Improving the Swiss Grid Proteomics Portal

Requirements and new Features based on Experience and Usability Considerations

Peter Kunszt, Lorenz Blum,
Béla Hullár, Emanuel Schmid,
Adam Srebniak, Witold Wolski
SystemsX.ch SyBIT, ETH Zurich
and Swiss Institute of Bioinformatics
Zurich, Switzerland

Bernd Rinn, Franz-Josef Elmer,
Chandrasekhar Ramakrishnan
Center of Information Sciences and
Databases, ETH Zurich, Department of
Biosystems Science and Engineering
and Swiss Institute of Bioinformatics
Basel, Switzerland

Andreas Quandt, Lars Malmström
Institute for Molecular Systems
Biology,
ETH Zurich, Department of Biology
Zurich, Switzerland

Abstract—**We have received feedback from our users and supporters on the functionality and usability of the Swiss Grid Proteomics Portal during its first year of operation. We have also realized which aspects of the portal could be improved upon through frequent monitoring and interaction with the production system under heavy use. In a second, highly upgraded version of the Swiss Proteomics Portal, called iPortal, we have introduced several new concepts based on this feedback and both user and supporter experience. In this paper we detail the requirements and the improvements we have made, and also give an outlook on future possible improvements.**

Keywords—gateway; ease of use; portal; proteomics

I. Introduction

Recent advances in observational technologies have turned the Life Sciences into a data-intensive science. Microscopy imaging, mass spectrometry, gene sequencing and other technologies are available at a relatively low cost to the research labs, turning many labs into large data producers. The precision with which biological processes can be observed today provides the researchers with a very large amount of complex information, which has to be analyzed, processed and understood. The relatively new field of Systems Biology aims to integrate and model several scales of observational data of a given biological system, which can be a cellular organism like yeast or an organ of a larger organism like the wing of the fruit fly. The system is analyzed as a whole and models are built to understand its behavior. Due to the many layers of complexity already involved, researchers are in need of specialized assistance to deal with the complexity of the digital infrastructure involved. The SyBIT project of SystemsX.ch, the Swiss National Initiative in Systems Biology, has been set up to provide this support to all research projects in the SystemsX.ch initiative, which involve over 200 research labs in Switzerland. With the additional SystemsX.ch funding, many new instruments were provisioned at the participating institutions. Several projects are producing raw observational data on a large scale, on the order of terabytes per instrument per month, week or in some cases daily. SyBIT collects the requirements on data processing and works with the local and central resource providers to make sure that the necessary infrastructure is available for data storage and data processing for all projects. SyBIT also provides and supports middleware to manage and catalog the large amounts of collected data. Finally, SyBIT maintains and supports a whole toolbox of software to be sure that all project's needs are met. Most of the middleware and tools are already well established community standard tools and libraries, to which SyBIT contributes wherever needed, improving the functionality using software engineering in the process to the benefit of these communities. SyBIT also provides training to the research groups in the usage of these tools and the integrated research infrastructure.

In a previous publication we have already described the Swiss Grid Proteomics Portal [1], aiming to provide an easy-to-use but powerful portal for standardized proteomics data analysis. In this paper we elaborate on the experiences of operating the portal, leading to new requirements and the implementation of new features. A lot of the considerations that led to better usability may be relevant to similar efforts, and are summarizing our best practices for sustained operations of the improved proteomics portal, that we now call iPortal. The name was inspired by the ease of use of Apple's products: we also want to give our users a fun experience and a self-explanatory portal interface.

II. Addressing Requirements in Proteomics

In several SystemsX.ch projects, proteomics data needs to be collected as part of the overall system biology analysis. Mass spectrometry is used to identify and quantify the protein content of a given biological sample. The analysis of the raw data collected by the mass spectrometers is a research domain on its own, and there is a very large number of community tools available to reconstruct the protein content from the mass spectra generated by the instruments. The complexity of proteomics data analysis is large, as the analysis itself depends strongly on the sample being observed, and the biological question at hand. Until recently, all analysis of mass spectrometry data was done through a series of manual steps, making use of individual command-line tools with different parameters for the heterogeneous data analysis. First raw data formats produced by the various instruments need to be converted to open standard formats. Then they are analyzed using one or several of the community tools available. Often these tools were produced as parts of a research project, with

This work was supported by the SystemsX.ch, the Swiss National Initiative for Systems Biology and by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no 283481 (SCI-BUS project).

