



Project Acronym: MuG

Project title: Multi-Scale Complex Genomics (MuG)

Call: H2020-EINFRA-2015-1

Topic: EINFRA-9-2015

Project Number: 676556

Project Coordinator: Institute for Research in Biomedicine (IRB Barcelona)

Project start date: 1/11/2015

Duration: 36 months

Deliverable 5.3: Final computational infrastructure and its components

Lead beneficiary: Barcelona Supercomputing Center (BSC)

Dissemination level: PUBLIC

Due date: 31/10/2018

Actual submission date: 09/11/2018

Copyright© 2015-2018 The partners of the MuG Consortium



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 676556.

Document history

Version	Contributor(s)	Partner	Date	Comments
0.1	Josep Ll.Gelpi	BSC	29/10/2018	First draft
0.2	Josep Ll.Gelpí Laia Codó Javier Conejero Javier Alvarez	BSC	5/11/2018	Complete draft
	Mark McDowall	EMBL-EBI		
0.3	Rosa M Badia	BSC	6/11/2018	Review
	Andy Yates	EMBL-EBI		
1.0	Josep Ll. Gelpi	BSC	7/11/2018	Final. Approved by Supervisory Board

Table of contents

1	EXECUTIVE SUMMARY	4
2	INTRODUCTION	4
3	OVERVIEW OF MUGVRE COMPUTATIONAL INFRASTRUCTURE	5
3.1	USER PERSPECTIVE	5
3.1.1	<i>User authentication</i>	5
3.1.2	<i>User workspace</i>	6
3.2	TOOLS REGISTRATION AND DEVELOPER ACCESS	7
3.3	MUGVRE INFRASTRUCTURE	7
4	UPDATE SINCE THE RELEASE OF MUGVRE V1.0	8
4.1	MUGVRE WORKSPACE	8
4.1.1	<i>Operations based management of the Workspace</i>	8
4.1.2	<i>Project oriented workspace</i>	9
4.1.3	<i>Improvement of user feedback</i>	10
4.2	SOFTWARE SCHEDULING	11
4.2.1	PMES	11
4.2.2	PyCOMPSS	11
4.3	SUPPORT TO DEVELOPERS	12
4.3.1	<i>Updated protocol for the preparation and inclusion of new tools</i>	12
4.3.2	<i>MuGVRE updates for developers</i>	13
5	MUGVRE SUSTAINABILITY	14
5.1	MUGVRE DEPLOYMENT	14
5.2	DEVELOPMENT IN COURSE	14
5.2.1	<i>Single data and execution space</i>	14
5.2.2	<i>Meta search on public repositories</i>	15
5.3	GALAXY	15
5.4	EOSC, ELIXIR, AND GA4GH	15
6	ANNEXES	16
6.1	AVAILABLE TOOLS AND VISUALIZERS	16
6.1.1	<i>Data visualizers</i>	16
6.1.2	<i>Analysis and simulation tools (Oct 2018)</i>	16
6.1.3	<i>MuGVRE list of operations</i>	19
6.2	KNOWN FILE TYPES AND FORMATS	20
6.3	SOFTWARE REPOSITORIES	20
6.4	MUGVRE DOCUMENTATION	21
6.5	LICENSE AND TERMS OF USE	21
6.6	RELATED DELIVERABLES AND MILESTONES	21
6.7	MUGVRE STATISTICS	22
7	REFERENCES	23



1 EXECUTIVE SUMMARY

MuG Virtual Research Environment (MuGVRE) provides the members of the 3D/4D genome community with an adequate combination of relevant information, data, and computational tools. The combination should help the researcher to analyse data, either from repositories, or obtained from experiment or simulation; combine and compare such analysis results with related studies and reference data.

The MuGVRE initial prototype was presented in Sept 2016 (MS17), and described, together with all design considerations in D5.1. The first release of MuGVRE was presented in November 2017, at the conference “Multidimensional Genomics: The 3D/4D organization of chromatin” [1] and documented in D5.2. This document describes the updates in MuGVRE infrastructure and its components since its first release. In brief, the portal is based on a central workspace that allows the user to find together data and tools related to research operations in 3D/4D genomics. A user is offered a series of tool and visualization options and may analyse together data coming from different levels of the 3D/4D genomics ecosystem. The portal backend is responsible for channelling the analysis or simulation operations to the appropriate infrastructure, manage the execution, and collect the results back to the workspace. MuGVRE is implemented in two cloud systems at IRB, and BSC premises, with an additional test installation at EMBL-EBI, and will be available in a larger infrastructure at BSC at Q1 2019. Most relevant updates of MuGVRE are the reorganization of user’s workspace to allow for the allocation of projects, an alternative method for accessing tools choosing them based on operations rather than just data, and a new workspace for developers. The protocols for adding tools to the infrastructure have been revised and completed, and comprehensive documentation for both workspace usage and tool development has been produced. MuGVRE v1.1 corresponds to the achievement of milestone MS20.

The document is organized as follows: [Section 3](#) will recall the design guidelines and provide an overview of the MuGVRE infrastructure. [Section 4](#) will highlight the most relevant improvements done in the transition from version 1.0 to version 1.1: section 4.1 describes in detail the improvements made on the user’s workspace, section 4.2 the improvements in software scheduling, and section 4.3 describes the protocols and services prepared for developers. [Section 5](#) briefly describes the current plan for the maintenance and further development of the infrastructure. Documentation and links to the software repositories are listed in the [Annex](#) section.

2 INTRODUCTION

3D/4D genomics community is a highly heterogeneous community where researchers focus their work in a specific scale, both from the scientific and methodologic points of view, usually without accessing to the others. The heterogeneity of data types and tools (see D3.1 for a more formal discussion) is the main reason for this problem. The MuG Virtual Research Environment (MuGVRE) has been designed to cover this heterogeneity with a common infrastructure that allow users to work at their respective level of expertise but also provide a seamless access to the other levels with the necessary degree of integration among data and tools. In summary, MuGVRE puts together data coming from atomistic simulations, genome annotation, middle and high scale 3D genomics, and cell biology imaging data, and establishes the necessary relationships among the different levels to build an integrated view of the biological phenomena under study. The computational infrastructure as it is designed, assures interoperability of analysis tools and generates an integrated environment with a seamless transition among the available data levels. Previous reports (D5.1 and D5.2) have already described the overall design and initial implementation of MuGVRE. MuGVRE

computational infrastructure has the mission of managing the components produced by the MuG consortia: Multiscale Visualizers (WP3), data management infrastructure (WP4), and the collection of tools generated wither by MuG partners or by third parties (WP6), and integrate them in a single user environment, assuring the best efficiency in data mobilization and processing. The chosen infrastructure (see D5.1 and D5.2) will allow MuGVRE users to i) browse the available data in an integrated way, ii) incorporate raw data to the VRE that will perform the appropriate analysis, and incorporate results to MuG's data repository, iii) use MuGVRE as an analysis platform using the available tools on existing or uploaded data, and iv) download data in the appropriate formats for further in-house analysis.

The first release of MuGVRE (v1.0) was presented to the community on the 15th Nov 2017. Since then, we have collected feedback from users, developers, and use cases. The original implementation has been reconsidered and updated, and the components of the infrastructure following the final design have been implemented. We present here details of such improvements, the present state of the infrastructure, and the roadmap of the future evolution of MuGVRE. MuGVRE is available at <https://vre.multiscalegenomics.eu>.

3 OVERVIEW OF MuGVRE COMPUTATIONAL INFRASTRUCTURE

The present implementation of MuGVRE corresponds to version 1.0, released and presented to the 3D/4D genomics community on the 15th Nov 2017. Details about the design and components of the infrastructure were fully documented in Deliverable D5.2. Improvements done since the initial release will be included in v1.1 that is expected to be released in Nov 2018. Here we include a short overview of the infrastructure.

