BIOINFORMATICS APPLICATIONS NOTE Vol. 22 no. 10 2006, pages 1280–1281
doi:10.1093/bioinformatics/btl101

*Data and text mining*

# XQTav: an XQuery processor for Taverna environment

Jacek Sroka[1,*], Grzegorz Kaczor[1], Jerzy Tyszkiewicz[1] and Andrzej M. Kierzek[2,*]

[1]Warsaw University, Warsaw, Poland and [2]SBMS, University of Surrey, Guildford, GU2 7XH, UK

Received on October 17, 2005; revised on February 28, 2006; accepted on March 15, 2006
Advance Access publication March 21, 2006
Associate Editor: Martin Bishop

**ABSTRACT**

Taverna workbench is an environment for construction, visualization and execution of bioinformatic workflows that integrate specialized tools available through the internet. It is gaining popularity fast, because of supporting the most important bioinformatic services and its simple, yet robust graphical notation. Here we present XQTav—an extension of Taverna that provides full integration with XQuery (the query language for XML) engine. XQTav allows execution of XQuery scripts in Taverna workflow diagrams. All existing Taverna processors can be accessed in the XQuery scripts. This provides an alternative way of specifying sub-workflows in Taverna and is useful when one deals with query-like algorithms (e.g. filters and inner joins). Moreover, XQtav may be used to automatically generate an XQuery script that is equivalent to Taverna's workflow. This constitutes another way of creating and enacting bioinformatic workflows: overall structure of a diagram is drawn in Taverna environment, XQuery code is generated and possibly adjusted by hand. It can be executed by XQuery engines or incorporated into other software environments.

**Availability:** XQtav is an open source software. It may be downloaded from http://xqtav.sourceforge.net/. The page also contains various tutorials and examples, including the one described in this report.

**Contact:** sroka@mimuw.edu.pl, a.kierzek@surrey.ac.uk

An application of internet-based resources for hypothesis generation and data analysis is an everyday reality in a molecular biology laboratory. It becomes problematic to perform those *in silico* experiments by manual execution of simple queries via www-based interfaces. Submission of data to tens of servers and manual integration of results is tedious and error prone. This motivates development of grid-based integration techniques in which a user communicates with client software that executes queries containing calls to web services provided by dedicated bioinformatic sites. Currently, many bioinformatic institutions including NCBI, EBI, Sanger Institute, Kyoto University expose their databases and software as web services for easy programmatic access.

As pointed out in Achard *et al*. (2001), the eXtensible Markup Language (XML)—a W3C standard for structuring documents—is ideally suited for data interchange between bioinformatic services. Similarly as in other domains, the adoption of XML in bioinformatics increases quickly, and majority of existing, as well as newly constructed bioinformatic tools, make XML their common denominator (Gordon, 2003, http://www.visualgenomics.ca/gordonp/xml/).

The Taverna software (Oinn *et al*., 2004, http://taverna.sourceforge.net/) is a front end of a grid-based integration platform that makes use of workflow technology. With the help of a simple graphical user interface, in which computational tasks (calls to web services) are represented as processors and data flow between them is symbolized by arrows, it allows to create workflows that describe data analysis tasks. Workflows are subsequently expressed in an internal format of Taverna, the SCUFL language and executed. During execution Taverna makes calls to the relevant web services and the user is presented with an integrated result. The system has been already used in molecular biology research (Stevens *et al*., 2003) and is gaining popularity fast.

Following the emerging standard, a considerable part of the information flow between tools used in Taverna workflows is implemented with the XML format. The XQuery (Boag *et al*., 2005, http://www.w3.org/TR/2005/WD-xquery-20050915/) is becoming *the* language for querying XML data. It can identify data items across XML files using XPath expressions and define operations on these data using FLWOR expressions and functional programming language syntax. Numerous competing engines for execution of XQuery scripts are being developed, optimized and used by industrial service providers in a growing number of applications. A simple XQuery processor based on one of such engines is also present in Taverna. Here, we present the XQTav, a middleware for further integration of Taverna environment with an XQuery engine. The current version has been tested with the SAXON-B engine. XQTav has two major, novel functionalities:

- Creation of Taverna processors that encapsulate XQuery scripts with complex queries. All of the processors of Taverna can be executed as an external function calls in the XQuery script.
- Automatic transformation of a whole Taverna workflow into an XQuery script for later execution by an XQTav processor or XQuery engine.

The first functionality allows for easy creation of new Taverna processors that perform queries, which are difficult to express in graphical notation. Such queries can involve some preprocessing of XML data transferred between processors (e.g. filtering), but also can be successfully used to define query-like algorithms (e.g. inner

*To whom correspondence should be addressed.