poor adherence to standard data formats, so there is a lot of data transformation involved to assure that the output of one tool can serve as the input of another. Scripts and specialized workflows have been built by the researchers and bioinformaticians in proteomics to automate some of their steps, but these were often not kept track of, or had hardcoded elements for specific environments, people or projects, not really intended or suitable to be reused by others

In SyBIT we needed to address several issues to enable reusable, traceable proteomics analysis workflows for large amounts of proteomics data. First, we had to make sure that all raw data is well tracked and annotated for future reference. Data needs to be searchable based on criteria like project, observer, timestamp, biological context and other user-defined parameters. We had to also make sure that the raw data is stored such that it can be retrieved easily for analysis and also future re-analysis. For traceable, large-scale data management we are using the open biology information system openBIS [3], which we are continuously improving to support our communities. Second, we need to find a way to process the data using the various analysis tools available in the community. We need to be able to adapt and change the data processing pipelines while keeping track of the steps and parameters involved, to assure that the results are traceable and reproducible. For this purpose we have built the Swiss Grid Proteomics Portal [1], based on the P-GRADE grid portal system [2]. This first portal has been put to production in 2010.

III. Experience with the First Proteomics Portal and New Requirements

We have made several observations and collected feedback from the users and the resource operators of the Swiss Proteomics Portal by interacting with the users, either in direct personal discussions or through email. We have presented the portal at several internal seminars and we have provided training in its usage. We have also collected all the requirements in our bug tracking system and have evaluated their relevance regularly in user meetings. The requirements are as follows:

1) Resource providers were not happy with the portal being operated under a single user name, not being able to distinguish who makes use of their resources, in our case the local HPC cluster managers. New users are required to sign the cluster usage policies and rules, and this did not occur through the portal.

2) The existing workflows did not have enough customizability for the end user. Also, setting of parameters was not straightforward.

3) Error tracing through the individual workflow steps was extremely difficult.

4) Running and rerunning the workflows was not intuitive for the user and it was not straightforward for an administrator to see what went wrong to be able help the user out in a short timeframe. Debugging took too much effort.

5) We also tried to use Grid Certificates, but too many users were not able to make use of them on their own.

6) There is a very large heterogeneity in input datasets that are needed for the protein identification workflows. It was very cumbersome and error-prone to select different input datasets.

7) Developers of completely new algorithms also need access to a portal-like infrastructure. This kind of easy deployment of high-turnaround custom workflows was not enabled with the first portal.

In the context of the EU FP7 SCI-BUS project we have upgraded the portal to the next-generation technology using Liferay portal technology and the gUSE/WS-PGRADE workflow managers [10,11]. By changing to a more modern, modular technology we could now start to address the issues observed by the users of the first portal, extending and improving the portal. We could formulate the following high-level requirements based on the feedback above:

1) Authentication and authorization: Each user needs to use their cluster account, to be requested and signed for separately, adhering to local usage policy.

2) The individual workflows need to be configurable to a much higher degree, giving much more possibilities to customize the workflow.

3) Input and output management from and to the individual workflow nodes needs to be managed at a lower level. Error reporting and logging needs to be standardized for the workflow system to be able to cope with the various failure modes.

4) The usability and intuitiveness of the portal needs to be heavily improved. Researchers not intimately involved in data processing algorithms need to be guided through the process of selecting and configuring a workflow, associating their data with it and retreiving and registering their results. Workflows need to be categorized by research topic. Also Monitoring of running workflows needs to be simplified for the end-user, but an administrator should be able to dig down into the relevant logs in case of failures. Administrators need to see the logs of all running workflows to be able to invoke procedures to rescue failed jobs also through the portal.

5) If Grid Certificates are used, they should be invisible to the user.

6) UniProt, SwissProt reference datasets used in protein identification workflows often need to be extended with specific proteins being searched in a given experiment. These reference datasets need to be easily managed and selected for identification workflows by the user.

7) Workflow algorithm developers need either a pluggable architecture to modify existing workflow nodes easily or a mechanism to submit workflows outside of the interactive portal.

IV. Implementation of the Requirements

For all of the requirements above we have implemented modules or extensions to the Proteomics Portal. We are calling the new Liferay/gUse/WS-PGRADE-based portal including all these extensions *iPortal* to distinguish it from the previous P-GRADE based Swiss Grid Proteomics Portal versions.

