Bioinformatics, 31(5), 2015, 785–787
doi: 10.1093/bioinformatics/btu694
Advance Access Publication Date: 24 October 2014
Applications Note

OXFORD

Systems biology

MP-GeneticSynth: inferring biological network regulations from time series

Alberto Castellini[1,2,3,*], Daniele Paltrinieri[4] and Vincenzo Manca[1,4]

[1]Center for BioMedical Computing, Verona University, 37134 Verona, Italy, [2]Max Planck Institute for Molecular Plant Physiology-Syst Bio & Math Modelling Group, 14476, Germany, [3]University of Potsdam, Institute for Biochemistry and Biology - Bioinformatics Group, 14476, Germany and [4]Department of Computer Science, Verona University, 37134 Verona, Italy

*To whom correspondence should be addressed.
Associate Editor: Alfonso Valencia

Received on January 27, 2014; revised on July 12, 2014; accepted on October 17, 2014

Abstract

Summary: MP-GeneticSynth is a Java tool for discovering the logic and regulation mechanisms responsible for observed biological dynamics in terms of finite difference recurrent equations. The software makes use of: (i) metabolic P systems as a modeling framework, (ii) an evolutionary approach to discover *flux regulation functions* as linear combinations of given primitive functions, (iii) a suitable reformulation of the least squares method to estimate function parameters considering simultaneously all the reactions involved in complex dynamics. The tool is available as a plugin for the virtual laboratory MetaPlab. It has graphical and interactive interfaces for data preparation, a priori knowledge integration, and flux regulator analysis.

Availability and implemention: Source code, binaries, documentation (including quick start guide and videos) and case studies are freely available at http://mplab.sci.univr.it/plugins/mpgs/index.html.

Contact: castellini@mpimp-golm.mpg.de

1 Introduction

Biological system modeling is a central methodology for unraveling complex interactions in molecular species and understanding the internal logic of metabolic, regulatory and signaling processes. Mathematical and computational models are used to formally represent knowledge of biological processes, which is usually acquired from experiments. A key goal is represented by the automatic inference of model structure and parameters from observed data. Several reverse-engineering methodologies and related software were proposed to solve this problem. Simulated annealing, particle swarm intelligence, genetic algorithms (GA) and other optimization techniques were used with both differential (Cho *et al.*, 2006; Goel *et al.*, 2008; Gonzalez *et al.*, 2007) and unconventional models (Besozzi *et al.*, 2009; Cao *et al.*, 2010; Manca and Marchetti, 2012).

Here we present *MP-GeneticSynth*, a tool based on GA and multivariate regression for the synthesis and the analysis of flux regulation functions from observed time series within the modeling framework of *Metabolic P systems* (*MP systems*) (Manca, 2013). The evolutionary approach on which the software is based provides new perspectives on the analysis of biological network regulations. Main applications range from metabolic to gene expression and signaling networks (Bollig-Fischer *et al.*, 2014), although the methodology lends itself to be applied also to other contexts where fluxes of matter or information have to be inferred from time series. The main strengths of the proposed tool are the possibility to integrate observed data with a priori knowledge, and to generate models even if exact reaction rates or parts of the stoichiometry are unknown.

2 Methods

MP systems are a deterministic and time-discrete modeling framework based on multiset rewriting grammars for computing the dynamics of metabolic phenomena. An *MP model* (also called *MP grammar*) is defined by: (i) a set of variables (called *substances*), a set of rewriting rules (called *reactions*) and a set of initial values for substances. Rewriting rules $\alpha \rightarrow \beta$ transform multisets of

variables α, β and are regulated by *flux regulation functions* which depend on the state of the system (i.e. instantaneous values of substances). Using a matrix-based representation of an MP grammar, the temporal evolution of its substances is described by a first-order recurrent system of finite difference equations (Manca, 2013).

Discovering sets of regulation functions able to regenerate an observed dynamics is a key problem in MP modeling which generalizes the parameter estimation problem described in section 1. In Castellini *et al.* (2013) an evolutionary approach for this problem is proposed, wherein regulation functions are assembled from a predefined dictionary of *primitive functions*, also called *regressors*. GA (Mitchell, 1998) act as primitive function selectors and use three main 'environmental pressures' to shape flux regulation functions: (i) simulation error, (ii) function complexity, in terms of number of primitive functions, (iii) biological soundness of computed fluxes (i.e. magnitude, relative variations, etc.). The tradeoff between model accuracy and complexity is optimized by a procedure called *w-adaptation algorithm* (Castellini *et al.*, 2014) which also prevents overfitting.

Multiple linear regression is used to compute the parameters of selected primitive components. A non-trivial matricial reformulation (Manca, 2013) of the regression problem enables to estimate simultaneously the parameters of every regulation function by means of least squares estimation. Pseudoinverse computation is a key in this context since it enables further simplification of the model. MP-GeneticSynth uses a method based on singular value decomposition (SVD) which achieves good performance.

3 Main features and functionalities

MP-GeneticSynth (Fig. 1) has four main functionalities summarized in the following. Further information, case studies and details about graphical interface elements are reported in the website.

3.1 Execution of new experiments

Experiments represent the generation of regulation functions for a system under investigation. They produce analytical results and log files, and involve four steps: (i) *data preparation:* substance time series are sampled and selected according to correlation values, and training/validation sets are defined; (ii) selection *of primitive functions:* a dictionary of primitive functions (e.g. first/second degree terms, Hill functions) is defined; (iii) a priori *knowledge definition:* the probability of each primitive function can be defined for each regulation function; (iv) *optimization parameters:* GA and w-adaptation algorithm parameters are set.

