ViKM - Vital-IT Knowledge Management

About

Most projects and facilities need a data management system to organise, analyse, visualise and share data. Common to all usage scenarios are the needs to handle file organisation into datasets, grouping of datasets into projects and manage file access permission, involving user account management, organisation of users into groups and setting access rights. These functionalities have been integrated in the ViKM system to enable the very quick deployment and customization of a secured data sharing system for any new project or research facilities. ViKM is in production to manage data for tens of research projects, from single lab project to multi-partner European projects. It is also at the heart of OncoBench, an NGS and LIMS bioinformatics pipeline installed at the molecular pathology lab of the Geneva Hospitals. For each of these installations, specific mining and visualisation modules have been integrated to the base ViKM system.

This document will describe the installation and usage of the base ViKM system. Examples of visualization modules and integration with R are also described.

Installation

The ViKM application is web-based and is composed of two parts: a server part and a browser front-end.

The server side back-end is a RESTful application written in PHP with the SLIM framework. It is connected to a database using the Medoo framework. This solution enables to choose between different relational database systems. By default, ViKM comes with a SQLite database, but a dump script to use MySQL is also provided. It should be possible to easily link ViKM to other RDBMS, like Oracle or PostgeSQL.

The web side is a JavaScript web application written with the AngularJS, Rectangular and Bootstrap frameworks.

The source code of both parts is versioned using GIT and hosted in the SIB Swiss Institute of Bioinformatics GitLab server.

Backend

> *Prerequisite: The server should have a web server (the system has been tested with Apache) with PHP >=5.6 and the PHP-PDO module installed.*

The SIB GitLab repository of the ViKM back-end is:
`git://gitlab.isb-sib.ch/wikmgroup/vikmapp-backend.git`

To install it:
`bash mkdir -p vikm; cd vikm; git clone git@gitlab.isb-sib.ch:wikmgroup/vikmapp-backend.git`

From now on the path of the newly created `vikm` folder will be referred as: `<path-to-vikm-root>` -

The cloning create a `vikmapp-backend` directory which contains for directories:

- **conf:** contains the configuration file config.php
- **data:** contains the sqlite3 database (database.sqlite) and a MySQL dump script (vikm.sql)
- **htdocs:** contains the server side scripts. The PHP library dependencies are managed with Composer. The libraries are in the `vendor` folder whereas the scripts are in the `api` folder.
- **tools:** contains two files: `db.inc.php` host the initialisation of the database connection. `include.php` contains generic functions.

Ideally, one has to create a web server virtual host pointing to the htdocs directory as a document root. For security reasons, the `conf`, `data` and `tools` directories must sit outside of the document root of the web server.

A tutorial to create a Virtual host is available: https://www.digitalocean.com/community/tutorials/how-to-set-up-apache-virtual-hosts-on-ubuntu-14-04-lts

In order to setup the system, one has to edit the `conf/config.php` file to match the server configuration.

```
<?php
// A random string used to crypt user password //
if(!defined('SALT')) define('SALT','O06i44wKq5Kjt.');

// The email address of the ViKM administrator
if(!defined("CONTACT_EMAIL")) define("CONTACT_EMAIL","first.last@email.com");

// The name of the ViKM instance
if(!defined("SITE_TITLE")) define("SITE_TITLE","ViKM APP");

// Path to the data directory. All datasets will be stored in this directory.
// This directory must be writable by the apache user.
if(!defined('DATA_PATH')) define('DATA_PATH',__DIR__."/../data");

// The type of RDBMS to use. Supported values: sqlite, mysql
if(!defined("DBTYPE")) define("DBTYPE","sqlite");

// For MySQL connection
// if(!defined("DBBASE")) define("DBBASE","mysql_database");
// if(!defined("DBSERVER")) define("DBSERVER","mysql_server");
// if(!defined("DBNAME")) define("DBNAME","mysql_username");
// if(!defined("DBPWD")) define("DBPWD","mysql_user_password");


// For Sqlite3 connection. Path to the DB file
if(!defined("DBFILE")) define("DBFILE",DATA_PATH."/database.sqlite");


// Path to the tools directory

if(!defined('INCLUDE_PATH')) define("INCLUDE_PATH",dirname(__FILE__)."/../tools/");


// Whether to server should accept Cross-Origin request or not. Should be set to false in p
roduction.
if(!defined('CORS')) define('CORS',true);

// Set the debug state of the application. Might be used to display some debugging messages
if(!defined('DEBUG')) define('DEBUG',false);
?>
```

If ViKM is used with a MySQL database, one has to change the DBTYPE value and un-comment the part related to MySQL connection.

IMPORTANT: be sure that the DATA_PATH is writable by the web server (apache) user.

The database connections are managed by Medoo framework. This framework supports various SQL databases. However, in case of usage with MySQL, we recommend the usage of Meekro which offers an easier interface and supports advanced MySQL specific features. ViKM uses the Composer package manager to handle the dependencies.

ViKM uses the SLIM PHP Framework (version 2) to handle the routing of the RESTful application.

To get the required external libraries:

```
cd <path-to-vikm-root>/vikm-backend/htdocs;
php composer.phar update;
```

Frontend

> *prerequisite: The ViKM frontend is developed with the following technologies:*
>
> - NodeJS JavaScript runtime
> - Bower package manager
> - Grunt task runner
>
> Make sure to have all these components installed on your system in order to deploy the web application._

The SIB GitLab repository of ViKM frontend is: `git://gitlab.isb-sib.ch/wikmgroup/vikmapp-ng.git`

To install it:
`bash cd <path-to-vikm-root>; git clone git@gitlab.isb-sib.ch:wikmgroup/vikmapp-ng.git`

Once this directory cloned, the necessary Node modules and bower components must be installed.

```
cd <path-to-vikm-root>/vikmapp-ng/;
npm install;
bower install;
```

This will install modules in two folders; `node_modules` and `bower components`. The html, JavaScript and css files of the application are in the `app` folder.

The development and deployment of the front-end is managed by Grunt. The included Gruntfile.js can be customized to fit your settings: especially, the `constants->serverURL`. In the following example, we assume that a virtual host named 'vikm' points to `<path-to-vikm-root>/vikm-backend/htdocs`.

```
    ngconstant: {
          // Options for all targets
          options: {
              space: '  ',
              wrap: '"use strict";\n\n {%= __ngModule %}',
              name: 'config',
          },
          // Environment targets
          development: {
              options: {
                  dest: '<%= yeoman.app %>/scripts/config.js'
              },
              constants: {
                  ENV: {
                      serverURL: 'http://vikm/api/index.php/',
                      withCredentials: true,
                      debugInfoEnabled: true,
                      CORS: true
                  }
              }
          },
          production: {
              options: {
                  dest: '<%= yeoman.app %>/scripts/config.js'
              },
              constants: {
                  ENV: {
                      serverURL: 'api/',
                      withCredentials: false,
                      debugInfoEnabled: false,
                      CORS: false
                  }
              }
          }
    },
```

This part contains two important sections: *development* and *production*. Depending on whether the app is under development or deployed in production, different constants are set accordingly:

- *serverURL:* the URL of the PHP backend. Must link to the api/index.php.
- *withCredentials:* in development, two different web servers are used (the front-end is served on port 9000 and the backend on port 80). To allow Cross-Origin-Resource Sharing, this value must be set to *true*. In production mode, both front-end and back-end are served by the same web server, so set this constant to *false*.
- *CORS*: same as for *withCredentials*
- *debugInfoEnabled:* internal constant to be used in development to log some messages on the console. We don't want those messages to appear in a production mode.

