ICT for Health Science Research
A. Shabo (Shvo) et al. (Eds.)

doi:10.3233/978-1-61499-959-1-46

Developing New Analysis Functions for a Translational Research Platform: Extending the cBioPortal for Cancer Genomics

Philipp UNBERATH [a,1], Christian KNELL [a], Hans-Ulrich PROKOSCH [a] and Jan CHRISTOPH [a]

[a] *Medical Informatics, University Erlangen-Nürnberg, Erlangen, Germany*

Abstract. *Background:* The cBioPortal is a prevalent open-source translational research platform, allowing private instances and extensions. *Objective:* Our aim was to build up an own instance of cBioPortal, identify missing functionality by interviewing researchers, and implementing these extensions. *Methods:* We examined the code base of the cBioPortal and conducted a requirements analysis with researchers. Then an own extension was implemented and a usability evaluation was performed. *Results:* We developed a new tab in the results view of cBioPortal adding the option to analyze the correlation of gene expression and mutation patterns. *Conclusion:* While extending the cBioPortal is possible, there are still some challenges to overcome. A plug-in concept and a more detailed documentation would greatly facilitate the development of own extensions.

Keywords. cBioPortal, genomics, translational research, MIRACUM, MI-I

1. Introduction

The cBioPortal for Cancer Genomics, developed by the Memorial Sloan Kettering Cancer Center (MSK) [1,2], has become a widely used research platform for exploring cancer genomics data, generating research hypotheses, and translating omics data into new biomedical insights and clinical applications. The MSK hosts a public version of the cBioPortal (cbioportal.org), which can be accessed to analyze the data of the constantly growing number of currently over 230 contained cancer studies.

Since cBioPortal is developed as an open-access, open-source resource, the whole study database as well as the source code are publicly available. Therefore, anyone can set up a private instance of cBioPortal and customize or extend it to suit their local needs. This can also be used for importing own data and analyzing it with built-in features.

Whereas for the setup of and data loading in cBioPortal extensive manuals and work-simplifying scripts exist, there are no publicly available resources on how to extend the

[1]Corresponding Author: Philipp Unberath, Chair of Medical Informatics, Friedrich-Alexander-Universität Erlangen-Nürnberg, Wetterkreuz 13, 91058 Erlangen, Germany. E-Mail: philipp.unberath@fau.de

portal with own analysis methods. As the cBioPortal currently also does not feature any sort of plug-in concept, the barrier of creating own extensions is significant.

While cBioPortal provides core functionality that is likely useful to the overall research community, we hypothesize that local groups may need or want more specialized analysis tools that are customized to their particular needs. Therefore, we aimed at analyzing the requirements of local researchers and at developing and evaluating an own extension to cBioPortal.

2. Methods

A requirements analysis was conducted with translational researchers to determine whether cBioPortal supported their uses cases out of the box or if extensions to cBioPortal had to be implemented. We interviewed in total six researchers of three different departments (Experimental Medicine (EM), Molecular Surgery, and Molecular Medicine) to present the current functionality of cBioPortal and to inquire and document additional needs.

An own local instance of cBioPortal was set up and the public database as well as data from a locally conducted study was imported. We then implemented our first prototypical modification in the front end, a new tab in the patient view. Our setup was based on cBioPortal v1.11.3, which was the latest version at the beginning of our work in February 2018.

We analyzed the software architecture of the cBioPortal using available publications of the developers from the MSK. Additionally the code base and the flow of data was manually examined using the Co-Expression tab of the query result view as an example.

Subsequently we took the Co-Expression tab as a template for our own analysis tab while adapting the visualization and parametrization in the front end as well as functionality and data access in the back end. The main focus of the analysis is generating new research hypotheses and not statistically analyzing the data.

To evaluate our extension, a small-scale usability evaluation with three researchers from the EM was performed. It consisted of a think-aloud protocol and conducting the System Usability Scale (SUS) questionnaire [3]. The interview focused on the question whether the analysis method of our new extension produces the anticipated results.

3. Results

The requirements analysis revealed that there was no need for major changes to existing cBioPortal features, but there were requests to import own study data to benefit from the already available extensive analyses. Researchers from the EM had a demand for special analyses, consisting of analyzing the correlation of gene expression and mutation patterns, a feature that is currently not supported in the cBioPortal[2].

