Renewing bioinformatics workflow systems by using a Web 2.0 approach

Roberto Colella[1]✉, **Bachir Balech**[2], **Antonella Vaccina**[3], **Pietro Leo**[3], **Gaetano Scioscia**[3]

[1]Istituto di Studi sui Sistemi Intelligenti per l'Automazione - Consiglio Nazionale delle Ricerche, Bari, Italy
[2]Istituto di Biomembrane e Bioenergetica - Consiglio Nazionale delle Ricerche, Bari, Italy
[3]GBS BAO Advanced Analytics Services and MBLab, IBM Italia S.p.A., Italy

Received 14 May 2015; **Accepted** 27 July 2015; **Published** 6 August 2015

Colella *et al.* (2015) *EMBnet.journal* **21**, e840. http://dx.doi.org/10.14806/ej.21.0.840

Competing Interests: none

Abstract

The use of "mashups" is expanding considerably in the business environment. Business mashups are usually adopted within integrating business and data-service frameworks to provide the ability to develop new integrated services quickly. Typically, mashups provide organisations with a pronounced and flexible commodity to combine internal with external services in order to create new services, usually accessed through user-friendly Web-browser interfaces. In this study, a Web 2.0 technology was adopted to promote a key field of bioinformatics research through the management and automation of bioinformatics workflows. Consumables (widgets and services) have been developed using the Lotus Widget Factory, an Eclipse plug-in providing an easy-to-use development environment enabling developers of all skill levels to create dynamic widgets rapidly. A workflow built from widgets works as follows: the core widget receives data from one or more widgets, invokes a generic Web service, performing iteration and/or recursion, and sends the results to all other connected widgets. The number of iterations and recursions depends on the input data-set dimension and user-defined parameter values related to each specific application. Some prototype workflows have been assembled and tested with a number of widgets created with algorithms from the European Molecular Biology Open Software Suite (EMBOSS), exposed as Web services. The adoption of recent Web 2.0 technologies, such as mashup platforms, has enabled rapid generation, sharing and discovery of reusable application building-blocks (widgets, feeds, mashups), and has shown to be a plausible alternative environment for supporting bioinformatics workflow design, management and execution.

Introduction

The execution of complex bioinformatics workflows is becoming increasingly important for advanced scientific research, given the huge amount of data output by next-generation sequencing technologies (*e.g.*, Marguiles *et al.*, 2005; Bentley *et al.*, 2008). In this context, analysis workflows are becoming more complex to build, requiring advanced technical skills, which end-users may not have.

Several software and platform products have been proposed to meet the typical needs of bioinformaticians. Examples include the integration of multiple data sources (*e.g.*, data stored on local file systems, query results from public or private databases, feeds), the availability of computational tools necessary to achieve specific research results, and workflow storage for reproducibility. Amongst the most frequently used tools for managing and executing workflows are Taverna (Wolstencroft *et al.*, 2013), Bioextract (Lushbough *et al.*, 2010) and Galaxy (Goecks *et al.*, 2010). The first of these requires installation as a standalone workbench, allowing workflow design and building, plus the ability to execute them on the cloud and share them on a public website[1]. The second is a Web-based application, which does not allow use of available operators to assemble workflows. Bioextract's main functionality is based on recording actions performed by users, and not on "drag and drop" propositions. Galaxy offers efficient online reuse of previously implemented applications, but its difficulty resides in the need for advanced programming skills to build new workflows. The main

[1] www.myexperiment.org

goal of this article is to evaluate the usability of a widget-based tool that facilitates and speeds up the development of bioinformatics workflows. To that end, we provide a complete description of this workflow management and storage system, presented as a prototype and tested on two locally assembled bioinformatics workflows in a mashup framework.

The mashup framework

Mashups are applications that integrate information from different data sources into a single new service. Data from different sources need to be represented in such a way that users can understand and analyse them. In enterprise IT management, there is an opportunity to mash up data from various products, keeping intact data behaviour and data flow, to provide new insights (Fichter, 2010).