> ***enabling bootstrap***
>
> Bootstrap is used as a layout framework By default, bower deploys the *.less* version, but it is easier to simply include the *.css* version. To do so, edit:

> `<path-to-vikm-root>/vikmapp-ng/bower_components/bootstrap/bower.json` and replace `less/bootstrap.less` by `dist/css/bootstrap.css`. Optionaly, we can add `dist/css/bootstrap-theme.css`.

The development environment can be started by running the following command in the vikmapp-ng directory. `grunt serve` This should open Chrome at `http://localhost:9000/#/`. If not, open it manually.

Customisation

- To rename the instance of ViKM, edit the file `app/scripts/app.js` at line: `.constant('siteTitle',{name: 'ViKM APP'})` Replace ViKM APP with the new name.
- To replace the generic logo of ViKM with the logo of the project or platform, replace the image file: `app/images/icon_webapp.png`. The image must have a height of at least 32px.
- If the installation of ViKM must be monitored by Google Analytics, put the Google Analytics code in the corresponding section of the `index.html` file.

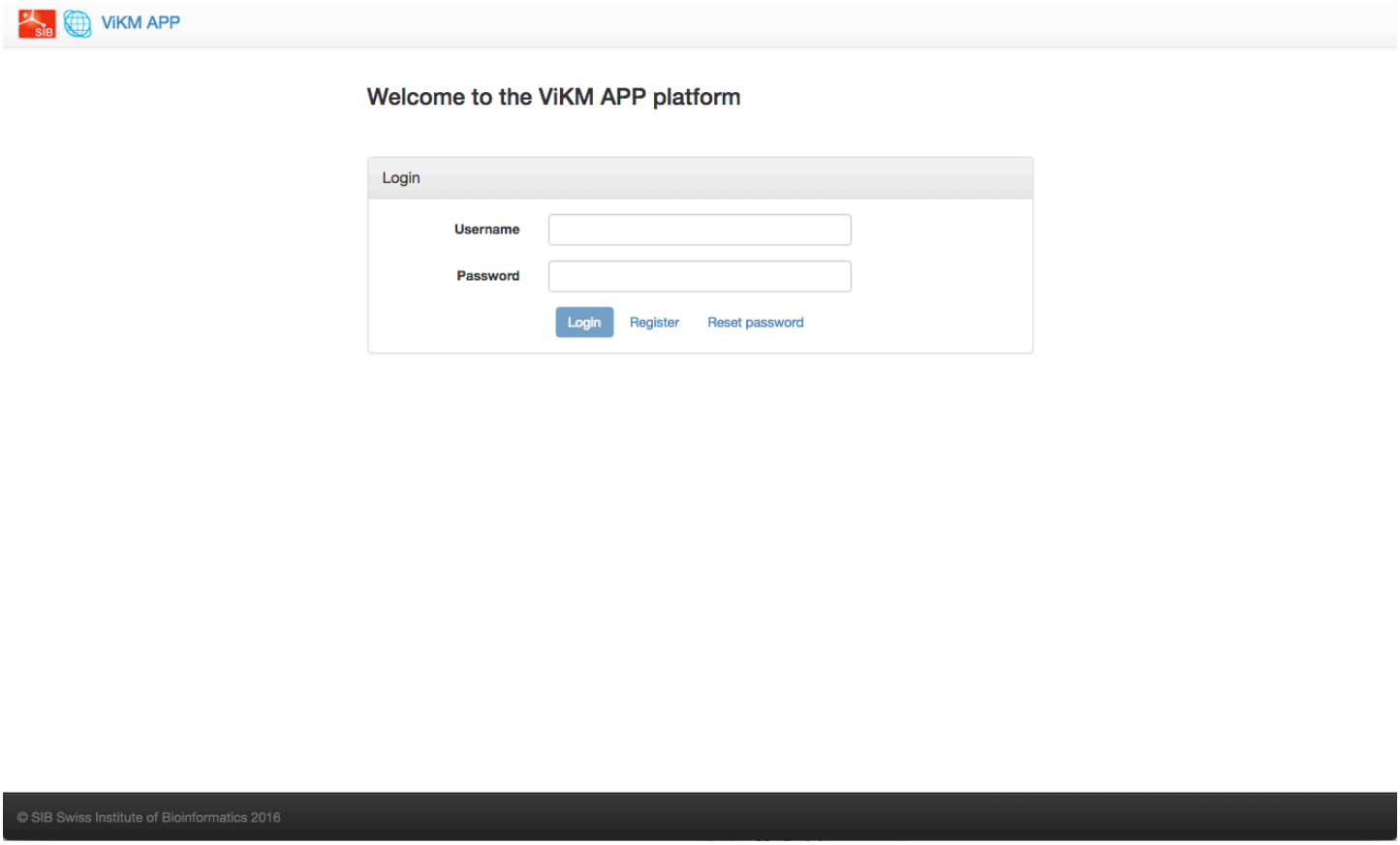
```
        <!-- Google Analytics: change UA-XXXXX-X to be your site's ID -->
        <script>
        (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
            (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
            m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
        })(window,document,'script','//www.google-analytics.com/analytics.js','ga');

        ga('create', 'UA-XXXXX-X');
        ga('send', 'page view');
        </script>
```

- ViKM is using Bootstrap v3 as a layout framework. Several themes (free or commercial) exist to customise the appearance of the web application.

