BIOINFORMATICS APPLICATIONS NOTE
Vol. 27 no. 18 2011, pages 2618–2619
doi:10.1093/bioinformatics/btr425

Databases and ontologies
Advance Access publication July 19, 2011

CalcTav—integration of a spreadsheet and *Taverna* workbench

Jacek Sroka[1,*], Łukasz Krupa[1], Andrzej M. Kierzek[2] and Jerzy Tyszkiewicz[1]

[1]Institute of Informatics, University of Warsaw, Warsaw, Poland and [2]School of Biomedical and Molecular Sciences, University of Surrey, Guildford GU2 7XH, UK

Associate Editor: Martin Bishop

ABSTRACT

Motivation: *Taverna* workbench is an environment for construction, visualization and execution of bioinformatic workflows that integrates specialized tools available on the Internet. It already supports major bioinformatics services and is constantly gaining popularity. However, its user interface requires considerable effort to learn, and sometimes requires programming or scripting experience from its users. We have integrated *Taverna* with *OpenOffice Calc*, making the functions of the scientific workflow system available in the spreadsheet. In *CalcTav*, one can define workflows using the spreadsheet interface and analyze the results using the spreadsheet toolset.

Results: Technically, *CalcTav* is a plugin for *OpenOffice Calc*, which provides the functionality of *Taverna* available in the form of spreadsheet functions. Even basic familiarity with spreadsheets already suffices to define and use spreadsheet workflows with *Taverna* services. The data processed by the *Taverna* components is automatically transferred to and from spreadsheet cells, so all the visualization and data analysis tools of *OpenOffice Calc* are available to the workflow creator within one, consistent user interface.

Availability: *CalcTav* is available under GPLv2 from http://code.google.com/p/calctav/

Contact: sroka@mimuw.edu.pl

Received on January 20, 2011; revised on June 27, 2011; accepted on July 13, 2011

1 MOTIVATION

Specialized systems for defining and conducting biological experiments *in silico*, such as *Taverna* (see Hull *et al*., 2006; Oinn *et al*., 2006) or *Kepler* (see Ludäscher *et al*., 2006), have clear advantages over software designed and written *ad hoc*. To name just the main reasons:

- Correctness is much easier to achieve by using software organized from carefully tested and verified components.
- The general design of the experiment is required to implement it and is separate from the software itself, thus providing easy insight into the experiment, for its author, later for the reviewers and finally for the readers of the publication (Mesirov, 2006).
- Experiments, once created, are portable (can be easily transferred to some other machine and executed there) and reusable (can be easily modified and executed again).

It is therefore beyond doubt that the use of scientific workflow systems should be advocated for and promoted. We would like to add another argument:

- Experiments can be created much easier and faster in a specialized environment than elsewhere.

However, this argument has one exception. The very first experiment, when the researcher has to learn the new system, creates a 'barrier', probably preventing many prospective users from even getting started. Besides that, from the user's of view 'easier and faster' means not only the computations, but also the analysis and visualization of their results. In order to assist those users, we offer a completely new form of using *Taverna*. We have created a plug-in for the spreadsheet *OpenOffice Calc*, which provides the functionality of *Taverna* within spreadsheet. One can then think of spreadsheet as a user interface for *Taverna*, which makes it easier to learn for new users. Moreover, data analysis tools of spreadsheets are now integrated with the functions of *Taverna*.

That integration is possible because the paradigms of spreadsheets and workflow systems are very close to each other on the conceptual level. In both, the computational task is realized bottom-up, by starting from input values and repeatedly using built-in functions, evaluated on initial inputs and intermediate, already computed values, until the desired result is eventually achieved. It is only a matter of convention, if one uses links specified by arrows in a graphical user interface to indicate on which values a newly added function should be computed (workflow systems), or writes a formula, which specifies the same function to be used and the very same arguments by the addresses of cells in which they are found (spreadsheets). The only real difference is that spreadsheets typically operate on numbers, while workflow systems on much more complicated data objects, such as sequences of strings, etc. However, this difference affects mainly the computation engine, but not the user interface—the user can well think of a whole database of protein sequences or any other complex collection of data objects as a single item, which should be processed, very much like a single integer.