Mashup techniques have been successfully adopted in several business areas. For instance, Boeing (Ayhan *et al.*, 2009), Wells Fargo, the UK's Kent County, AMEC Paragon and the New York State Department of Labor (Sezici, 2009) are examples of the use of mashups for fast application delivery and improved decision-making. Recently, the mashup approach has also been suggested for use in bioinformatics (Gong, 2013; Hogan *et al.*, 2011; Cheung *et al.*, 2008), but to our knowledge a bio-mashup editor is still lacking. The kind of issues cited above suggested the adoption of IBM Mashup Center, an end-to-end enterprise mashup platform supporting rapid assembly of widgets, which are dynamic miniature Web applications embedded within HTML pages. This tool includes a Mashup Builder, a widget-based browser interface that contains all the necessary components for creating, assembling, configuring and designing objects, such as widgets, mashup pages and spaces. Moreover, it provides a set of out-of-the-box, business-ready widgets, which jump-start mashup creation and enhance information visualisation options, such as charting.

The uniqueness of this system lies in the simplicity of extending the mashup environment by incorporating custom IT widgets from the IBM Mashup Catalog, or widgets from external Web resources, including any of the thousands of Google Gadgets. Furthermore, Mashup Center allows bioinformaticians to work with feeds, which can be mixed and transformed into new feeds, also known as data mashups. Using the Data Mashup builder, a visual browser-based tool, information and business analysts can re-mix, merge, group, sort, annotate, filter and transform feeds in a variety of ways, creating a single view of disparate sets of information in a very short time. Once a mashup is assembled, it can be easily shared and, by means of some embedded visual tools, the workflow owner can define users or groups of users who can view or edit their various pages. Additionally, with just a few clicks, Mashup Center allows users to customise widgets and pages, and then copy-and-paste the scripts behind them into a Web page, all without writing additional code. Mashups can also be published to the *Mashuphub* catalogue, a shared environment where other users can easily reuse them. Figure 1 shows the context diagram of the adopted platform.

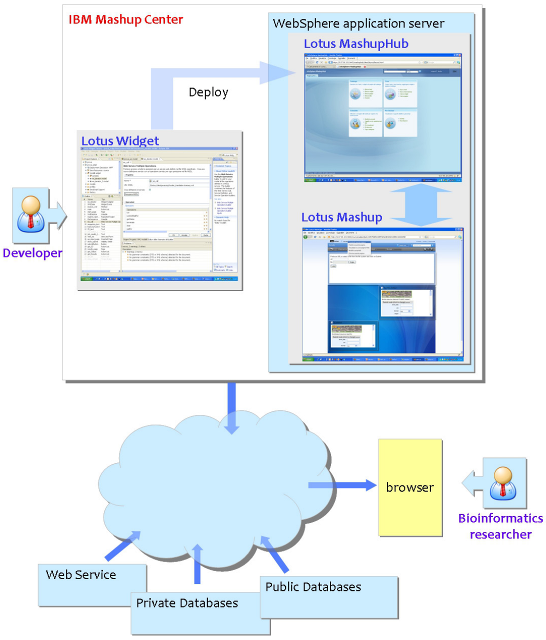


Figure 1. Context diagram of the mashup platform, showing the developer and the bioinformatics researcher interaction with the mashup framework. At the top, the developer deploys the widget(s) on the WebSphere application server to become available to the bioinformatician (bottom) to create a custom workflow.

Implementation

The development of widgets has been carried out using the Lotus Widget Factory[2]. This is an Eclipse plug-in based on the concept of models that a developer assembles from basic bricks

[2] https://www.ibm.com/developerworks/lotus/documentation/widgetfactory

called builders. The builders are generic components that encapsulate a given capability. Lotus Widget Factory comes with a large number of predefined builders, ranging from user-interface components, such as buttons, to components responsible for fetching data from remote Web services. A user-friendly wizard interface is associated with each builder, and lets the developer specify its characteristics, such as input data. Once the development has been completed, widgets are deployed as ".WAR" (Web-application ARrchives) in Lotus MashupHub and can be added to a mashup page. Our goal was to obtain detailed insights into the usability of this framework for the assembly, execution and management of bioinformatics workflows. To this end, we implemented separate widgets for some bioinformatics algorithms in order to offer users easy assembly of their own workflows. In addition, we used these widgets to assemble some prototype workflows. In the following paragraphs, we detail the widgets implemented, covering generic and/or specific user-defined requisites for DNA/protein sequence analysis.