Testing

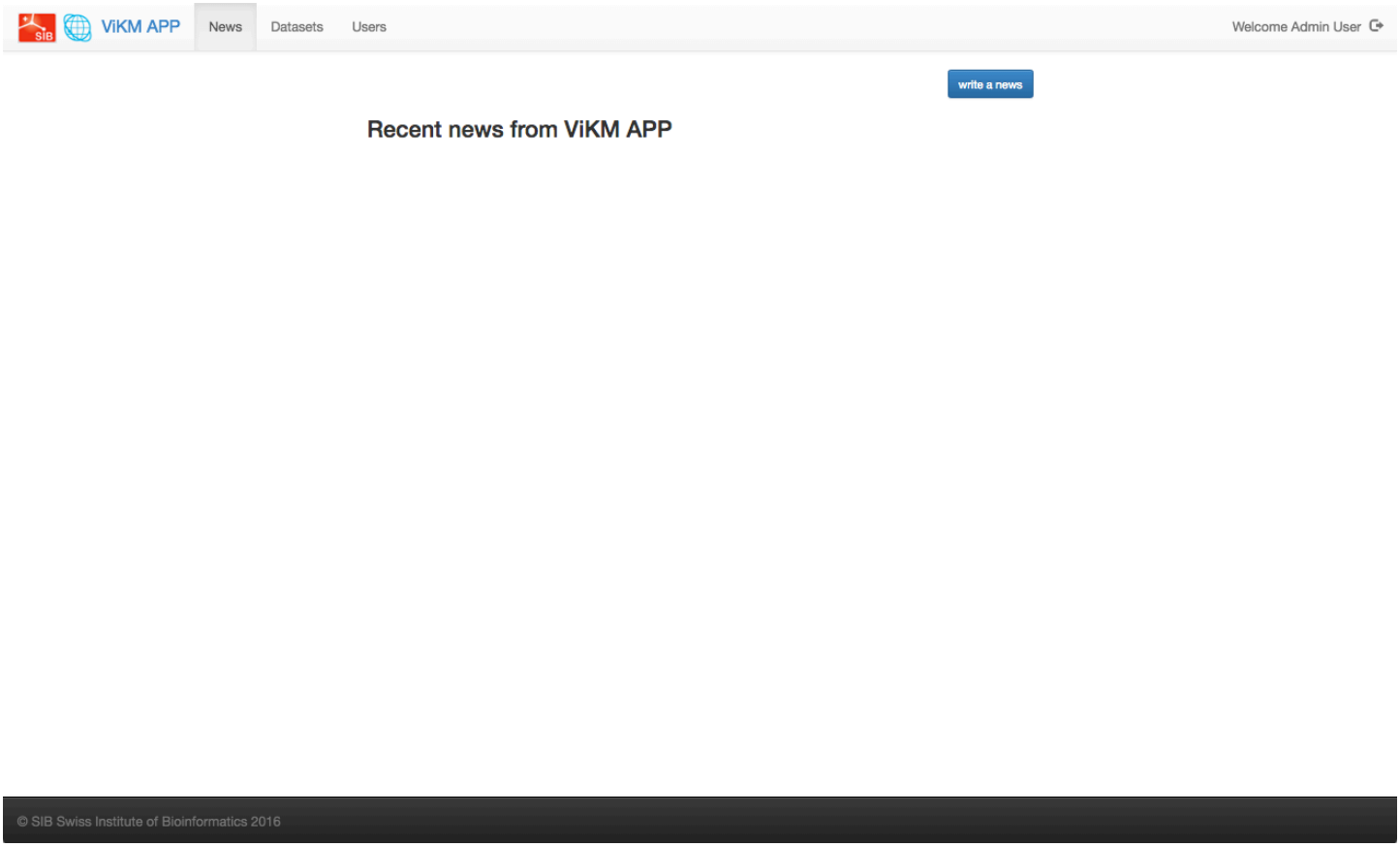
The login interface should be displayed in the browser window:



By default, an *admin* account is created with:

- login: *admin*
- password: *admin*

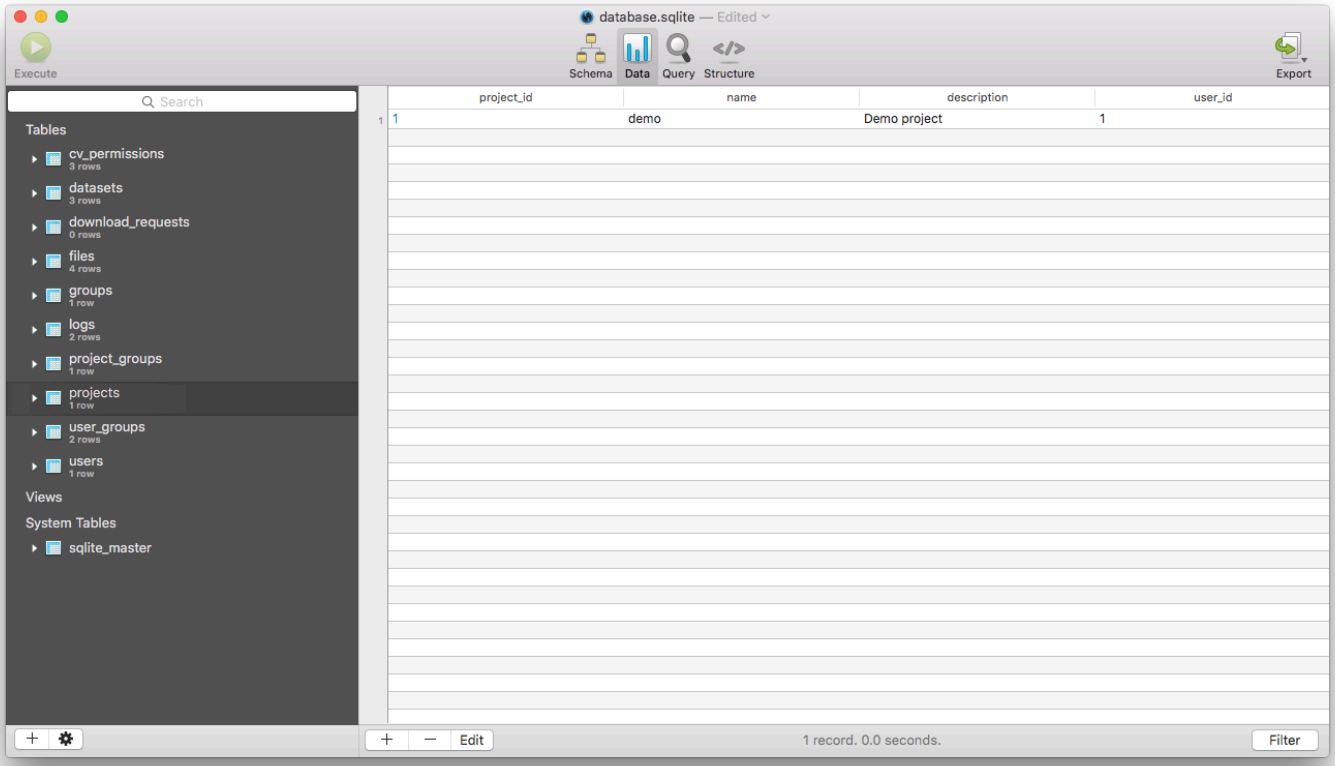
You should be able to log in and access the `News` section.



