www.emif.eu

European Medical Information Framework

Grant Agreement nº115372

D13.3 Data analysis and visualisation tools, including workflows for linkage with omics data v1

WP13 – Analysis, processing & visualisation methods and tools

V2.0
[Final version]

Lead beneficiary: EMBL
Date: 16/11/2015
Nature: P
Dissemination level: PU

TABLE OF CONTENTS

DOCUMENT INFORMATION

Grant Agreement Number	115372	**Acronym**	EMIF
Full title	European Medical Information Framework		
Project URL	http://www.emif.eu		
IMI Project officer	Ann Martin (Ann.Martin@imi.europa.eu)		

Deliverable	**Number**	13.3	**Title**	Data analysis and visualisation tools, including workflows for linkage with omics data v1
Work package	**Number**	13	**Title**	Analysis, processing & visualisation methods and tools

Delivery date	**Contractual**	Month 30	**Actual**	16/11/2015
Status	Current version / V2.0		Draft ☐	Final ☑
Nature	Report ☐ Prototype ☑ Other ☐			
Dissemination Level	Public ☑ Restricted ☐ Confidential ☐ (See note[1])			

Authors (Partner)	Natalja Kurbatova (EMBL)			
Responsible Author	Natalja Kurbatova		**Email**	natalja@ebi.ac.uk
	Partner	EMBL	**Phone**	+44 (0) 1223 492 597

[1] The title of the deliverable will be available at the public website (http://www.emif.eu). The availability of the deliverable document will depend on the degree of confidentiality that is assigned:

- **Public (PU):**
  - The executive summary will be available as soon as the document is submitted to the IMI JU.
  - A request button leading to a form will serve to request the full document.
  - After filling in some contact information, the user will be allowed to directly download the document through the website.
- **Restricted (RE):**
  - The executive summary will be available as soon as the document is submitted to the IMI JU.
  - A request button leading to a form will serve to request the full document.
  - Upon request, the PM will forward the request to the main author(s) to decide how to proceed.
  - In the Platform, active approval from the main author(s) and passive consent from partners will be needed to accept a deliverable sharing request.
  - A Non-Disclosure Agreement (NDA) may be placed, if needed.
- **Confidential (CO):** The title and the executive summary will be displayed on the website as soon as the document is submitted to the IMI JU.

DOCUMENT HISTORY

NAME	DATE	VERSION	DESCRIPTION
Natalja Kurbatova	13/05/15	1.0	First draft
Rudi Verbeeck	18/06/15	1.1	Review and changes
Paul Avillach	06/07/15	1.2	Review and comments
Natalja Kurbatova	14/07/15	1.3	Review and changes
Anna Bauer-Mehren, Nigel Hughes	06/11/15	1.3	Feedback from consortium review
Natalja Kurbatova	16/11/15	2.0	Final version

