as BIND and DIP as well as function databases such as TRANSFAC. Several dozen bio-circuit simulation programs such as GEPASI [35], E-Cell [36], Cellerator [37] MetaFluxNet [12], and Virtual Cell [38,39] have been drawing increasing attention from system biologists. Integrated platforms such as Systems Biology Workbench [40], Cytoscape [41], and Genomic Object Net [42] have also been developed, where a variety tools can be integrated to achieve complex analysis functions in unified environments.

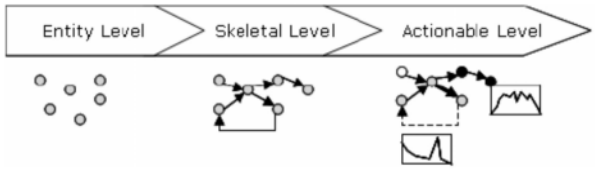

Fig. 1. Transformation of augmented circuit templates.

Reverse Engineering for Actionable Model Exploration

This section presents our component-based system architecture for biosystem reverse engineering. There are several design principles underlying the architecture. First, it should provide an environment where autonomous components are loosely coupled. Depending on the reverse engineering task in hand, the way of component assembly can vary widely. There can be several alternatives for the same subtask. Thus, tightly coupled integration can place critical burdens against such flexible requirements. Second, it could accommodate existing solutions as well as newly developed components in minimal adaptation overhead. Third, it could provide a unified user interface metaphor even though participating components have their own interface styles.

Augmented Circuit Model

We propose a central information structure that bridges gaps between disparate user interface styles, and provides heterogeneous components with the functional closure property. It is referred to as an augmented circuit template, where relevant raw data and derived information such as activation levels of bio-entities and inter-entity regulation strengths (as well as structural constraints of biological circuits are represented in a unified framework). Reverse engineering can be mapped using a process of transformation, enrichment, annotation, and refinement of the augmented circuit templates as depicted in Fig. 1.

A disconnected collection of biological entities such as genes and proteins constitutes an entity level template. Experimental measurements such as gene expressions and metabolic profiles can be augmented to this template. The application of a particular structure identification technique such as Bayesian network learning can transform this model into the skeletal level. Depending on the characteristics of applied techniques, the connections in the skeletal level templates can be deterministic or stochastic. Various structural or dynamics analysis techniques upgrade the skeletal level templates to the actionable level. The actionable level templates can be further refined by incorporating additional analysis results.

Overall System Architecture

Fig. 2 shows the overall system architecture for biosystem reverse engineering. The architecture consists of 11

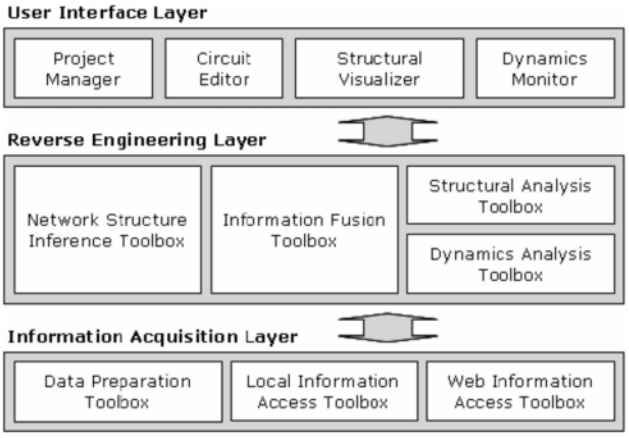

Fig. 2. Three-layered component-based architecture.

components, which are divided into three layers. The layered architecture reduces communication complexity by restricting interacting pairs of components. Most components come in the form of toolboxes; in other words, a set of functional units sharing common interfaces. The underlying communication infrastructure among components is the Service Oriented Architecture (SOA) based on XML-based protocols WSDL (Web Service Definition Language) and SOAP (Simple Object Access Protocol) [8]. This infrastructure facilitates the loosely coupled integration of existing functional units, as well as allowing isolated invocation of them. The following subsections describe the functions of each component layer by layer.

