B-Fabric: The Swiss Army Knife for Life Sciences

Can Türker, Fuat Akal, Dieter Joho, Christian Panse,
Simon Barkow-Oesterreicher, Hubert Rehrauer, Ralph Schlapbach

Functional Genomics Center Zurich, Winterthurerstrasse 190, CH–8057 Zurich, Switzerland
Contact: tuerker@fgcz.ethz.ch

ABSTRACT

This paper demonstrates B-Fabric, an all-in-one solution for two major purposes in life sciences. On the one hand, it is a system for the integrated management of experimental data and scientific annotations. On the other hand, it is a system infrastructure supporting on-the fly coupling of user applications, and thus serving as extensible platform for fast-paced, cutting-edge, collaborative research.

1. INTRODUCTION

Life sciences research more and more aims at characterizing complex biological organisms and functions at the systems level. To achieve this ambitious goal, i) data produced in different research projects and groups must be linked together and ii) the data must easily be usable as input of arbitrary applications that are coupled with the system on-the-fly. At the Functional Genomics Center Zurich (FGCZ) we have implemented such a "swiss army knife" for life sciences. B-Fabric [1] allows storing and annotating all data produced at FGCZ and offers a structured way of data retrieval for the user. Besides internal storage capacity, any external data store can be attached and made accessible via B-Fabric. Users do not need to care about where and how the data are kept. B-Fabric captures and provides the data transparently and in access-controlled fashion through a Web portal. Using its search and browse features, inter-experiment and inter-project analyses become possible. Since experimental data is captured together with annotations like instrument and processing parameters, experiments become reproducible for third parties. Besides, B-Fabric allows to dynamically couple external applications with the system, whether be commercial packages or in-house developments. Once an application is registered with B-Fabric, users may invoke and feed the application via B-Fabric. Through application registration, the functionality of B-Fabric can be extended at run-time without changing the core code base. Research centers like the FGCZ that have both researchers with bioinformatics background and users without profound computer knowledge can benefit from such an infrastructure since it provides an easy way for the users to invoke and feed applications and scripts from the bioinformaticians with data from the B-Fabric repository.

For the interpretation and reuse of experimental data, scientific annotations are crucial. Based on our experience with many bioinformaticians and researchers and their practical experiences and difficulties with using standard schemas MIAME or Gene Ontology, we decided to apply a "minimal" metadata schema approach for B-Fabric (see Figure 1).

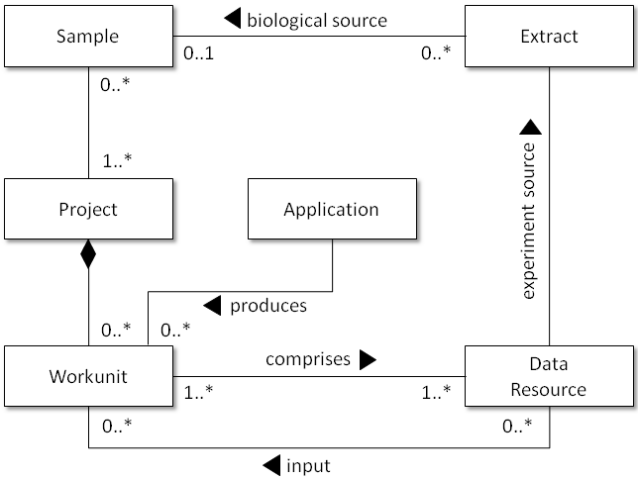

Figure 1: Core of B-Fabric's Metadata Schema

A *data resource* is an abstraction of a file or link to a file. Examples for data resources are raw files produced from a mass spectrometer or cel files generated from an array scanner. Each data resource is connected to an *extract* representing the biological input into the experiment or measurement that produced the data resource. We distinguish between samples and extracts describing the biological sources at different levels. The *sample* contains general information about the biological source while the extract represents an extraction of that source which actually is used for the experiment or measurement. There might be several extracts of one sample. These extracts might be the result of different extraction procedures. To ease the finding and reuse of sample and extract information, each sample and indirectly each extract are associated with a *project*. This information helps to significantly reduce the set of values in drop-down menus, for instance, to associate a data resource with an