MuG computational infrastructure was originally designed to fulfil the following principles (taken from D5.1):

- 1 Flexible environment, able to adapt to the specific needs of the analysis tools (from WP6), both in terms of software requirements, or computational resources.
- 2 Software scheduler(s), able to manage analysis workflows, and computational resources in a transparent and adaptable manner. This will be an elastic infrastructure with automatic adaptation to user loads.
- 3 Multi-scale execution. Analysis workflows could be executed either at the cluster level, in HPC environments, or distributed infrastructures like EGI [2].
- 4 Web-based access centred in the MuG multi-scale browser (designed in WP3). User access will integrate the Authentication and Authorization Infrastructure being designed within the Elixir initiative [3].
- 5 The infrastructure will be eventually interface to European e-infrastructures, including the EGI for computation, and EUDAT [4] for shared storage, and aligned to initiatives like EOSC, ELIXIR, and GA4GH [5]

3.1 User perspective

3.1.1 User authentication

The access to MuGVRE is open. The default access is anonymous and does not require registration. However, to obtain a more efficient experience of MuGVRE, users are recommended to register. MuGVRE uses a Keycloak authentication and authorization server [6] to handle all internal communications and user access. Keycloak implements OpenID Connect [7] which allows for the Web access on the code authorization flow of OAuth2 [8], and a token-based authentication for the



REST services. Authentication schemes based on username/password, but also third-party identity providers (Google, LinkedIn, Elixir) are accepted. Anonymous users may convert their temporary account into a permanent one, upon registration (new in v1.1).

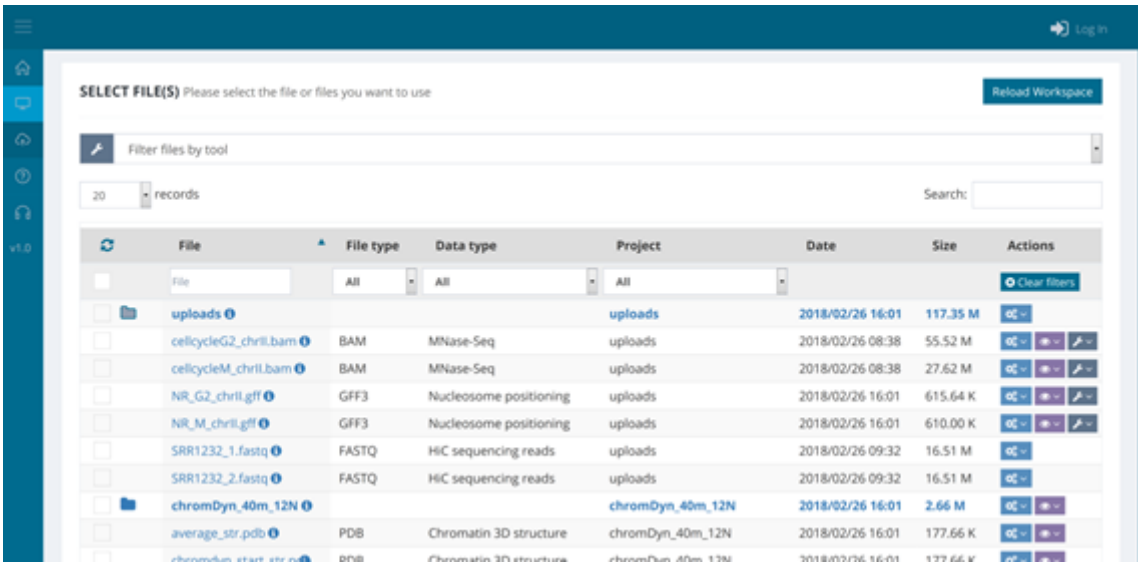
3.1.2 User workspace

MuGVRE personal workspace is the central environment for user activity. It is based on a filesystem-based layout (see [Figure 1](#)), where uploaded data and analysis results both are available. Uploaded data should be annotated to specify data and formats types. The analysis of this information enables MuGVRE workspace to offer an adapted toolkit for each file, including only compatible tools and visualizers (see annex '[Available tools and visualizers](#)', and annex '[Known file types and formats](#)'). A user's workspace is organised into projects (new in v1.1, see Section '[Project oriented workspace](#)'). Each of these projects hold an independent workspace, that in turn is organized in folders: Uploads (uploaded data), Repository (data obtained from public repositories), and several Analysis folders (results for pipelines and analyses). File lists can be filtered by any of the fields (name, format, data type, or project). Additionally, a tools-based filter is available to select only valid input files for a given tool.

Three interactive toolkits containing the following options are available:

- File toolkit: Download data or folder, edit metadata, delete, pack and compress, rename, move.
- Visualization toolkit: Available visualizers for the specific data type and format.
- Tools toolkit: Available tools for the specific data type and format.

The original MuGVRE workspace was purely data-centric, and tools should be accessed from their input data. This mode of action was relatively complex for tools requiring many input files and was considered inconvenient by non-experienced users. In v1.1 an alternative way of accessing tools is provided. Tools can be accessed directly through a comprehensive description of operations, and data is added once the tool is invoked. Section '[Operations based management of the Workspace](#)' provides a more detailed description of the operations-based access.



The screenshot shows the MuGVRE workspace interface. At the top, there's a 'SELECT FILE(S)' prompt and a 'Reload Workspace' button. Below this is a search bar and a table of files. The table has columns: File, File type, Data type, Project, Date, Size, and Actions. The 'File' column includes a folder icon for 'uploads' and several individual files like 'cellycycleG2_chrl1.bam', 'cellycycleM_chrl1.bam', 'NR_G2_chrl1.gff', 'NR_M_chrl1.gff', 'SRR1232_1.fastq', 'SRR1232_2.fastq', 'chromDyn_40m_12N', and 'average_str.pdb'. The 'File type' column shows formats like BAM, GFF3, FASTQ, and PDB. The 'Data type' column shows data types like MNase-Seq, Nucleosome positioning, HiC sequencing reads, and Chromatin 3D structure. The 'Project' column shows projects like 'uploads' and 'chromDyn_40m_12N'. The 'Date' column shows dates like '2018/02/26 16:01'. The 'Size' column shows file sizes like '117.35 M', '55.52 M', '27.62 M', '615.64 K', '610.00 K', '16.51 M', '2.66 M', and '177.66 K'. The 'Actions' column contains icons for file operations like download, delete, and move.

File	File type	Data type	Project	Date	Size	Actions
File	All	All	All			Clear filters
uploads			uploads	2018/02/26 16:01	117.35 M	Actions
cellycycleG2_chrl1.bam	BAM	MNase-Seq	uploads	2018/02/26 08:38	55.52 M	Actions
cellycycleM_chrl1.bam	BAM	MNase-Seq	uploads	2018/02/26 08:38	27.62 M	Actions
NR_G2_chrl1.gff	GFF3	Nucleosome positioning	uploads	2018/02/26 16:01	615.64 K	Actions
NR_M_chrl1.gff	GFF3	Nucleosome positioning	uploads	2018/02/26 16:01	610.00 K	Actions
SRR1232_1.fastq	FASTQ	HiC sequencing reads	uploads	2018/02/26 09:32	16.51 M	Actions
SRR1232_2.fastq	FASTQ	HiC sequencing reads	uploads	2018/02/26 09:32	16.51 M	Actions
chromDyn_40m_12N			chromDyn_40m_12N	2018/02/26 16:01	2.66 M	Actions
average_str.pdb	PDB	Chromatin 3D structure	chromDyn_40m_12N	2018/02/26 16:01	177.66 K	Actions

Figure 1. MuGVRE workspace

3.2 Tools registration and Developer access

MuGVRE is designed as an infrastructure open to any application designed for the analysis of 3D/4D genomics data. Tools installed at MuGVRE should assure free, unrestricted usage. Guidelines are available for developers (see annex '[MugVRE documentation](#)'). Developers wishing to include their applications are granted a specific interface to manage tool definitions and execution details, and to edit tool's help pages. In summary, the inclusion of new tools in MuGVRE requires the developers to prepare a Virtual Machine with the tool itself, wrapped within a specific python skeleton (see D4.6 for a detailed explanation). Additionally, developers should provide a series of documents (JSON format) including the necessary metadata for MuGVRE be able to link tools and data (see D5.2 for a detailed explanation or check [MugVRE documentation](#)). Examples of such information are the type of input file(s) accepted by the tool, the arguments, the expected output files, the type of application (serial, COMPSs), the MuG cloud infrastructure(s) in which the tool VM is installed, the identifier of the tool VM, the application callable to be invoked inside the VM, the computational resources (number of cores, memory size) or type of process manager (PMES [9], SGE [10] -Oneflow [11]) that should be used. With all this information the VRE is able to:

- Suggest the tool given a set of input files in the user workspace
- Create a web form so that the user can specify the arguments before execution
- Invoke the application callable via any of the process managers (See next section)
- Register the tool results in the Data Management protocol (DMP) so output files can be found in the user's workspace.
- Recognize the ownership of the tool, so that tool developers have the adequate administrative permissions over their tools

In MuGVRE v1.1 the whole process has been revised and automated and a specific workspace for developers created (see section '[Support to developers](#)'). The workspace allows developers to track the state and usage statistics of their tools, register new tools and provide in a automated way the necessary metadata.