DEFINITIONS

- **Analysis (Broad definition).** Any "analytical method" used to get insights from data, based on descriptive or predictive statistics, modelling, simulation, graphs and other visualisation methods.
- **API.** Stands for application programming interface. In computing, API specifies how some software components should interact with each other.
- **Cluster.** A computer cluster consists of a set of loosely or tightly connected computers that work together so that, in many respects, they can be viewed as a single system.
- **Cloud computing.** Cloud computing is defined as a type of computing that relies on sharing computing resources rather than having local servers or personal devices to handle applications.
- **HTTP.** It stands for Hypertext Transfer Protocol – a protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web (Web).
- **HTTPS.** It stands for Secure Hypertext Transfer Protocol – a protocol for secure communication over a computer network. HTTPS is the result of layering the Hypertext Transfer Protocol (HTTP) on top of the SSL (Secure Sockets Layer) or TLS (Transport Layer Security) protocol, thus adding the security capabilities of SSL/TLS to standard HTTP communications. The main motivation for HTTPS is to provide authentication of the visited website and to protect the privacy and integrity of exchanged data.
- **Interface (In computing).** It is a device or program enabling a user to communicate with a computer.
- **Jump server.** It is a special-purpose computer on a network typically used to manage devices in a separate security zone. A jump server is a hardened and monitored device that spans two dissimilar security zones and provides a controlled means of access between them. User access is tightly controlled and monitored.
- **Pipeline (In the scope of this document).** Pipeline is a chain of data analysis components. Terms "pipeline" and "workflow" are interchangeable.
- **Platform (In the scope of this document).** The meaning of the term "platform" is very similar to the term "framework" – any base of technologies on which other technologies or processes are built. Platform in most of the cases has tools for developers and may provide computational power.
- **URL.** It stands for uniform resource locator is a reference to a resource that specifies the location of the resource on a computer network and a mechanism for retrieving it. A URL is a specific type of uniform resource identifier (URI).
- **VM (In computing).** In computing, VM stands for a virtual machine. Virtual machine is an emulation of a particular computer system. Virtual machines operate based on the computer architecture and functions of a real or hypothetical computer and their implementations may involve specialized hardware, software, or a combination of both.
- **Workflow (In the scope of this document).** A series of computational steps usually programmed to run at once. Terms "pipeline" and "workflow" are interchangeable.

EXECUTIVE SUMMARY

The main driver of deliverable D13.3 is the needs of the EMIF verticals. During our face-to-face meetings and conference calls the EMIF AD vertical requested a multi-omics data sharing solution that allows metadata attachment to the data, a pipeline sharing solution and high performance computing. We have incorporated all these requirements into a "Multi-omics Research Environment" developed by using EMBL-EBI Embassy Cloud that allows us to provide high performance computing.

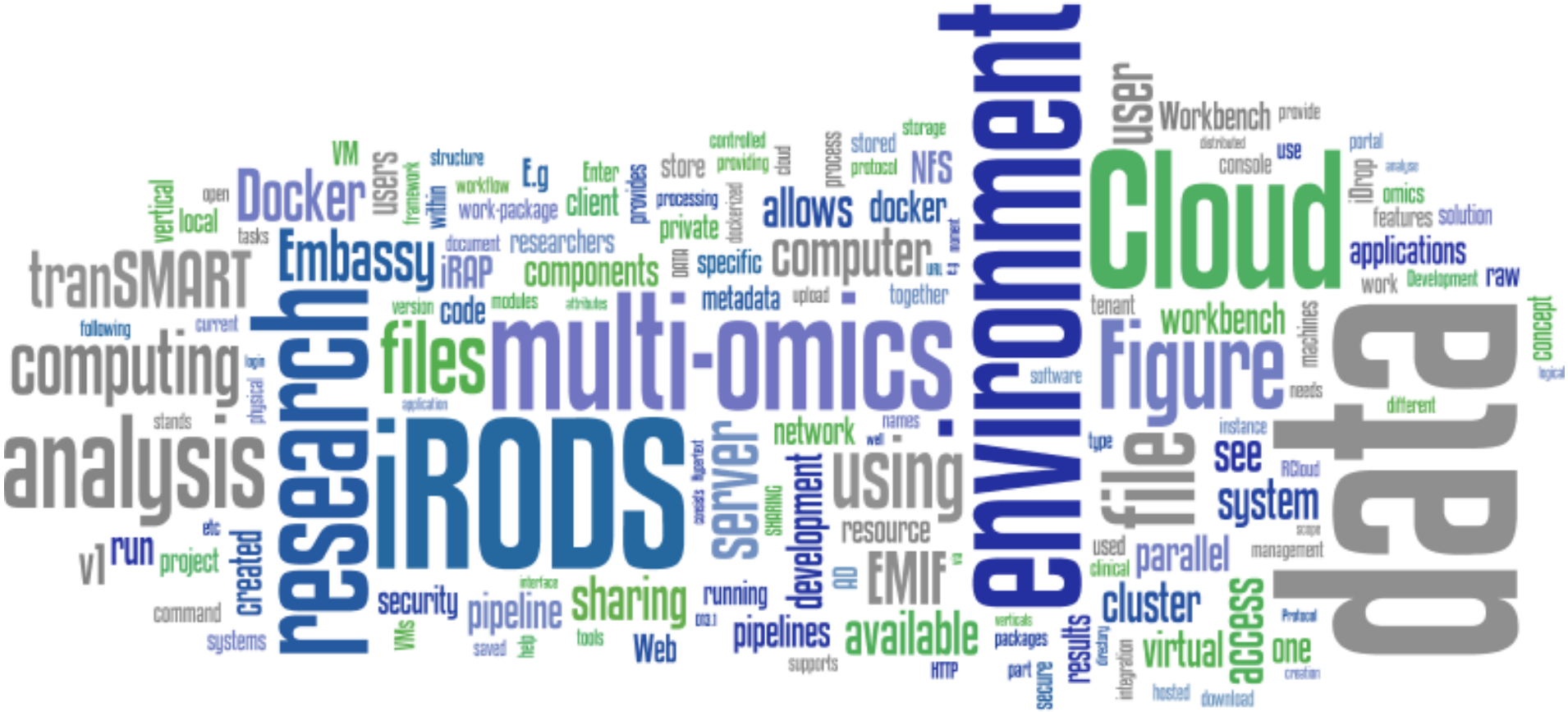
"Multi-omics Research Environment" at the current stage (v1) consists of the following components:

- iRODS for data sharing;
- Docker cluster for pipeline sharing;
- tranSMART for clinical data;
- RCloud for R parallel computing (in development).

All components are connected and use a shared file system. Docker cluster and RCloud benefit from cluster computing usage.

Deliverable D13.3 is a continuation of WP13 previous deliverables D13.1 ("Evaluation of technologies and tools available for data analysis and visualisation") and D13.2 ("Data analysis tools for vertical projects v1").

KEY WORDS (Wordle style)[2]



[2] http://www.wordle.net/.

1. INTRODUCTION

EMIF work-package WP13 is about developing analysis, processing and visualization methods and tools, in particular, to aid EMIF verticals. Now we are focused on the development of a "Multi-omics Research Environment" to meet the needs of the EMIF AD vertical. The environment is built on EMBL-EBI's "Embassy Cloud" - a secure, private virtual data-centre that uses VMware technology, more details here: http://www.embl-em.de/downloads/5/EMBL-EBI_Embassy_Cloud.pdf.

Deliverable D13.3 is the document describing “Multi-omics Research Environment” and its components available at the first version v1:

- iRODS for data sharing;
- Docker cluster for pipeline sharing and running;
- tranSMART instance for clinical data;
- RCloud for R parallel computing (in development).

The "Multi-omics Research Environment” components are installed on EMBL-EBI’s Embassy Cloud with an allocation of CPU, RAM and storage resources. Potentially the same components can be installed on any other cloud e.g. Amazon Web Services. In order to have access to your own EMBL-EBI's Embassy Cloud it is necessary to be in collaboration with EMBL-EBI and to sign an agreement.

The users of the environment are the researchers from the EMIF AD vertical. User accounts are created on individual basis by request and verification. All components are interconnected to provide solutions for the integrative data analysis (see Figure 1).

Initially all multi-omics and clinical data we are dealing with are anonymised. The environment is focused on -omics data and only the patient stratification for the -omics data analysis (e.g. AD, MCI, healthy) is absolutely necessary. The usage of more advanced patient level data components like tranSMART depends on the legal agreements signed by EMIF-AD with data custodians meaning that we can provide secure environment and appropriate documentation, however the legislation issues with data custodians have to be solved by research coordinators before data are loaded into the components of "Multi-omics Research Environment”.