[2]Technically, part of the specific analysis is possible in the cBioPortal, but involves multiple individual queries with different sample lists and is therefore neither efficient nor easily parameterizable.

Table 1. Locations of the used objectives to extend the result view

Category	Objective	Location
Front end	Layout	portal/src/main/webapp/WEB-INF/jsp/
	Functionality	portal/src/main/webapp/js/src/
	Register tab	portal/src/main/webapp/WEB-INF/jsp/visualize.jsp
Communication	Register servlet	portal/src/main/webapp/WEB-INF/web.xml
Back end	Servlet	core/src/main/java/org/mskcc/cbio/portal/servlet/
	Utility	core/src/main/java/org/mskcc/cbio/portal/util/
	Data access	core/src/main/java/org/mskcc/cbio/portal/dao

3.1. Software Architecture

The cBioPortal is structured as a typical Java web application, consisting of the presentation tier running in the client's browser, the business tier containing all the back end logic at the server, and the persistence tier, responsible for data access and storage. The corresponding locations in the code base are outlined in Table 1.

Since late 2016 the cBioPortal is also split in the code base between front and back end. As this separation is an ongoing process, all of the tabs belonging to the results view page are still in the back end project.

3.2. Extension

Our implemented extension provides a parameterizable analysis of the correlation of gene expression and mutation patterns in a given cohort.

Front End The tabs of the result view are mounted as Java Server Pages from the back end to the front end. The functionality is implemented using Javascript while D3.js is used for generating the plots.

Our extensions consists of three sections: parametrization, table, and plot view (Figure 1). For maximal consistency to the overall style of cBioPortal we tried to reuse existing parts of the Co-Expression tab. Multiple drop down selections for further parametrization were added at the top of our tab. Since our analysis includes comparing two subgroups the existing table view was duplicated and customized to show the results of the back end analysis. The existing Co-Expression plot was adapted and extended for visualizing our analysis results.

Communication & Back End Requests to the back end are submitted to the internal REST interface via GET or POST requests. They are processed in the Java Servlets, which contain the logic of the specific analysis. Parameters can be transmitted as the payload of the request and analysis results are transferred in the request response.

Data Access The cBioPortal uses a MySQL database for storing and managing all study data. For retrieving mutations or Copy Number Alterations (CNA), cBioPortal takes the approach of querying either specific genes or getting alterations in one slice after the other. For our analysis all mutations/CNAs for all samples of the selected study were needed, which is not supported by the cBioPortal data access layer. Therefore a custom method using SQL to retrieve all alterations of all samples at once was implemented.