EDBT 2010, March 22–26, 2010, Lausanne, Switzerland.

extract. As a result of extensive discussions between the bioinformaticians and researchers at FGCZ about the question what primary (and secondary) data should be stored and especially what this data actually represents, the generic concept of a workunit was found in B-Fabric. A *workunit* is an abstraction that can be used to represent the result of an experiment, a measurement, an analysis, a search etc. In principle, a workunit is a container referencing to data resources that logically form a unit. Some of these data resources are marked as input resources meaning that they were the inputs of the processing step (*application*) that created the remaining data resources. The scientist individually decides what a workunit should represent.

2. DEMONSTRATION

In this demonstration, we sketch some central features of B-Fabric. As example scenario, we use a scientist who is working on a plant named Arabidopsis Thaliana with the goal to figure out the effect of certain gene and the effect on light on it. For this purpose, he registers his samples and extracts with B-Fabric, loads his data into B-Fabric and defines his experiment. Afterwards, he runs his experiment and stores the results in B-Fabric. To complement this scenario, retrieval and administrative issues are being demonstrated as well. Screen-shots are used to illustrate the different steps. Due to space limitations and to avoid clutter, the figures are clipped accordingly.

Register Samples/Extracts. Users register their samples and extracts through intuitively designed forms (see Figure 2 and 3). Data entering is facilitated by providing as much drop-down menus as possible to select annotations from the system vocabularies and by dynamically drawing forms according to selected annotation values.

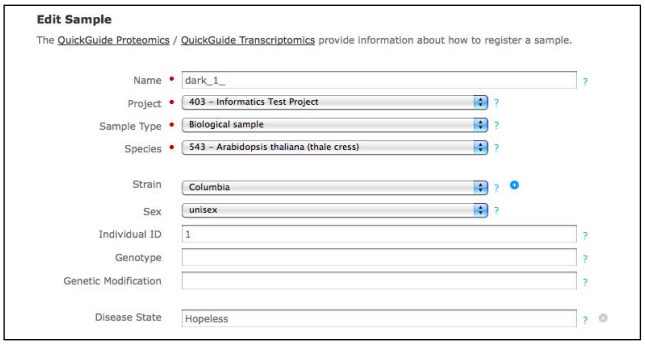

Figure 2: Register Sample

In addition, users typically register several samples and extracts where only a few attributes differ. In order to further ease the registration of them, cloning as well as batch registration of samples and extracts are supported.

Annotation Management. B-Fabric provides extensible vocabularies for the different annotations. If a user does not find a needed annotation in the corresponding drop-down list, the user can create a new one. In Figure 2, the scientist adds a new annotation *Hopeless* for the attribute *Disease State* of the sample. All annotations created by users must be reviewed by an expert (in our case by an FGCZ employee). The expert checks the annotation and releases it if it is correct, as depicted in Figure 4.

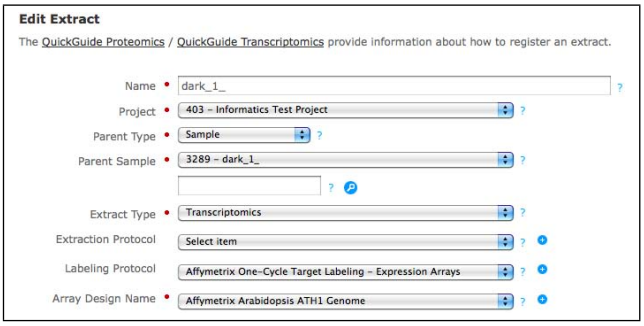

Figure 3: Register Extract

Figure 4: Release Annotation

Annotation reviewing can be a tedious task due to similarly written versions of the same annotation. In the example, we assume that some another scientist looked for the disease state annotation given above while registering his sample. Let us assume that he cannot find the annotation and recreates it. Besides, he misspells the annotation as *Hopeles*. In such cases, B-Fabric automatically detects similar annotations and recommends merging them, as seen in Figure 5. If asked so, similar annotations can be merged easily to maintain the annotation consistency system wide.