3.2 Continuation of experiments

This functionality enables the user to load an experiment file and to continue the optimization process from the last generation.

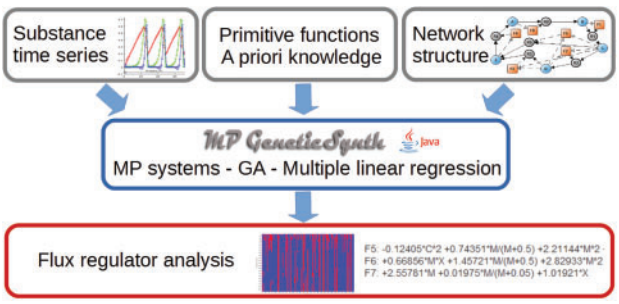

Fig. 1. MP-GeneticSynth workflow: input, output and methods

Some parameters, such as crossover and mutation rates, can be modified.

3.3 Analysis of single experiments

Experiment files can be individually analyzed by the *ExpAnalyzer* (see screenshots in tutorials and main guide), a graphical tool showing complete information about each generation of the evolutionary process, with specific focus on the best solution. A global view displays the evolution of simulation error and number of regressors, and enables the user to select specific generations by a slider. The local view is updated accordingly, showing regulation functions, substance/flux simulated dynamics and other measures. A heatmap shows the usage rate of each primitive function for each flux regulation function, and enables to relate these values to flux accuracy. GA individuals and related distances/statistics can be visualized and exported.

3.4 Comparison of solutions

Solutions coming from different experiments can be compared by a graphical tool called *ExpComparator* (see screenshots in the main guide). It enables to select sets of solutions according to error values and number of regressors used, and to identify, by graphical and statistical parameters, common sets of flux regulators.

4 Implementation and case studies

MP-GeneticSynth is released as a plug-in for MetaPlab 1.3 (http://mplab.sci.univr.it/). The code is written in Java (JDK 6 and it employs the following open source libraries: *ejml-0.23* (https://code.google.com/p/efficient-java-matrix-library/) for matrix computation, *jep-2.4.1* (http://sourceforge.net/projects/jep/) for function parsing, *jfreechart-1.0.14* (http://www.jfree.org/jfreechart/) for chart visualization.

The plugin is supported on Linux, Windows and Mac OS and needs only the Java Runtime Environment (JRE 6u45 or later) to be used. Step-by-step instructions for installing MetaPlab 1.3 and running the plugin are available in the MP-GeneticSynth website (see section *Availability* in the abstract). This simple process is explained in the Quick Start Guide. The website provides also four case studies, namely, the mitotic oscillator in early amphibian embryos (with clean and noisy time series), the predator-prey system, the chaotic logistic map and Vega, a synthetic model with very complex dynamics. Textual and video tutorials explain how to load and analyze case study experiments.

The time required to generate regulation functions depends on: the number of substances n, the number of reactions m, the number of primitive functions d, the length of time series t, the number of individuals of the GA population N and the number of generations g. The main bottlenecks are the computation of the SVD, whose complexity is $O((n \cdot t)^3)$, and the computation of the simulation error, whose complexity is $O(k \cdot n \cdot m \cdot t)$, where constant k is large because the computation of flux values requires *jep* function parsing. These operations are performed for each individual of the population and for each GA generation (Castellini *et al.*, 2013). On a laptop with 8 GB of ram and a processor Intel(R) Core(TM) i5-3340 M 2.70 GHz about 30 min were required to generate regulation functions for a system having 6 substances, 10 reactions and 100 observations, performing 10 000 generations.

Future developments of this project mainly concern: (i) use of regularization methods for regression, (ii) code parallelization,

(iii) improvement of constraints for biological soundness of inferred fluxes, (iv) application to new biological systems.

Funding

The first author was financially supported by CBMC (Center for Biomedical Computing), University of Verona.

Conflict of interest: none declared.

References

Besozzi,D. *et al*. (2009) A comparison of genetic algorithms and particle swarm optimization for parameter estimation in stochastic biochemical systems. *LNCS*, **5483**, 116–127.

Bollig-Fischer,A. *et al*. (2014) Modeling time-dependent transcription effects of HER2 oncogene and discovery of a role for E2F2 in breast cancer cell-matrix adhesion. *Bioinformatics*, **30**, 3036–3043.

Cao,H. *et al*. (2010) Evolving cell models for systems and synthetic biology. *Syst. Synth. Biol.*, **4**, 55–84.

Castellini,A. *et al*. (2013) From time series to biological network regulations: an evolutionary approach. *Molecular BioSystems*, **9**, 225–233.

Castellini,A. *et al*. (2014) An evolutionary procedure for inferring MP systems regulation functions of biological networks, *Nat. Comput*. doi: 10.1007/s11047-014-9421-1.

Cho,D.Y. *et al*. (2006) Identification of biochemical networks by S-tree based genetic programming. *Bioinformatics*, **22**, 1631–1640.

Goel,G. *et al*. (2008) System estimation from metabolic time-series data. *Bioinformatics*, **24**, 2505–2511.

Gonzalez,O.R. *et al*. (2007) Parameter estimation using simulated annealing for S-system models of biochemical networks. *Bioinformatics*, **23**, 480–486.

Manca,V. (2013) *Infobiotics: Information in Biotic Systems*. Springer, Berlin.

Manca,V. and Marchetti,L. (2012) Solving dynamical inverse problems by means of metabolic P systems. *BioSystems*, **109**, 78–86.

Mitchell,M. (1998) *An Introduction to Genetic Algorithms*. MIT Press, Cambridge.