BIOINFORMATICS APPLICATIONS NOTE Vol. 24 no. 2 2008, pages 287–289
doi:10.1093/bioinformatics/btm578

Systems biology

Automated manipulation of systems biology models using libSBML within Taverna workflows

Peter Li[1,*], Tom Oinn[2], Stian Soiland[3] and Douglas B. Kell[1]

[1]School of Chemistry and Manchester Centre for Integrative Systems Biology, Manchester Interdisciplinary Biocentre, University of Manchester, M1 7DN, [2]EMBL European Bioinformatics Institute, Hinxton, Cambridge, CB10 1SD and [3]School of Computing Science, University of Manchester, M13 9PL, UK

Received on October 11, 2007; revised on November 13, 2007; accepted on November 18, 2007

Advance Access publication December 1, 2007

Associate Editor: Chris Stoeckert

ABSTRACT

Summary: Many data manipulation processes involve the use of programming libraries. These processes may beneficially be automated due to their repeated use. A convenient type of automation is in the form of workflows that also allow such processes to be shared amongst the community. The Taverna workflow system has been extended to enable it to use and invoke Java classes and methods as tasks within Taverna workflows. These classes and methods are selected for use during workflow construction by a Java Doclet application called the API Consumer. This selection is stored as an XML file which enables Taverna to present the subset of the API for use in the composition of workflows. The ability of Taverna to invoke Java classes and methods is demonstrated by a workflow in which we use libSBML to map gene expression data onto a metabolic pathway represented as a SBML model.

Availability: Taverna and the API Consumer application can be freely downloaded from http://taverna.sourceforge.net

Contact: peter.li@manchester.ac.uk

Supplementary information: Supplementary data and documentation are available from http://www.mcisb.org/software/taverna/libsbml/index.html

There are often processes involving the manipulation and analysis of biological data that we would wish to automate due to their frequent and essentially repetitive invocation. This is particularly the case when the structure of such data adheres to standardized specifications that are supported by software tooling (Brazma *et al*., 2006; Strömbäck *et al*., 2007). This is true in the case of the Systems Biology Mark up Language (SBML), which may be used to represent a biological system as a network of reactions (Hucka *et al*., 2003). Software libraries such as libSBML have been developed to read, write, manipulate and validate SBML files and data streams. libSBML (http://sbml.org/software/libsbml/) has been implemented in C and C++ but is also provided with language bindings in, for example, Python, Matlab and Java.

Workflow software such as Taverna may be used for automating processes that are applied to data in the life sciences (Oinn *et al*., 2004, 2006, 2007), and systems biology represents a prime candidate for such automation via loosely coupled workflows (Kell, 2006a, b, 2007). Workflows in Taverna consist of a pre-defined series of tasks that are performed by processors. A number of processors are available for accessing data and applications with different invocation mechanisms including Web Services (http://www.w3.org/2002/ws/). Taverna consists of a number of modules such as the workflow enactor engine and workbench that together allow one to construct and execute scientific workflows (Hull *et al*., 2006). This application note reports on how Taverna can be utilized for writing and enacting workflows involving the manipulation of SBML data by making direct use of the classes and functions in the libSBML programming library.

Taverna has been extended with a processor that is capable of invoking methods within Java classes. The set of methods for use in workflows is configured using a Doclet (http://java.sun.com/j2se/1.4.2/docs/tooldocs/javadoc/overview.html) program called the API consumer. This API consumer Doclet presents a user interface for selecting the subset of methods of an API, such as libSBML, that is to be exposed to the Taverna workbench. This selection is stored as a definition in XML format which can be imported into Taverna to present the selected classes and methods of the API as available services for inclusion when constructing a workflow. This definition file can be further distributed together with the actual API implementation to third party workflow designers for enabling the usage of the API as tasks within their workflows. We illustrate this approach in what follows with a specific example.

A common and useful means of visualizing transcriptome data is to map them onto pathway diagrams (Chung *et al*., 2004; Dahlquist *et al*., 2002). This can be performed as an automated pipeline using a Taverna workflow with SBML-compliant tools so that, for example, diagrams of metabolic pathways can be rendered with microarray data such that the nodes corresponding to proteins are colored according to the expression levels of the genes that encode them. The microarray data may be stored in a database from which they may be retrieved as part of the Taverna workflow. Such a workflow is shown in the Supplementary Material as well as in Figure 1A involving the automated editing of a SBML model of the glycolysis pathway to incorporate gene expression data from

*To whom correspondence should be addressed.