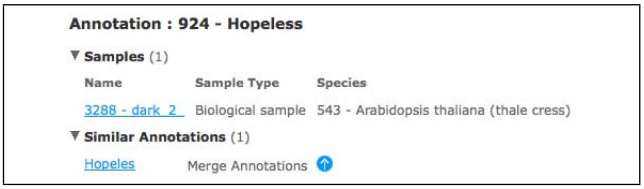

Figure 5: Annotation View

If the expert decides to merge two annotations, B-Fabric provides a form where he can easily select the attributes of the resulting merged annotation (see Figure 6).

Figure 6: Merge Annotations

When the two annotations merged, B-Fabric automatically associates the samples which were previously associated with the misspelled annotation (see Figure 7).

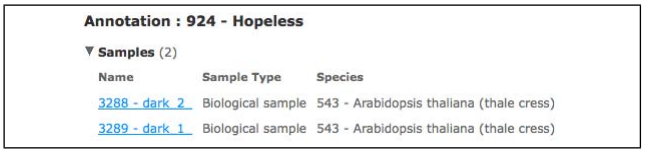

Figure 7: Merged Annotations

Task Orientation. B-Fabric is a task-oriented system that reminds its users about open tasks, awaiting to be performed next. In the create annotation example above, the question for the experts is when to release annotations. In fact, as soon as a new annotation is added to the vocabulary, a new task to release this annotation appears in the task list of the corresponding expert (see Figure 8).

Figure 8: Tasks List

Data Import. B-Fabric supports two ways of data import: 1) physically copying and 2) linking data files. To import data from a data source, a proper data provider must be configured. The B-Fabric deployment at FGCZ allows importing data files from local file systems as well as several instruments available at FGCZ. New data providers can be added to the system easily. With the configuration of a data provider the selection of the data files in corresponding data stores can be restricted to the ones that are potentially relevant for the user. This is a crucial feature since the number of the data files can be huge. An import results in a workunit. A workunit thus represents a unit of logically related data files. Figure 9 shows the screen where a workunit is created by fetching files from the Affymetrix GeneChip instrument, which is an instrument already known to B-Fabric.

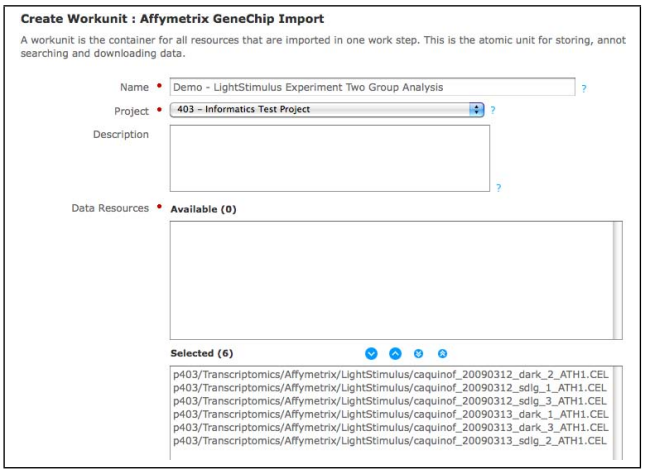

Figure 9: Create Workunit

B-Fabric implements the data import via workflows. With the initiation of a data import, the corresponding workflow becomes visible to the user. The next step to be taken by the user is highlighted in the graphical representation of the workflow. In data import workflow, for instance, the user must assign extracts to the imported files. The workflow-driven approach of B-Fabric is very useful in practice to reduce human mistakes and avoid skipping steps.

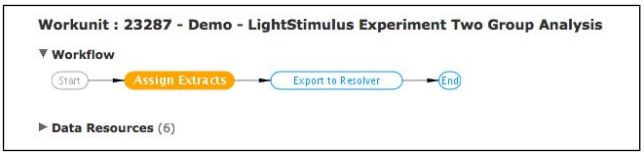

Figure 10: Assign Extracts Workflow

Assigning extracts to data resources also comes with some intelligence in B-Fabric. When the scientist goes to the assign extracts screen, he gets already the best matches between data resources and extract names. Typically he just needs to press the save button and continue.

