

## JavaEE for breakfast: start off on the right foot developing biological Web applications

Arnaud Ceol<sup>✉</sup>, Heiko Mülle

Center for Genomic Science of IIT@SEMM, Istituto Italiano di Tecnologia (IIT), Milan, Italy

Received 15 September 2013; Accepted 17 September 2013; Published 14 October 2013

Competing interests: the authors have declared that no competing interests exist.

### Abstract

Bioinformaticians may be intimidated by the apparent complexity of programming languages such as Java and stick to efficient but less suited languages for building rich applications and web sites. Nevertheless, with Java Enterprise Edition (EE), the part of programming is drastically reduced and many tools and frameworks allow for building professional applications rapidly. Indeed in order to access local or remote data, to extend existing software or build user friendly, and good looking, web interfaces, most of the pieces are already provided. The developer can assemble and finalise the applications with his/her own business classes, focusing on the biological concepts. Here we present an overview of some useful and easy to use tools and frameworks based on Java programming, and provide as an example a simple web interface to browse annotations associated with cancer samples provided by the Cancer Genome Atlas (TCGA).

### Motivation and Objectives

Bioinformaticians may be intimidated by the apparent complexity of programming languages such as Java and stick to efficient but less suited languages for building rich applications and web sites. Nevertheless, with Java Enterprise Edition (EE) the part of programming is drastically reduced, and many tools and frameworks allow for building professional applications rapidly. Indeed in order to access local or remote data, to extend existing software or build user friendly, and good looking, web interfaces, most of the pieces are already provided. The developer can assemble and finalise the applications with his/her own business classes, focusing on the biological concepts. Here we present an overview of some useful and easy to use tools and frameworks based on Java programming and provide as an example, a [simple web interface](#)<sup>1</sup> to browse annotations associated with cancer samples provided by the Cancer Genome Atlas (TCGA).

### Methods

Many tools and frameworks help Java programmers to build professional applications. Thanks to these tools, bioinformaticians with less experience in software development may forget about complex implementations of many features that are already provided and focus on the goal and design of the applications.

### Managing data easily

Biological data do not always require the complexity of a relational database. [Solr](#)<sup>2</sup>, a Java open source search platform, provides a complete and easy solution for storing data from heterogeneous formats (including text, XML, doc and pdf). The data is indexed and searchable in a "Google" like fashion. In addition, Solr provides a web interface to query and manage the data, and a web service to access it locally or remotely from a browser or a remote application (Figure 1a).

### Remote access to data

In the era of cloud computing, it is increasingly less necessary to store local copies of external voluminous databases. Indeed, many biological repositories provide computational access (web service) to their data that can be used to query and retrieve information from them directly from within a program or a web site. [Java EE](#)<sup>3</sup> provides a simple interface to implement a client to a web service (Figure 1b).

### Building an application and managing dependencies

One of the most tedious tasks in software programming, and even more so for the end user during the installation, is the management of dependencies. [Maven](#)<sup>4</sup>, a software management and comprehension tool allows to drastically simplify both processes. A configuration file stores the names of the packages which the ap-

