BioNessie$_G$ - A Grid Enabled Biochemical Networks Simulation Environment

Xuan Liu[1], Jipu Jiang[2], Oluwafemi Ajayi[2], Xu Gu[1], David Gilbert[1], Richard Sinnott[2]

[1] Bioinformatics Research Centre,
University of Glasgow,
Glasgow, G12 8QQ, Scotland, UK
{xliu, gux, drg}@dcs.gla.ac.uk
[2] National e-Science Centre,
University of Glasgow,
Glasgow, G12 8QQ, Scotland, UK
{j.jiang, o.ajayi, r.sinnott}@nesc.gla.ac.uk

Abstract. The simulation of biochemical networks provides insight and understanding about the underlying biochemical processes and pathways used by cells and organisms. BioNessie is a biochemical network simulator which has been developed at the University of Glasgow. This paper describes the simulator and focuses in particular on how it has been extended to benefit from a wide variety of high performance compute resources across the UK through Grid technologies to support larger scale simulations.

1 Introduction

Simulation and modelling are essential techniques for understanding complex biochemical processes. These techniques necessarily require the construction of models of the underlying biochemical processes [19]. To support this, software tools to both develop and analyse models are crucial to shed insight on the underlying biochemical processes which take place.

Reasoning about biochemical processes can often be a computationally expensive activity, e.g. when multiple biochemical factors are being considered or a variety of complex networks are being simulated simultaneously. In such cases, access to and use of large scale high-performance computing (HPC) resources are essential. The Grid provides one paradigm by which large scale HPC resources can be seamlessly accessed by end users. Past history has shown however [33,34] that end users such as bioinformatians and biochemists often do not wish to understand the underlying technologies associated with Grids, e.g. the X509 certificates and their use to support Public Key Infrastructures. Instead they are more concerned (naturally so!) with their own research. Key to this is making access to and usage of heterogeneous Grid resources transparent to the end user scientists.

In this paper we present a biochemical simulation tool developed as part of the Department of Trade and Industry funded *BPS: A Software Tool for the Simulation and Analysis of Biochemical Networks* project. We introduce the background to this tool and the approach that has been taken for biochemical network simulation. We show how this tool has been extended to make use of a variety of heterogeneous computational resources through the Grid.

1.1 Related Work

Several tools have been implemented to simulate biochemical network processes. The most popular tools include COPASI [19], GEPASI [26], CellDesigner [16], and SimBiology [1]. Most of these tools provide specific functionalities, *e.g.* constructing and simulation of biochemical networks [29]; parameter scanning where a simulation is run several times and a parameter is changed for each run [31]; model parameter estimation [27], and parameter sensitivity analysis [21] of the biochemical networks.

Mathematical modelling and parameter estimation require a set of parameters (kinetic constants and physiological concentration of the cellular components), however, there are no experimental methods to measure all such parameters with accuracy. The estimation of biological parameters based on any observable wet-lab data is essential for construction of the model. Moreover, this estimation process often requires huge computer resources which one organisation cannot afford [22]. Since parameter scanning usually involve running the same simulation many times, the whole process can be very time-consuming. The user can benefit if the parameter estimation and parameter scanning phases can be parallelised by utilising HPC resources.

Boys et. al [12] proposed a computational GRID facility - BASIS [2] designed for inferring parameters of biochemical models using Bayesian methods. The user submits the model in SBML [20] format to BASIS and retrieves the simulation results using standard web-service protocols. Kimura et.al [22] developed OBIYagns (yet another gene network simulator) which is a biochemical system simulator that comprises a multiple-user web-service graphical interface, an ordinary differential equation(ODE) solver and a Genetic Algorithm(GA) based parameter estimators distributed over an open bioinformatics grid (OBIGrid) [3].