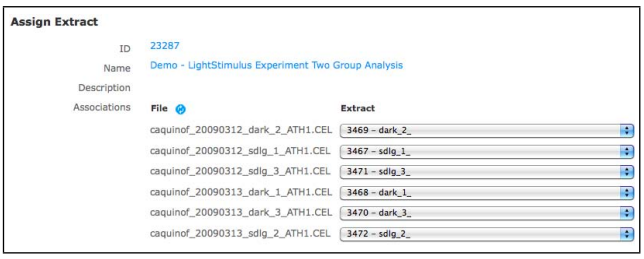

Figure 11: Assign Extracts

Application Integration. Integration of external functionality into B-Fabric is done via *application registration*. First, a connector is written for a certain type of application, e.g., for running *R* scripts on an *Rserve* system. Then, a small interface is defined to describe how the application gets its input (see Figure 12). Finally, the scientist writes the application in any language. This on-the-fly coupling of external applications is a crucial feature of B-Fabric, which allows fast evolution of the system.

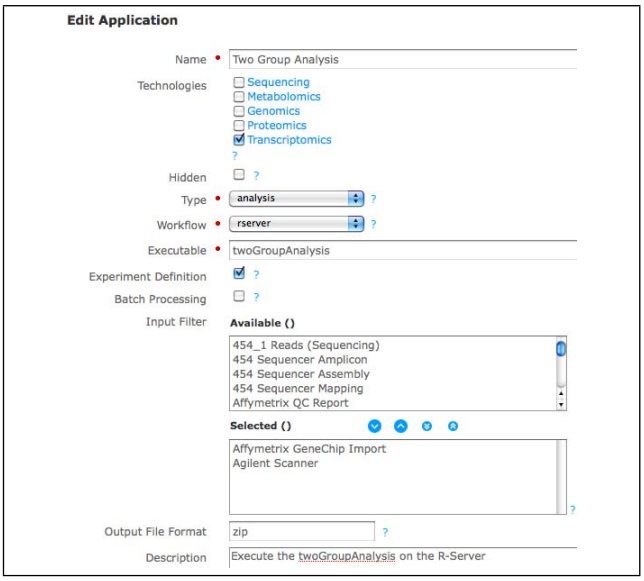

Figure 12: Application Registration

Once an application is registered, an experiment can be created to run this application. As an example, Figure 13 shows the definition of the experiment that will be conducted on the Arabidopsis Thaliana plant as mentioned earlier in this section. Defining an experiment consists of a selection of data resources, samples, extracts, and arbitrary number of attributes (e.g. species and treatment in the example.) that will be used as input for the application.

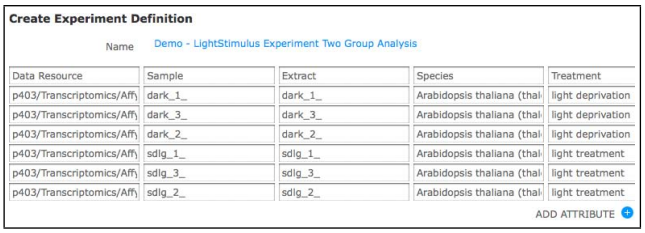

Figure 13: Create Experiment Definition

Figure 14 shows how easily a previously registered application (two group analysis) can be invoked to conduct the desired experiment. This step requires a name for the resulting workunit which contains the result files of the application along with specific parameters regarding the experiment, e.g. reference group.