3.3 MuGVRE infrastructure

MuGVRE is a fully virtualized system aimed to run in a standard cloud system. VMs are compatible with both OpenNebula and OpenStack cloud managers. See [Figure 2](#) for a schema of the infrastructure.

MuGVRE uses two complementary layouts for process management: i) Sun Grid Engine, in combination with OneFlow, a component of the OpenNebula framework that allows managing multi-VM applications and auto-scaling. SGE is used to manage applications where no complex workflows are necessary, requiring only to deploy additional workers on peaks of demand, ii) the COMPS Superscalar (COMPSs) programming model (and its python binding PyCOMPSs [12]), managed by the Programming Model Enactment Service (PMES), which interacts with cloud infrastructures through Open Cloud Computing Interface (OCCI [13]) servers. PMES/PyCOMPSs are used to control complex workflows and distributed execution.

Data is provided following the MuG's Data Access API Specification described in D4.5.

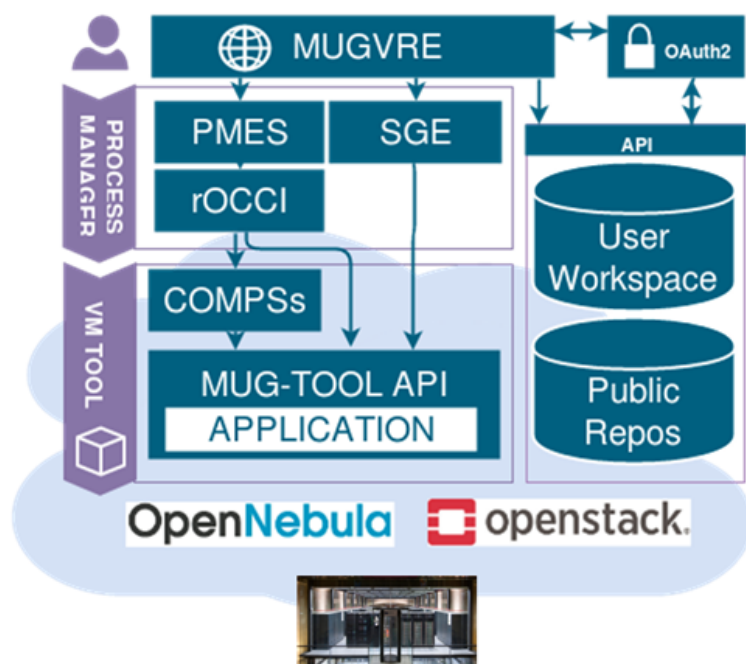


Figure 2 shows a general schema of MuGVRE infrastructure.

4 UPDATE SINCE THE RELEASE OF MuGVRE v1.0

4.1 MuGVRE Workspace

4.1.1 Operations based management of the Workspace

MuGVRE workspace is the main landing zone for the users. It constitutes a central place where data (either provided by the users, obtained from repositories, or produced by running processes) is maintained. The Workspace automatically matches data and tools based on the curated set of metadata. In MuGVRE v1.0, users could decide on the workflow based on the data files that had been selected. Feedback from users during the initial phases showed that this approach could be complemented by providing a workflow first approach. Although non-experienced users may find it useful that all decisions about the possible pairing of data and tools was handled by the workspace, actual users do have a clear idea of the operations included in the pipeline. Initial work had been done to address this point, already available on v1.0 was to start the calculation submission from tool selection, however, the documentation of the interface was not clear enough as only tool names were used and no actual functionalities were described. To address this issue a new, and complementary approach for the identification of relevant workflows has been added. First, developers were asked to state the functionalities of their tools as a collection of bioinformatics operation. That information has been used to document the operations available at the MuGVRE workspace ([Figure 3](#)), and to provide a search engine allowing the users to decide which operation to perform. After this decision, the tool (or tools) fulfilling the desired operations are selected, and compatible data present in the workspace is offered for selection as input. At this point, users may also add additional data ([Figure 4](#)).