The advantage of the integration of a scientific workflow system with a spreadsheet, is, in our opinion, very significant: already during the first use of such a system, the user feels familiar with the layout of the interface, can intuitively predict how to carry out certain manipulations and what effects they might have, as opposed to studying the semantics of a workflow system as the one in Sroka *et al*. (2010), which can be overwhelming for users with no programming background. Furthermore, the spreadsheet provides many additional features, like the built-in visualization tools, data analysis functions and, last but not least, many editing and

[*]To whom correspondence should be addressed.

control tools (like the *Detective* functions), which help in designing and debugging computations. Spreadsheets also have very intuitive user interface for working with collections of data based on the *fill handle* mechanism, which is a method of copying formulas to the neighboring cells, with suitable reference modifications. Portability of spreadsheet workflows is obvious, while creating spreadsheet templates without data offers a convenient method to achieve reusability. It is also important that there exists vast literature concerning spreadsheets in general and *OpenOffice Calc* in particular, and that the majority of design methods and tips described there apply to our spreadsheet interface, as well.

2 RESULTS

Technically speaking, *CalcTav* is a plugin for *OpenOffice Calc*, which provides the computational functionality of *Taverna* within spreadsheet. It extends the spreadsheet system in the following ways:

(1) External data sources and computing services of *Taverna* are wrapped as spreadsheet functions. Tools for managing these services are provided.

(2) Complex data objects, such as lists and nested lists can be stored in cells, transferred between cells and presented to the user. Basic construction and deconstruction operations for those objects are provided, e.g. elements of a list can be extracted into a row or column of cells.

(3) A tool for visualization and editing of complex data objects located in cells is provided.

(4) To increase performance, calls to remote data sources and computing facilities are executed in multiple threads.

(5) Extended control over recomputation of the data sheet is provided, so it can be made automatic, blocked or manually enforced for selected cells.

(6) A workflow structure view, to supplement the standard spreadsheet mechanism for presenting cell function dependencies with arrows, is provided.

Certain XML formats of complex data objects are presently not supported. *CalcTav* does not limit quantities of data that can be processed. However, it does not offer *implicit iteration* present in *Taverna*. This mechanism of the original workflow system works as follows: if a processor expects a single input item and returns a single output, but receives a list of items as input, it is invoked for every element of that list, and produces a list of its individual outputs.

In *CalcTav*, we have adopted the spreadsheet paradigm, that each cell is evaluated only once. Iteration is achieved with the spreadsheet fill handle mechanism by making copies of the cells with function calls, one for each item of the list to be processed. This way, *CalcTav* spreadsheets can be extended by copying cells to process list of variable length limited only by the number of cells prepared beforehand. This mild limitation, however, helps very much in debugging spreadsheet-workflows, as every processor call is executed in a separate cell and can be debugged individually. This should help novice users, for whom *CalcTav* is particularly intended.

In order to test and demonstrate how *CacTav* compares to *Taverna*, we have implemented a computational procedure based on a real-life experiment described in article Sassetti and Rubin (2003), in the form of a bionformatic workflow using both systems, and executed them. It involves annotation of genes, which according to high-throughput mutagenesis experiment are required for mycobacterial survival during infection. The workflow analyses published list of genes which are differentially represented according to statistical analysis of microarray data. The gene names written in the spreadsheet column are automatically assigned gene orthology and metabolic pathway annotation according to the KEGG database. The user uses the fill handle tool to iterate over the list of genes and *CalcTav* plugin connects to the web services of KEGG and executes queries. Subsequently, spreadsheet functions are used to calculate frequency of the functional descriptors assigned to the list of differentially represented genes. The functional categories are then sorted to identify those which are most frequently associated with the genes required for mycobacterial survival during infection.

3 AVAILABILITY AND IMPLEMENTATION

OpenOffice and *LibreOffice* are supported under Windows and Linux. MacOS X is currently not supported. The Web page of *CalcTav* is http://code.google.com/p/calctav/. It contains the software distributed under GPLv2, both example workflows mentioned above, together with documentation, including flash movies.

ACKNOWLEDGEMENT

The authors are grateful to M. Dopiera, A. Kawa, P. Krewski, T. Weksej and M. Zawadzki for the initial contributions to the project.

Funding: Polish *National Science Centre*.

Conflict of Interest: none declared.

REFERENCES

Hull,D. *et al*. (2006) Taverna: a tool for building and running workflows of services. *Nucleic Acids Res.*, **34**, 729–732.

Ludäscher,B. *et al*. (2006) Scientific workflow management and the Kepler system. *Concurr. Comput. Pract. Exp.*, **18**, 1039–1065.

Mesirov,J.P. (2010) Accessible reproducible research. *Science*, **327**, 415–416.

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

Sassetti,C.M. and Rubin,E.J. (2003) Genetic requirements for mycobacterial survival during infection. *Proc. Natl Acad. Sci. USA*, **100**, 12989–12994.

Sroka,J. *et al*. (2010) A formal semantics for the Taverna 2 workflow model. *J. Comput. Syst. Sci.*, **76**, 490–508.