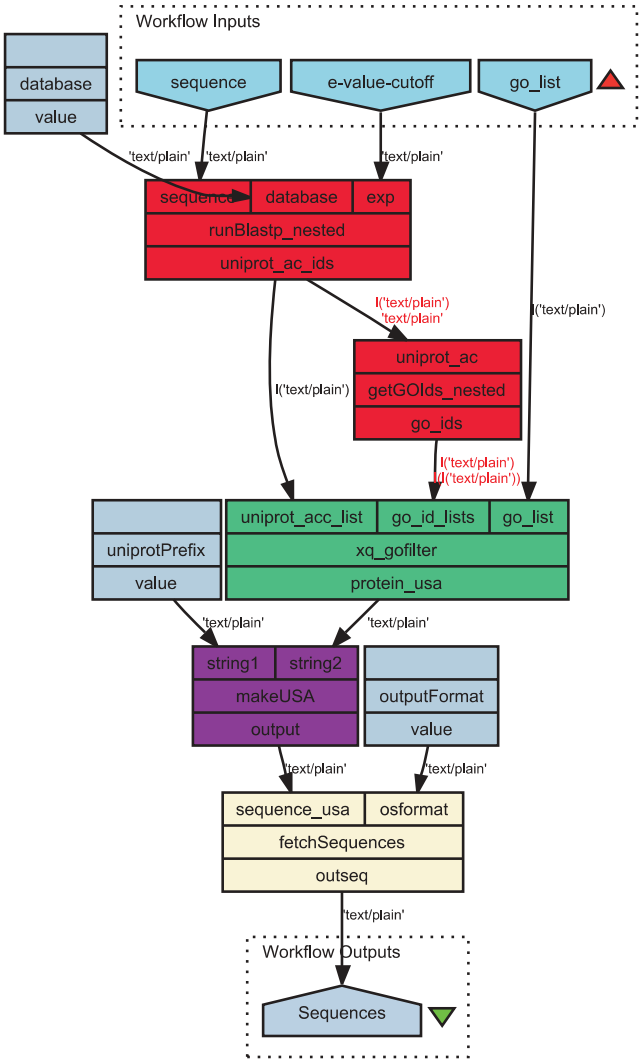


**Fig. 1.** Taverna workflow example that uses XQTav processor for executing XQuery scripts. The processor xq_gofilter extracts those sequences from BLASTP output, for which Uniprot database assigns at least one of the user-specified GO terms.

joins of lists), for which a query language is better suited than a general graphical notation (such algorithms are often met in bioinformatics). In the latter case, the involved processors do not need to use XML formats. Queries can be easily implemented by programmers and stored in the form of ready-to-use processors that can be re-used in Taverna workflows by other users that do not need to have programming experience.

For example, the workflow from Figure 1 represents a simple data analysis task in which the user searches for an evidence supporting hypothesis that a protein sequence performs certain function. Input of the workflow is the sequence in question and a list of Gene Ontology (Ashburner *et al*., 2000) terms which specify functions. The program executes a BLAST search to find a protein sequences in the Uniprot (Bairoch *et al*., 2005) database. With the use of a processor which executes another Taverna workflow, the second list is created. Its elements are lists of GO terms that correspond to subsequent Uniprot identifiers from the BLAST output list. That along with the sequences list (from which it originated) and the input list of GO terms are processed by an XQTav processor. It returns only the sequences that are described by at least one GO term from the input list. It is easier to express this operation in a query language than by a nested diagram. The use of XQuery allows also for query optimization, which may improve efficiency of the operation.

The second functionality allows for a new way of creating and executing bioinformatic workflows. The overall idea of the workflow can be effectively specified in Taverna's simple graphical notation and saved as a SCUFL document. Then SCUFL file is automatically converted into an XQuery script by XQTav and possibly adjusted 'by hand' to implement complex features that are difficult to express graphically. The resulting query is executed in an XQTav processor or incorporated into other software environments using SAXON-B engine. Complex SCUFL documents involving calls to all Taverna's processors can be automatically transformed into XQuery and enacted. There are some mild limitations of the automatic transformation process. They are described, along with the possible work-arounds, in the XQTav Reference Guide (Chapters 5.3 and 6.2, Sroka *et al*. 2006, http://xqtav.sourceforge.net/refguide.pdf). Transformation of SCUFL documents into XQuery scripts can be used to construct WWW servers which offer complex analysis protocols generated from Taverna's workflows, and whose HTML user interface is generated by the same XQuery scripts. This way all of the server side programming could be done with the use of XQuery (Ivanov, 2003, http://www.xml.com/pub/a/2003/05/14/xquery.html).

To summarize, the XQTav is an interface between Taverna and XQuery engines. It offers an alternative to those complex nested workflows that are hard to represent as diagrams, by providing access to Taverna processors from XQuery scripts and extending the XML processing functionalities of Taverna. As both the Taverna system and XQuery standards and engines will mature, their integration via XQTav middleware will give additional flexibility for the design of distributed computing applications in the area of biological data analysis.

## ACKNOWLEDGEMENTS

This research has been supported by the bilateral Polish-Flemish project 'Foundations of Databases for Bioinformatics' and by the Polish KBN grant 4 T11C 042 25.

*Conflict of Interest*: none declared.

## REFERENCES

Achard,F. *et al*. (2001) XML, bioinformatics and data integration. *Bioinformatics*, **17**, 115–125.

Ashburner,M. *et al*. (2000) Gene Ontology: tool for the unification of biology. *Nat. Genet*., **25**, 25–29.

Bairoch,A. *et al*. (2005) The Universal Protein Resource (UniProt). *Nucleic Acids Res*., **33**, D154–D159.

Boag,S., Chamberlin,D., Fernández,M.F., Florescu,D., Robie,J. and Siméon,J. (Eds) (2005) XQuery 1.0: An XML Query Language.

Gordon,P. (2003) XML for molecular biology as compiled by Paul Gordon.

Ivanov,I. (2003) Interactive web applications with Xquery.

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

Sroka,J., Kaczor,G., Tyszkiewicz,J. and Kierzek,A.M. (2006) XQTav Reference Guide.

Stevens,R.D. *et al*. (2003) Exploring Williams–Beuren syndrome using myGrid. *Bioinformatics*, **20**, i303–i310.