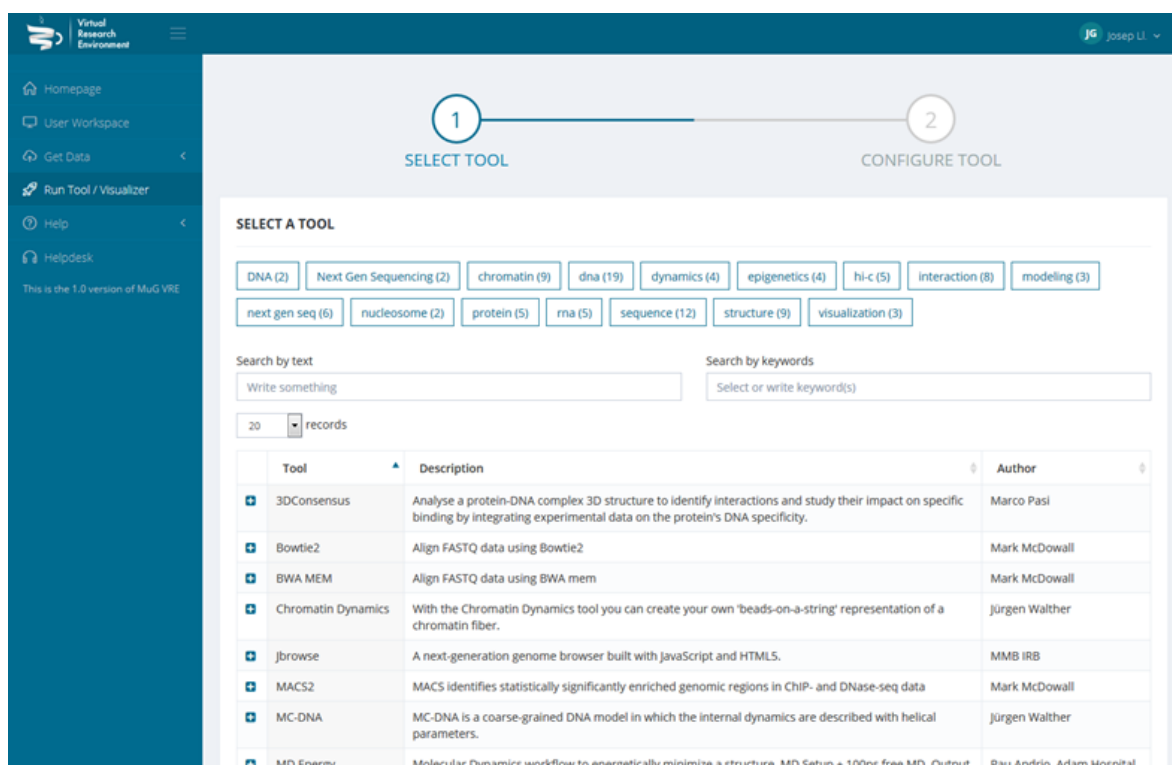


Figure 3. Search of MuGVRE operations

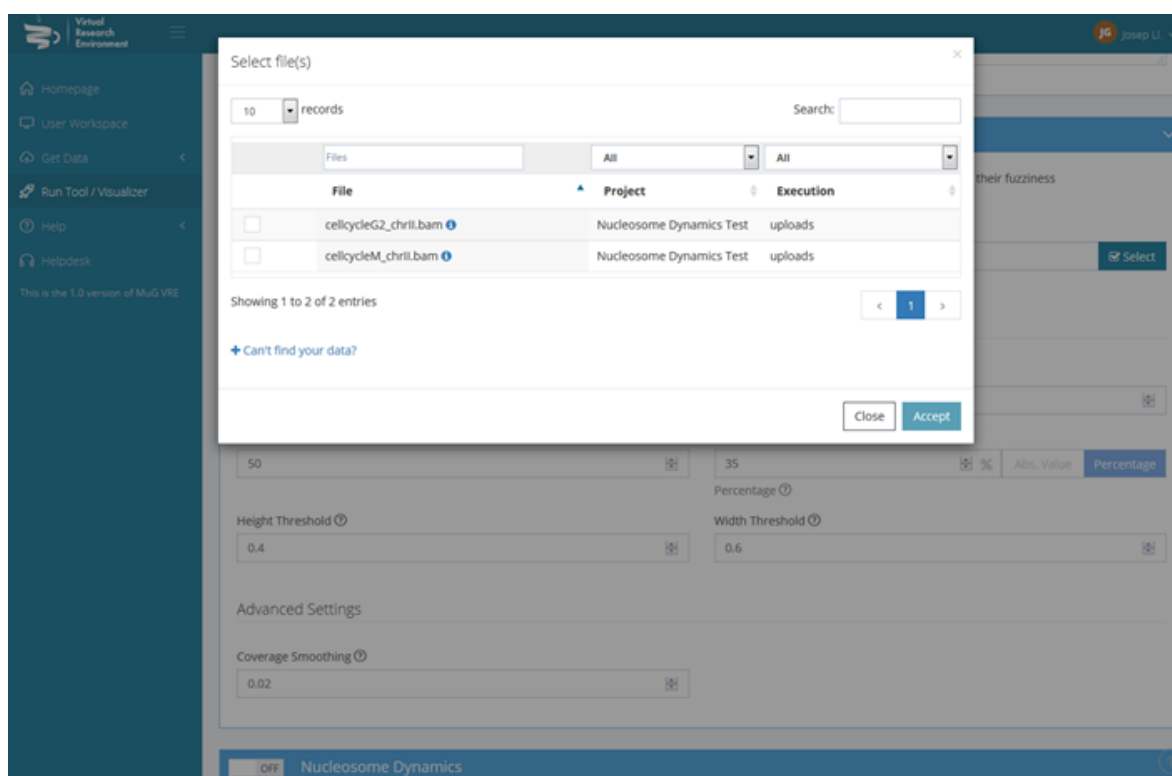


Figure 4. Selection of input data from the workspace, after tool selection.

4.1.2 Project oriented workspace

The workspace has been extended to allow registered users to better manage their data. In v1.1 the workspace is classified into “projects”, that can be accessed as separate entities. This allows users to maintain several datasets in parallel with an adequate management of the data. A specific

section of the workspace allows users to manage projects ([Figure 5](#)) : Creation of new project, modification of project's metadata ([Figure 6](#)), switching among projects.

The classification of data as separated projects will allow in a future to define levels of access to the data including data sharing among user groups.

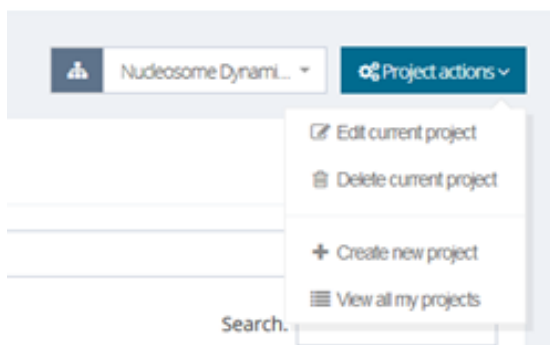


Figure 5. Project Management

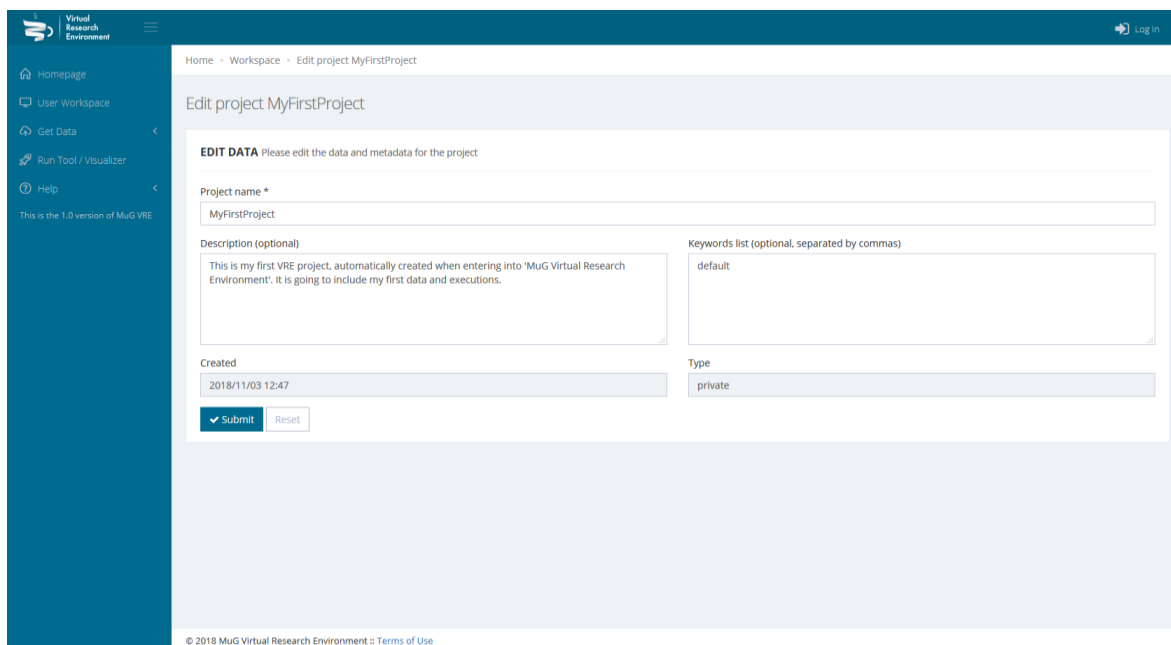


Figure 6. Modification of project's metadata

4.1.3 Improvement of user feedback

The feedback provided by the system on the progression of a process has been improved ([Figure 7](#)). The information provided includes the current operations, the state of completeness, and eventually the state of the executions. The new interface also provides access to the raw log files of the executions, as provided by the tools.

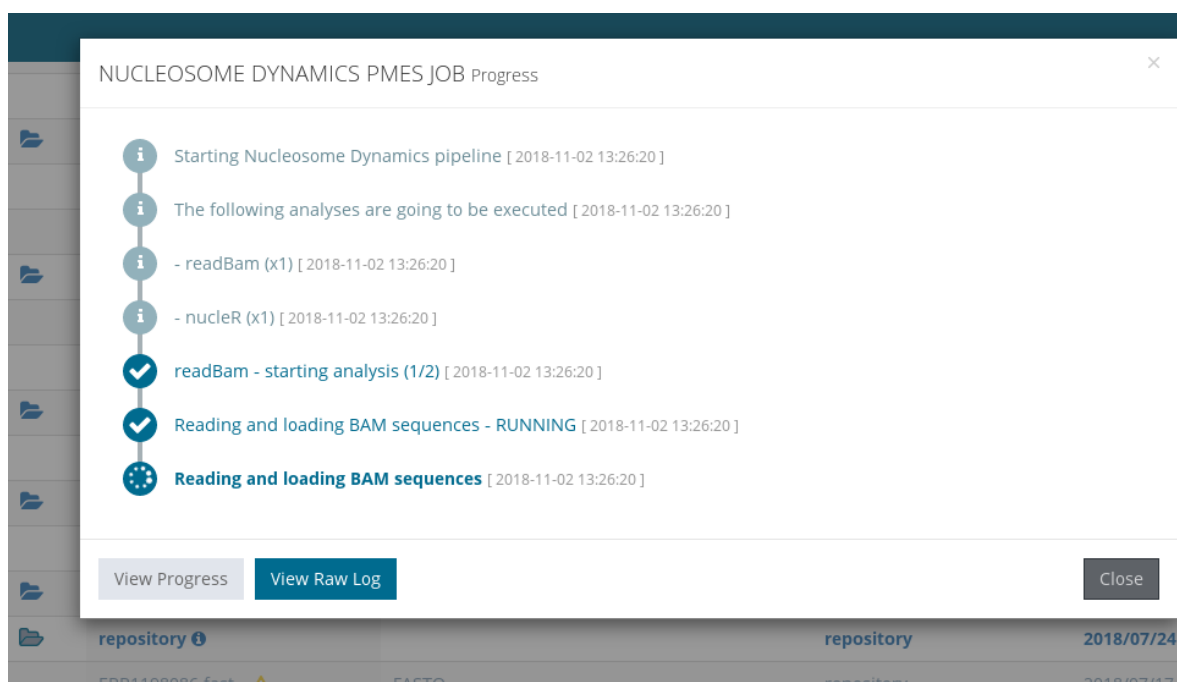


Figure 7. Example of execution progression feedback screen.