A. Integrating with Cluster Authorization

The first requirement is to make use of individual user accounts on the local cluster. Having individual user accounts was the requirement of the cluster operators at the ETH Zurich. Each user has to request an account on the local Brutus cluster through the usual means, by filling out a web form and agreeing to the terms of use. We have built a portlet into iPortal that comes into play whenever a new user is registered. This portlet is activated at first time log-in of a new user.

For each user, we create a new openssl secure public-private keypair. We store the private key in the secure portal database. At the first-time login, the user is asked to log into the cluster (a popup window asks him or her to enter their cluster username and password), and a session is established to their cluster user account. The portal copies the new public key into their ssl directory as a new authorized user. Also, a new configuration file is added that will be sourced whenever the portal is submitting jobs on the user's behalf by making use of their account.

From now on, the interaction with the cluster is always through the individual user's accounts. Of course, the iPortal users need to apply for and receive a cluster account before being able to register for the iPortal. The popup window requesting their cluster account credentials informs them of this fact and provides the link to the cluster registration page. Through this mechanism we can completely fulfill the requirement for cluster registration and running cluster jobs using the individual user accounts. We are in close collaboration with the cluster administrators to make sure that our security mechanisms are trusted and adhere to their policies.

B. Using Grid Certificates

We have implemented a mechanism to make use of Grid Certificates based on SAML assertions in a previous project called GridCertLib [4]. It can create a proper X.509 certificate based on the user's AAI login, which is available to all researchers in Switzerland based on the SWITCH-AAI countrywide service [5]. This would fulfill requirement 5), that the users do not need to deal with Grid Certificates when submitting to distributed infrastructures.

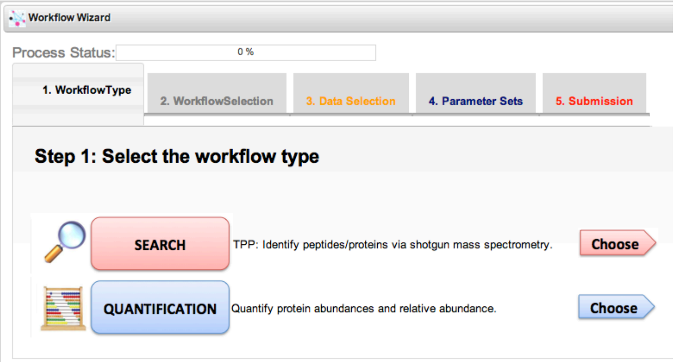
However, the GridCertLib implementation relies on the delegation feature of Shibboleth, which is a new feature and not yet available on the current infrastructure. All SWITCH-AAI enabled institutions would need to upgrade their identity provider service and would need to configure this service accordingly. This has proven to be an insurmountable administrative hurdle for the past 2 years, unfortunately. As elegant as this solution is, it is not usable in practice. In the production Swiss Proteomics Portal, the users are therefore still expected to upload their proxy certificates to a myproxy server outside of the portal if they want to make use of Grid resources.

We are now exploring other technologies that could be used also in the context of cloud infrastructures, but they are not mature enough yet. So this requirement is, unfortunately, still not met.

C. The Workflow Wizard

The main change to the way the Swiss Grid Portal is now perceived by the end-user was the introduction of the Workflow Wizard in the iPortal. This has been implemented as another Liferay portlet. The users can select the Workflow Wizard as one of the top level tabs on the main page of the portal. The Workflow Wizard guides the user through a series of steps:

1) *Workflow type selection*. The first step is to select the type of workflow that the user wants to run. Currently there are two workflow types, search and quantification. The search workflow implements several flavors of the trans proteomic pipeline [6,7], that is used to identify the peptides and proteins in the raw data as received from the mass spectrometers in a proteomics experiment. The quantification workflows make use of the result of a search workflow and additional information based on historical reference data, to quantify the abundance of the proteins with respect to one another in a given sample. Also here there are several workflows that can be selected based on the type of experiment and the type of sample being analyzed. The picture below shows this first step as shown in the iPortal.



Step 1: Select the workflow type

TPP: Identify peptides/proteins via shotgun mass spectrometry.