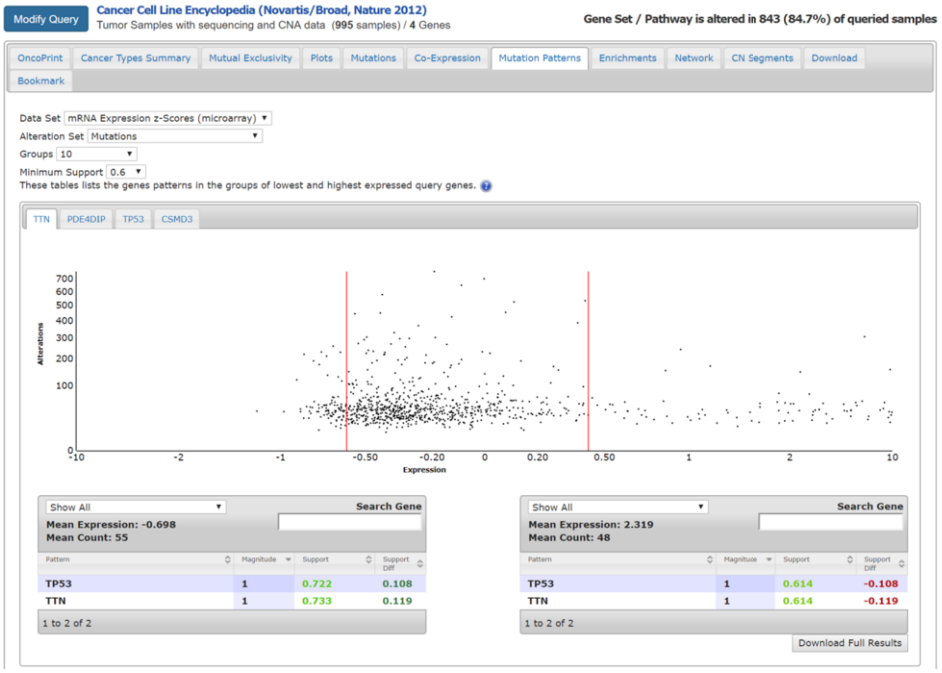

Figure 1. The new tab in the result view of cBioPortal.

3.3. Evaluation

The three interviews revealed several smaller issues with the layout of the results view and it was reported that some description labels should be phrased more clearly. The results were described as promising, although they still have to be statistically validated. The evaluation of the three questionnaires yielded SUS score of 82.5 with a standard deviation of 6.6.

4. Discussion

A major challenge when developing extensions for the cBioPortal is the lack of a plug-in concept for simplified extensibility. Another challenge consists of getting a detailed overview of the software architecture, as rapid growth and change in the cBioPortal obstruct extensive documentation.

While our current usability evaluation with three participants only generates limited insights in the usability of our extension, it provided valuable feedback to be able to expose serious flaws in the user interface. Therefore we plan on performing an in-depth usability evaluation after our extension passed this first user test.

Since the cBioPortal is an active open source project, the code base is constantly changing and being improved. In June 2018 with the release of cBioPortal v1.14.0 the Co-Expression tab, which was used for the code analysis and as an template, was moved to the new architecture, meaning it was translated from Javascript to Typescript and moved to the front end project. This basically eliminates the critique of having to imple-

ment front end code in the back end, but for us it also means that our extension has to be translated and adapted as well.

4.1. Related Work

Prior to implementing a local instance of the cBioPortal, an own instance of tranSMART was introduced at our medical faculty to serve as a platform for translational research. While both platforms have different areas of focus, tranSMART has one evident benefit: the SmartR framework allows users to create plug-in based extensions and therefore facilitates the implementation of own extensions [4].

4.2. Conclusion and Outlook

For the future development of cBioPortal it would be highly desirable to keep major architecture changes - including the change of programming languages - to a minimum. This would facilitate the integration of longer-lasting implementations of new features, without the need to contribute them as early as possible.

The whole implementation process would be greatly simplified with a plug-in concept in cBioPortal. This would enable the generation of new analysis methods without a deep insight into the overall cBioPortal architecture.

We plan on using the cBioPortal further on as the base for the development of a platform to support the molecular tumor board. Therefore we would greatly benefit of an extensible design to more efficiently implement the needed extension for clinical use [5].

5. Acknowledgments

We thank Prof. Dr. Thomas Brabletz for providing feedback during the requirements analysis and the evaluation and the team of the MSK for support with technical questions. This research has been conducted within the MIRACUM consortium. MIRACUM is funded by the German Federal Ministry of Education and Research (BMBF) under the Funding Number FKZ 01ZZ1801A. The present work was performed in (partial) fulfillment of the requirements for obtaining the degree "Dr. rer. biol. hum." from the Friedrich-Alexander-Universität Erlangen-Nürnberg (PU).

References

[1] Cerami, Ethan, et al. "The cBio cancer genomics portal: an open platform for exploring multidimensional cancer genomics data." Cancer Discov 2 (2012): 401-404.

[2] Gao, Jianjiong, et al. "Integrative analysis of complex cancer genomics and clinical profiles using the cBioPortal." Sci. Signal. 6.269 (2013): pl1-pl1.

[3] Brooke, John. "SUS-A quick and dirty usability scale." Usability evaluation in industry 189.194 (1996): 4-7.

[4] Knell, Christian, et al. "Developing interactive plug-ins for tranSMART using the SmartR framework: the case of survival analysis." Stud Health Technol Inform 236 (2017): 375-382.

[5] Hinderer, Marc, et al. "Supporting Molecular Tumor Boards in Molecular-Guided Decision-Making–The Current Status of Five German University Hospitals." Stud Health Technol Inform 236 (2017): 48-54.