User Interface Layer

This layer is primarily responsible for interactive conversation with the users. It adopts the word processing metaphor where an augmented circuit template is mapped to a text document. Composing, editing, printing, formatting text documents can be mapped to constructing, modifying, analyzing structurally and dynamically, and visualizing augmented circuit templates, respectively. Since most potential users are familiar with common word processor (WP) systems, this metaphor could provide one of the easiest user interface styles.

Project Manager

Depending on the characteristics of the analysis target in hand, various scenarios can be applicable. For example, if a user is going to reverse engineer genetic regulatory circuits for yeast cell differentiation, the first task might be to acquire gene expression profiles and pre-process them.

The pre-processed profiles should be fed into a network inference unit such as Bayesian network learning. Since the edges in the inferred network are likely to contain significant amounts of false positives, information fusion units should be invoked to filter them. If the user's objective is to unravel the static structure of the circuit, this could fulfill the requirement. Otherwise, the user would have to perform more sophisticated structural analysis or dynamic simulations. This whole scenario can be mapped into successive and parallel invocations of corresponding functional units in various toolboxes. Since the scenario itself is complex and has its own value of knowledge, a user interface layer component-Project Manager-stores, manages, and reuses them using its hierarchical storage structure.

Circuit Editor

This component is the main dialogue agent with the users. It utilizes Structural Visualizer to present biological circuits in the graphical user interface environment. It also utilizes a Dynamic Monitor to monitor the circuit behavior vs time. The biological circuits can be obtained from the Reverse Engineering layer or imported from external sources. Though its internal data structures are derived from the augmented circuit templates mentioned in the previous section, it can parse and transform external bio-pathway formats such as Systems Biology Markup Language (SBML) [43] and Matabolic Flux Analysis Markup Language (MFAML) [44]. The Circuit Editor also provides the users with various edit functions such as circuit modification, merger of partial circuits, and consolidation of different levels of circuits, as well as circuit browsing functions.

Structural Visualizer

When a biological circuit contains several hundreds or thousands of bio-entities, it is not straightforward to visualize it. Structural Visualizer provides various modalities of visualization including hierarchical decomposition, extraction of critical pathways and reverse traverse so that the users can comprehend the circuit and focus on essential parts.

Dynamics Monitor

Dynamics analysis of biological circuits producesrious types of results such as temporal trajectories of multiple variables, state transition diagrams and stability diagrams. Since large biological circuits have many variables, and there are interrelationships among variables, Dynamic Monitor provides user-comprehensible means of monitoring dynamics.

Reverse Engineering Layer

This layer contains four components for reverse engineering: network structure inference, information fusion, structural analysis, and dynamics analysis.

Network Structure Inference Toolbox

Depending on the problem at hand, various techniques can be applied to infer network structures of biological circuits. This toolbox provides representative techniques such as Bayesian-network learning, Boolean-network learning, Correlation Matrix Construction (CMC), and association-rule learning. Data Preparation Toolbox processes experiment data result in an augmented circuit template in its entity level. This toolbox transforms the model to its skeletal level where associations among bio-entities

are elicited. Note that there are many actual cases where this skeletal level with an augmented circuit template cannot avoid significant amounts of false positive associations due to dimensionality problems. In other words, the available experiment data is likely to be too few to draw reliable network structures. For example, at most, a few hundred microarray gene expression profiles are applicable to network inference for several thousands of genes.

Information Fusion Toolbox

In order to identify the false positives mentioned above, and assign biological meanings to the associations in an augmented circuit template, it is indispensable to incorporate extra information from independent sources. The Information Fusion Toolbox implements two different techniques for this purpose. First, it incorporates prior biological knowledge. For example, if we know cellular localization, gene sequence motifs of transcriptional regulation information in advance, it helps to identify invalid associations effectively. Second, the technique applies to multi-modality analysis where different aspects of the same circuit are separately elaborated, and then consolidated. For example, a genetic regulatory network, a protein interaction map, and a metabolic pathway on the same part of cellular phenomena can be consolidated. The latter technique requires several information fusion techniques, including synchronization, scaling and confidence coordination, as well as ontological mapping.