1 <https://cru.genomics.iit.it/TcgaAnnotationBrowser>

2 <https://lucene.apache.org/solr/>

3 <http://www.oracle.com/technetwork/java/javaee/>

4 <https://maven.apache.org/>

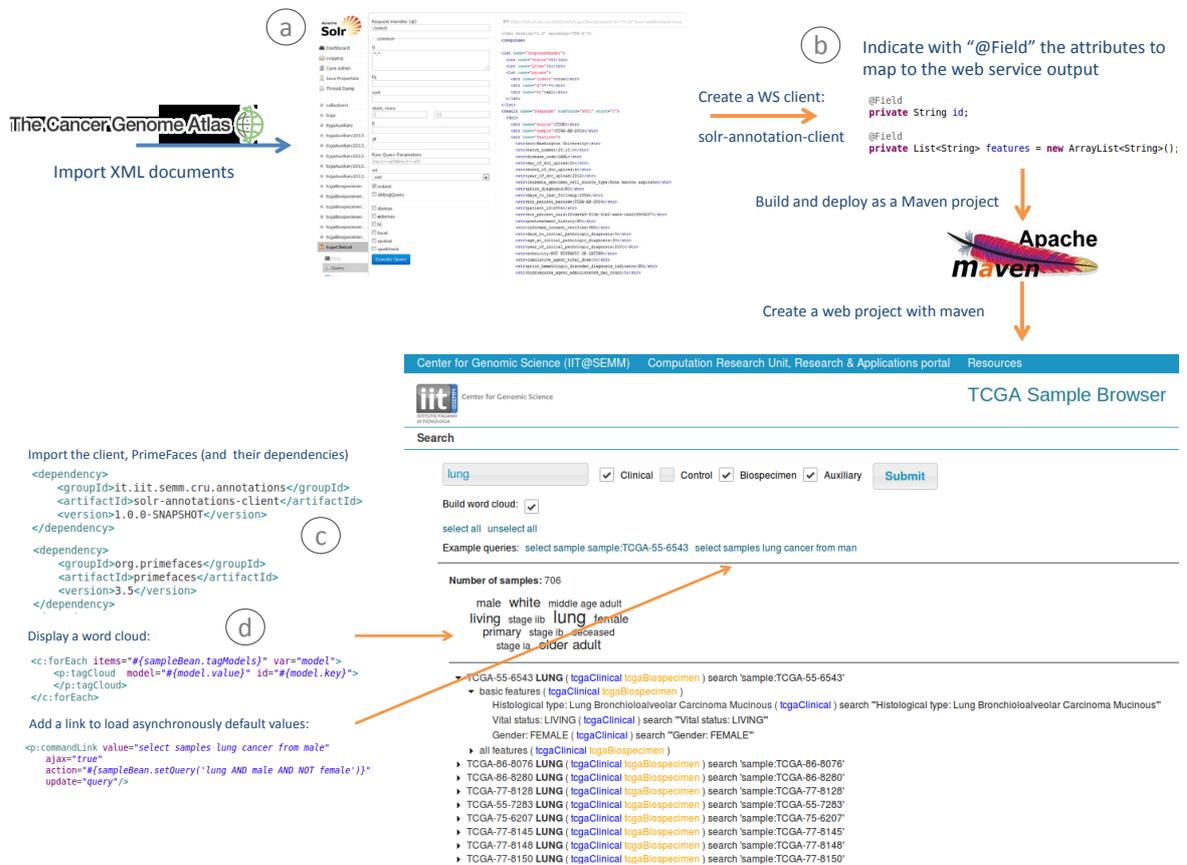


Figure 1. Building a web application. a) Annotations from TCGA are downloaded in XML format and imported into Solr. Solr includes a web interface that allows to query the data. b) Create a client to access the data on Solr. An annotation is enough for the client to understand which attributes should be mapped to the output of the web service (WS). c) Import the client and PrimeFace into the web application. With Maven, it is enough to add them into the configuration file. Maven will resolve and download all the dependencies for those packages. d) Create asynchronous links (ajax) or word cloud with PrimeFaces.

application directly depends on. Maven then takes care of downloading all those packages and their own dependencies. Any piece of software developed with maven can be successively imported in other Maven projects (Figure 1c).

### Building dynamic and good looking, web applications

When it comes to building dynamic and professional websites, the inexperienced bioinformatician may soon be lost in developing workflow of data between the server and the client, or digging and adapting complex JavaScript pieces of code. The last versions of Java EE include the JavaServer Faces (JSF) specification to help building web sites. On top of JSF, PrimeFaces<sup>5</sup> for example provides a user interface kit that can be

easily integrated into the user web pages to allow for drawing tables that can be ordered or filtered by the user, to update asynchronously part of the pages, or displaying word clouds (Figure 1d). In addition, it includes a component tool kit oriented to mobile devices.