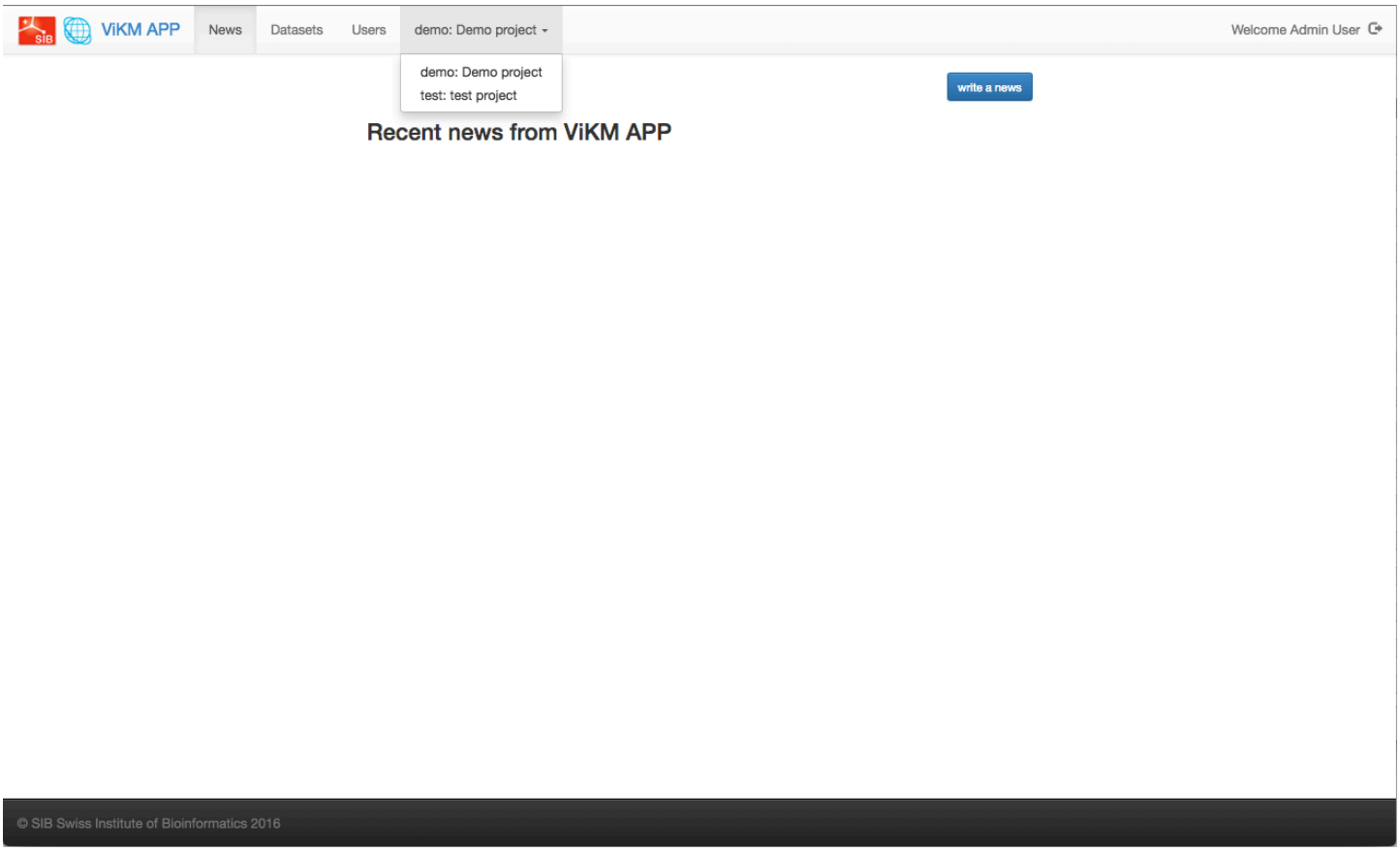
The default `admin` account must be edited; to do so, click on the `Admin User` link at the top right of the window. A form allows to modify the user details:

Database Setup

By default, ViKM is configured with a single project. To use it as a multi-project (or work-package) system, new entries must be added to the projects table of the database. Use your favorite database managing software to edit the database schema (either SQLite or MySQL). Select the `projects` table and add new projects with their name, description and project leader (`user_id` = a reference to a registered user in the `users` table).



You must then add a new entry in the table `project_groups` to enable `group_id=1` to access to the new project. At the moment, no web interface is available to create or edit projects. By reloading the web page, a new navigation menu should be displayed at the top of the window, allowing to switch projects.



Each user must be part of at least one group. Each group must be led by a registered and active user. The leader of a group will be responsible to activate new accounts linked to his group. The preferred method to start working with ViKM is to manually create user entries in the `users` SQL database:

```
INSERT INTO `users` ( `login`, `first name`, `last name`, `password`, `phone`, `email`, `is
_admin`, `is_active`, `code`, `activation_code`, `is_password_reset`)
VALUES
    ('jdoe', -- login name
     'John', -- firstname
     'Doe', -- lastname
     NULL, -- empty password value
     '+41211234567', -- phone number
     'john.doe@email.com', -- email address
     'Y', -- is active
     'N', -- is admin
     'tg333xRsmg0pC1', -- random string for API code
     '555Zh09nXWgmOJ', -- temporary password
     'Y' -- enable use of temporary password
    );
```

Note: once setup, user jdoe can login with the temporary password. ViKM will prompt to setup a permanent password.

Then, a group must be created. Set `leader_id` = `user_id` assigned to user `jdoe`. Future members of the group of John Doe will be able to register with the web interface and assign themselves to the correct group. John Doe will be notified by email and will be responsible to activate or reject the new account.

Deployment

To deploy the ViKM instance on a production server, `grunt` is used to package the front-end and minify its code. In the web application root directory, launch the command:

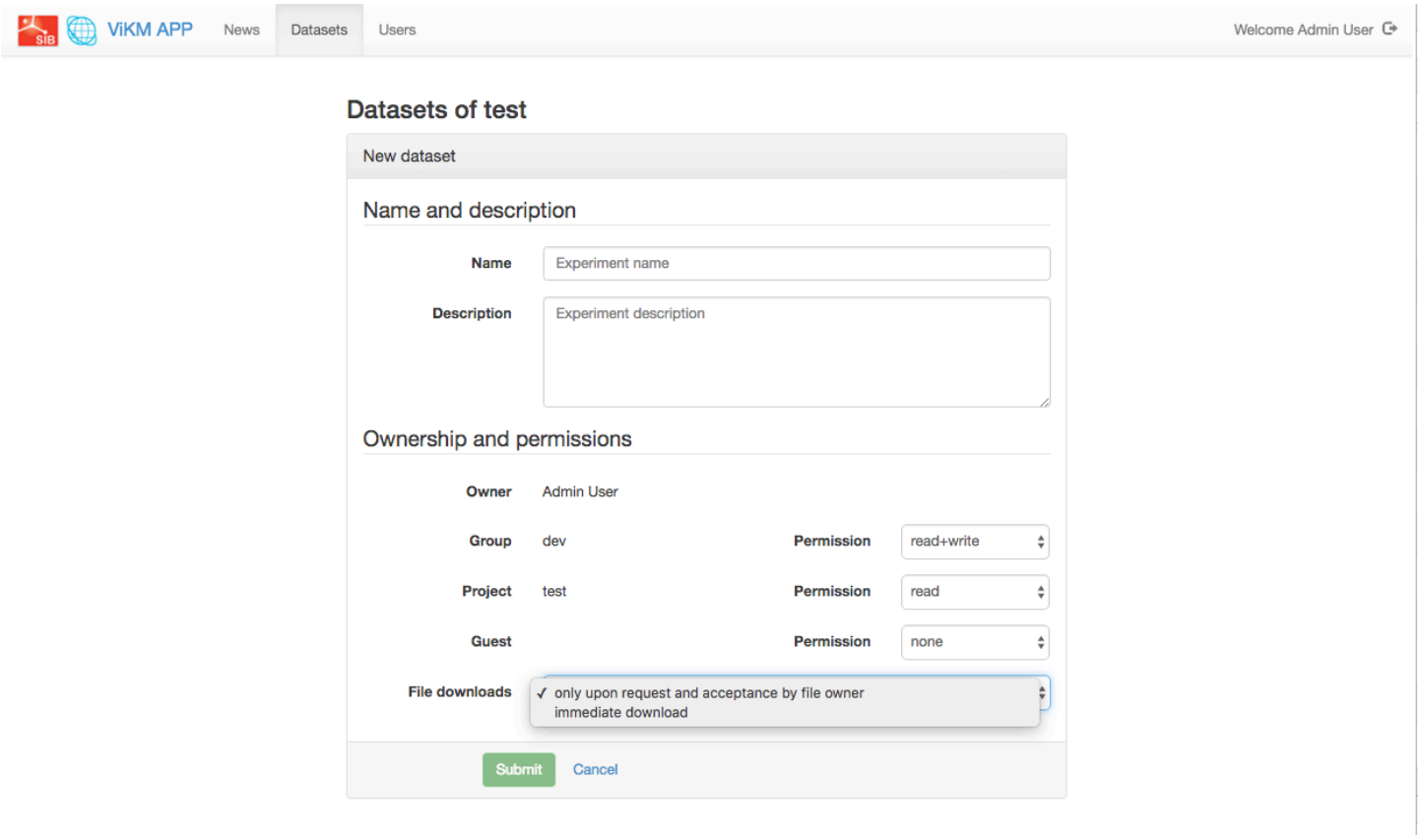
```
grunt build;
```

The result of this command is available in the `dist` folder of the web application. Copy the content of this folder in the `htdocs` directory of the production server. Copy (or clone) there as well the back-end part of the ViKM application.

Uploading documents and data

Each member of a project can create a dataset for that project. A dataset can contain different types of data and documents e.g. Excel, Word, PDF, Powerpoint. To browse available datasets, go to the Datasets tab and click on an existing Dataset.

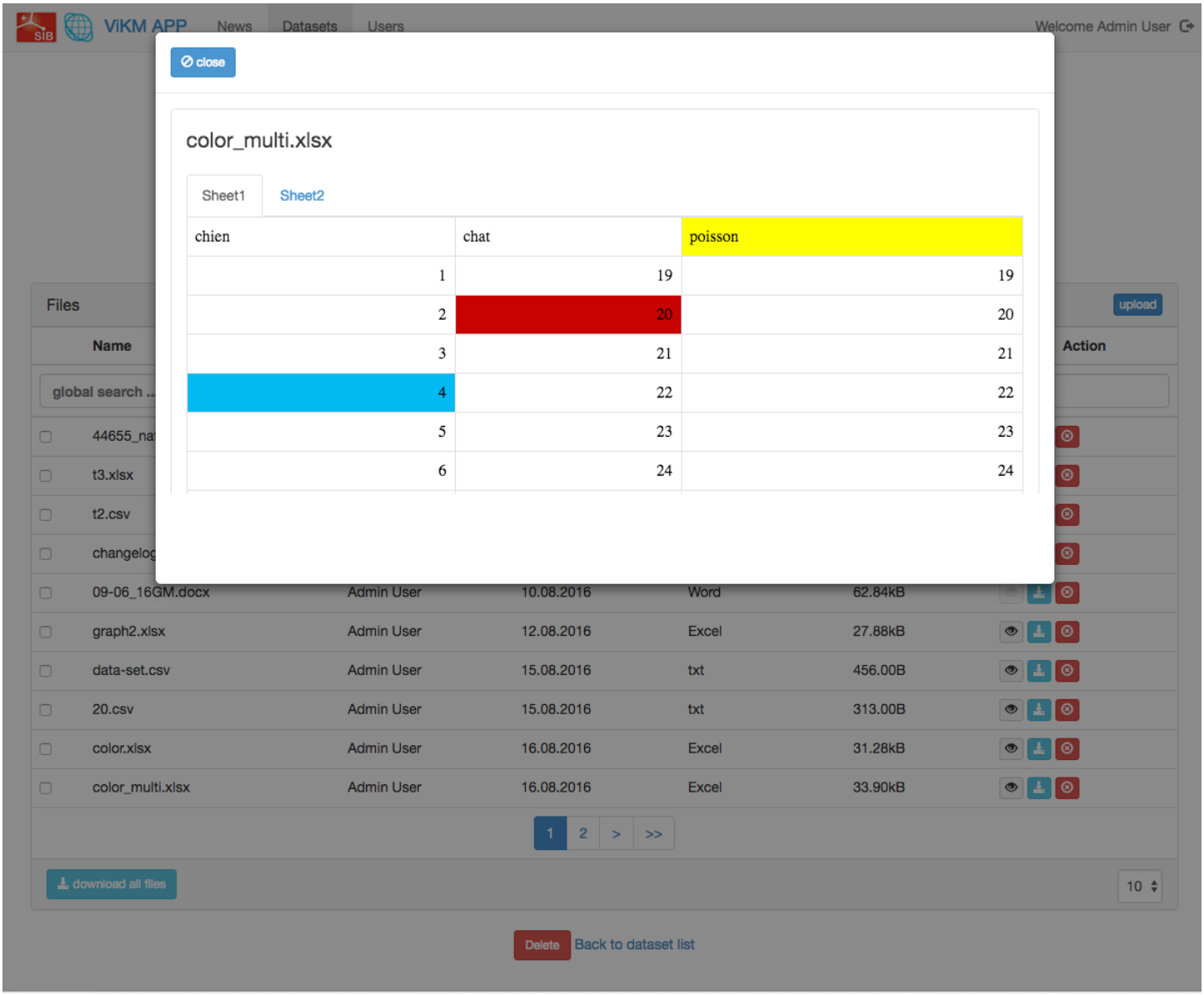
To create a new dataset, click on ‘NEW DATASET’ and fill out the information to describe the dataset as shown below. The permission for the uploaded data can be set. For example, different levels of permission can be set for the group and for other members of the project. This enables to upload preliminary data that must only be shared within the user own group, before sharing with the rest of the project.



If a file is set to be '*only be downloadable on request and acceptance by the file owner*', this means that, when someone in the same project tries to download the file, an email will be sent to the data owner asking to grant permission to access the file. This is a mechanism to ensure that the data owner is aware of where the uploaded data are being used and that the data owner is in direct contact with the person who wishes to use the data.

File preview

ViKM has a functionality to preview the file content without the need to download the file. The supported file types are: images, text, csv and Excel files. The preview opens in a modal upon clicking on the '*preview*' button in the dataset file list table.



Data visualization module

vikmGroupedBarplot

VikmGroupedBarplot is an AngularJS component to represent experimental data with a grouped barplot. This enables the comparison of different sub-groups of the population (for example, control vs treated) across different conditions or time points. Each bar represents the mean of all values of a sub-group of one condition. Clicking on the `'display values' button` will display individual measurements. The numeric value is displayed by positioning the cursor over a point. The figure can be downloaded as a PNG image by clicking on the `'download' button`.

Installation

> _Prerequisite: - lodash should be installed. You can install it with bower `bower install lodash --save` - d3js V3 should be installed. Only compatible with version 3 (`<script src="https://d3js.org/d3.v3.min.js"></script>`). Not yet with the latest version 4.