Therefore, there is demand for a web based biochemical software tool which can directly communicate with web-services, and then is convenient for the users to be able to retrieve, manage, edit and analyse the biochemical models and experimental results from a range of HPC resources. Grid Cellware - a grid-based modelling and simulation tool, has been developed by the Systems Biology Group of the Bioinformatics Institute in Singapore [14]. Grid Cellware incorporates the Swarm algorithm for parameter estimation jobs. The Swarm algorithm is based on simulation of social behaviour where each individual in a swarm adjusts its flight according to its own flying experience as well as that of its companions' flying experience [30].

Here we present a new software tool -BioNessie$_G$: *A Grid Enabled Biochemical Networks Simulation Environment* [4] which has all of the above standard methods for processing the biochemical network models and offers in addition grid enabled parameter estimation and parameter scanning.

1.2 Background to Biochemical Pathways

A biochemical system is a non-linear and complex network, where interactions of different pathways and dynamics of information processing within the pathway produce a multitude of biological outputs. Biochemical pathways take a variety of forms including metabolic pathways, signalling pathways and gene transcriptional pathways. *Metabolic pathways* exist within the cell, which can be viewed and modelled in terms of complex networks of chemical reactions catalysed by enzymes; *Signalling pathways* describe the movement of signals within or across the cell, which contains a sequence of phosphorylation or enzyme-related activities; *Gene regulatory pathways* control and regulate the process of gene expression in nucleus in response to enzymatic signals. Living organisms rely on the interaction of those pathways to coordinate cellular activities and accommodate to environmental changes, for example, via modulating kinase activities associated with cell signalling, to control the metabolism of cellular substances and energy, to implement cellular functions (e.g. muscle contraction), and to modulate cellular cycle into a state of cell division or cell proliferation by modulating gene expression.

Biochemical pathways derived through scientific experiments and data analysis, not only capture the current knowledge of biological systems but also serve as a local point to integrate other diverse related information, such as literature citations, research notes and experimental data [32]. The study of these pathways has become an important theme in Systems Biology, which benefits post-genomics research and also has a significant influence on industrial applications.

2 Modelling and Simulation

As most biochemical pathways of interest involve components connected through interlocking loops, an intuitive understanding of their dynamics is hard to obtain. Formal methods and software tools for the modelling and analysis of biochemical pathways are indispensable, both for gaining better comprehension and obtaining experimentally testable predictions. As the body of data about biological systems is rapidly accumulating, many groups have been prompted to develop models for representing and analysing these systems. A *Model* is an approximate representation of a biochemical system, used to study, design or control a biological system. It usually incorporates known quantities, including the number or amount of the components, rates at which these components interact, together with physical laws that govern the reactions. *Simulation* is the process of exploring and analysing the information contained in the model, often through

the description of the dynamics of the system, which provides the abstraction needed for grasping knowledge of biochemical systems.

2.1 ODE-based Modelling Method

ODEs (Ordinary Differential Equations) are a commonly used modelling technique for modelling continuous systems with a large number of molecules. This approach is concerned with constructing a set of rate equations to describe reactions of biochemical pathways with concentrations of chemical species as variables.

A basic biochemical reaction is the *decay* of substance A to substance B at rate k, depicted by the simple mass-action equation $A \xrightarrow{k} B$. This can be described in differential equations as follows:-

$$\frac{d[A]}{dt} = -\frac{d[B]}{dt} = -k * [A] \tag{1}$$

where $[A]$ stands for the concentration of substance A.

