Computers in Biology and Medicine 43 (2013) 1415–1427

Contents lists available at ScienceDirect

Computers in Biology and Medicine

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

A pattern-oriented specification of gene network inference processes

Nestor W. Trepode, Cléver R.G. de Farias*, Junior Barrera

Department of Computer Science and Mathematics (DCM), Faculty of Philosophy, Sciences and Letters at Ribeirão Preto (FFCLRP), University of São Paulo (USP), Av. Bandeirantes, 3900, Monte Alegre, Ribeirão Preto 14040-901, SP, Brazil

ARTICLE INFO

Article history:
Received 2 February 2011
Accepted 6 July 2013

Keywords:
Genetic regulatory networks
Dynamical system identification
Gene network inference
Microarray data analysis
Process modeling
Patterns

ABSTRACT

Patterns have been widely used in Computer Science. A pattern describes a generic solution to an existing problem in a more readable and accessible form. A pattern-oriented process specification consists of a generic and abstract description of a process. This paper presents a pattern-oriented specification of a genetic regulatory network inference process performed from microarray data and prior biological knowledge. The proposed specification was conceived based on prior work on gene inference networks. The adequacy of the proposed solution was then evaluated with respect to modern tendencies of the genes network inference literature.

1. Introduction

Bioinformatics concerns the use of advanced computation techniques and biological knowledge, including internet distributed databases and processors, parallel processing and data mining, to promote knowledge advances in (molecular) biology. One of the greatest challenges of the field consists in the development of mathematical models and corresponding inference techniques for mining databases and extracting new knowledge. These models and processes are continuously created, improved and modified by experts in the area. The applied research community profits from these advances with the use of the corresponding software when it becomes available in specialized repositories.

Genes and their protein products, generated by gene expression through transcription and translation, form complex signaling networks, which control metabolic pathways, cellular functions and the future expression of genes themselves. A network of this kind is known as a *gene regulatory network* or *genetic regulatory network* (GRN). In a GRN, the level of expression of a gene depends on the expression values of a gene subset and on external stimuli, both at previous instants of time. This property characterizes GRNs as dynamical systems [1]. Understanding the structure and dynamics of gene regulatory networks is one of the biggest challenges that Biology faces today. For example, the ability of intervening in a GRN in order to make it reach given states and avoid others (such as those associated with disease) would have strong impact in human therapy [2,3] as well as in animal breeding [4,5] and farming [6–8].

Many approaches have been proposed for the inference of genes network from gene expression data, for example [9–16]. Reviews of such methods can be found in [1,17–20]. In general, these approaches consist in the execution of a sequence of procedures (i.e., databases creation and consulting, signal processing, system inference, etc.) until a new biological hypothesis can be inferred from available data and previous knowledge. Since these tools are usually not integrated, researchers of the field are frequently compelled to re-implement pieces of software in order to create a coherent analysis pipeline [21].

The composition of individual system parts to create an integrated software solution can be accomplished using general purpose integration environments, such as Taverna [22,23], Bio-jETI [24,25] and Magallanes [26]. These environments provide great flexibility for the integration of any given set of individual programs. Since these environments are not targeted to a specific domain they neither provide guidelines nor define data structures and interfaces to facilitate the development of individual contributions in any given domain. We believe that this lack of facilities for the development of individual contributions in a particular domain hinders the effective use of these environments as platforms for collaborative scientific research.

We believe that general purpose integration environments would benefit greatly from the availability of abstract process models for specific domains described using a sequence of abstract activities, each one characterized by its input–output data and expected behavior. Based on the abstract specification, different computational methods could then be developed by researchers as possible realizations of these abstract activities. These contributions would easily be integrated with the contributions of other

* Corresponding author. Tel.: +55 1636020564.
E-mail addresses: walter@usp.br (N.W. Trepode), farias@ffclrp.usp.br (C.R.G. de Farias), jb@ime.usp.br (J. Barrera).

0010-4825/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compbiomed.2013.07.008

researchers to create comprehensive analysis solutions. A similar, albeit simpler, approach was adopted by Khoros [27,28], which was a well-known software environment widely used by the image processing research community.

This paper concretely applies this idea by presenting the specification of an abstract genes network inference process that can be used as basis for collaborative research on gene regulatory network inference. We call this abstract process specification a *pattern*, since it describes a reusable solution to a problem in a given context. Our specification was designed based on three previous works on gene regulatory network inference processes [29–32]. The quality of the proposed pattern specification was checked by studying its adequacy to describe representative techniques of different modern tendencies of the genes network inference literature [19,33–35].

Following this introduction, Section 2 recalls the main activities involved in gene regulatory network inference. Section 3 presents a review of concepts and properties of a pattern-oriented process specification and provides an overview of a standard modeling language used in our specification. Section 4 presents an overview of the proposed gene regulatory network inference pattern specification and describes in detail its main activity, *Network Construction*. Section 5 discusses the genesis and robustness the proposed specification. Finally, Section 6 presents some conclusions and outlines future works of this research. Appendix A provides a detailed description of the activities specified in the proposed gene regulatory network inference pattern specification.