The SIB GitLab repository of the ViKM Visualization module is:
`git@gitlab.isb-sib.ch:wikmgroup/vikm-Visualization.git`

To use vikmGroupedBarplot, you need to add Vikm_Visualization module in your app.js file.

```
angular.module('myApp', ['Vikm_Visualization'])
```

and create the `vikm-grouped-barplot` component on a `<div>` on your html.

```
<div vikm-grouped-barplot data='data'></div>

<div vikm-grouped-barplot data='data' height='500' point='true' ylabel='"Normalized counts"
' maxrange='300' usenull='false'></div>
```

Parameters

- **data** - object containing the data to display. Values must be of type number. If values are not numbers, they just are ignored.
- **height** - *optional* (*400px* by default)
- **point** - *optional* (true/false, *true* by default) - to display individual values
- **ylabel** - *optional* - label for Y axis.
- **maxrange** - *optional* - maximum of Y axis (useful if to compare several graphs with same Y scale).
- **usenull** - *optional* (true/false, *false* by default) - if data contains values = 0, these values are ignored for the mean calculation of the barplot. If you set keepnull to true, 0 values will be used to compute mean.
- **simple** - *optional* If present, a minimal plot is produced, without any title, label and buttons.

Data example

The data file is a json-based format, with a title, a description and an array of all categories (`x`). Each category has a `name` attribute and a `cat` array containing the subgroups of this category. Each sub-group has a `name` attribute and a `values` array containing all the numerical values.

```
{
    "title":"Title of barplot",
    "description":"description of barplot",
        "x":[
            {"name":"x1",
                "cat":[
                    {"name":"A","values":[16.7069]},
                    {"name":"B","values":[10.7069]},
                    {"name":"C","values":[1]},
                    {"name":"D","values":[31.068,0]}
                ]
            },
            {"name":"x2",
                "cat":[
                    {"name":"A","values":[10.5769,10.368,0.8526]},
                    {"name":"B","values":[20.5769,10.368,0.8526]},
                    {"name":"C","values":[27.4099,40.7495]}
                ]
            },
            {"name":"x3",
                "cat":[
                    {"name":"A","values":["N/A"]},
                    {"name":"B","values":[20.4099,36.7495]},
                    {"name":"C","values":[23.4099,30.7495]}

                ]
            }
        ]
}
```


Example

An example implementation is available in the GIT repository.

[NEW] Linking ViKM with R using RServe

To enable the linking of the web interface with an R session, ViKM uses Rserve (https://rforge.net/Rserve/) with FastRWeb (https://rforge.net/FastRWeb/). Rserve must be started in daemon mode on the server and each connection from the web server will fork the Rserve process. In order to persist the connection between the web server and the Rserve instance, an intermediate socket manager (socketeer) must be set up.

> *Prerequisite installation of R packages*
>
> - Rserve (https://rforge.net/Rserve/)
> - FastRWeb (https://rforge.net/FastRWeb/)
> - jsonlite (https://cran.r-project.org/package=jsonlite)

The SIB GitLab repository of the ViKM-rserve is: `git@gitlab.isb-sib.ch:wikmgroup/vikm-rserve.git`

Clone this repository in `<path-to-vikm-root>`.

```
cd <path-to-vikm-root>;
git clone git@gitlab.isb-sib.ch:wikmgroup/vikm-rserve.git;
mkdir vikmapp-backend/tools/run;
mkdir vikmapp-backend/log;
mv vikm-rserve/socketeer.c vikmapp-backend/tools/;
mv vikm-rserve/FastRWeb_config.php vikmapp-backend/conf/;
mv vikm-rserve/FastRWeb* vikmapp-backend/tools/;
```

Before compiling the `socketeer.c` file two paths must be edited:

```
#define RSERVE_SOCK "<path-to-vikm-root>/vikmapp-backend/
tools/run/Rserve.sock"
```

```
#define LOG_NAME "<path-to-vikm-root>/vikmapp-backend/log/socketeer.log"
```

Compile the socketeer.c file `gcc -O3 -Wall -o socketeer socketeer.c`

Edit `<path-to-vikm-root>/vikmapp-backend/tools/FastRWeb/code/rserve.conf` to set the correct path:

```
socket <path-to-vikm-root>/vikmapp-backend/tools/run/Rserve.sock
source <path-to-vikm-root>/vikmapp-backend/tools/FastRWeb/code/rserve.R
```

Edit `<path-to-vikm-root>/vikmapp-backend/tools/FastRWeb/code/start`

```
ROOT=<path-to-vikm-root>/vikmapp-backend/tools/FastRWeb
```

To ease the call to Rserve from the web server, it is recommended to add an Alias to the `R.php` file in your virtual host configuration:

```
<VirtualHost *:80>
    ServerAdmin email@domain.com
    DocumentRoot <path-to-vikm-root>/vikmapp-backend/htdocs/
    ServerName vikm
    ErrorLog <path-to-vikm-root>/vikmapp-backend/log/vikm-error_log
    CustomLog <path-to-vikm-root>/vikmapp-backend/log/vikm-access-log common

    Alias /R "<path-to-vikm-root>/vikmapp-backend/tools/

    FastRWebPHP/R.php"
    <Directory "<path-to-vikm-root>/vikmapp-backend/">
           AllowOverride All
           Order allow,deny
           Allow from all
    </Directory>
</VirtualHost>
```

Restart the web server in order to apply the modifications to its configuration.

To start RServe:

```
cd  <path-to-vikm-root>/vikmapp-backend/tools/FastRWeb/code;
./start;
```

Rserve demo

A minimal example of ViKM integration with R is available in the `demo` directory:

- **rserve-service.js**: an Angular service to interact with RServe through socketeer and R.php.
- **rserve-directive.js**: an Angular directive to integrate in an Angular view to display the data generated by R.

Setup:

```
cp <path-to-vikm-root>/vikm-rserve/demo/rserve-service.js\
 <path-to-vikm-root>/vikmapp-ng/app/scripts/services/;
cp <path-to-vikm-root>/vikm-rserve/demo/rserve-directive.js\
 <path-to-vikm-root>/vikmapp-ng/app/scripts/directives/;
```

edit `<path-to-vikm-root>/vikmapp-ng/app/index.html` and add:

```
<script src="scripts/services/rserve-service.js"></script>
<script src="scripts/directive/rserve-directive.js"></script>
```

Using the web interface, upload the file `demo/Rdemo.RData` to the dataset named `test`;

edit the file: `<path-to-vikm-root>/vikm-ng/app/views/dataset/view.html` and add the angular directive `<rserve-directive>` at the bottom of the page:

```
<rserve-demo
    filepath='"project_"+vm.dataset.project_id+
    "/dataset_"+vm.dataset.dataset_id+"/Rdemo.RData"'>
</rserve-demo>
```

To run the example, navigate to the *test* dataset page and click on the `get data` button. This will register a new socket with socketeer, call the `load_data.R` script to load the `Rdemo.RData` file, which contain a single dataframe (expData) and call the `Rdemo.R` script with the number of rows to retrieve as argument.

The sequence of requests from Angular to the back-end is listed in the figure below:

1. The first request is for `newskt` : this gets a unique socket ID (6159753270272802) in this case. Any following request must use this ID.
2. The second request, `load_data` tells Rserve to load the file `project_1/dataset_4/Rdemo.RData` . The PHP script R.php checks whether the corrent user is allowed to access this file.