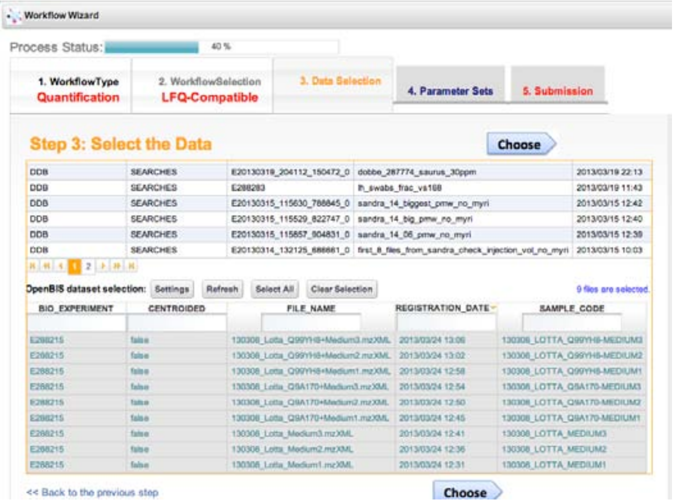
Quantify protein abundances and relative abundance.

2) The second step selects the actual workflow of the given type. Each workflow comes with a name and a one-paragraph description, with a link to further information and detailed workflow description in the project wiki pages. All of these workflows have been created and tested by workflow developers, by making use of the gUse workflow editor. Now the regular users do not need to interact with this editor anymore but are given the choice among many predefined workflows. The user can choose one of the workflows by clicking on the corresponding 'choose' button.

3) The third step is the selection of the data that needs to be analyzed with the workflow. Depending what workflow was chosen, the user is given the right type of data that he or she has access to, presented in a table format. This list is generated on the fly by submitting a query to the openBIS information system [3], where all the data are indexed and annotated with the relevant metadata. Users only see their own datasets or data they have been given access to by others.
We have decided to use openBIS as the data management hub for all of the proteomics data already for the first

version of the Swiss Proteomics Portal. Data that is being generated at the mass spectrometers is automatically ‘uploaded’ into openBIS: a monitoring process checks the contents of the directory into which new datasets are generated, scans these for automatically available metadata, registers the new data in the openBIS database and moves the data files to the central data store from where they can be made available to the data consumers. These can be processing steps as part of a workflow by making use of the rich openBIS API or it can be accessed interactively by the user over the web interface or directly using the APIs, for example in Matlab.

The data registration is highly configurable: a python script can be customized to extract the available metadata from the raw data and to register it in openBIS. For the researchers it is very convenient to see their newest mass spectrometry run already cataloged and available soon after their sample has been processed by the instrument. Users interact with openBIS through its web interface where they can browse and access the data. We have consciously decided to keep the data management interface separate from the processing interface over iPortal, in order not to overload either one. Depending on what kind of workflow was chosen, the data selection can be a two-step process. In the case of quantification workflows for example, first an experiment context needs to be chosen, then the datasets from within that experiment can be specified which should be analyzed with the given workflow. The next picture shows this data selection step.



Step 3: Select the Data

DDB	SEARCHES	E20130319_204112_150472_0	dobbe_287774_saurus_30ppm	2013/03/19 22:13
DDB	SEARCHES	E288283	lh_swabs_frac_vs188	2013/03/19 11:43
DDB	SEARCHES	E20130315_115630_788845_0	sandra_14_biggest_pmw_no_myri	2013/03/15 12:42
DDB	SEARCHES	E20130315_115529_822747_0	sandra_14_big_pmw_no_myri	2013/03/15 12:40
DDB	SEARCHES	E20130315_115857_804831_0	sandra_14_06_pmw_no_myri	2013/03/15 12:39
DDB	SEARCHES	E20130314_132125_688661_0	first_8_files_from_sandra_check_injection_vol_no_myri	2013/03/15 10:03

BIO_EXPERIMENT	CENTROIDED	FILE_NAME	REGISTRATION_DATE	SAMPLE_CODE
E288215	false	130308_Lotta_Q99YH8+Medium3.mzXML	2013/03/24 13:06	130308_LOTTA_Q99YH8-MEDIUM3
E288215	false	130308_Lotta_Q99YH8+Medium2.mzXML	2013/03/24 13:02	130308_LOTTA_Q99YH8-MEDIUM2
E288215	false	130308_Lotta_Q99YH8+Medium1.mzXML	2013/03/24 12:58	130308_LOTTA_Q99YH8-MEDIUM1
E288215	false	130308_Lotta_Q8A170+Medium3.mzXML	2013/03/24 12:54	130308_LOTTA_Q8A170-MEDIUM3
E288215	false	130308_Lotta_Q8A170+Medium2.mzXML	2013/03/24 12:50	130308_LOTTA_Q8A170-MEDIUM2
E288215	false	130308_Lotta_Q8A170+Medium1.mzXML	2013/03/24 12:45	130308_LOTTA_Q8A170-MEDIUM1
E288215	false	130308_Lotta_Medium3.mzXML	2013/03/24 12:41	130308_LOTTA_MEDIUM3
E288215	false	130308_Lotta_Medium2.mzXML	2013/03/24 12:36	130308_LOTTA_MEDIUM2
E288215	false	130308_Lotta_Medium1.mzXML	2013/03/24 12:31	130308_LOTTA_MEDIUM1