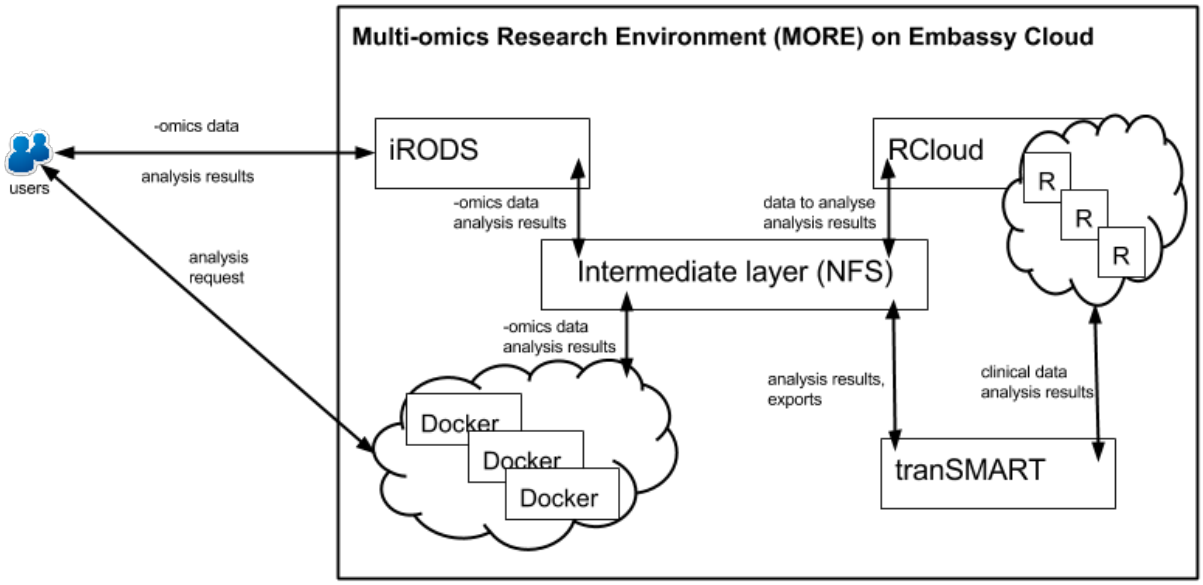


Figure 1. *Components of the "Multi-omics Research Environment" v1*

We have opened the discussion with work-package 14 concerning any possible security issues, especially in the light of the ongoing security and privacy requirements analysis. At the current version of the "Multi-omics Research Environment" (v1) our efforts for providing a secure solution include the following:

1. Embassy Cloud provides private virtual datacentres to tenants.

   Each tenant's Embassy Cloud is a secure, private virtual datacentre hosted within our VMware installation with an allocation of CPU, RAM and storage resources to each tenant. Each tenant is able to specify the internal and external network configuration of his virtual datacentre with firewall and VPN functions. The tenant takes on full programmatic control and integration of its virtual datacentre and each tenant organization is solely in charge of their own systems administration inside their Embassy.

   These are the general rules of Embassy Cloud and we are aware that it may pose a security risk if not managed properly. However, the tenant (in our case by the "Multi-omics Research Environment") may restrict the configurations for security reasons if needed. For example, if we are to use this environment as EMIF "private research environment".

   The Embassy Cloud is operated and hosted by EMBL-EBI in the same UK datacentres where EMBL-EBI's other services are hosted and yet it is logically outside the EMBL-EBI's LANs. EMBL-EBI's staff does not have access to the virtual machines run by a tenant.

2. The "Multi-omics Research Environment" is hosted within an Embassy tenancy providing its own private networks within Embassy Cloud. We employ a Jump server security model to secure access from the internet into the "Multi-omics Research Environment" network of VMs so user access is tightly controlled and monitored. Individual users access to the environment is organised through public-key cryptography protocol.