4.2 Software scheduling

4.2.1 PMES

Besides from various bug fixes, PMES logging system has been improved to give users as much information about running jobs as possible. PMES now includes the PyCOMPSs task graph in job reports and updates these reports periodically with the job output to provide real-time monitoring of the job progress.

4.2.2 PyCOMPSs

MuGVRE v1.1 uses the last update of PyCOMPSs (v2.3 release), which includes the following new features and improvements relevant for MuG:

- New features:
 - Persistent storage API implementation based on Redis (distributed as default implementation with COMPSs)
 - Support for Python 3
 - Support for Python virtual environments [14]
 - Support for running PyCOMPSs as a Python module
 - Support for tasks returning multiple elements (returns=#)
 - Automatic import of dummy PyCOMPSs API
- Improvements:
 - Software distribution as docker images
 - Support for sharing objects in memory between tasks (no file serialization is required now with persistent workers)
 - Source Code and example applications distribution on GitHub [15]
 - Automatic inference of task return
 - Improved obsolete object clean-up
 - Improved tracing support for applications using persistent memory
 - Improved finalization process to reduce zombie processes

4.3 Support to developers

4.3.1 Updated protocol for the preparation and inclusion of new tools

A revised version of the tool's integration protocol ([figure 8](#)) has been implemented in order to better support the process of wrapping, submission and annotation of user's pipelines or applications into MuGVRE. The whole process can be divided in two main steps, a wrapping process for preparing the application to be invoked from MuGVRE, and the actual integration of such code into the infrastructure.

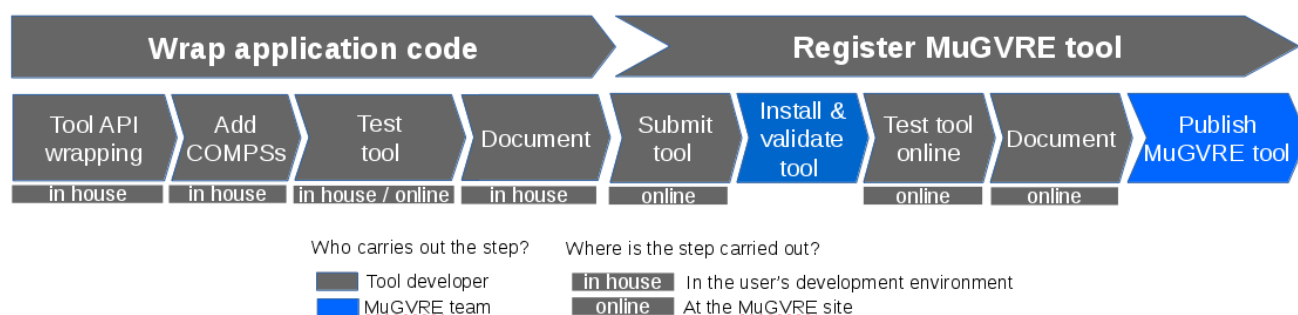


Figure 8: Updated protocol for integrating tools

Application wrapping is based on the Tool API, a Python library that provides a common access interface for MuGVRE. It is formalized as a tool skeleton on top of which user adapts the application. The user clones the Tool API repository in their own development environment and fills in the skeleton according to its pipeline's requirements (see D4.6 for a detailed explanation of the procedure). If parallelisation is to be enabled, PyCOMPSs decorators should also be added here. Once the Tool has been set up, we suggest a functional testing by which a MuGVRE execution is emulated at user's in-house installation by creating a set of test job configuration files (job configuration and input's metadata JSON files - see annex '[MugVRE documentation](#)'). The process can be performed online through the new developer's workspace. When the Tool locally passes the functional tests, the code is ready to be documented and made available on an accessible software repository, as MuG coding guidelines suggest.

From the user's perspective, integrating a new Tool into the MuGVRE infrastructure essentially implies the definition and documentation of such Tool. Through a developer's workspace, the user provides the code location, tool's descriptive metadata (i.e. title, keywords, etc), deployment details (i.e. CPUs, memory, Tool main script path, etc), logo images, etc. to eventually "Submit" the tool, at this point the ticketing system opens a communication channel with the user, and MuGVRE support team is made aware of the Tool proposal. The submission is evaluated and validated by MuGVRE team, who deploys a virtual machine instance with the Tool's code at the MuG cloud. After provisioning, the new tool is activated under the testing mode, and tool-related web pages are generated automatically based on the developer's tool definition. Finally, the tool is debugged, refined and tested on MuGVRE, example data sets are made available from the 'Get Sample Data' menu, and the tool help pages are prepared through the MuGVRE online markdown editor.

The whole integration protocol is fully described and documented step by step, including training material (see annex '[MugVRE documentation](#)').

4.3.2 MuGVRE updates for developers

The revised protocol for preparing MuGVRE tools offers an improved support and guidance to tool developers. Accordingly, MuGVRE has been adapted, creating new user roles, implementing a new developer's workspace, and assigning tool development status.

Apart from the regular account, a user can be granted with the “tool developer” role, by which developers are entitled to create and manage their own tool instances. Regular users can upgrade their account by editing their profile settings, which produces a petition to be processed by the MuGVRE support team. For debugging purposes, tool developers have extended access to the metadata of workspace files and jobs, being able to access job configuration JSON files, expected job output files, file metadata documents, etc. Moreover, they are granted access to the developer's workspace.

The developer's workspace is split in two sections, one meant to create and help developing new MuGVRE tools, and a second one dedicated to manage already integrated tools. In the first ([Figure 9](#)), tool developers freely initiate a new tool entry, represented as a new line in the central table. Each column represents the integration protocol steps described above for which MuGVRE either offers support or gathers information from the tool developer. These steps are: (1) the generation of test files for the in-house testing, a downloadable TAR file with the set of JSON configuration files and a bash script with the very command MuGVRE will use to invoke the tool. To this end, tool developer needs to provide the definition of the input files, arguments and expected output files of its tool. (2) The URL where the tested tool code is to be found. (3) The JSON validation of the tool definition fields required for the registration (deployment details, tool descriptions, etc). (4) The storage or automatic generation of tool logo images. Finally, the last column represents the submission status that can take values like “in preparation”, “submitted”, “to be reviewed”, “rejected” or “accepted”.

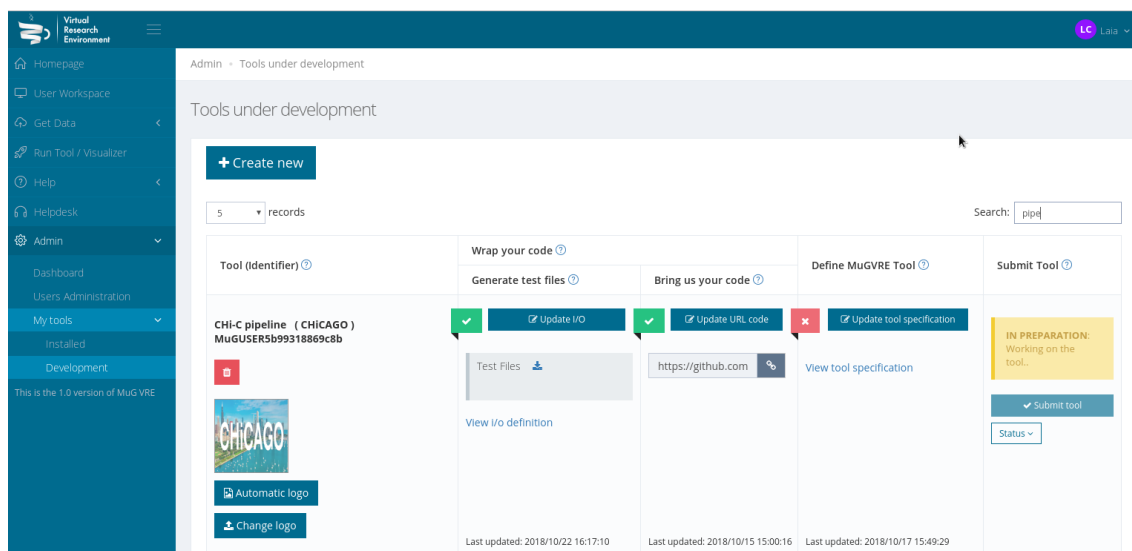


Figure 9: Tool developer's workshop for integrating new tools.