Structural Analysis Toolbox

Once the network structure of a biological circuit is unraveled, we can apply various structural analysis techniques. Path and cycle analysis is able to provide critical paths, possible system state transitions, duplicate paths, and feedback loops. Global connectivity analysis can show the submodules of the circuit, each of which takes distinguishable functions, while loosely being associated with other submodules. Structural comparison of multiple network structures can enlighten evolutionarily conserved parts as well as provide clues for their functions. All of these analysis results are incorporated into its corresponding augmented circuit template for subsequent analysis.

Dynamics Analysis Toolbox

Typical techniques for dynamics analysis are quantitative simulation, trajectory analysis, and perturbation stability analysis. Since it is hard to attain precise kinetic models and parameters, fuzzy theoretic approximate analysis and qualitative analysis techniques have been investigated to add to this toolbox as functional units. Though approximate analysis cannot bring precise quantitative values for particular situations, its coarse-grained results can shed light on global dynamic characteristics as well as suggest further fine-grained experiments. Applicable to the Structural Analysis Toolbox, the analysis results are incorporated into its corresponding augmented circuit template for subsequent analysis.

Information Acquisition Layer

This layer has components for information acquisition from in-house experiments, local databases, and external web accessible bioinformatics databases.

Data Preparation Toolbox

Raw experiment data requires several steps of preprocessing including normalization, interpolation, extrapolation, and transformation. Specific procedures for the preprocessing tasks vary depending on the characteristics of the raw data. Raw data types in the current version include profiles of the microarray gene expressions, metabolism, and proteins. A data preparation toolbox adds relevant raw data profiles to the corresponding bioentities in the entity level augmented circuit template.

Local Information Access Toolbox

Local information for biological circuit reverse engineering includes in-house experiment data, downloaded bioinformatics databases and preprocessed secondary databases. Extensible objectrelational schemas are designed to store and manage this information in a unified way. Some data types can be transformed to object-relational formats, while others are stored in their own formats along with corresponding file descriptors in the central database. An augmented circuit template can have pointers to or copies of relevant data items. This attachment is a part of the augmented circuit template enrichment process.

Web Information Access Toolbox

Currently, over 500 public bioinformatics databases are accessible through the Web, which encompasses DNA sequences, biological pathways, and disease information. It is crucial and demonstrates the essential power of bioinformatics to utilize this information. This toolbox connects to various Web search engines to find relevant information, building indices for rapid and focused accesses. Text mining capabilities are also included in this toolbox to extract relevant information from textual information segments like Medline abstracts.

Discussion and Concluding Remarks

This paper has introduced existing techniques for biosystem reverse engineering. It describes the representative approaches for network structure inference including Correlation Matrix Construction (CMC), Boolean network and Bayesian network-based methods.

It also explains the different techniques for structural and dynamic analysis of target biosystems, introducing some commonly used software tools in the field. Based on the three proposed design principles, a component-based software architecture for biosystem reverse engineering is proposed. The architecture identifies essential functions, and organizes them into 11 components in three layers. By adopting the Web Services protocol, it has a higher extensibility level and requires less effort for accommodating existing solutions.

Currently, several components are being plugged into the architecture. A Bayesian network inference component called MONET, which itself is a two-tier distributed

system utilizing 32-way parallel processing, is being adapted for network structure inference. A web search engine equipped with text mining capabilities referred to as "UNICORN" is also being adapted for the information acquisition and fusion functions. Our own components for structural and dynamic analysis functions are being developed while existing tools such as MetaFluxNet are integrated.

Acknowledgement This work was supported by the National Research Laboratory Grant (2005-01450) from the Ministry of Science and Technology. We would like to thank CHUNG Moon Soul Center for BioInformation and BioElectronics and the IBM SUR program for providing research and computing facilities.