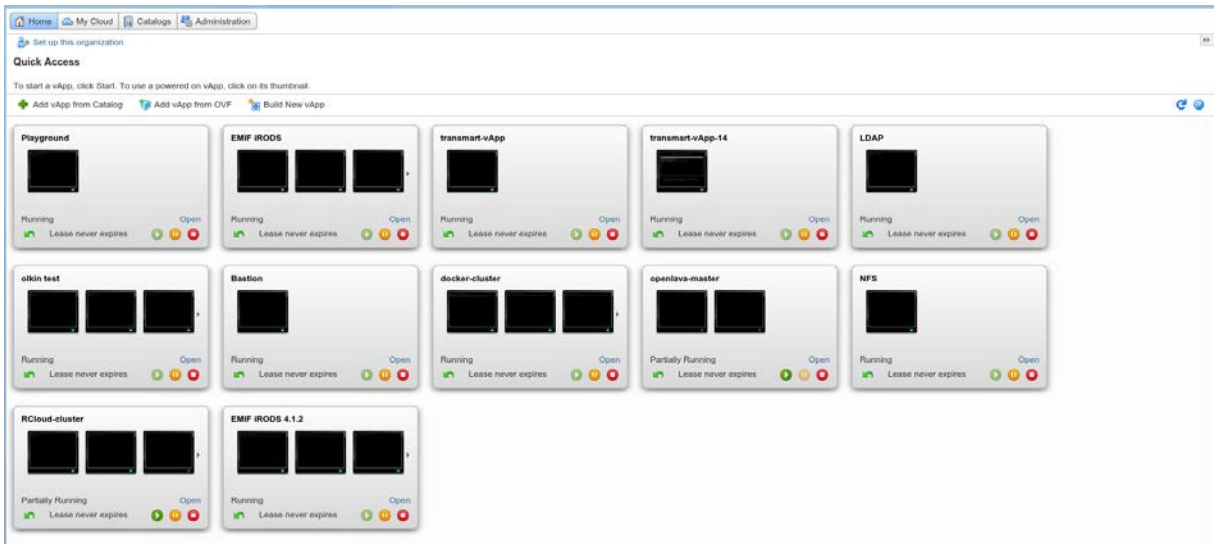


Figure 2. *Embassy cloud with "Multi-omics Research Environment" VMs*

2. DATA SHARING – iRODS

2.1.iRODS concept

iRODS is the integrated Rule-Oriented Data-management System, a community-driven, open source, data grid software solution. Fundamentally, iRODS helps researchers, archivists and others manage (organize, share, protect, and preserve) large sets of computer files. iRODS is highly configurable and easily extensible for a very wide range of use cases through user-defined Micro-services, without having to modify core code. It includes a set of features that blend well and augment each other to form a comprehensive whole. iRODS major features include:

- **High-performance network data transfer.** iRODS transfers data across the network in an integrated manner (get/put, read/write; parallel threads for large files), efficiently using up to 70% of available bandwidth.
- **A unified view of disparate data.** iRODS uses unique logical names that are separate from the names as stored physically, providing a global 'logical name-space'. The system (via the iCAT Metadata Catalog in a DBMS) keeps track of the names and locations of files so users do not have to.
- **Support for a wide range of physical storage.** iRODS accesses files stored in various systems including Unix and Windows files systems, archival storages systems etc; and does this in the same manner for each (i.e. to users they all look the same).
- **Easy backup and replication.** iRODS provides easy, automated replication and backup to multiple storage devices/locations at the physical level. Therefore, users access the files via the logical names and the system finds and gets the physical files.

- **Manages metadata** (data about data). iRODS metadata is both system (automatic) and user-defined, and stored in the iCAT Metadata Catalog running in a DBMS. Users can query the system to find, use, verify, etc. files with particular attributes (metadata).
- **Controlled access**. iRODS provides fine-grained controlled access, by user or group.
- **Policies, Rules and Micro-services.** iRODS innovative Rule Engine applies local and community Policies expressed as Rules and executed via server-side Micro-services.
- **Workflows**. A workflow is a series of steps to be done to process data. These can be executed as part of normal operation (e.g. a Rule can be run as a file is initially stored to automatically make an offsite replica) or as delayed or periodic Rules.
- **Management of large collections.** Various features, including: irsync (to check and synchronize between iRODS collections and local storage), audit trails (to record activity, verify authenticity, show compliance with human subjects access controls, etc.), metadata (to help organize and find data), bulk ingestion, etc.

iRODS administrators set the limits for maximal size of one file.