A second Developer's Workspace is used once the tool is eventually accepted. It consists on a panel listing all the tools belonging to that user ([Figure 10](#)). The workspace displays the definitive tool definition as stored at MuGVRE database, allows to download the tool usage statistics, and administrate the tool status on the infrastructure:

- Active: the tool is eligible to be run for all users
- Inactive: the tool is not eligible to be run
- Testing: the tool is eligible only by the tool developer owing the tool

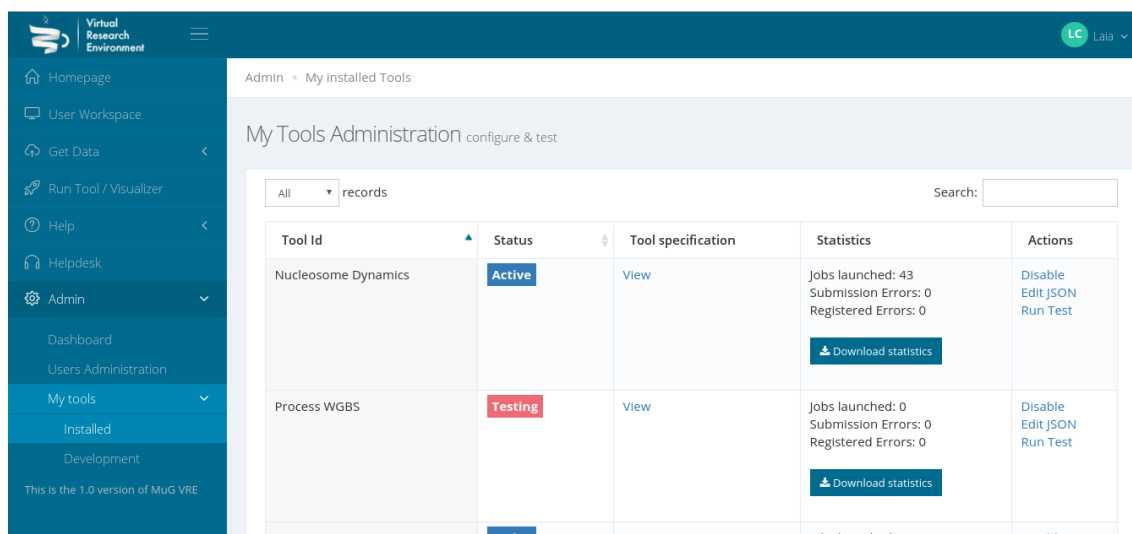


Figure 10: Tool developer's workshop for managing integrated tools.

5 MuGVRE sustainability

MuGVRE v1.1 will be released by Nov 2018. It has been built as a stable infrastructure open to the community and has already a significant number of users (see [Statistics](#) section). This section describes the current plans for maintaining and MuGVRE infrastructure active and updated, and developments already ongoing to improve its functionality.

5.1 MuGVRE deployment

The current layout of MuGVRE deployments include three cloud systems, IRB, BSC, and EMBL-EBI. IRB constitutes the production deployment, BSC is used for development, and EMBL-EBI is used for testing remote execution and data sharing. The immediate plan for deployment is to install MuGVRE 1.1 in a new production cloud at the BSC's StarLife system [16] (2,160 cores, expected 140 TFLOPS, 9.5PB of combined storage), presently being tested, and expected to be fully operative by Nov 2018. This will dramatically increase the calculation and storage capabilities of MuGVRE. Starlife also provides HPC capabilities that will allow the management of large-scale calculations. Starlife will also support data management for our PDB and UniProt mirrors, and will host BigNASim [17], a database for nucleic acids molecular dynamics simulations.

A few projects starting after MuG have already chosen MuGVRE as their starting infrastructure. Developments in these projects will be included in MuGVRE infrastructure to keep it updated.

5.2 Development in course

5.2.1 Single data and execution space

Envisioning a distributed computational environment, we have focused on OneData [18] as a solution to build a single virtual space for transparently and efficiently storing MuGVRE data across several MuG deployments. With the idea of setting a unique OneZone (federation) supported by 3

OneProviders (IRB, BSC, and EMBL-EBI's Embassy), a pilot installation of OneData has been set up locally at the BSC, with three virtual machines emulating the three cloud instances, and a fourth one acting as the zone manager. The installation is configured with the MuG authentication server, and it already uses the server to manage OneData group permissions. Also, MuG user's data access have been modelled to map OneData spaces so that it would be possible to share public and private datasets among MuG users. The design includes a unique OneData space called 'public' accessible to all users and containing public MuG datasets. Private data will be spread in N OneData spaces corresponding to the N user's workspaces and projects. There are going to be the shareable data unit so that a user will be able to share a project with a collaborator. Currently, the connectivity is being added to the BSC pilot installation.

Regarding distributed execution, PMES and rOCCI VMs and several tools have been installed and tested at EMBL-EBI Embassy cloud. Additionally, in collaboration with the BioExcel Center of Excellence, several MuG VMs can be deployed at BioExcel portal [19], and hence, deployed at EGI infrastructure.

5.2.2 *Meta search on public repositories*

One clear enhancement of MuGVRE workspace is the possibility of including data already deposited in public repositories and include such data in the analysis pipelines. MuGVRE v1.0 includes a collection of metadata from selected publications deposited in ArrayExpress [20]. Data from those publications can be downloaded into the workspace. We plan to improve this procedure by providing a meta-search service including ArrayExpress, but also ENA [21], dbGAP[22], BigNASim [17], and other possible sources of data related to 3D/4D Genomics. We will interface the original search engines at those sites and provide a consolidated report. Users could then include selected data into the integrated data space. The system will evaluate the convenience of either downloading the data locally or submit the calculation to the appropriate cloud (see section 5.2.1). This activity will be largely supported by the project EuCANSShare [23] (EC. H2020 no. 825903).

5.3 Galaxy

Galaxy [24] is a well-known interface to provide a general-purpose virtual research environment. Although the specificity of MuGVRE for 3D/4D genomics makes it more suitable for this project, Galaxy has long term support, and a large community of users and developers. In the original design the provision of a Galaxy interface to MuGVRE was already envisioned. We will extend the data management and pipeline execution APIs in MuGVRE to allow calling MuGVRE execution engine from any Galaxy interface. This will allow the use of the Galaxy workflow management system to build complex pipelines in MuGVRE, and also provide Galaxy with the complex and versatile execution system of MuG. This work will be supported by the deployment of a usegalaxy.es site, coordinated with the main European site (usegalaxy.eu), made by the National Institute of Bioinformatics [25] (Spanish node of ELIXIR, coordinated by BSC), and hosted by the Starlife system.

5.4 EOSC, ELIXIR, and GA4GH

MuGVRE has been developed with a close relationship with European and Global tendencies for cloud computing in life sciences. In particular, MuG partners are active in ELIXIR (BSC, IRB, CRG, and EMBL-EBI), the European Infrastructure for Bioinformatics, EOSC (BSC, CRG, EMBL-EBI), the European Open Science Cloud Initiative, and GA4GH (BSC, CRG, EMBL-EBI), the Global Alliance for Genomics and Health. Although cloud computing specification by these initiatives are still to be fully defined, the relationship of MuG with them has guided the decisions taken in the building of MuGVRE. The long-term objective is the full integration of MuGVRE in the European infrastructures to assure its maintenance. The global layout of MuGVRE including distributed data and execution is



fully aligned with the combined plans of the ELIXIR compute platform and the GA4GH, through a series of specifications [1]: Tools Registry (TRS), workflow management (WES), task execution (TES), and data management (DRS). This combined work is the basis of the cloud computing management for life science at EOSC and is supported by the EOSCLife project (EC H2020 no. 824087). The alignment implies that converting MuGVRE (in fact already providing more functionality) in an implementation of such standards only requires small adjustments with no major changes to the underlying code. This work will be supported by BSC participation in EOSCLife as coordinator of the ELIXIR-ES node.

6 ANNEXES

6.1 Available tools and visualizers

6.1.1 Data visualizers

Visualizer	Description	Author	Tool Status
Jbrowse [27]	A next-generation genome browser built with JavaScript and HTML5.	IRB	Active
NGL [28]	NGL Viewer is a web application for molecular visualization. WebGL is employed to display molecules like proteins and DNA/RNA with a variety of representations.	IRB	Active
TADkit [29]	3D genome browser and TADbit front-end	CNAG	Active
CytoScape [30]	Software platform for visualizing molecular interaction networks and integrate them state data like gene expression profiles, chromatin assortativity, etc.	Centre de Recherches en Cancérologie de Toulouse	Testing

(*) Tool/visualizer are: (1) active: the tool is eligible to be run for all users (2) Inactive: the tool is not eligible to be run. (3) Testing: the tool is eligible only by the tool developer owing the tool. (4) Submitted: the tool has started the integration process but is not yet integrated.