Biochemical reactions are often catalysed by enzymes (which are not consumed in the process). We extend our example above with the following mass-action description where $A|E$ is the substrate-enzyme complex and E is the enzyme.

$$A + E \underset{k_2}{\overset{k_1}{\rightleftarrows}} A|E \xrightarrow{k_3} B + E \tag{2}$$

This reaction can be described by the following four differential equations:

$$\begin{aligned}
\tfrac{d[A]}{dt} &= \quad -k_1 \times [A] \times [E] + k_2 \times [A|E] \\
\tfrac{d[A|E]}{dt} &= k_1 \times [A] \times [E] \quad - k_2 \times [A|E] - k_3 \times [A|E] \\
\tfrac{d[B]}{dt} &= \qquad\qquad\qquad\qquad + k_3 \times [A|E] \\
\tfrac{d[E]}{dt} &= \quad -k_1 \times [A] \times [E] + k_2 \times [A|E] + k_3 \times [A|E]
\end{aligned} \tag{3}$$

The Michaelis-Menten equation is often used in practice to model enzymatic reactions, and is given by

$$V = V_{\max} \times \frac{[A]}{K_M + [A]} \ , \ k_{\text{cat}} = k_3 = \frac{V_{\max}}{[E_T]} \ , \ K_M = \frac{k_2 + k_3}{k_1} \tag{4}$$

where $[E_T]$ is the total enzyme concentration, V is the reaction velocity, $V_{\max}$ is the maximum reaction velocity, and K_M the *Michaelis constant* is the concentration of the substrate at which the reaction rate is half its maximum value. The Michaelis-Menten equation makes the explicit assumption that there is no product present, and only holds for the initial rate of the enzymatic reaction.

However, the advantage of using this forumulation is that it enables a single differential equation to be used to describe the system above:

$$\frac{d[A]}{dt} = -\frac{d[B]}{dt} = -kcat \times [E_T] \times \frac{[A]}{(K_M + [A])} \tag{5}$$

The differential equations in equations 3 and 5 can be solved using a variety of numerical integration techniques.

A *biochemical network* is described by several biochemical equations, and thus represented by a set of ordinary differential equations in our model. The number of equations and parameters will determine the complexity of the numerical solution, and the speed of the solution can be also be adversely affected by the degree of 'stiffness' in the system. Thus the next step when considering an ODE model will be to apply an appropriate analysis method to extract and interpret information from the model, where computer-aided techniques are necessary due to the nonlinear nature of the differential equations. Simulation is often the first choice towards the end. Methods including parameter scans and sensitivity analysis are equally important in understanding the dynamics of biochemical systems.

2.2 Simulation

Simulation permits the exploration and understanding of the behaviour of models [15]. It is a technique well suited to studying complex system of equations where closed-form analytical solutions cannot be obtained by hand. In a simulation environment, equations are evaluated under different parameter settings. Implementation of these equations provides a first intuitive impression of how these equations relate to dynamic responses. There might be many circumstances where simulation is necessary. Neelamkavil [28] enumerated some of these, for example, when there is a need to study the past, present, or future behaviour of the system in real time. This might include side-effects of new drugs when expected accuracy of simulation results is consistent with the requirements of the particular problem, e.g. accuracy of radiation dosage for treating cancer patients is critical compared to the accuracy of forecasts on sample species. Moreover, computational simulation is necessary when there is a need to investigate the sensitivity of biological systems to parameter changes such as environmental noise and minor mutations. Kitano [23] points out that a number of networks for biological oscillations and transcriptional regulations have been shown to be tolerant against noise, but only computer simulation could have shown the degree to which the gene regulatory networks for segmentation during *Drosophila* embryogenesis remain robust over a large range of kinetic parameters.

3 BioNessie

BioNessie is a platform-independent environment for modelling biochemical networks, and simulating and analysing the dynamic behaviour of biochemical models.

BioNessie uses a modular architecture that allows the developer to easily plug-in various components and update them. It is developed using Java technology and can run on many platforms that support JRE (Java Runtime Environment 1.5.x or higher).

BioNessie provides a full user-friendly Graphical User Interface (GUI) which allows the user to import, create, edit and export the biochemical models conforming to the SBML standard [20]. LibSBML [5] has been used to read, write, manipulate, translate, and validate SBML files and data streams. The unique Concurrent Versions System (CVS) design helps users to keep track of the version history of their SBML models during construction and subsequent modification.