2. Gene regulatory network inference

A time-course microarray experiment aims at quantitatively measuring levels of gene expression for a large set of genes (usually thousands) at successive time-intervals and at experimental conditions at which the biological phenomenon under study can be observed. The output of a time-course microarray experiment is the main input to the gene regulatory network identification process. It consists of a $n \times m$ matrix measuring the expression levels of the n genes at m consecutive time instants, usually spaced at regular intervals. Each of the n rows in the matrix represents the expression value of a single gene at the m time instants, and each of the m columns represents the expression values of the whole gene set at a given time, i.e., the values in each column usually come from a different microarray experiment performed at a given time[1] (see Fig. 1a).

These time-course microarray data, after adequate pre-processing (i.e., normalization and quantization), are used for estimating the probabilistic dependence of a *target* gene to a set of *predictor* genes (see Fig. 1a–c). In general, these inference methods are based on the estimation of mutual information [36] or similar concepts such as CoD (Coefficient of Determination) [37], which essentially measures the degree of mass concentration of the conditional distributions due to the observation of a given set of genes. The best predictor sets are exactly the ones that produce a stronger regulation, i.e., the ones that concentrate more substantially the probability mass of the conditional distributions. When evaluating these dependencies from time-course microarray data, predictor genes are observed in samples taken before the target gene sample in order to infer dynamic dependencies and temporal evolution regulation. Nevertheless, in many cases, like in higher eukaryotes, where temporal gene expression data are difficult to obtain, samples taken from different individuals are supposed to capture steady states of the underlying dynamics, and predictor–target dependencies are evaluated inside the same sample [13,30,33,38]. These target–predictor dependencies, identified in the data, are the building blocks for inferring the GRN architecture and dynamics.

Fig. 1 provides an overview of how GRN inference from microarray data is carried out. The main sources of information are periodic samples of gene expression obtained by time-course microarray experiments (Fig. 1a). Based on this information we try to determine dependencies of the type: which (predictor) genes regulate – directly or indirectly according to the data – a (target) gene – for simplicity, in Fig. 1b we assume cardinality two for the predictor set. In order to infer these gene dependencies (also called network *wiring connections*), we estimate from the microarray data the probabilities of occurrence of each possible target value given each possible state of the predictor set (Fig. 1c). We use those probability distributions to estimate some dependence or predictability measure (like CoD, entropy or mutual information) to evaluate the strength of connection $SC(g_1, g_2 \rightarrow g_3)$ from a predictor gene set g_1, g_2 to a target gene g_3 (Fig. 1b).

The network construction method starts with a *seed* subset of genes, which are known to be involved in the phenomenon under study, as the initial gene layer (initial network), and adds to the growing network, at each successive step, a new gene layer formed by the genes most significantly connected to the genes in the previous layer.[2] At each network growing step, we compute the strength of connection "*from*" the previous layer G to each candidate[3] gene h, $SC(G \rightarrow h)$, and the strength of connection from each candidate gene h "*to*" the previous layer G, $SC(h \rightarrow G)$ (Fig. 1d). Candidate genes are ranked by their *overall* strengths of connection with the previous layer G, $SC(h \leftrightarrow G) = \max\{SC(h \rightarrow G), SC(G \rightarrow h)\}$ (upper Fig. 1e indicates this ranking). Candidate genes ranked over the predictability threshold T_{p_l} are automatically included in the next layer, while candidate genes between thresholds T_{p_l} and $T_{p_{min}}$ are included in the next layer *only* if they belong to the auxiliary subset of genes S_F known (or plausibly supposed) to be related to the phenomenon under study (lower Fig. 1e indicates gene selection for the next layer). The most recently added gene layer will be considered the previous layer G in the next growing step. This process is iterated a number of times until some stop condition is reached (in Fig. 1f each color indicates a different gene layer).

3. Pattern-oriented process specification

3.1. Pattern specification

A pattern describes a generic solution to an existing problem in a more readable and accessible form [42]. Patterns capture proven solutions to real problems and generalize these solutions so that they can be reused in similar contexts. Historically, patterns emerged as a discipline in Computer Science somehow influenced by the pioneer work of Christopher Alexander, a professor of architecture at the University of California at Berkely, who first wrote a series of books cataloging a number of architectural patterns and describing their application [43–45]. Patterns in Computer Science have been widely adopted in the different phases of system development, e.g., [46–49].

A process can be specified using different (standard) modeling notations. In the context of this work, we have adopted the Business Process Modeling Notation (BPMN) [50]. BPMN is a process modeling

[1] Sometimes this matrix is presented in its transposed form.

[2] The concept of *layer* presented here is inherent to the construction method. The initial gene layer depends on which set of seed genes we start with and the following layers will depend on the information obtained from the data and previous biological knowledge. Seed genes selection is carried out based on actual biological knowledge, which can be obtained from ontologies and/or functional annotation databases, such as Gene Ontology (GO) [39], KEGG [40] and REACTOME [41].

[3] *Candidate genes* are all genes not already included in the growing network.