Future Generation Computer Systems 63 (2016) 1–14

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

MultiGrain/MAPPER: A distributed multiscale computing approach to modeling and simulating gene regulation networks

Alexandru E. Mizeranschi [a,*], Martin T. Swain [b], Raluca Scona [a], Quentin Fazilleau [a], Bartosz Bosak [c], Tomasz Piontek [c], Piotr Kopta [c], Paul Thompson [a], Werner Dubitzky [a,*]

[a] University of Ulster, Coleraine, UK

[b] Aberystwyth University, Aberystwyth, UK

[c] Poznań Supercomputing and Networking Center, Poznań, Poland

HIGHLIGHTS

- This paper presents MultiGrain/MAPPER—a novel concept, framework and tool for modeling and simulation based on a multiscale computing paradigm.
- MultiGrain/MAPPER has been designed to tackle the computational challenges of large-scale gene-regulatory networks (GRN) model modeling and simulation tasks.
- In particular, MultiGrain/MAPPER realizes a distributed computing solution to reverse-engineering of GRN models from gene-expression data.
- The solution is based on a distributed multi-swarm (multi-island) particle swarm optimization algorithm we implemented, where PSO islands are mapped to CPU cores.
- A detailed evaluation of MultiGrain/MAPPER's concepts and performance is provided in the paper, with a particular emphasis on the tool's computational aspects.

ARTICLE INFO

Article history:
Received 28 July 2015
Received in revised form
15 February 2016
Accepted 2 April 2016
Available online 12 April 2016

Keywords:
Gene-regulatory networks
Reverse-engineering of gene-regulation models
Distributed multiscale computing

ABSTRACT

Modeling and simulation of gene-regulatory networks (GRNs) has become an important aspect of modern systems biology investigations into mechanisms underlying gene regulation. A key task in this area is the automated inference or *reverse-engineering* of *dynamic mechanistic GRN models* from gene expression time-course data. Besides a lack of suitable data (in particular multi-condition data from the same system), one of the key challenges of this task is the *computational* complexity involved. The more genes in the GRN system and the more parameters a GRN model has, the higher the computational load. The computational challenge is likely to increase substantially in the near future when we tackle larger GRN systems. The goal of this study was to develop a distributed computing framework and system for reverse-engineering of GRN models. We present the resulting software called MultiGrain/MAPPER. This software is based on a new architecture and tools supporting multiscale computing in a distributed computing environment. A key feature of MultiGrain/MAPPER is the realization of GRN reverse-engineering based on the underlying distributed computing framework and multi-swarm particle swarm optimization. We demonstrate some of the features of MultiGrain/MAPPER and evaluate its performance using both real and artificial gene expression data.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Many complex phenomena occur across multiple spatial and/or temporal scales. Such phenomena are difficult to model and simulate within a single, monolithic approach. Multiscale modeling and simulation adopts a divide-and-conquer philosophy that solves a complex problem by decomposing it into a set of simpler scale-specific sub-models and combining these into a global integrated model, called a multi-scale model [1]. A central modeling task in multiscale modeling is how information specific to one scale-specific sub-model is transformed to another. This information transformation is called *scale-bridging*. A computing challenge in multiscale modeling and simulation is the coupling

* Corresponding authors.
E-mail addresses: alex.mizeranschi@gmail.com (A.E. Mizeranschi), w.dubitzky@ulster.ac.uk (W. Dubitzky).

http://dx.doi.org/10.1016/j.future.2016.04.002
0167-739X/© 2016 Elsevier B.V. All rights reserved.

and coordinated execution of multiple codes representing the submodels of a multiscale model. We refer to this kind of computing as *multiscale computing*. This article presents a comprehensive computing framework and tool called MultiGrain/MAPPER. This approach has been designed for modeling and simulation of large gene-regulatory networks. The tool was developed as part of the European FP7 project Multiscale Applications on European e-Infrastructures (MAPPER) [2–5]. The goal of MAPPER was to develop a general framework and technology facilitating the development, deployment and execution of *distributed* multiscale modeling and simulation applications [1,6].

Based on tools and services developed in the MAPPER project, MultiGrain/MAPPER realizes a gene-regulation model reverse-engineering process based on a multi-swarm *particle swarm optimization* (*PSO*) algorithm. In this approach, the overall particle swarm is partitioned into multiple sub-swarms (which assume the role of sub-models in the MAPPER framework), which are then mapped onto the processor cores of a (potentially distributed) computing resource. The rationale behind this approach is threefold: First, we expect that the multi-swarm PSO approach is less prone to converge to suboptimal solutions. Second, by casting the reverse-engineering problem into the multiscale modeling and simulation framework of MAPPER, we will be able to develop future versions of MultiGrain/MAPPER which will decompose very large gene-regulation networks into modules of sub-networks and synthesize these into a coupled multi-network overall solution. And third, by building this approach on the distributed multiscale computing capabilities of MAPPER, we should be able to process large gene-regulatory network problems by taking advantage of computing resources in distributed computing environments.