The core of BioNessie comprises the SBML ODE Solver Library (SOSlib) [25]. This provides a programming library for symbolic and numerical analysis of a system of ODEs derived from a chemical reaction network encoded in SBML format. SOSlib employs LibSBML's Abstract Syntax Tree (AST) for formula representation to construct ODE systems, their Jacobian matrix and other derivatives. SUNDIALS' version of CVODE [17] is incorporated for numerical integration and sensitivity analysis of stiff and non-stiff ODE systems. BioNessie can generate the changes of species amounts and parameter values over time by simulating the SBML model numerically with SOSlib. The simulation results can be generated in two ways: plots and report text files.

BioNessie is not only a editor and simulator, but also an analyser. It supports *parameter scans, sensitivity analysis* and *parameter estimation* for biochemical networks. These methods are commonly used approaches for understanding the dynamics of a biochemical model.

Parameter scanning permits the exploration of a model's behaviour over different ranges of parameter values. This is achieved by running simulations for specific values in the range.

Sensitivity analysis is the study of the response of system variables to changes in parameter values, which can be used to establish the contribution of individual parameter values to the overall performance of a complex biochemical system. Without loss of generality, the sensitivity gain can be written (for finite changes δ) as

$$S_P^M = \frac{\delta M/M}{\delta P/P} \tag{6}$$

where P represents the parameter that may be varied and M the response of the overall system [24].

Parameter estimation is a method used to perform an estimation of biological parameters based on any observable wet-lab data. We use a Genetic Algorithm [18] to estimate the parameters by searching different rate constant sets in a predefined range to minimise the difference between the time-course data (obtained from wet lab) and simulation results of the model.

4 BioNessie$_G$ - A Grid Enabled BioNessie

4.1 Why Grid?

Parameter scanning and parameter estimation are quite time-consuming activities. In the parameter scanning process, BioNessie has to perform simulations many times. For example, Raf-1 and RKIP are both proteins in RKIP pathway [13], and react with rate k_1. If we want to carry out the parameter scan on some of variables in this reaction, e.g. the initial concentrations for Raf-1 and RKIP and the rate k_1, then the simulations will be performed on all the possible combinations of values for these three parameters. Suppose each value will be changed over a range of 100 values, resulting in a total of $100^3 = 1000000$ different combinations for which the simulation is being run. The whole scanning process took 4 hours to be accomplished on a computer with an AMD Athlon 64 4800+ X2 Dual Core CPU and 2GB RAM. However, all the simulations can be run in parallel allowing for considerable speed up.

The same situation also applies to the parameter estimation. Due to their trivially parallel nature, large scale parameter scans and parameter estimation are well suited to execution on HPC cluster resources such as National Grid Service (NGS - www.ngs.ac.uk), ScotGrid (www.scotgrid.ac.uk) and Condor pools. We recognise that scientists do not wish to deal with the issues of access to and usage of these clusters directly or through dealing with the complexities of individual job schedulers/job submission systems associated with these clusters.

Grid computing provides an infrastructure in which it is possible to share and aggregate computing resources from many diverse sources and view them as one virtual resource. These resources can be CPU, memory and storage amongst other things.

4.2 BioNessie$_G$ Architecture - An Overview

BioNessie has a modular architecture and extensions can be easily implemented by pluging a module into the BioNessie application. The client side of BioNessie$_G$ has been designed as a graphical user interface for specifying the model and parameters while the Grid side is the computational engine which is fully Grid enabled. The client side has cross-platform compatibility because Java has been chosen as the software developing language.

For the Grid side, the computational engine has been implemented on different Grid resources including the NGS, ScotGrid and Condor pools as shown in Figure 1. The system has been designed to be extensible so that other resources can be integrated seamlessly.