Data Source

The Data-Source widget allows selection of an input file from a local file system or a URL invoking a REST service, and then parses the fetched data. The parser can interpret different file formats (EMBL, FastA, *etc.*) to extract all the contained DNA/protein sequences and display them in a tree view. At this step, users can choose which of the sequences will be sent to the next workflow block. Note that the data are converted to FastA format and then arranged in an XML structure, which facilitates communication amongst the consecutive widgets embedded in the workflow.

Merge and Split

The Merge and Split widgets operate on the XML data-flow between widgets. The Merge widget converts input data (sequences, matrices, *etc*...) into a unique output data-flow. It is useful to create a single XML file containing sequences from many files (*i.e.*, coming from a Data-Source widget). In contrast, the Split widget is used to separate the elements of the XML data according to a given regular expression, to facilitate recursive usage of the subsequent workflow widgets.

WSDL-described Web-service widget

The WSDL-described (Web-Service Description Language) Web-service widget is the core application of mashup techniques for bioinformatics workflow building. Its main aim is the execution of an algorithm remotely exposed as a Web service, implemented by means of the following Widget Factory builders:

- *Web service multiple operation and HTML page* builder invoke the Web service, get the available operations and create the user interface;
- *widget event* builder, together with a data-decoding Java method, receive and parse XML data from the previous widget;
- *repeated region* builder iterates over the XML structure and enables recursive invocation varying according to user-defined parameter values;
- *action list* builder executes the "run" Web-service action for all the items found in the XML input data, getting a job ID for each of them;
- *HTML page* builder creates a results page and invokes the "waitfor" and "getResults" Web-service operations (action list);
- *another action list* builder stores the results of the executed jobs in an XML output;
- *widget event* builder sends the XML structure to all the widgets connected to it.

The Web-service widget has been implemented to accept the WSDL file describing EMBOSS[3] (Rice *et al.*, 2000) bioinformatics tools exposed as Web services. With very few customisations, mainly regarding variable names and eventual multiple inputs, it was possible to create ~200 widgets corresponding to applications in the entire EMBOSS suite.

REST Web-service widget

The widget executes a REST service call, stores the results in the XML format output (action list builder), and sends it to all the widgets connected to it (*widget event* builder).

Recursion widget

The Recursion widget can be wired to a Web-service widget, and can collect all the parameters from it. This widget subsequently displays a menu with all the relevant application-specific parameters, allowing users to set their corresponding values for execution during the recursion.

[3] emboss.sourceforge.net

Weblogo widget

Weblogo[4] is a Web application that can be used if a graphical representation is needed to summarise one or more sequence alignments obtained by a given algorithm. The application can be installed locally or exposed as a REST service.

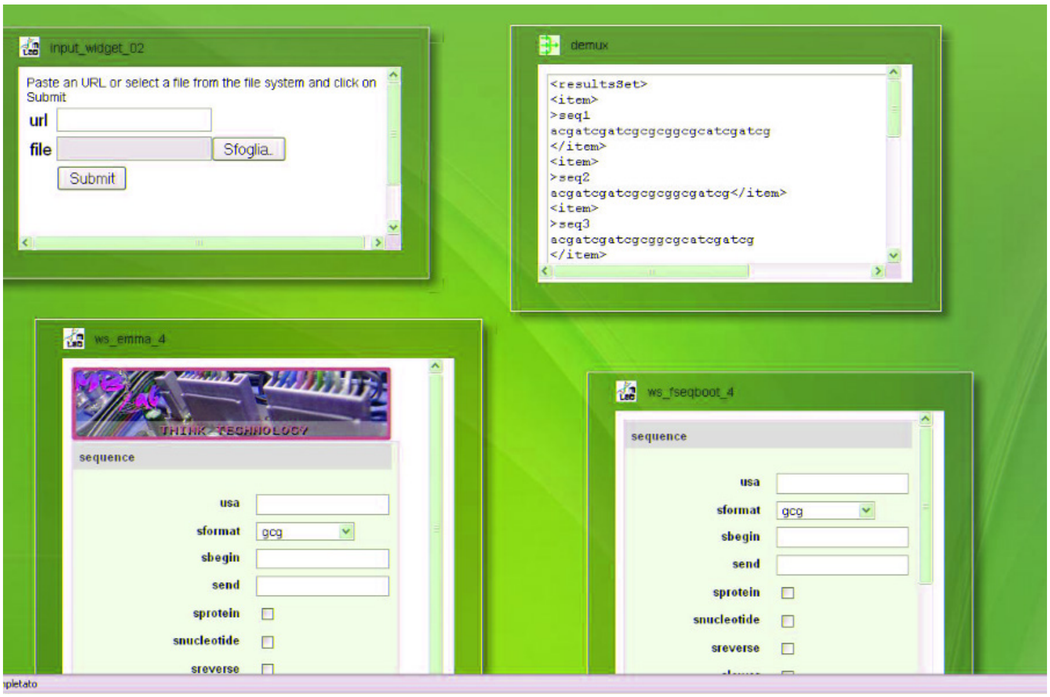
In summary, the result of creating the widgets described above is that users can assemble their own workflows by choosing widgets from a drop-down menu and dragging them onto the application page of the mashup editor and connecting them. They can also choose which workflow steps are to be executed automatically, simply by checking a box on each user interface. Another important aspect of this system is the ability to inspect intermediate results, as each widget included in the workflow shows the results it has produced. This can be useful for trouble-shooting and further adapting bioinformatics workflows.