4) The fourth step is the parameterization of the workflow. Each workflow comes with a number of predefined parameterizations, provided by the workflow developers, that are suitable for a various workflow usage scenarios. The users can create their own parameterization by un-hiding the detailed parameter settings. They can then save their own parameter sets under a new name, which can be used for future workflow parameterizations. These parameterizations may also be shared among users. For search workflows, the parameterization step also includes the selection of the input database (see BioDB section below).

5) Finally, the workflow is ready for submission. The user receives an overview of the workflow to quickly check that everything is in order or whether changes need to be made, in which case the ‘back to previous step’ link can be used to go back to the corresponding step to change the settings. If the workflow and its configuration are found to be correct, the workflow can be submitted by clicking on the ‘submit’ button. The wizard makes use of the gUSE Application Specific Module (ASM) interface to select and execute the predefined workflows through gUSE.

Once a workflow was successfully submitted, the workflow wizard asks the user whether another workflow should be created with identical settings of the current workflow. This was one of the requests we have received from the users, often they want to submit the same workflow several times but with different datasets, and this helps them to do so more quickly as they only need to select the dataset, all the other settings are remembered by the wizard and are provided as default settings for the next session. The Workflow Wizard was a very large improvement in terms of usability for our users, and has improved the acceptance of the iPortal. We are continuously extending the wizard with new workflows and are planning also new types of workflows.

The Workflow Wizard addresses requirement 2), ie. the request for more customization possibilities in the workflow. In step 4, especially with the ability to store custom parameter sets, users can adjust every parameter of the workflow. Together with the new monitoring portlet described below, the very important usability requirement 4) has been addressed as well to a large degree.

D. Improved Monitoring

The workflow monitoring page as provided by WS-PGRADE has been perceived as very overloaded by many of the proteomics portal users, and we have also experienced first-hand that many users simply did not find the information they were looking for. We have therefore decided to make use of the gUSE ASM interface again to provide a more intuitive view on the current state of the user’s workflows by implementing a monitoring portlet. It can be accessed through another main tab on the portal at any time. In the initial view, the users see a list of their workflows in a simple table, color coded whether they are running, completed successfully or aborted with an error. Clicking on the one of the workflow lines will open a second table below the first one, where the users get a detailed list of all job types (workflow nodes) in the given workflow, with an indication of success or failure on this level. Again the user can click on one of these items to get access to the detailed logs of that particular node of the workflow. Usually one is most interested in the node that shows warnings or errors. The detailed log view opens three new panes on the page, displaying the standard output, the standard error and the gUSE logs of the job. There are only very rare cases when this view is not sufficient to understand why a certain failure has occurred.

However, as mentioned in the requirements, we have realized early on that many users cannot extract the necessary information from the logs to understand why a particular workflow has failed, simply because they do not know what to look for. Very often a user needs a supporter to help them to browse the logs and to understand the root cause of the problem. The reason for a failed workflow is often just a random cluster node failure or data access issue, or a job that ran out of time or memory for some reason unrelated to the job itself. More rarely we see wrong parametrizations, erroneous datasets or input data selections. In the case of cluster failures, we have started to build in automatic resubmissions and retries, which are very common in such environments. Another addition we made is a 'monitoring administrator' role, which can be assigned to supporters. With such a role, a supporter not only monitors his own workflows, but the workflows of all users in the portal. If workflows of a user fail, a monitoring administrator can check the logs of that user within his own monitoring view and take action to rescue the failed workflows directly. This way the users often do not even realize that something went wrong and also the supporters do not need to spend a lot of time trying to understand the issues over email indirectly. For failures that involve user error (like selecting wrong parameters) the users of course are being contacted directly. This mode of operation is much appreciated by the user community, and allows for a tight interaction between the supporters and the users.

E. Workflow Node Wrapping

The proteomics workflows are making use of many open source tools. All of them have different ways of managing input and output files and they also are not standardized on how they report successful processing or errors. Some return with a nonzero error code, others write messages into standard output, standard error, or into specific log files. We have to parse these outputs, validate them for correctness and scan for errors. Additionally, at each step there are output files in a workflow node that usually serve as input to the next node. Often these need some conversion and validation into the suitable format for the next step.