ScotGrid is one of the four CERN-LHC Tier-2 [6] centres in the UK. It has 140 worker nodes of dual core CPU Opteron 2.6 GHz with 8GB RAM, more than 1 Million Spec Int 2000 compute capacity and 100TB of storage. NGS Leeds and NGS Oxford have 2 x 64 compute nodes of dual 3.06 GHz Xeon with 2GB RAM and 2 x 120 GB of storage. The NESC Condor pool has 20 compute nodes of Intel CPU 2.20GHz with approximately 300GB of storage. Each of these resource

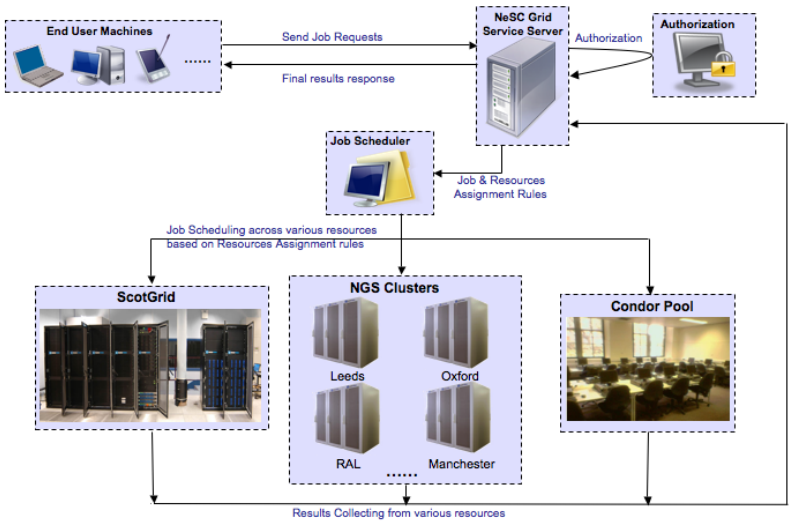


Fig. 1. Grid Architecture

headnodes have been configured with SOSlib, LibSBML and SUNDIALS, as well as to accept inputs encoded in SBML formats.

The communication between the Grid resources and the client side of BioNessie$_G$ was implemented through web services. The web services were developed using Apache Axis 2 [7] and Globus Toolkit 4.0 [8]. The job scheduler was also developed by them which evaluates the suitability and availability of Grid resources before submitting the job to appropriate computer resources.

So, a typical scenario for a BioNessie$_G$ end user is as follows: the end user wishes to submit a batch of parameter scans (jobs) to the Grid service through the GUI interface. BioNessie$_G$ then divides the parameter scan into several jobs (simulations), each job being represented by a SBML file and saved in a local directory folder BioNessie$_G$. The web service then checks if the authority of the client who sent the job has sufficient privileges, and if this is the case the jobs are decompressed and distributed to the available resources by the job scheduler. Each job is then executed and the corresponding results are sent back to the web service in text file format, which are compressed there and sent back to the BioNessie$_G$ client. BioNessie$_G$ decompresses the result file and merges all results file to a single XML scanning result file which can be recognised by BioNessie$_G$. Finally, BioNessie$_G$ generates a report either as a plot or text file.

4.3 Two Key Issues in BioNessie$_G$

Two key parts of this architecture are the authorisation service and the job scheduler.

The approach we have taken in the authorisation service is to have users register only once for usage of BioNessie$_G$. The back-end Grid operations such as creation of proxy certificates defining the security context through which jobs can be submitted to HPC resources is hidden from the end users.

With this model, all authorised BioNessie$_G$ users can share the same security context, which avoids each BioNessie$_G$ user having to individually register for their own Grid certificate and for accounts on the HPC resources. This simplifies the end user experience greatly. However having only one security context implies that all jobs being executed on NGS resources may appear as one Grid user. In other projects we are providing ways in which pooled accounts can be used by specific virtual organisations exploiting technologies such as MyPROXY [10] and VOMS [11]. At the time of writing, we have shown how end users can submit jobs to shared accounts on the ScotGrid resource. Work is on-going through the VPman (www.nesc.ac.uk/hub/projects) to roll this solution out to the NGS nodes. This will provide multiple security contexts that can be shared amongst all BioNessie$_G$ users, e.g. there will be accounts set up with the appropriate ODE libraries needed for all/some BioNessie$_G$ users as deemed appropriate.