REFERENCES

[1] Csete, M. and J. Doyle (2002) Reverse engineering of biological complexity. *Science* 295: 1664-1669.
[2] Kitano, H. (2002) Systems biology: a brief overview. *Science* 295: 1662-1664.
[3] International Human Genome Sequencing Consortium (2004) Finishing the euchromatic sequence of the human genome. *Nature* 431: 931-945.
[4] Nam, J. W., K. H. Han, E. S. Yoon, D. I. Shin, J. H. Jin, D. H. Lee, S. Y. Lee, and J. W. Lee (2004) *In silico* analysis of lactate producing metabolic network in *Lactococcus lactis*. *Enzyme Microb. Technol.* 35: 654-662.
[5] Jin, J. H, U. S. Jung, J. W. Nam, Y. H. In, S. Y. Lee, D. H. Lee, and J. W. Lee (2005) Construction of comprehensive metabolic network for glycolysis with regulation mechanisms and effectors. *J. Microbiol. Biotechnol.* 15: 161-174.
[6] Di Bernardo, D., M. J. Thompson, T. S. Gardner, S. E. Chobot, E. L. Eastwood, A. P. Wojtovich, S. J. Elliott, S. E. Schaus, and J. J. Collins (2005) Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks. *Nat. Biotechnol.* 23: 377-383.
[7] Kim, W., K. C. Kim, E. K. Hong, and D. Lee (2000) A component-based architecture for preparing data in data warehousing. *J. Obj. Oriented Prog.* 13: 43-47.
[8] Web Services Activity (2002) http://www.w3c.org/2002/ws.
[9] Arkin, A., P. D. Shen, and J. Ross (1997) A test case of correlation metric construction of a reaction pathway from measurements. *Science* 277: 1275-1279.
[10] Weaver, D. C., C. T. Workman, and G. D. Stormo (1999) Modeling regulatory networks with weight matrices. *Proc. Pac. Symp. Biocomput.* 112-123.
[11] Akutsu, T., S. Miyano, and S. Kuhara (2000) Algorithms for inferring qualitative models of biological networks. *Proc. Pac. Symp. Biocomput.* 293-304.
[12] Lee, D. Y., H. Yun, S. Park, and S. Y. Lee (2003) MetaFluxNet: the management of metabolic reaction information and quantitative metabolic flux analysis. *Bioinformatics* 19: 2144-2146.
[13] Liang, S., S. Fuhrman, and R. Somogyi (1998) REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. *Proc. Pac. Symp. Biocomput.* 18-29.
[14] Akutsu, T., S. Miyano, and S. Kuhara (1999) Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. *Proc. Pac. Symp. Biocomput.* 17-28.
[15] Kauffman, S. A. (1969) Metabolic stability and epigenesist in randomly constructed genetic nets. *J. Theor. Biol.* 22: 437-467.
[16] N. Friedman, M. Linial, I. Nachaman, and D. Pe'er (2000) Using Bayesian networks to analyze expression data, *J. Comp. Biol.,* 7: 601-620.
[17] Hartemink, A. J., D. K, Gifford, T. S. Jaakkola, and R. A. Young (2002) Combining location and expression data for principled discovery of genetic regulatory network models. *Proc. Pac. Symp. Biocomput.* 437-449.
[18] Tamada, Y., S. Kim, H. Bannai, S. Imoto, K. Tashiro, S. Kuhara, and S. Miyano (2003) Estimating gene networks from gene expression data by combining Bayesian network model with promoter element detection. *Bioinformatics* 19(S2): II227- II236.
[19] Pe'er, D., A. Regev, G. Elidan, and N. Friedman (2001) Inferring subnetworks from perturbed expression profiles. *Bioinformatics* 17(S1): S215-S224.
[20] Yoo, C., V. Thorsson, and G. F. Cooper (2000) Discovery of causal relationships in a gene regulation pathway from a mixture of experimental and observational DNA microarray data. *Proc. Pac. Symp. Biocomput.* 498-509.
[21] Lee, P. H. and D. H. Lee (2005) Modularized learning of genetic interaction networks from biological annotations and MRNA expression data. *Bioinformatics* 21: 2739-2747.
[22] Segal, E., M. Shapira, A. Regev, D. Peer, D. Botstein, D. Koller, and N. Friedman (2003) Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat. Genet.* 34: 166-176.
[23] Phuong, T. M., D. Lee, and K. H. Lee (2004) Regression trees for regulatory element identification. *Bioinformatics* 20: 750-757.
[24] Voit, E. (2000) Computational analysis of biochemical systems, Cambridge Univ. Press.
[25] De Jong, H., J. Geiselmann, C. Hernandez, and M. Page (2003) Genetic network analyzer: qualitative simulation of genetic regulatory networks. *Bioinformatics* 19: 336-344.
[26] Goss, P. J. and J. Peccoud (1998) Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets. *Proc. Natl. Acad. Sci. USA* 95: 6750-6755.
[27] Peleg, M., I. Yeh, and R. B. Altman (2002) Modeling biological processes using workflow and Petri Net models. *Bioinformatics* 18: 825-837.
[28] Kauffman, S. A. (1991) Antichaos and adaptation. *Sci. Am.* 265: 78-84.
[29] Kauffman, S. A. (1993) The Origins of Order: Self-Organization and Selection in Evolution. Oxford University Press, Oxford, UK.
[30] Thomas, R. (1991) Regulatory networks seen as asynchronous automata: A logical description. *J. Theor. Biol.* 153: 1-23.
[31] Shimulevich, I., E. R. Dougherty, S. Kim, and W. Zhang (2002) Probabilistic Boolean Networks: a rule-based uncer-