6.1.2 Analysis and simulation tools (Oct 2018)

Tool	Description	Author	Tool status (*)
3DConsensus	Analyse a protein-DNA complex 3D structure to identify interactions	UNOT	Active
Bowtie2 [31]	Align FASTQ data using Bowtie2	EMBL-EBI	Testing
BWA MEM [29]	Align FASTQ data using BWA mem	EMBL-EBI	Testing



Chromatin Dynamics	With the Chromatin Dynamics tool you can create your own 'beads-on-a-string' representation of a chromatin fiber.	IRB	Active
MACS2 [32]	MACS identifies statistically significantly enriched genomic regions in ChIP- and DNase-seq data	EMBL-EBI	Active
MC-DNA	MC-DNA is a coarse-grained DNA model in which the internal dynamics are described with helical parameters.	IRB	Active
MD Energy Refinement [33]	Molecular Dynamics workflow to energetically minimize a structure. MD Setup + 100ps free MD. Output last structure from the 100ps.	IRB	Active
NAFlex analyses [34]	Set of analyses to extract Nucleic Acids flexibility properties from Molecular Dynamics trajectories	IRB	Active
Nucleosome Dynamics [35]	Nucleosome Dynamics Tools for performing nucleosome-related analysis based on MNase-seq experimental data	IRB	Active
Process ChIP-seq [29]	Align ChIP-seq data, filtering with BioBamBam and Peak Calling using MACS2.	EMBL-EBI	Testing
Process Genomes [29]	Generates BWA, Bowtie2 and GEM indexes for a given genome	EMBL-EBI	Active
Process RNA-seq [29]	Align RNA-seq data, gene expression calling with Kallisto	EMBL-EBI	Testing
Process WGBS [29]	Align WGBS data, uses BS Seeker2 and Bowtie2	EMBL-EBI	Testing
PyDockDNA [36]	Docking Protein-DNA	BSC	Active
PyDock [36]	Protein-Protein Docking	BSC	Active
TADbit bin [37]	TADbit Hi-C binning.	CNAG	Active
TADbit map, parse and filter [37]	TADbit Hi-C mapping, parsing mapped reads and filtering of artifactual reads.	CNAG	Active
TADbit model [37]	TADbit 3D modeling.	CNAG	Active
TADbit normalize [37]	TADbit Hi-C normalize.	CNAG	Active
TADbit segment [37]	TADbit Hi-C segment (TADs and compartments).	CNAG	Active

CHiCAGO [38]	CHiCAGO pipeline for calling significant interactions in Capture HiC data, such as Promoter Capture HiC	EMBL-EBI	Submitted
Chromatin Assortativity [39]	Calculation of chromatin assortativity to integrate the epigenomic landscape of a specific cell type with its chromatin interaction network	Centre de Recherches en Cancérologie de Toulouse	Testing
Process DamID-seq [29]	Align DamID-seq data, filtering with BioBamBam and Peak Calling using iDEAR	EMBL-EBI	Submitted
BioBamBam2 Filtering [29]	Mark technical duplicates using BioBamBam2 and then remove them with samtools	EMBL-EBI	Submitted
FASTQ Trimming [29]	Trimming of single and paired end FASTQ reads using TrimGalore	EMBL-EBI	Submitted
BWA ALN [29]	Aligns single and paired end data using the BWA ALN method	EMBL-EBI	Submitted
Analyse FASTQ [29]	Analyse the quality of reads within a fastq file and provide relevant statistics	EMBL-EBI	Submitted
BS Seeker 2 Peak Caller [29]	WGBS BS Seeker2 Methylation Peak Caller	EMBL-EBI	Submitted
Process BS Seeker2 Aligner [29]	Align WGBS data using BS Seeker2 and Bowtie2	EMBL-EBI	Submitted
Process BS Seeker2 Filter[29]	Filter WGBS data, uses BS Seeker2	EMBL-EBI	Submitted
Process WGBS BS Seeker 2 Indexer [29]	Create the custom Bowtie2 index required by BS Seeker2	EMBL-EBI	Submitted

(*) Tool/visualizer are: (1) active: the tool is eligible to be run for all users (2) Inactive: the tool is not eligible to be run. (3) Testing: the tool is eligible only by the tool developer owing the tool. (4) Submitted: the tool has started the integration process but is not yet integrated.

6.1.3 MuGVRE list of operations

Operation	Tool	DNA	Next	Chromatin	Dynamics	Epigenetics	Hi-C	Interactions	Modelling	Nucleosomes	Protein	RNA	Sequence	Structure	Visualization
Analyse Protein-DNA specificity data	3D consensus	X						X			X				
Align Single/Paired end reads	Bowtie2	X	X												
	BWA MEM	X	X												
Align Single/Paired end reads custom assembly	Bowtie2	X	X												
	BWA MEM	X	X												
Create 3D representations of chromatin fiber from Sequence	Chromatin Dynamics	X		X	X									X	
Create 3D representations of chromatin fiber from NucleR output															
Browse genome	JBrowse	X				X					X	X			X
Call peaks on Chip-seq reads	MACS2	X	X						X				X		
Build DNA structure from sequence	MC-DNA		X		X				X			X			
Set up a 3D structure from MD	MD energy refinement				X						X			X	
Nucleic Acid Flexibility Analyses from Trajectory	NAFlex analyses	X			X							X		X	
Nucleic Acid Flexibility Analyses from Structure															
View 3D molecular structure	NGL										X				X
Analyse MNase-seq data	Nucleosome Dynamics	X		X						X					
Process Chip-seq reads	Process Chip-	X	X			X				X			X		

6.4 MuGVRE documentation

MuGVRE landing page, <http://multiscalegenomics.eu>, holds documentation about MuG project and events, MuGVRE tools and tutorials, as well as a section targeted for MuG tool developers under the 'Bring your tool' section. In there, it can be found:

- Guideline for tool developer's : <http://multiscalegenomics.eu/MuGVRE/instructions/>
- Description of MuGVRE files: <http://multiscalegenomics.eu/MuGVRE/integration-of-tools/>
- Tool developers's workshop material: <http://multiscalegenomics.eu/MuGVRE/training/>

6.5 License and Terms of Use

<http://www.multiscalegenomics.eu/MuGVRE/terms-of-use/>

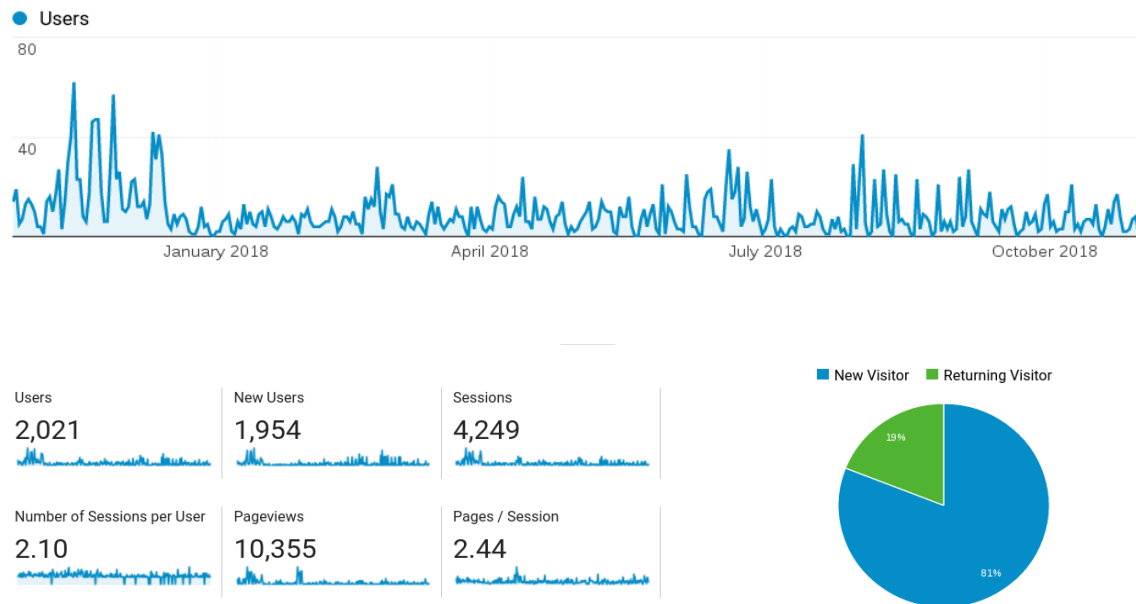
6.6 Related deliverables and Milestones

- D3.1. [A critical evaluation of the problems on data structure the browser has to solve](#)
- D4.5. [Data access API specification and implementation](#)
- D4.6. Benchmarks and documentation (Oct 2018)
- D5.1. [Computational infrastructure set-up](#)
- D5.2. [Computational infrastructure components implementation](#)