We can simplify the error parsing and error processing as well as the complexity of the workflows by wrapping each executable serving as a workflow node with a python script that standardizes their behavior. Our wrapper, called *applicake*, implements the following new behavior for the workflow nodes:

- Each node has only one input file and only one output file. The output of the previous node serves as the single input file of the next node.
- These files contain only metadata, ie. a set of key-value pairs, describing the properties of the previous and current workflow nodes, their actual input parameters and data files. Currently, this is implemented as a MS-Windows-like '*ini'* properties file, containing the relevant key value pairs grouped into sections.
- The nodes only extend the previous *ini* file, they do not remove data, leaving a trace of the whole workflow process.
- The nodes all validate their input and output and are exiting with standardized error codes or warnings.
- All messages and errors are written to configurable log files or the standard output and standard error.
- The error messages themselves are standardized and allow for automated error management in the future.
- The *ini* files can be stored with the result of the workflow for future reference, allowing for the complete tracking of the workflow for future reference.

Using *applicake* wrappers [8], the gUSE workflow now only needs to be configured with a single input and a single output port, specifying the configuration file of the wrapper. This simplifies also the construction of the workflows in the gUSE workflow editor, and allows for the collection of all relevant messages in the standard output, standard error and gUSE log files as displayed in the monitoring portlet.

With *applicake*, we are addressing requirement 3) to a large extent. *Applicake* needs to be extended continuously, as new workflows are being implemented with new node types, ie. with executables that have not been wrapped yet with a validator and using the *ini* file managing input-output data. This seems like additional work at first, but it is worth the effort as it provides us with traceability as well as with unified validation and error handling and a better possibility for testing.

F. BioDB

For the proteomics search workflows, reference data has to be available for the identification algorithms. The reference datasets are usually a subset of the publicly available UniProt/SwissProt database, but there are also other custom reference datasets. These data need to be made available to the search workflow jobs on the execution machines, but they are quite large and therefore it makes sense to pre-stage them to a well-defined location for read-only access.

The reference data is usually 'enriched' with special proteins of interest to the given experiment, with contaminants (like keratin, often found in human skin or hair that is often found as contamination in the observed sample) and with so-called decoy proteins, ie. protein sequences that do not exist in nature, which are later used to estimate false discovery rates [21]. There are several algorithms available to build decoy proteins, one of the popular ones is to simply reverse the order of amino acids the sequence of an existing protein, as this will provide decoys with identical overall mass to real proteins.

In the case of UniProt, there is also a new release every month that needs to be downloaded from the UniProt Knowledgebase server. We need to keep track of the different versions of the UniProt datasets to be able to reproduce previous results, or for larger experimental projects that would be too expensive to re-search every time a new UniProt version is available.

Since every search workflow is run on different samples, very often specific reference datasets need to be constructed for the corresponding organisms, contaminants and the most suitable decoy algorithm. So we end up with a large number of flavors for the UniProt knowledgebase. We have realized early on that we need to provide an easy-to-use mechanism for users to select their reference dataset in their workflow. Currently this is implemented as a drop-down list in step 4 of the Workflow Wizard.

We provide an automated tool we call *BioDB* to regularly download the UniProt Knowledgebase, provide a versioning based on the date of the download, and to enrich it with a default set of contaminants and decoys, splitting it by the most commonly used organisms in the iPortal.

The *BioDB* has four components: a download agent that fetches the original data from the public data providers, a publication agent that makes the enriched dataset available, a subscriber agent that downloads and installs the data on the local resource and finally a central registry that keeps track of all publishers and subscribers. We are running a *BioDB* subscriber agent on the ETH Brutus cluster to assure that all datasets are available on the cluster scratch filesystem in a well-known location so that all search jobs can just access the right reference dataset based on the user's selection (which is kept in the *ini* file provided through *applicake*). The users can run providers on their own custom dataset and register it in the system. Such a design assures that *BioDB* is an independent module that simply makes the necessary datasets available at the right resource without any intervention from the user. With BioDB we address requirement 6).

G. *Workflow Development using Ruffus*

The final requirement was to allow workflow developers to run workflows in a pluggable manner, also outside of the portal. By making use of *applicake* and the Python Ruffus package [9], workflow developers can quickly test their workflows locally or directly on the cluster. Ruffus is a lightweight workflow library that can deal with dependencies, parallelism and also provides some error handling. Once a workflow has been sufficiently tested with Ruffus, it is very straightforward to build a gUSE workflow. In fact, the developer of the original Ruffus workflow can usually hand over the code to a gUSE expert who has no difficulty to turn it into a proper gUSE workflow. Ruffus is also very useful for automated regression testing of the *applicake* nodes.

