EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Data Management for Quantitative Biology

Project Paper

Project 1: Data Modelling and Visualization

SS 2015

Sebastian Goerges
Benjamin Schroeder
Nils-Oliver Schliebs

1 Background

Data management in life sciences connects different fields and people with different scientific backgrounds. One challenge, especially when data needs to be collected and presented, is to keep track of the metadata that is important for analysis and reproducibility of an experiment, while exposing only informative and easy to understand data and metadata to users that are not concerned with computation or data management. Hence, for a given data set, the project task was to define what parts of data should be shown to end users (life scientists) and to implement a web-based Graphical User Interface, using Java, to visualize given data. Users should also be able to add some annotation on their own and download a summarizing report of the results. In addition, the user interface should be intuitive and well documented to guide even first-time users easily through the task of data annotation.

2 Material and Methods

2.1 Data

The main data structure is provided in form of TSV-files. The file *projects.tsv* contains multiple projects together with several attributes, describing the project. Projects themselves can contain multiple different experiments (*experiments.tsv*), which in turn contain two different types of samples (*QCOFF.tsv* and *QMOUSE.tsv*). *QCOFF* is storing samples of a coffee diversity project, whereas *QCOFF* consists of samples for a mouse knockout project. The set of samples can be structured hierarchically, with patients/organisms at the top (Entity samples) from which tissue/cell samples are derived. Experiments or samples can contain datasets, stored in *datasets.tsv*. Datasets link to one or more files. These files are for instance further descriptions in plain text, images, quality control HTML documents or FASTQ files. All structures of the system contain unique identifiers to connect the set of TSV-files together in a logical way.

2.2 Java and Vaadin

The graphical user interface (GUI) was developed using the open source web application framework Vaadin 7. Vaadin enables the possibility to build single page web apps in server-side Java. All of the browser–server communication and data transfer objects are automated by the framework. The app's state resides on the server, but the end-users use an HTML5 web app in their browsers. In addition, Vaadin's default component set can be extended with custom Google Webtoolkit Widgets (GWT) and designed over cascading style sheets (CSS) [1]. In addition to the common vaadin distribution, an extra addon, that provides with basic PDF and Excel export functionality, was used for the report generation [2].

3 Results

3.1 Data Parser

The parser was programmed to be able to import any database which shares the same structure of the bot example databases. The input for the parser is a file system, containing .tsv files. The necessary .tsv files are datasets.tsv, experiments.tsv and projects.tsv. Additionally some .tsv files containing a sample list. The parser creates out of these .tsv-files lists containing objects, which are further processed for the visualisation.

3.2 Project Visualizer

The resulting user interface tries to simplify the provided collection of data. Therefore, an clear navigation has been designed, to enable also new users handling all components of the web application in a fairly easy way. When the tool is started, all existing data is loaded in the background and available projects are displayed in a select-able list. If one choose a particular project, additional project information is shown in a extra table. In addition, a further select-able list appears, displaying all experiments under the project (Fig. 1).