3. The third request `Rdemo` simply gets the content of the `expData` R dataframe as json.

To add a new R script (`my_script.R`), place it in the `<path-to-vikm-root>/vikm-backend/tools/FastRWeb/web.R` directory. The skeleton of the file must be:

```
run <- function(param1,param2,...) {
    oclear();

    # do something clever here

    out(json_result):
    done():
}
```

`json_result` is a JSON formatted variable, generally created with the `toJSON()` function of the `jsonlite` R package.

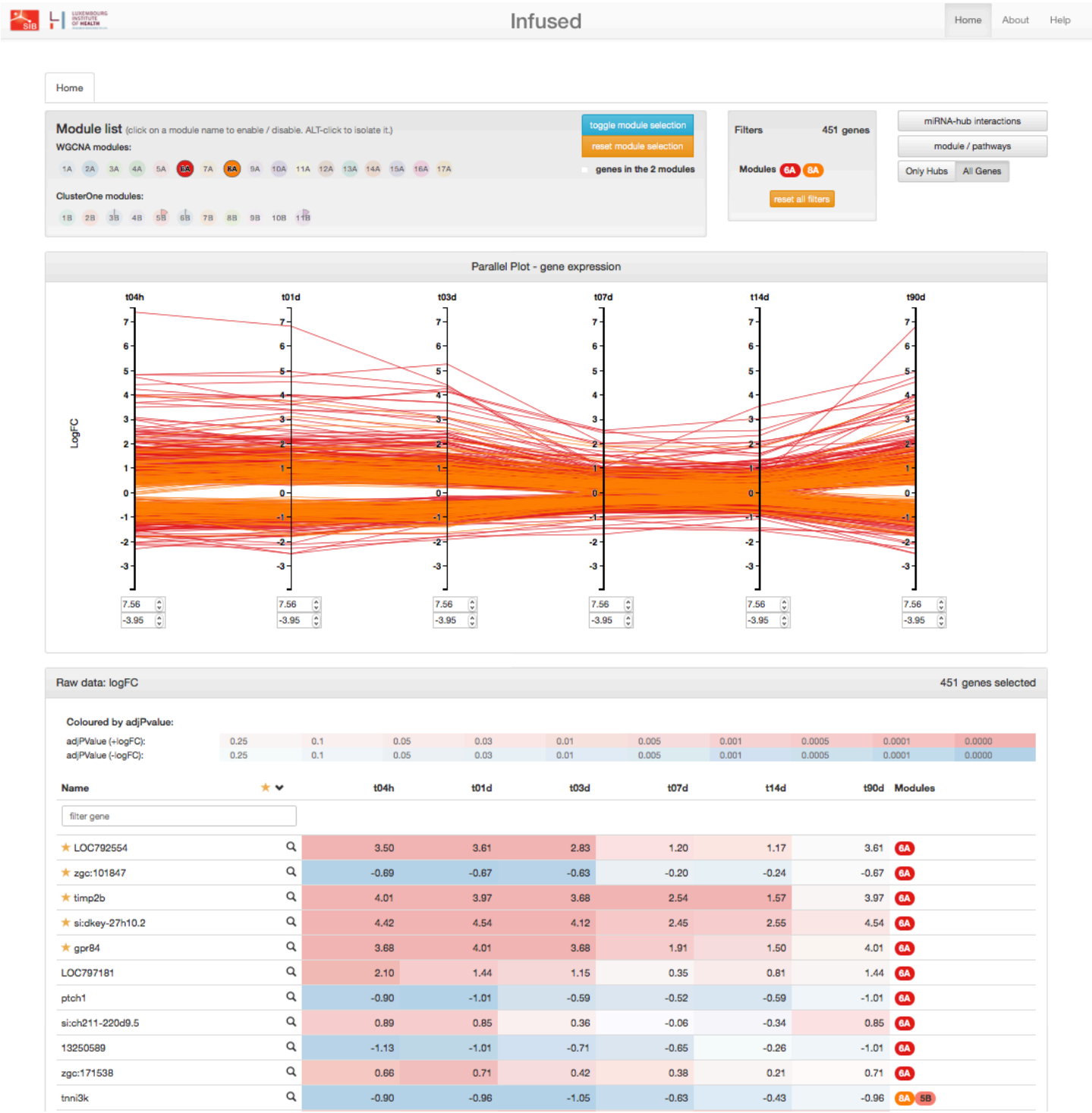
To call the script from the front-end, the link syntax is `http://vikm/R/<socketID>/my_script` .

More information on FastRWeb is available here: https://rforge.net/FastRWeb/

Examples of Installed Instances

ViKM serves as a base system to store, manage and control access to data grouped into datasets and projects. Thanks to the use of the `AngularJS` framework, it is highly modular and can be complemented with specific modules. Those could be devoted to display and navigate through tabular data or display interactive graphs or combine both. Some examples are highlighted in the following screenshots:

Infused: http://infused.vital-it.ch - publication



OncoBench: a LIMS and Variant Annotation Pipeline for the HUG

« RETOUR AU RUN

Résultat du run du 12.05.2016 - Analyse BM16_243

› Informations imprimer un rapport

› Aide

4 gènes d'intérêt

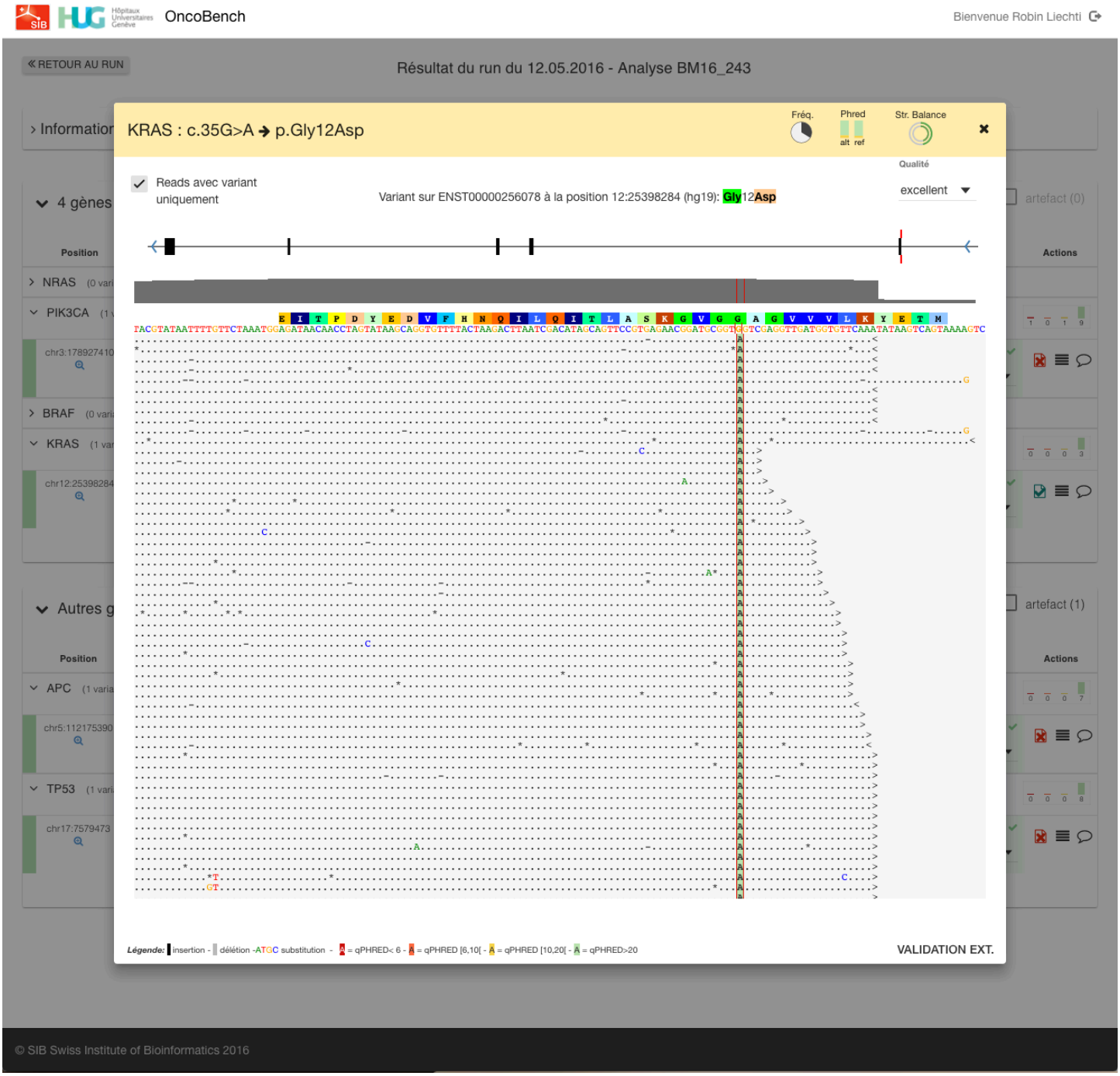
silencieux (1) artefact (0)