With Ruffus we address the final requirement 7) to a sufficient degree. Ruffus cannot be used for more complex workflows, but it is very adequate for the testing of new ideas and quick prototyping, as well as for automated testing.

We are operating three portals, the production iPortal, a development portal and a testing portal. New workflows, new functionality and capabilities can be easily installed and operated on the development and the testing portal. Also end-users can log in and make use of new functionality when the developers work together with the end-users to build workflows for new projects. Once development has finished and sufficiently tested, the new items can be deployed on the production server.

V. Summary and Future Work

The new iPortal is addressing several requirements that we have collected by interacting with the users, the workflow developers and also the experts operating the initial Swiss Proteomics Portal. By moving to a modern technology (Liferay, gUSE and WS-PGRADE from GridSphere and P-GRADE), we could make use of a modular architecture to improve existing components and interfaces and to implement several new parts at all layers of the portal.

For the end-user we have created a workflow wizard where the user can select from several predefined workflows, but with the capability to customize in detail all parameters of the workflow, and to store and share parameter sets. For the supporters and portal administrators it was essential to see the monitoring information of all users so that they can quickly understand and fix problems, even before the users themselves realize them. For this, we have also introduced a node wrapper framework to homogenize node input-output management, for result validation and to unify error messages. Now the users receive better support and more meaningful error messages.

Finally, for the developers of new algorithms and workflows, we have also provided new ways for quick prototyping and made the porting of workflows straightforward to the production portal.

For future work, we can improve further on all aspects mentioned above. In terms of security, we need to find new ways to enable federated identity management frameworks, as currently we are still using certificates to access distributed grid resources. We also want to be able to access public cloud infrastructures through the gUSE DCI-Bridge interfaces as provided through the SCI-BUS project. In the Wizard we are continuously improving the intuitiveness based on user feedback, changing the look and feel of the wizard in the process.

In BioDB, we need to improve the management of the personal datasets that vary from user to user, as now the list of BioDB databases has grown too long. In the applicake framework we are looking to replace the *ini* files with the common tool description (CTD) format developed by the OpenMS team [16]. This would enable us also to make use of KNIME [17] instead of Ruffus for workflow development and testing.

VI. Related Work

The Swiss Grid Proteomics Portal was itself based on the more experimental Swiss Protein Identification Toolbox swissPIT [12.13]. For proteomics analysis, there are several commercial (like Mascot [14]) and open source (like Corra [15]) resources available, where users can upload and process data. Our portal differs from these as it allows for more automation and parallelism, with the ability to process much more data simultaneously since there is much less user interaction involved. We also keep track of how the analysis was conducted, keeping all parameters and settings of the workflows for further reference.

In the SCI-BUS project there are many portals being built or extended using the same technology we use, like the MosGrid molecular life science portal [18]. Another very popular and easy to use gateway is Galaxy [19], which can easily be extended with bioinformatics tools, but is also not geared towards large-scale analysis. Galaxy however already has a cloud binding called CloudMan [20] that allows the usage of Amazon for the processing of certain workloads.

REFERENCES

[1] P. Kunszt, L. Espona Pernas, A. Quandt, E. Schmid, E. Hunt and L. Malmström, “The Swiss Grid Proteomics Portal”, Proceedings of the Second International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering, P. Iványi and B.H.V. Topping, (Editors), Civil-Comp Press, Stirlingshire, Scotland (2011)

[2] P. Kacsuk, G. Sipos, “Multi-Grid, Multi-User Workflows in the P-GRADE Grid Portal”, Journal of Grid Computing, 3(3-4): 221–238, 2005.

[3] A. Bauch, I. Adamczyk, P. Buczek, F-J. Elmer, K. Enimanev, P. Glyzewski, M. Kohler, T. Pylak, A. Quandt, C. Ramakrishnan, C. Beisel, L. Malmstrom, R. Aebersold, B. Rinn, “openBIS: a flexible framework for managing and analyzing complex data in biology research”, BMC Bioinformatics (2011) Vol.12, Issue: 1, 468.