Results and Discussion

The main result of the solution described here is the availability of a prototype workbench system to develop and build either classical analysis workflows or more complex ones. Our experience in building this prototype has shown that bioinformatics researchers can easily design and develop their own workflows and application pages using different tools and data sources. Apart from the existing default widgets, including those mentioned above, a palette of widgets providing the EMBOSS suite applications has been added. In addition, the system flexibility allows advanced users to add new applications, and therefore create new widgets. To validate the functionality of this system, two workflow case-studies are presented in the following paragraphs: i) a phylogenetic inference workflow, and ii) a universal primer-design and validation workflow.

Phylogenetic inference workflow

Our first example of a workflow assembled using the Mashup Center is phylogenetic inference using the neighbour-joining (Saitou and Nei, 1987; St John *et al.*, 2003) or UPGMA (Reguant and Bordons, 2003) methods, commonly used in molecular-evolution studies. The workflow constructs a consensus phylogenetic tree (Figure 2 shows the first steps of the workflow), starting from a set of DNA sequences, and assigns a



Paste an URL or select a file from the file system and click on Submit

url

file

```
<resultsSet>
<item>
>seq1
acgatcgatcgcgcggcgcatcgatcg
</item>
<item>
>seq2
acgatcgatcgcgcggcgatcg</item>
<item>
>seq3
acgatcgatcgcggcgcatcgatcg
</item>
```

sequence

usa

sformat gcg

sbegin

send

sprotein □

snucleotide □

sreverse □

sequence

usa

sformat gcg

sbegin

send

sprotein □

snucleotide □

sreverse □

Figure 2. Partial representation of the phylogenetic inference workflow in the mashup editor. Input and split widgets are shown at the top, while the bottom ones correspond to emma and eseqboot, the first two steps of the workflow.

[4] weblogo.berkeley.edu

bootstrap value to each node of the tree. It has been tested on a data-set comprising 600 DNA sequences (600 bp long) of the cytochrome oxidase subunit-one (COI) mitochondrial gene (Janzen *et al.*, 2005) belonging to organisms of the *Hesperiidae* family. Our workflow comprises a Data-Source widget and five WSDL Web-service widgets, each invoking one EMBOSS application:

- **emma** executes a multiple alignment across DNA sequences provided in FastA format;
- **eseqboot** generates multiple data-sets (alignments), which are resampled versions of the input data-set, necessary to compute the statistical significance of the final output phylogenetic tree;
- **ednadist** computes the distance matrix corresponding to the input alignment;
- **eneighbor** estimates phylogenies from distance-matrix data using the neighbour-joining or the UPGMA clustering methods;
- **econsense** returns the consensus phylogenetic tree.