Position	Variant	HGVS_p	Effet	Transcript	Fréquence	Phred	Strand balance	Position	Databases	Artefact	Pathogénicité	Actions
NRAS (0 variant)												
PIK3CA (1 variant, 1 variant silencieux)												1 0 1 9
chr3:178927410	c.1173A>G	p.Ile391Met	missense_variant	ENST00000263967 ATTATGATATA|A|TACATTCCTG		alt ref			ClinVar		unlikely pathogenic	
BRAF (0 variant)												
KRAS (1 variant)												0 0 0 3
chr12:25398284	c.35G>A	p.Gly12Asp	missense_variant	ENST00000256078 GCCTACGCCA|C|CAGCTCCAAC		alt ref			ClinVar		pathogenic	

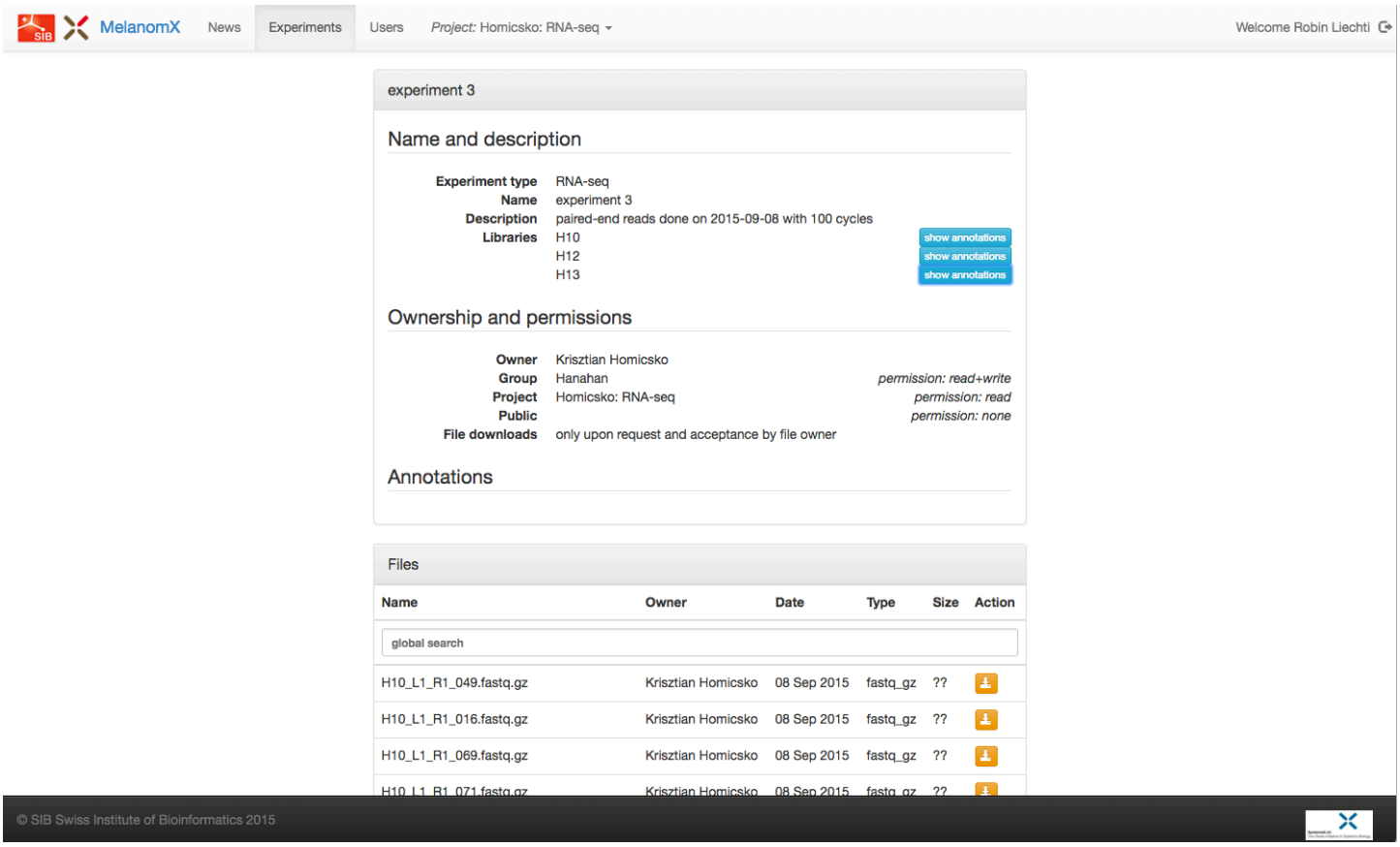
Autres gènes

silencieux (14) artefact (1)

Position	Variant	HGVS_p	Effet	Transcript	Fréquence	Phred	Strand balance	Position	Databases	Artefact	Pathogénicité	Actions
APC (1 variant, 1 variant silencieux)												0 0 0 7
chr5:112175390	c.4099C>T	p.Gln1367Ter	stop_gained	ENST00000457016 AAGTGGTGCT|C|AGACACCCAA		alt ref			ClinVar		unknown pathogenicity	
TP53 (1 variant, 1 artefact)												0 0 0 8
chr17:7579473	c.214C>G	p.Pro72Ala	missense_variant	ENST00000269305 GGGGCCACGG|G|GGGAGCAGCC		alt ref			ClinVar		not pathogenic	



MelanomX: a SystemsX.ch project. ViKM is linked to a NGS LiMS at the UNIL.



Licence

This program is free software: you can redistribute it and/or modify it under the terms of the GNU Affero General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Affero General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see http://www.gnu.org/licenses/.