- MS17. [Early prototypes of the computational infrastructure](#)
- MS18. [User support tools available](#)
- MS19. [Programming models release](#)

6.7 MuGVRE statistics

User's statistics (Oct 2017 - Oct 2018)



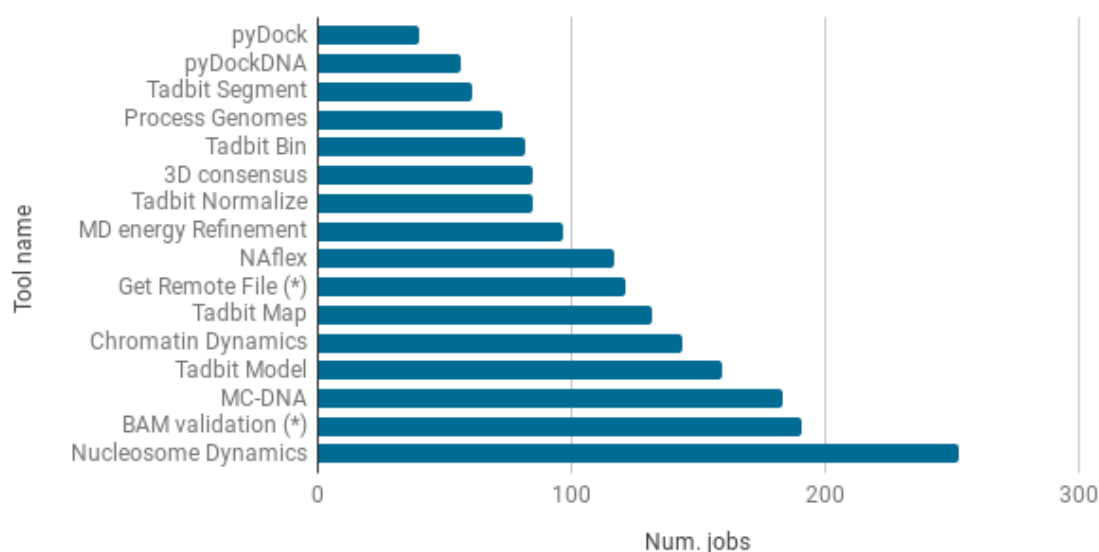
- **Users:** Users who have initiated at least one session during the date range.
- **New Users:** The number of first-time users during the selected date range
- **Sessions:** Total number of Sessions within last year. A session is the period time a user is actively engaged with your website
- **Number of Sessions per User:** The average number of Sessions per user.
- **Pageviews:** total number of pages viewed. Repeated views of single pages are counted.
- **Pages/Session (Average Page Depth):** average pages num. viewed during session. Repeated views of single pages are counted.

Tool job's statistics (Oct 2017 - Oct 2018)

Registered users	153
Total user's data storage	190 GB

Tools usage distribution

(Nov 17' - Nov 18')



7 REFERENCES

1. <https://www.irbbarcelona.org/en/events/multidimensional-genomics-the-3d4d-organization-of-chromatin>
2. <https://egi.eu>
3. <https://www.elixir-europe.org>
4. <https://eudat.eu>
5. <https://www.ga4gh.org/>
6. <http://www.keycloak.org>
7. <https://openid.net/>
8. <https://oauth.net>
9. Lordan, F., Tejedor, E., Ejarque, J., Rafanell, R., Alvarez, J., Marozzo, F., Lezzi, D., Sirvent, R., Talia, D. and Badia, R.M. (2013) ServiceSs: An Interoperable Programming Framework for the Cloud. J. Grid. Comput., 12, 67-91
10. <https://sourceforge.net/projects/gridscheduler>
11. https://docs.opennebula.org/5.4/advanced_components/application_flow_and_auto-scaling/appflow_use_cli.html
12. Tejedor, E., Becerra, Y., Alomar, G., Queralt, A., Badia, R.M., Torres, J., Cortes, T. and Labarta, J. (2017) PyCOMPSs: Parallel computational workflows in python. Intl. J. High Perf. Comput. Appl., 31, 66-82
13. <http://occi-wg.org>
14. <https://docs.python.org/3/tutorial/venv.html>
15. <https://github.com/bsc-wdc/compss>
16. <https://www.bsc.es/marenostrum/star-life>
17. Hospital, A., Andrio, P., Cugnasco, C., Codo, L., Becerra, Y., Dans, P.D., Battistini, F., Torres, J., Goñi, R., Orozco, M. et al. (2016) BIGNASim: a NoSQL database structure and analysis portal for nucleic acids simulation data. Nucleic Acids Res, 44, D272-278
18. <https://onedata.org/>

19. <https://bioexcel.ebi.ac.uk>
20. Kolesnikov, N., Hastings, E., Keays, M., Melnichuk, O., Tang, Y.A., Williams, E., Dylag, M., Kurbatova, N., Brandizi, M., Burdett, T. et al. (2015) ArrayExpress update--simplifying data submissions. *Nucleic Acids Res*, 43, D1113-1116
21. <https://www.ebi.ac.uk/ena>
22. <https://www.ncbi.nlm.nih.gov/gap>
23. <https://www.bsc.es/research-and-development/projects/eucanshare-eu-canada-joint-infrastructure-next-generation-multi>
24. <https://usegalaxy.eu>
25. <https://inb-elixir.es>
26. <https://github.com/ga4gh/wiki/wiki>
27. Buels, R., Yao, E., Diesh, C.M., Hayes, R.D., Munoz-Torres, M., Helt, G., Goodstein, D.M., Elisk, C.G., Lewis, S.E., Stein, L. et al. (2016) JBrowse: a dynamic web platform for genome visualization and analysis. *Genome Biol*, 17, 66
28. Rose, A.S. and Hildebrand, P.W. (2015) NGL Viewer: a web application for molecular visualization. *Nucleic Acids Res*, 43, W576-579
29. <http://3DGenomes.org/tadkit>
30. <http://js.cytoscape.org>
31. <https://github.com/Multiscale-Genomics/mg-process-fastq>
32. <https://github.com/Multiscale-Genomics/mg-process-macs2>
33. Hospital, A., Andrio, P., Fenollosa, C., Cicin-Sain, D., Orozco, M. and Gelpí, J.L. (2012) MDWeb and MDMoby: an integrated web-based platform for molecular dynamics simulations. *Bioinformatics*, 28, 1278-1279
34. Hospital, A., Faustino, I., Collepardo-Guevara, R., Gonzalez, C., Gelpi, J.L., Orozco, M. (2013) NAFlex: a web server for the study of nucleic acid flexibility. *Nucleic Acids Research*. 41, W47-W55.
35. <http://mmb.irbbarcelona.org/gitlab/NuclDynamics/nucleServ>
36. Jiménez-García, B., Pons, C. and Fernández-Recio, J. (2013) pyDockWEB: a web server for rigid-body protein-protein docking using electrostatics and desolvation scoring. *Bioinformatics*, 29, 1698-1699
37. Serra, F., Baù, D., Goodstadt, M., Castillo, D., Filion, G.J., and Marti-Renom, M.A. (2017), Automatic analysis and 3D-modelling of Hi-C data using TADbit reveals structural features of the fly chromatin colors. *PLoS Comput Biol*, 13, p. e1005665.
38. <https://github.com/Multiscale-Genomics/Chi-C>
39. Pancaldi V., Carrillo-de Santa-Pau E., Javierre B.M., Juan D., Fraser P., Spivakov M., Valencia A., Rico D. (2016) Integrating epigenomic data and 3D genomic structure with a new measure of chromatin assortativity. *Genome Biol*. 17:152.