[4] R. Murri, P. Kunszt, S. Maffioletti, V. Tschopp, “GridCertLib: A Single Sign-on Solution for Grid Web Applications and Portals“, Journal of Grid Computing, December 2011, Volume 9, Issue 4, pp 441-453

[5] M.A. Steinemann, C. Graf, T. Braun, M. Sutter, “Realization of a Vision: Authentication and Authorization Infrastructure for the Swiss Higher Education Community” (Educause 2003).

[6] A. Keller, A.I. Nesvizhskii, E. Kolker, R. Aebersold, “Empirical Statistical Model To Estimate the Accuracy of Peptide Identifications Made by MS/MS and Database Search”, Analytical Chemistry, 74(20): 5383–5392, 2002.

[7] A.I. Nesvizhskii, A. Keller, E. Kolker, R. Aebersold, “A Statistical Model for Identifying Proteins by Tandem Mass Spectrometry”, Analytical Chemistry, 75 (17): 4646–4658, 2003.

[8] The applicake software, http://sybit.net/software/1344307-applicake

[9] L. Goodstadt, “Ruffus: a lightweight Python library for computational pipelines”, Bioinformatics (2010) 26 (21): 2778-2779.

[10] P. Kacsuk, K. Karoczkai, G. Hermann, G. Sipos, J. Kovacs, "WS-PGRADE: Supporting parameter sweep applications in workflows," Workflows in Support of Large-Scale Science, WORKS 2008. pp.1,10, 17-17 Nov. 2008

[11] P. Kacsuk, “P-GRADE portal family for grid infrastructures”, Concurrency and Computation: Practice and Experience, Special Issue: IWPLS 2009, Volume 23, Issue 3, pages 235–245, 10 March 2011

[12] A. Quandt, P. Hernandez, P. Kunszt, C. Pautasso, M. Tuloup, C. Hernandez, R.D. Appel, “Grid-based analysis of tandem mass spectrometry data in clinical proteomics.”, Stud Health Technol Inform, 126: 13–22, 2007.

[13] A. Quandt, A. Masselot, P. Hernandez, C. Hernandez, S. Maffioletti, R.D. Appel, F. Lisacek, “SwissPIT: An workflow-based platform for analyzing tandem-MS spectra using the Grid.”, Proteomics, 9(10): 2648–2655, May 2009

[14] D.N. Perkins, D.J.C. Pappin, D.M. Creasy, J.S. Cottrell, “Probability-based protein identification by searching sequence databases using mass spectrometry data”, Electrophoresis, 20(18): 3551–3567, 1999.

[15] M.Y. Brusniak, B. Bodenmiller, D. Campbell, K. Cooke, J. Eddes, A. Garbutt, H. Lau, S. Letarte, L. Mueller, V. Sharma, O. Vitek, N. Zhang, R. Aebersold, J. Watts, “Corra: Computational framework and tools for LC-MS discovery and targeted mass spectrometry-based proteomics”, BMC Bioinformatics, 9(1): 542, 2008, ISSN 1471-2105, URL

[16] O. Kohlbacher, K. Reinert, “OpenMS and TOPP: Open Source Software for LC-MS Data Analysis”, in Proteome Bioinformatics, Volume 604 of Methods in Molecular Biology, Chapter 14, pages 201–11. 20

[17] M.R. Berthold, N. Cebron, F. Dill, T.R. Gabriel, T. K¨otter, T. Meinl, P. Ohl, K. Thiel, B. Wiswedel, “KNIME - the Konstanz information miner: version 2.0 and beyond”, SIGKDD Explorations, 11(1): 26–31, 2009.

[18] M. Wewior, L. Packschies, D. Blunk, D. Wickeroth, K. D. Warzecha, S. Herres-Pawlis, U. Lang, et. al, “The MoSGrid Gaussian Portlet–Technologies for the Implementation of Portlets for Molecular Simulations”. In Proceedings of the International Workshop on Science Gateways (IWSG10) (pp. 39-43).

[19] B. Giardine, C. Riemer, R.C. Hardison, R. Burhans, L. Elnitski, P. Shah, A. Nekrutenko, “Galaxy: a platform for interactive large-scale genome analysis”. Genome research, 15(10), 1451-1455. (2005)

[20] E. Afgan, D. Baker, N. Coraor, B. Chapman, A. Nekrutenko, J. Taylor, “Galaxy CloudMan: delivering cloud compute clusters”. BMC bioinformatics, 11(Suppl 12), S4. (2010)

[21] A. Keller, AI. Nesvizhskii, E. Kolker, R. Aebersold, “Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search”. Anal Chem. 2002;74:5383–92.