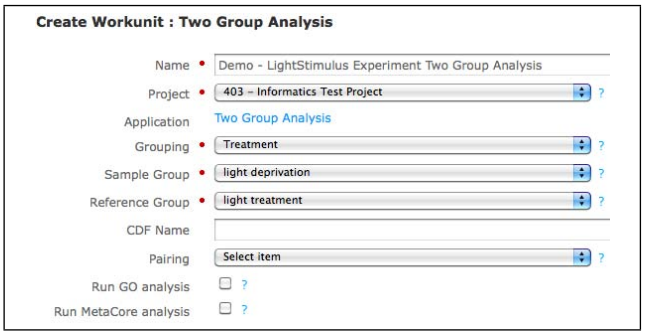

Figure 14: Run Experiment

Once the experiment is started, a corresponding workflow is initiated. The graphic presentation of the workflow is also used to show what is happening underneath in the system. The example workflow (generate an R report) is quite simple and consists of a single step (see Figure 15). Note that B-Fabric supports arbitrary complex workflows based on its underlying workflow engine (OSWorkflow).

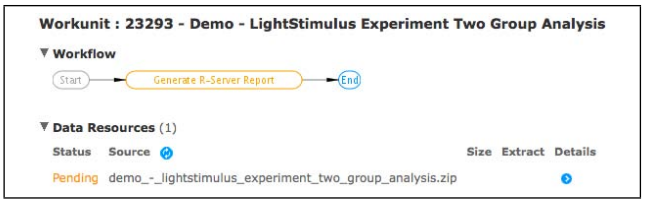

Figure 15: Run Experiment - Pending State

When the experiment is done, the scientist can view the experiment results by clicking the proper link on the screen (see Figure 16). The results of the experiment is also presented to the user as a zip file so that they can easily be transferred to another medium.

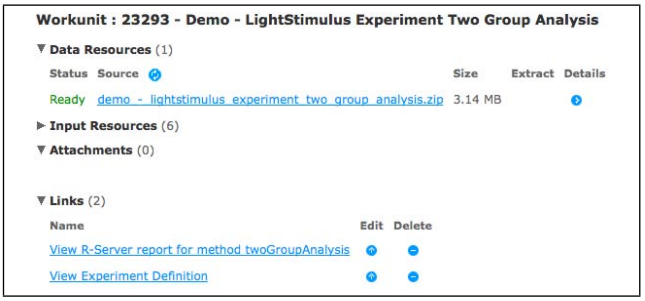

Figure 16: Run Experiment - Ready

Full-text Search. B-Fabric provides full-text search capabilities. A search may vary from certain attributes of certain objects to the content of readable attachments and data resources. The system provides quick search boxes on the main screen as well as more refined advanced search form. Searches done by the user are kept in the search history during his session and can be executed easily by selecting a search query from the search history. A query can also be saved for future reuse. A later invocation of such a saved query will of course include all objects satisfying the query at run-time. Another important feature of B-Fabric is that search results can be exported into files.

Miscellaneous Functions. In addition to all major aspects presented above, B-Fabric provides some additional functionality. Especially, B-Fabric supports a view on the main data objects in a networked fashion. Users can simply browse bidirectionally through all objects linked together. In addition, all data manipulation operations (create/update/delete) are logged in the system such that the user can remember what he did in the past and the system can be monitored. Last but not least, B-Fabric provides a bunch of administrative functions to manage objects, workflows, errors, and maintain the system.

Final Remark. B-Fabric is running in daily business at FGCZ since beginning of 2007. Here are some figures about the FGCZ deployment as of January 2010:

Users	1555	Samples	3151
Projects	750	Extracts	3642
Institutes	224	Data Resources	40005
Organizations	59	Workunits	23979

Acknowledgement: The further development of B-Fabric is partially supported by SWITCH (Swiss Academic and Research Network) within the AAA/Switch project "Generalizing B-Fabric towards an Infrastructure for Collaborative Research in Switzerland" (UZH.5, June 2009-May 2011).

3. REFERENCES

[1] C. Türker, E. Stolte, D. Joho, and R. Schlapbach: B-Fabric: A Data and Application Integration Framework for Life Sciences Research. *Data Integration in the Life Sciences, DILS 2007*, LNCS 4544, pp. 37–47. Springer-Verlag, 2007.

[2] B-Fabric. `http://www.bfabric.org/`