Universal primer-design and validation workflow

In order to implement and accomplish the universal primer-design workflow (in Figure 3), we combined several bioinformatics tools able, on the one hand, to design universal primer-sets based on multiple DNA sequence data and, on the other, to validate the primer pairs obtained on the starting data-set. Primer universality is a crucial step in environmental sequencing studies, as the maximum number of organisms is targeted during PCR enrichment prior to sequencing. The workflow has been tested on a data-set of 64 DNA sequences, corresponding to the gene *ITS-1* of *Pucciniastraceae*, extracted from ITSoneDB (Santamaria *et al.*, 2012). A detailed description of the workflow steps is provided below:

- **emma** aligns the initial DNA sequence data-set;
- **cons** defines a consensus sequence corresponding to the multiple alignment;
- **einverted** controls inverted repeats on the consensus sequence;
- **extractseq** extracts a new consensus sequence free from repeated patterns, and keeps its length intact;
- **eprimer3** performs primer design, taking the newly obtained consensus sequence as template, and outputs a number of primer pairs having different characteristics (*e.g.*, GC content, linguistic complexity, PCR product length, *etc.*). At this step, users can choose the best primer pairs that fit their experiment;
- a final universality validation step is performed on the initial data-set by *in silico* PCR using the **primersearch** program. It is important to note that, in this last step, the mis-match percentage value can be changed according to experimental needs.

IBM Mashup Center is a flexible platform that can readily resolve bioinformatics issues. It can be seen as a collection of different tools and sources expressed as Java code, Web services, databases, Web applications and portals to fulfil the typical needs of bioinformatics researchers. The main benefit of the proposed platform

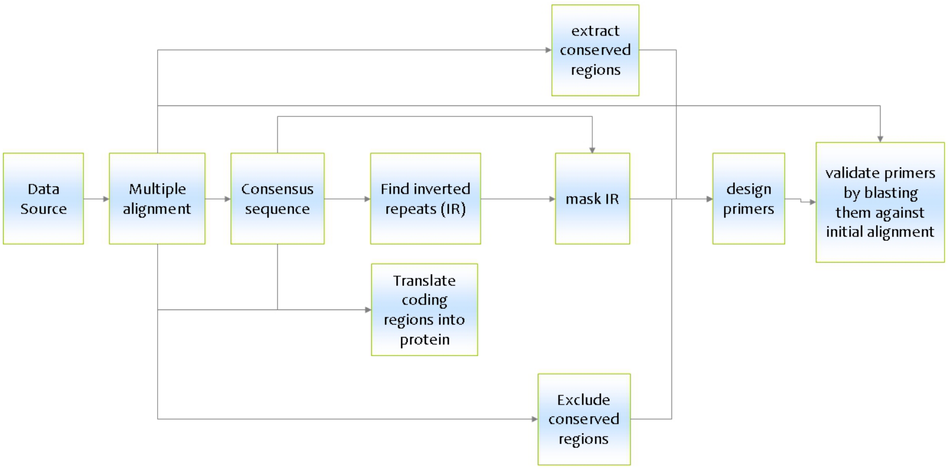

extract conserved regions
Data Source
Multiple alignment
Consensus sequence
Find inverted repeats (IR)
mask IR
design primers
validate primers by blasting them against initial alignment
Translate coding regions into protein
Exclude conserved regions

Figure 3. Schematic representation of the Primer Design Workflow, illustrating the basic actions computed by the workflow.

is its user-friendly interface to rapidly assemble tools and sources into a single workflow, and as an interface to different features provided by the MashupHub catalogue. Currently ongoing enhancements include optimisation of the implemented widgets by improving their performance, and the development of new widgets. In addition, the user interface will provide, in future, the possibility of easily creating user-defined Web services. This would facilitate the assembly of complex workflows completely tailored to users' needs.

Availability and requirements

The system was tested locally and is currently still a prototype. It will be released with its complete documentation and requirements once the above-mentioned optimisations have been achieved.

Key Points

- Bioinformatics workflows are built from different tools, each executing their own bio-computational tasks, working together in a standardised manner.
- Bioinformatics widgets are core dynamic elements of graphical user interfaces that contain embedded bioinformatics applications.
- Mashups are applications that integrate information from different data sources into a single new service.
- Bioinformatics widgets can be connected within a mashup framework to form a bioinformatics workflow.

Acknowledgements

The authors acknowledge support of the Ministero dell'Università e della Ricerca (MIUR), under the project DM19410 "Laboratorio di Bioinformatica per la Biodiversità Molecolare".