The current study focuses on the third aspect. In particular, we are interested in how well the computational performance behaves under different problem and computing environment configurations. We have evaluated the performance of MultiGrain/MAPPER based on data from real biological and artificial gene-regulatory networks.

The remaining part of this article is organized as follows: Section 2 recapitulates the basic biological aspects of gene-regulatory networks and the main“ingredients” of reverse-engineering gene-regulation models from gene-expression data. In Section 3.1, we review some of the state-of-the-art tools used to address the reverse-engineering task. Section 3 presents our approach, including a description of our multi-swarm PSO algorithm and its implementation in MultiGrain/MAPPER. This is followed by Section 4, in which we describe and discuss our performance evaluation experiments and their results. Finally, Section 5 provides some concluding remarks and a brief look at future developments in this area.

MultiGrain/MAPPER software resources are available from this repository: https://apps.man.poznan.pl/svn/sbml-toolbox/GRNApplication/.

2. Reverse-engineering GRN models

Cells regulate the expression of their genes to create functional gene products (RNA, proteins) from the information stored in genes (DNA). Gene regulation is a complex process involving the transcription of genetic information from DNA to RNA, the translation of RNA information to make protein, and the post-translational modification of proteins. Gene regulation is essential for life as it allows an organism to respond to changes in the environment by making the required amount of the right type of protein when needed. Complex gene-regulatory processes are coordinated by multiple genes, whose mutual influences are organized as a *gene-regulation network* (*GRN*). Developing *quantitative models of gene regulation* is essential to guide our understanding of complex gene-regulatory processes. For instance, understanding gene-regulatory processes in the context of diseases is increasingly important for the development of treatment and prevention strategies. Automated reverse-engineering of dynamic mechanistic GRN *models* from gene-expression time-series data is becoming an area of growing interest in systems biology research [7–11].

Reverse-engineering quantitative dynamic mechanistic GRN models with accurate structure and high predictive performance is a long-standing problem [9]. Currently, some of the main challenges include:

- A lack of sufficient amounts of relevant gene expression time-course data. (1) While the number of sampling points is important (typically, 10–50 time points are measured), far more important is to have multiple stimulus–response datasets from the same system under different stimuli [7]. This is a challenging requirement for current experimental practice. (2) GRN stimulus–response data from GRN systems with experimentally confirmed GRN regulatory structure. A good positive example is the study of Cantone and colleagues [12].
- A lack of reverse-engineering algorithms and methods that are able to incorporate existing biological knowledge effectively.
- A lack of algorithms and tools whose computations scale well when the number of genes in the GRN system is increased. Currently, most algorithms and tools are applied to systems involving only 5- to 10 genes.
- Systematic validation studies similar to the work by Cantone et al. [12] that evaluate all aspects (predictive and structure inference aspects) of the algorithm [12].

Reverse-engineering dynamic mechanistic GRN models from gene-expression time-series data involve the following main “ingredients”: *data*, *model*, *simulation* and *reverse-engineering algorithm*.

1. **Data**. We reverse-engineer a GRN model from gene-expression *data* that contains measurements over a set of consecutive time points. Thus, a GRN system with n genes corresponds to a dataset with $i = 1, \dots, n$ time series, each representing the mRNA abundance of gene i at time point $k = 1, \dots, m$.
2. **Model**. A system model (short: *model*) is a mathematical specification that *represents* the genes of a GRN system and their mechanistic regulatory relationships in terms of *variables* and *parameters*. A model variable represents the time-*variant* mRNA abundance expressed by a gene, and a model parameter represents time-*invariant* biological and experimental conditions of the GRN system (e.g. the maximal expression rate of a gene).
3. **Simulation**. Based on the initial values of all variables, called the *initial condition*, a system simulation (short: *simulation*) computes the variables' response to the initial condition over a specified time interval.
4. **Algorithm**. We use a *reverse-engineering algorithm* to create (infer) a concrete GRN model from gene-expression data, by iteratively generating candidate models until we have found one that meets certain quality criteria.

The quality of a GRN model depends on two factors: the model's *explanatory power* (or model completeness) and the model's *predictive power* (or model correctness).

A model's explanatory power depends on how well the elements of a mathematical model specification correspond to the salient features of the modeled system. A model's predictive power is estimated by simulating the system's response to the initial condition captured in an independent validation dataset [13]. The greater the deviation (error) between the response time courses predicted by the model and the actual time courses in the validation data, the lower the predictive power of the model.

Download English Version:

https://daneshyari.com/en/article/424504

Download Persian Version:

https://daneshyari.com/article/424504

Daneshyari.com