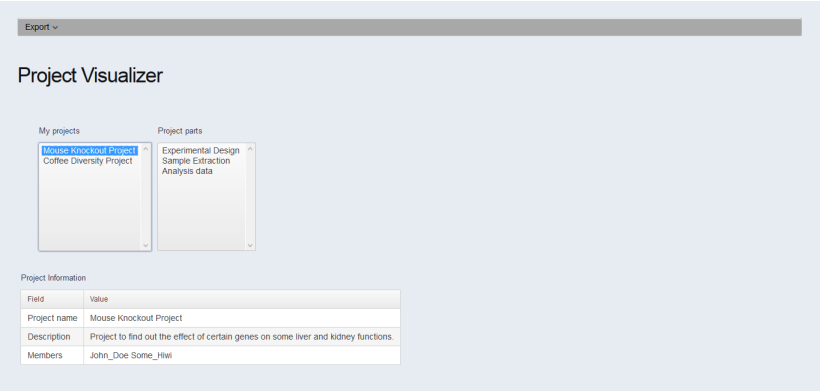
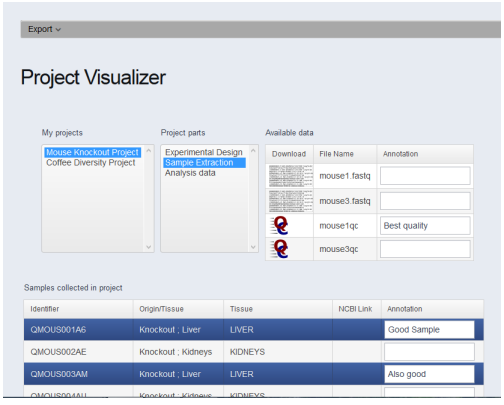
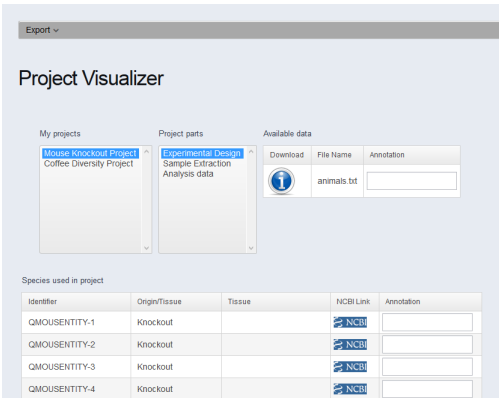


Figure 1: The GUI at the project level. Here, users can make a decision between various available projects and inspect additional information.

If the user clicks on a specific experiment, two additional tables are visualized (Fig. 2). One table showing available data files that are stored under a specific experiment and another one providing all existing experimental samples. The sample table has a multi selection feature, enabling the possibility to inspect the data sets that available for multiple samples. For instance, this can be seen in Fig. 2b. Here, two samples (QMOUS001A6 and QMOUS004AM) are selected, leading to a view at fastQ-files and quality control sites for each of the selected samples. All those datasets are click- and downloadable directly from the GUI.



(a) Example for a experimental design.



(b) Example for a sample extraction.

Figure 2: The GUI at the sample level. Here, users can inspect the collection of samples together with stored data sets.

A additional feature is the possibility to add some annotations for samples and datasets behind each row. Here end-users might leave some additional information, that is later included into to report for

exportation. Additionally, the sample table comes up with an external connection to the NCBI Taxonomy Browser [3], if an taxonomy ID is deposited for a sample (Fig. 2b).

3.3 Data Exporter

With the aim to provide users with a printed summary of the visualized information, a data exporter has been developed. On clicking at the export menu bar, a new subwindow pops up, showing summarized information about a specific project in a new table. For instance, the project details, as well as the number of species together with their conditions for the specific experiments, are presented. As an extra feature, all samples/datasets, that have been annotated by the user, are also included together with their annotation into the summary table. An example for this can be seen in (Fig. 3) based on the navigation and annotation of Fig. 2b). At the end, the user has two option in form of press-able buttons. One button exports the summary table in Excel format, the other one generates a PDF download.

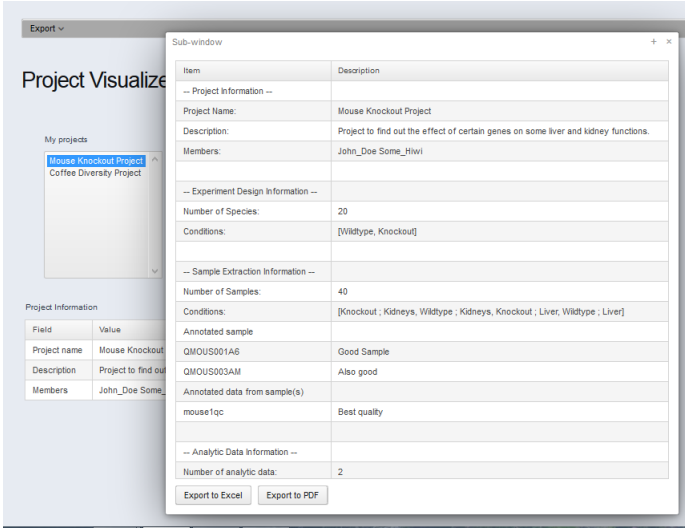


Figure 3: The export function. Here, inspect and download a project summary in Excel or PDF format.