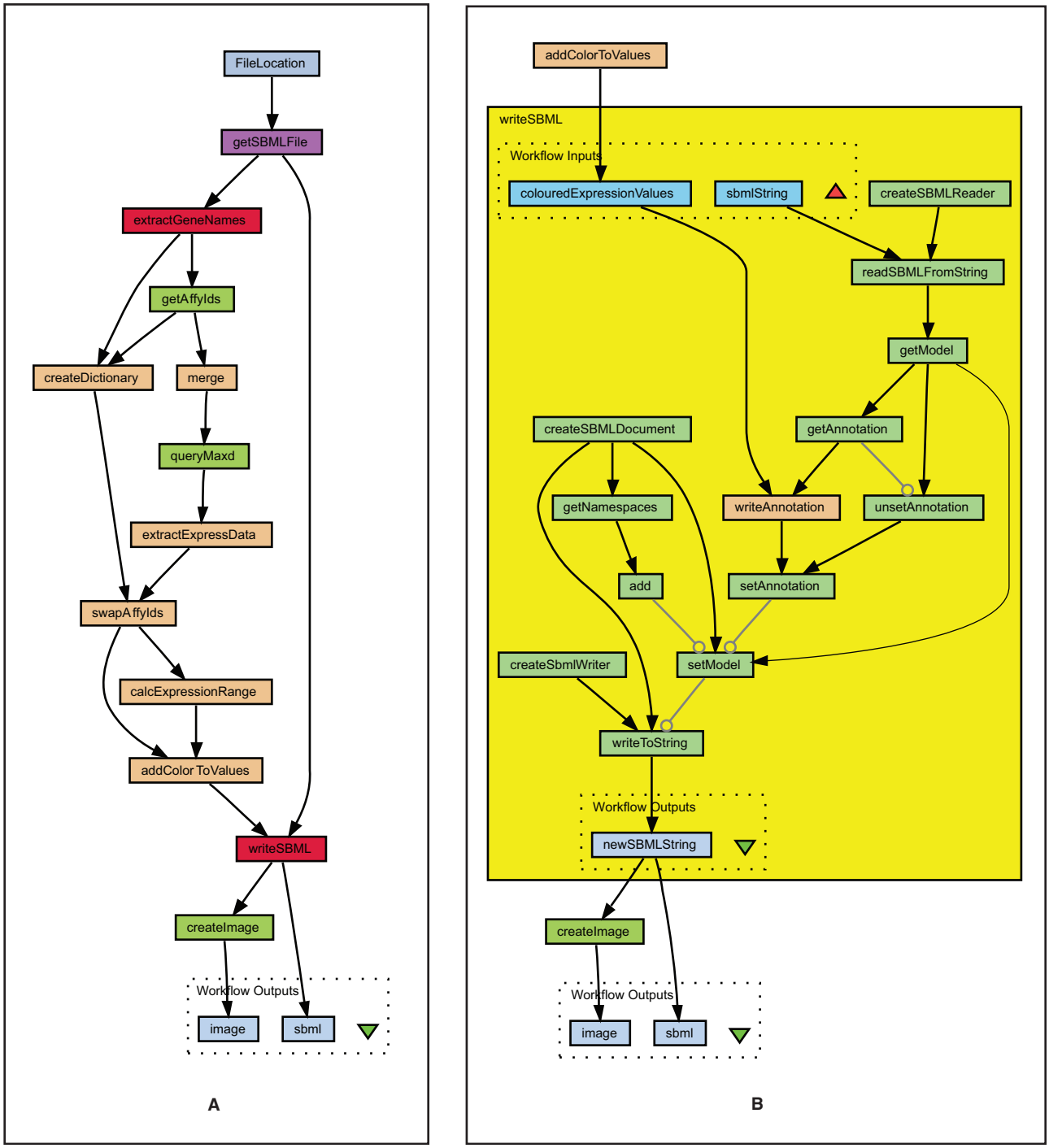

Fig. 1. A screenshot of the SBML microarray data mapping workflow is shown in Figure 1A. This workflow contains two sub-workflows labelled extractGeneNames and writeSBML. Figure 1B shows a screenshot of the writeSBML sub-workflow which has been expanded to show API consumer processors calling methods from libSBML. These methods from libSBML, together with a beanshell processor called writeAnnotation are used to create a new SBML model containing mapped microarray gene expression data.

the Maxd database (Hancock *et al.*, 2005) onto layout information embedded within SBML, which can then be visualized with Cell Designer (Funahashi *et al.*, 2003). The workflow has two sub-workflows which contain API consumer processors that use methods from libSBML for parsing the names of proteins in the SBML file as well as generating a new SBML file incorporating the mapped gene expression data (Fig. 1B). Beanshell processors (http://www.beanshell.org) can be used to provide application logic for further processing of the data in the SBML model. These beanshell processors were

used to determine how entities in the microarray data matched with those in the SBML model; this was done using information on how genes in the microarray data identified by their Affymetrix probe set identifiers mapped onto enzyme modifier species that were labeled using yeast gene names in the SBML model.