tainty model for gene regulatory networks. *Bioinformatics* 18: 261-274.
[32] Brutlag, D. L., A. R. Galper, and D. H. Millis (1991) Knowledge-based simulation of DNA metabolism: prediction of enzyme action. *Comput. Appl. Biosci.* 7: 9-19.
[33] Hofestadt, R. and F. Meineke (1995) Interactive modelling and simulation of biochemical networks. *Comput. Biol. Med.* 25: 321-334.
[34] Fukuda, K. and T. Takagi (2001) Knowledge representation of signal transduction pathways. *Bioinformatics* 17: 829-837.
[35] Mendes, P. (1997) Biochemistry by numbers: simulation of biochemical pathways with Gepasi 3. *Trends Biochem. Sci*. 22: 361-363.
[36] Tomita, M., K. Hashimoto, K. Takahashi, T. S. Shimizu, Y. Matsuzaki, F. Miyoshi, K. Saito, S. Tanida, K. Yugi, J. C. Venter, and C. A. 3rd. Hutchison (1999) E-Cell: software environment for whole-cell simulation. *Bioinformatics* 15: 72-84.
[37] Shapiro, B. and E. Mjolsness (2001) Developmental Simulations with Cellerator. *Proc. Second International Conference on Systems Biology*, Pasadena, CA, USA.
[38] Schaff, J. and L. M. Loew (1999) The Virtual Cell, *Pac. Symp. Biocomput*. 228-239.
[39] Loew, L. M. and J. Schaff (2001) The Virtual Cell: A Software Environment for Computational Cell Biology, *Trends Biotechnol.* 19: 401-406.
[40] Hucka, M., A. Finney, H. M, Sauro, H. Bolouri, J. Doyle, and H. Kitano (2002) The ERATO Systems Biology Workbench: enabling interaction and exchange between software tools for computational biology. *Proc. Pac. Symp. Biocomput*. 450-461.
[41] Shannon, P., A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.* 13: 2498-2504.
[42] Nagasaki, M., A. Doi, H. Matsuno, and S. Miyano (2003) Genomic Object Net: A platform for modelling and simulating biopathways. *Appl. Bioinformatics* 2: 181-184.
[43] Hucka, M, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, A. P. Arkin, B. J. Bornstein, D. Bray, A. Cornish-Bowden, A. A. Cuellar, S. Dronov, E. D. Gilles, M. Ginkel, V. Gor, I. I. Goryanin, W. J. Hedley, T. C. Hodgman, J. H. Hofmeyr, P. J. Hunter, N. S. Juty, J. L. Kasberger, A. Kremling, U. Kummer, N. Le Novere, L. M. Loew, D. Lucio, P. Mendes, E. Minch, E. D. Mjolsness, Y. Nakayama, M. R. Nelson, P. F. Nielsen, T. Sakurada, J. C. Schaff, B. E. Shapiro, T. S. Shimizu, H. D. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner, and J. Wang (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* 19: 524-531.
[44] Yun, H., D. Y. Lee, J. Jeong, S. Lee, and S. Y. Lee (2005) MFAML: a standard data structure for representing and exchanging metabolic flux models. *Bioinformatics* 21: 3329-3330.

[Received August 5, 2005; accepted October 6, 2005]