4 Discussion

The complete project was structured in different phases. The first phase was the theoretical planning, followed by the practical execution of the plan. The planing of the project discussed first: how to visualize and integrate the data. Many questions arose in this phase, like: which data has to be visualised and which should not be visualised, how data is made easy accessible and what is a logic structure for a user to navigate through the file system. In general only data serving for an effective information gain, is important to the user on a general interface. All other data, might be important for the program or further research, and therefore should not be visualised. Out of this statement results a new problem, as the decision between data for the program and biological information is easy.But it is not easy to decide, what is categorized as additional information and what is information which should be visible on first sight. In context of universality, we decided to visualize only information which is necessarily contained in each database. This is not an optimal information density on the gui, but the only possible in this short time period. For further projects we would advice to define different experiment types. For example a mouse experiment, which shows the number of mice used or the wild types and so on. Generally the databank design which was provided is discussable. One problem is, that there could be more than one

Experiment for one project. Our suggestion is to add a new branch experiments between “My projects” and “Project parts”, which should get “experiment-parts”. To the new branch the experiment contents like design, samples and analysis should be outsourced to the experiment parts (4). For the navigation through the database we decided to use a guide tree, which only shows the children of a node. This is in respect of the overview of the data. Different approaches like showing complete lists were neglected, because there would be to many rows.

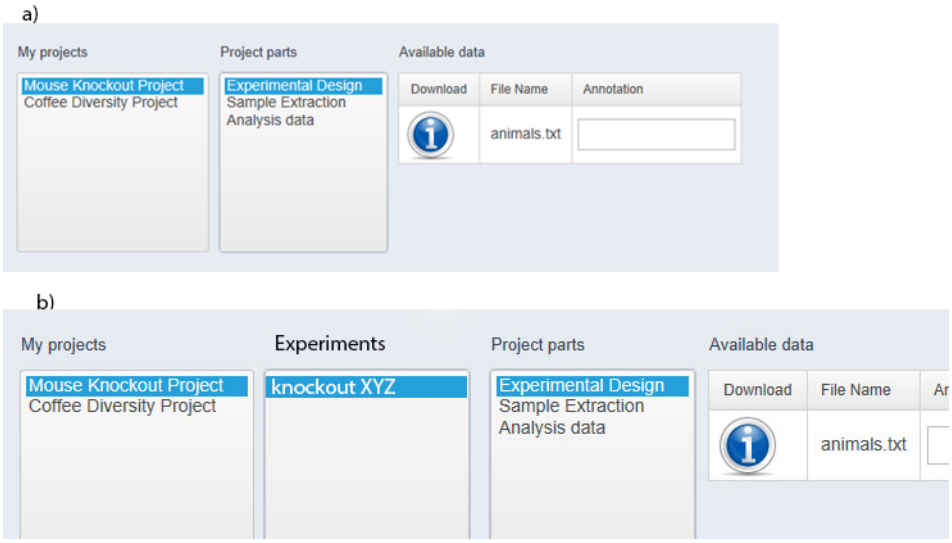


Figure 4: The current state is shown in a), fo a better system would be b).

References

[1] Vaadin Ltd. vaadin - user interface components for web apps, 2015.

[2] Haijian Wang. vaadin - exporter version 0.0.5.5, 2014.

[3] Eric W. Sayers, Tanya Barrett, Dennis A. Benson, Stephen H. Bryant, Kathi Canese, Vyacheslav Chetvernin, Deanna M. Church, Michael DiCuccio, Ron Edgar, Scott Federhen, Michael Feolo, Lewis Y. Geer, Wolfgang Helmberg, Yuri Kapustin, David Landsman, David J. Lipman, Thomas L. Madden, Donna R. Maglott, Vadim Miller, Ilene Mizrachi, James Ostell, Kim D. Pruitt, Gregory D. Schuler, Edwin Sequeira, Stephen T. Sherry, Martin Shumway, Karl Sirotkin, Alexandre Souvorov, Grigory Starchenko, Tatiana A. Tatusova, Lukas Wagner, Eugene Yaschenko, and Jian Ye. Database resources of the national center for biotechnology information. *Nucleic Acids Research*, 37(suppl 1):D5–D15, 2009.