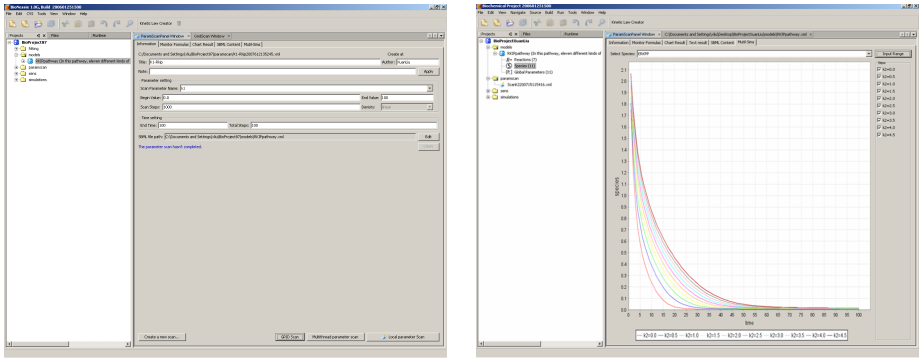
Based on the users privileges, the job scheduler service distributes jobs to available resources that users have been authorised. The scheduler also periodically checks the status of the jobs and notifies the grid server when each job finishes. On notification the Grid server fetches job outputs from various resources. Likewise, the client periodically checks the Grid server for job outputs and fetches results when they are available.

A key aspect of our work on grid enabling BioNessie$_G$ is to limit the design impact on the original BioNessie tool itself, so that the Grid-based job submission and data collection systems at the back end of the BioNessie$_G$ will not require that any Grid based middleware be integrated into BioNessie itself. For example, we specifically set out to avoid having to incorporate X509 digital certificates and associated Grid based security into the BioNessie application since this would have added significantly to the complexity of the tool both for developers and end users. Instead the tool generates standard SBML which is submitted to a client side Grid service, and it is this Grid client which deals with the credentials and secure access to the service for job submission and data collection across the Grid.

One artefact of this was how to support restricted access to BioNessie$_G$ itself and the back end resources used (since the users do not themselves have X509 certificates for identifying themselves on resources such as the NGS). To address these kinds of scenarios, we are integrating BioNessie$_G$ into a Shibboleth protected portal and delivering it to the user desktop via Java Web Start. Through this system, when users initially attempt to access BioNessie$_G$ in the portal, they are redirected and asked to log in to their home institution. Once logged in, a signed SAML (Security Assertion Markup Language) [9] assertion is sent

to the portal along with security attributes. One of these is the BioNessie$_G$ license and some form of persistent identity of the end users themselves. Through this information, we are able to create short lived proxy credentials used for job submission to sites requiring user identity through trusted X509 certificates issued for example through the UK e-Science Certificate Authority at RAL. Thus we will be able to support fine grained access to and usage of resources such as the NGS. This will also allow for example, data models associated with biochemical networks to be protected. The allocation of BioNessie$_G$ specific attributes needed by end users will exploit results from the DyVOSE project (www.nesc.ac.uk/hub/projects/dyvose) in particular the delegation issuing service.

The interface to BioNessie$_G$ is shown in Figure 2(a). Figure 2(b) shows the whole trace of selected species - ERKPP for a parameter scan in RKIP pathway of parameter K_2 from 0 through 4.5 in steps of 0.5 with linear density for the time course of 100 and times steps of 100 time units.