Through the use of the API consumer, Taverna can make direct use of the functionality residing within Java classes and methods as workflow tasks. This is accomplished without the need for deploying the services in the API as Web Services. This may be more suitable since the underlying services may perform trivial tasks, making the overhead of invocation through a Web Services interface impractical. Whilst we have shown the use of the API consumer in systems biology, it is a generic tool in that it can be used with other Java APIs such as the Chemistry Development Kit (Steinbeck *et al*., 2003), enabling Taverna to be tailored to different scientific domains. This generic nature of the API consumer means that Taverna can work with SBML using new releases of libSBML as and when they become available, with no extra coding being required to make use of new releases of libSBML in Taverna. This said, the use of APIs can make workflows more difficult to compose as the functions are more fine-grained than are operations in Web Services, requiring expert knowledge of using the API and Taverna. Whilst this results in more complex workflows, the complexity can be hidden from users within Taverna using nested workflows (Fig. 1A). Also, once such workflows have been written, there is the great benefit that they may be saved and shared for use within the systems biology community.

ACKNOWLEDGEMENTS

We thank Prof. Hiroaki Kitano and Dr Akira Funahashi for very useful discussions. P.L. and D.B.K. thank the BBSRC for financial support, and D.B.K. acknowledges the financial support of the BBSRC and EPSRC in the Manchester Centre for Integrative Systems Biology (www.mcisb.org).

Conflict of Interest: none declared.

REFERENCES

Brazma,A. *et al*. (2006) Standards for systems biology. *Nat. Rev. Genet*., **7**, 593–605.

Chung,H.J. *et al*. (2004) ArrayXPath: mapping and visualizing microarray gene-expression data with integrated biological pathway resources using Scalable Vector Graphics. *Nucleic Acids Res*., **32**, W460–W464.

Dahlquist,K.D. *et al*. (2002) GenMAPP, a new tool for viewing and analyzing microarray data on biological pathways. *Nat. Genet*., **31**, 19–20.

Funahashi,A. *et al*. (2003) CellDesigner: a process diagram editor for gene-regulatory and biochemical networks. *BIOSILICO*, **1**, 159–162.

Hancock,D. *et al*. (2005) maxdLoad2 and maxdBrowse: standards-compliant tools for microarray experimental annotation, data management and dissemination. *BMC Bioinformatics*, **6**, 264.

Hucka,M. *et al*. (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, **19**, 524–531.

Hull,D. *et al*. (2006) Taverna: a tool for building and running workflows of services. *Nucleic Acids Res*., **34**, W729–W732.

Kell,D.B. (2006a) Metabolomics, modelling and machine learning in systems biology: towards an understanding of the languages of cells. The 2005 Theodor Bücher lecture. *FEBS J*., **273**, 873–894.

Kell,D.B. (2006b) Systems biology, metabolic modelling and metabolomics in drug discovery and development. *Drug Discov. Today*, **11**, 1085–1092.

Kell,D.B. (2007) The virtual human: towards a global systems biology of multiscale, distributed biochemical network models. *IUBMB Life*, **59**, 689–695.

Oinn,T. *et al*. (2004) Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, **20**, 3045–3054.

Oinn,T. *et al*. (2006) Taverna: lessons in creating a workflow environment for the life sciences. *Concurrency Comput. Pract. Exper*., **18**, 1067–1100.

Oinn,T. *et al*. (2007) Taverna/myGrid: aligning a workflow system with the life sciences community. In Taylor,I.J. *et al*. (ed.) *Workflows for e-Science: Scientific Workflows for Grids*. Springer, Guildford, pp. 300–319.

Steinbeck,C. *et al*. (2003) The Chemistry Development Kit (CDK): an open-source java library for chemo- and bioinformatics. *J. Chem. Inf. Comput. Sci*., **43**, 493–500.

Strömbäck,L. *et al*. (2007) A review of standards for data exchange within systems biology. *Proteomics*, **7**, 857–867.