References

Ayhan S, Comitz P, Stemkovski V (2009) "Aviation Mashups" Digital Avionics Systems Conference. DASC '09. IEEE/AIAA 28th, 6.D.5-1, 6.D.5-9. http://dx.doi.org/10.1109/DASC.2009.5347436

Bentley DR, Balasubramanian S, Swerdlow HP, Smith GP, Milton J *et al.* (2008) Accurate whole human genome sequencing using reversible terminator chemistry. *Nature* **456**(7218), 53-59. http://dx.doi.org/10.1038/nature07517

Cheung KH, Yip KY, Townsend JP, Scotch M (2008) HCLS 2.0/3.0: Health care and life sciences data mashup using Web 2.0/3.0. *J Biomed Inform*, **41**(5), 694-705. http://dx.doi.org/10.1016/j.jbi.2008.04.001

Fichter D (2009) "What is a Mashup." In: Engard N (Ed.) *Library Mashups. Exploring new ways to deliver library data.* Medford, N.J: Information Today, Inc.

Goecks J, Nekrutenko A, Taylor J and The Galaxy Team (2010) Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol.* **11**(8), R86. http://dx.doi.org/10.1186/gb-2010-11-8-r86

Gong P. (2013). Dynamic integration of biological data sources using the data concierge. *Health Inf Sci Syst*, **1**(1), 1-19. http://dx.doi.org/10.1186/2047-2501-1-7

Hogan JM, Sumitomo J, Roe P, Newell F (2011). Biomashups: the new world of exploratory bioinformatics? *Concurr Comput*, **23**(11), 1169-1178. http://dx.doi.org/10.1109/eScience.2008.92

Janzen DH, Hajibabaei M, Burns JM, Hallwachs W, Remigio E *et al.* (2005) Wedding biodiversity inventory of a large and complex Lepidoptera fauna with DNA barcoding. *Philos Trans R Soc Lond B Biol Sci* **360**(1462),1835-1845. http://dx.doi.org/10.1098/rstb.2005.1715

Lushbough C, Bergman MK, Lawrence CJ, Jennewein D, Brendel V (2010) BioExtract server--an integrated workflow-enabling system to access and analyze heterogeneous, distributed biomolecular data. *IEEE/ACM Trans Comput Biol Bioinform* **7**(1), 12-24. http://dx.doi.org/10.1109/TCBB.2008.98

Margulies M, Egholm M, Altman WE, Attiya S, Bader JS *et al.* (2005) Genome sequencing in microfabricated high-density picolitre reactors. *Nature* **437**(7057), 376-380. http://dx.doi.org/10.1038/nature03959

Reguant C, Bordons A (2003) Typification of Oenococcus oeni strains by multiplex RAPD-PCR and study of population dynamics during malolactic fermentation. *J Appl Microbiol* **95**(2), 344-353. http://dx.doi.org/10.1046/j.1365-2672.2003.01985.x

Rice P, Longden I, Bleasby A (2000) EMBOSS: the European Molecular Biology Open Software Suite. *Trends Genet* 16(6), 276-277. http://dx.doi.org/10.1016/S0168-9525(00)02024-2

Saitou N, Nei M (1987) The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol Biol Evol* **4**(4), 406-425.

Sezici E (2009) New IBM Mashup Capabilities Bring Business Analytics to the Desktop. SYS-CON Media. http://sap.sys-con.com/node/1160750 (accessed 7 May 2015).

Santamaria M, Fosso B, Consiglio A, De Caro G, Grillo G et al (2012) Reference databases for taxonomic assignment in metagenomics. *Brief Bioinform* **13**(6):682-695. http://dx.doi.org/10.1093/bib/bbs036

St John K, Warnow T, Moret BME, Vawter L (2003) Performance study of phylogenetic methods: (unweighted) quartet methods and neighbor-joining. *J Algorithm* **48**(1), 173-193. http://dx.doi.org/10.1016/S0196-6774(03)00049-X

Wolstencroft K, Haines R, Fellows D, Williams A, Withers D *et al.* (2013) The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. *Nucleic Acids Res* **41**(Web Server issue), W557-561. http://dx.doi.org/10.1093/nar/gkt328