(a) User Interface Executing Grid Jobs (b) Results

Fig. 2. Screenshots for BioNessie$_G$

5 Conclusions

In this paper we have presented a grid enabled software tool called BioNessie$_G$ (BIOlogical NEtworkS SImulation Environment). The tool provides users with the scalability and availability needed for simulation of large scale biochemical models incorporating parameter estimations and parameter scans leveraging the computational power of the Grid. The tool core comprisess the SBML ODE Solver Library (SOSlib), LibSBML's AST and SUNDIALS, for symbolic and numerical analysis of ODE systems.

The tool abstracts users from the intricacy and complexity of the Grid without sacrificing usability and functionality. The purpose of the Grid however is

to show how larger scale simulations and parameter scans can be supported. At the time of writing we are in the processing of extending the basic Grid enabled BioNessie$_G$ prototypes to support much larger simulations. The Grid enabled BioNessie$_G$ has shown how it is possible to run jobs across Condor pools and the NGS. Supporting large scale job submission with potentially thousands of jobs requires work on meta-scheduling and optimised resource usage to be done. We expect to complete these enhancements in drawing on results from the BRIDGES project.

Acknowledgements

This research was supported by grants from the DTI Post-Genomics Beacons Bioscience programme for the 'Software Tool for Simulation and Analysis of Biochemical Networks' project, and from the European Commission STREP programme for the SIMAP project.

References

1. http://www.mathworks.com/access/helpdesk/help/toolbox/simbio/.
2. http://www.basis.ncl.ac.uk/index.html.
3. http://www.obigrid.org/OBIGrid/.
4. http://www.bionessie.org/.
5. http://sbml.org/software/libsbml/.
6. http://lcg.web.cern.ch/LCG/.
7. http://ws.apache.org/axis2/.
8. http://www.globus.org.
9. http://saml.xml.org/.
10. Myproxy credential management service. myproxy.ncsa.uiuc.edu.
11. R. Alfieri, R. Cecchini, V. Ciaschini, L. dell'Agnello, A. Gianoli, F. Spataro, F. Bonnassieux, P. Broadfoot, G. Lowe, L. Cornwall, J. Jensen, D. Kelsey, A. Frohner, D. L. Groep, W. S. de Cerff, M. Steenbakkers, G. Venekamp, D. Kouril, A. McNab, O. Mulmo, M. Silander, J. Hahkala, and K. Lhorentey. Managing dynamic user communities in a grid of autonomous resources. *CHEP 2003*, 2003.
12. R. Boys, D. Henderson, T. Kirkwood, D. Wilkinson, and W. E. Wolski. Calibayes: Integration of grid based simulation and data resources for bayesian calibration of biological models. *BMC Bioinformatics*, 6(3):14, 2005.
13. K.-H. Cho, S.-Y. Shin, H. W. Kim, O. Wolkenhauer, B. McFerran, and W. Kolch. Mathematical modeling of the influence of rkip on the erk signaling pathway. In *CMSB '03: Proceedings of the First International Workshop on Computational Methods in Systems Biology*, pages 127–141, London, UK, 2003. Springer-Verlag.
14. P. K. Dhar, T. C. Meng, S. Somani, L. Ye, K. Sakharkar, A. Krishnan, A. B. Ridwan, S. H. K. Wah, M. Chitre, and Z. Hao. Grid cellware: the first grid-enabled tool for modelling and simulating cellular processes. *Bioinformatics*, 21(7):1284–1287, 2005.
15. D. Endy and R. Brent. Modelling cellular behaviour. *Nature*, 409:391–395, 2001.
16. A. Funahashi, Y. Matsuoka, A. Jouraku, H. Kitano, and N. Kikuchi. Celldesigner: a modeling tool for biochemical networks. In *WSC '06: Proceedings of the 38th conference on Winter simulation*, pages 1707–1712. Winter Simulation Conference, 2006.