### Results and Discussion

We have presented a series of tools based on Java technology that can help bioinformaticians to build professional applications. The tools and their usage is not exhaustive. We did not mention the possibility of extending existing Java applications with plugins (e.g. Cytoscape) (Saito *et al.*, 2012) and the Integrated Genome Browser (Nicol *et al.*, 2009), or the advantages of mapping relational databases to objects. These tasks have been made easy with the Java Enterprise Edition.

5 <http://primefaces.org/>

To illustrate this, we developed a [web interface for browsing the annotations of cancer samples from TCGA](#)<sup>6</sup>. The XML documents were imported into a local installation of Solr and implemented a client for the web service provided that was with Maven. This package is used by a web page that we developed using the PrimeFaces toolkit. Our client allows the web application to retrieve the annotations in a table, and the components from PrimeFaces display the results both as a table and as a word cloud that highlight the annotations most commonly associated to the cancer samples associated to the query.

Other tools and frameworks exist for other languages. [Django](#)<sup>7</sup> and [Symphony](#)<sup>8</sup>, for instance, make it easy to build dynamic websites with Python or PHP, and JavaScript libraries like [jQuery](#)<sup>9</sup>, on which PrimeFaces relies, can be integrated in any web page. But in our opinion none of those languages provides the same amount of tools and frameworks as Java does. They are easy to install and manage and are well supported by bioinformatics institutes like the European Bioinformatics Institutes (EBI). [EBI's Maven repository](#)<sup>10</sup> contains packages for accessing and managing data about proteins (Patient *et al.*, 2008), chemicals (Deng *et al.*, 2011), molecular interactions (Aranda *et al.*, 2011; Kerrien *et al.*, 2012) and many others. Biological libraries like BioJava (Prlic

*et al.*, 2012) or the structure visualization tool Jmol (Herráez, 2006) are similarly available.

We believe that the adoption of the technologies presented here can benefit to the whole scientific community by improving the quality of web tools while diminishing the time spent on software development.

## References

- Aranda B, Blankenburg H, *et al.* (2011) PSICQUIC and PSIScore: accessing and scoring molecular interactions. *Nat Methods* **8**(7), 528-529. doi:10.1038/nmeth.1637.
- Deng N, Zhang J, *et al.* (2011) Phosphoproteome analysis reveals regulatory sites in major pathways of cardiac mitochondria. *Mol Cell Proteomics* **10**(2), M110.000117. doi:10.1074/mcp.M110.000117.
- Herráez A (2006) Biomolecules in the computer: Jmol to the rescue. *Biochem Mol Biol Educ.* **34**(4), 255-61. doi:10.1002/bmb.2006.494034042644.
- Kerrien S, Aranda B, *et al.* (2012) The IntAct molecular interaction database in 2012. *Nucleic Acids Res.* **40**(Database issue), D841-846. doi:10.1093/nar/gkr1088.
- Nicol JW, Helt GA, *et al.* (2009) The Integrated Genome Browser: free software for distribution and exploration of genome-scale datasets. *Bioinformatics (Oxford, England)* **25**(20), 2730-2731. doi:10.1093/bioinformatics/btp472.
- Patient S, Wieser D, *et al.* (2008) UniProtJAPI: a remote API for accessing UniProt data. *Bioinformatics* **24**(10), 1321-1322. doi:10.1093/bioinformatics/btn122.
- Prlic A, Yates A, *et al.* (2012) BioJava: an open-source framework for bioinformatics in 2012. *Bioinformatics* **28**(20), 2693-2695. doi:10.1093/bioinformatics/bts494.
- Saito R, Smoot ME, *et al.* (2012) A travel guide to Cytoscape plugins. *Nat Methods* **9**(11), 1069-1076. doi:10.1038/nmeth.2212.

6 <https://tcga-data.nci.nih.gov/tcga/>

7 <https://www.djangoproject.com/>

8 <http://symfony.com/>

9 <http://jquery.com/>

10 <http://www.ebi.ac.uk/~maven/m2repo/>