17. A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward. Sundials: Suite of nonlinear and differential/algebraic equation solvers. *ACM Trans. Math. Softw.*, 31(3):363–396, 2005.
18. J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
19. S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes, and U. Kummer. Copasi - a complex pathway simulator. *Bioinformatics*, 22(24):3067–3074, 2006.
20. M. Hucka, A. Finney, B. Bornstein, S. Keating, B. Shapiro, J. Matthews, B. Kovitz, M. Schilstra, A. Funahashi, J. Doyle, and H. Kitano. Evolving a lingua franca and associated software infrastructure for computational systems biology: the systems biology markup language (sbml) project. *Syst Biol (Stevenage)*, 1(1):41–53, 2004.
21. A. Ihekwaba, D. Broomhead, R. Grimley, N. Benson, and D. Kell. Sensitivity analysis of parameters controlling oscillatory signalling in the nf-κb pathway: the roles of ikk and i κ b α. *Systems Biology*, 1(1):93–103, 2004.
22. S. Kimura1, T. Kawasaki, M. Hatakeyama, T. Naka, F. Konishi, and A. Konagaya. Obiyagns: a grid-based biochemical simulator with a parameter estimator. *Bioinformatics*, 20(10):1646–1648, 2004.
23. H. Kitano. Computational systems biology. *Nature*, 420:206–210, 2002.
24. B. C. Kuo. *Automatic Control System*. Prentice-Hall, New Jersey, 1987.
25. R. Machn, A. Finney, S. Mller, J. Lu, S. Widder, and C. Flamm. The sbml ode solver library: a native api for symbolic and fast numerical analysis of reaction networks. *Bioinformatics*, March 2006.
26. P. Mendes. Biochemistry by numbers: simulation of biochemical pathways with gepasi 3. *Trends Biochem Sci*, 22(9):361–363, September 1997.
27. P. Mendes and D. B.Kell. Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. *Bioinformatics*, 14:869–883, 1998.
28. F. Neelamkavil. *Computer Simulation and Modelling*. John Wiley and Sons, 1987.
29. A. Pettinen, T. Aho, O.-P. Smolander, T. Manninen, A. Saarinen, K.-L. Taattola, O. Yli-Harja, and M.-L. Linne. Simulation tools for biochemical networks: evaluation of performance and usability. *Bioinformatics*, 21(3):357–363, 2005.
30. T. Ray and P. Saini. Engineering design optimization using a swarm with an intelligent information sharing among individuals. *Eng. Opt.*, 33:735C748, 2001.
31. S. Sahle, R. Gauges, J. Pahle, N. Simus, U. Kummer, S. Hoops, C. Lee, M. Singhal, L. Xu, and P. Mendes. Simulation of biochemical networks using copasi: a complex pathway simulator. In *WSC '06: Proceedings of the 38th conference on Winter simulation*, pages 1698–1706. Winter Simulation Conference, 2006.
32. P. Saraiya, C. North, and K. Duca. Visualizing biological pathways: requirements analysis, systems evaluation and research agenda. *Information Visualization*, 4(3):191–205, 2005.
33. R. Sinnott, O. Ajayi, J. Jiang, A. Stell, and J. Watt. User-oriented security supporting inter-disciplinary life science research across the grid. *Special Issue on Life Science Grids, New Generation Computing Journal*, May 2007.
34. R. Sinnott, A. Stell, M. Bayer, and J. Koetsier. Grid infrastructures for secure access to and use of bioinformatics data: Experiences from the bridges project. *The First International Conference on Availability, Reliability and Security (ARES-2006)*, 2006.

Author/s:
LIU, XUAN;Jiang, Jipu;Ajayi, Oluwafemi;Gu, Xu;Gilbert, David;SINNOTT, RICHARD

Title:
BioNessie: a grid enabled biochemical networks simulation environment

Date:
2008

Citation:
Liu, X., Jiang, J., Ajayi, O., Gu, X., Gilbert, D., & Sinnott, R. (2008). BioNessie: a grid enabled biochemical networks simulation environment. In HealthGrid 2008 Conference, Chicago, USA.

Publication Status:
Published

Persistent Link:
http://hdl.handle.net/11343/28858