2.2.iRODS in "Multi-omics Research Environment"

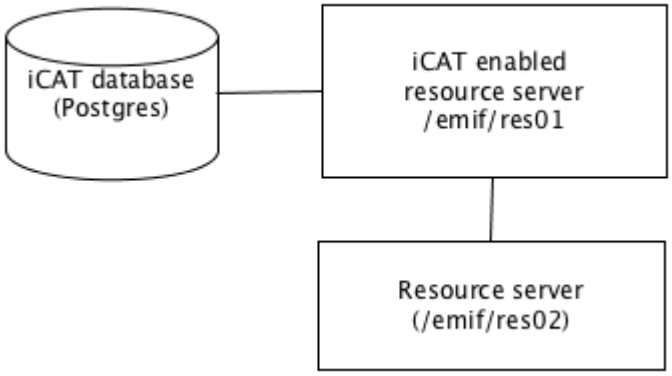


Figure 3. *iRODS implementation in the "Multi-omics Research Environment" v1*

In the current version (v1) of "Multi-omics Research Environment" iRODS consists of three parts: iCAT Postgres database to store meta-data, iCAT enabled resource server to store actual files and one more resource server for the data replication (see Figure 3). All three components are located on different physical volumes to make sure that uploaded data are securely stored and will not be lost if one of the hard disks is down.

iRODS is set up with 500MB for the file transfer through iDrop Web client and 50GB for the command line client. The user guide available on EMIF-AD wiki page explains to the users how to speed up file transfer process by allowing multiple threads.

We have configured iRODS to store metadata attached to the data files depending on the type of a file (see Table 1). This approach allows us firstly, to search through the data using attributes; and secondly, to facilitate data deposition into archives in the later stages of EMIF.

Attribute Name	Attribute Value(s) / Examples
file type:	raw spectral data file
	derived spectral data file
	metabolite assignment file
	free induction decay data file
	raw genomic data file
	derived genomic data file
	variant calling file
sample/aliquot id	E.g. Ex1-Col0-48h-Ag-1, Cecilia_AA_rerun05
experiment id	E.g. MTBLS2
raw spectral data file specific	
instrument	E.g. micrOTOF-Q, Thermo Electron Trace DSQ quadrupole
ion source	E.g. electrospray ionization, electron ionization
scan polarity	E.g. positive
scan m/z range	E.g. 100-1000, 50-450
mass analyzer	E.g. hybrid QTof, quadrupole

Table 1. *Attributes for EMIF AD multi-omics data*

Besides, each file has the following attributes: user created and timestamp created.

NFS server is used to share the iRODS data between running tasks in the cluster (see Figure 1).

There are different iRODS clients available in order to download/upload data files. For example, the “iCommands” is the command-line utility that interface to the iRODS system; *“iDrop Desktop”* is a desktop client and “iDrop Web” client can be accessed using a common web browser.

3. PIPELINES SHARING AND RUNNING - DOCKER

3.1. Docker concept

Docker is an open-source project that automates the deployment of applications inside software containers, by providing an additional layer of abstraction and automation of operating-system-level virtualization.

Docker is an open modern platform for developers and system administrators to build, ship, and run distributed applications. It Consists of Docker Engine, a portable, lightweight runtime and packaging tool, and Docker Hub, a cloud service for sharing applications and automating workflows.

The Docker Engine container comprises just the application and its dependencies. It runs as an isolated process in userspace on the host operating system, sharing the kernel with other containers. Thus, it enjoys the resource isolation and allocation benefits of VMs but is much more portable and efficient.

A cluster of docker machines gives the possibility to run dockerized analysis pipelines in parallel.

3.2. Docker in "Multi-omics Research Environment"

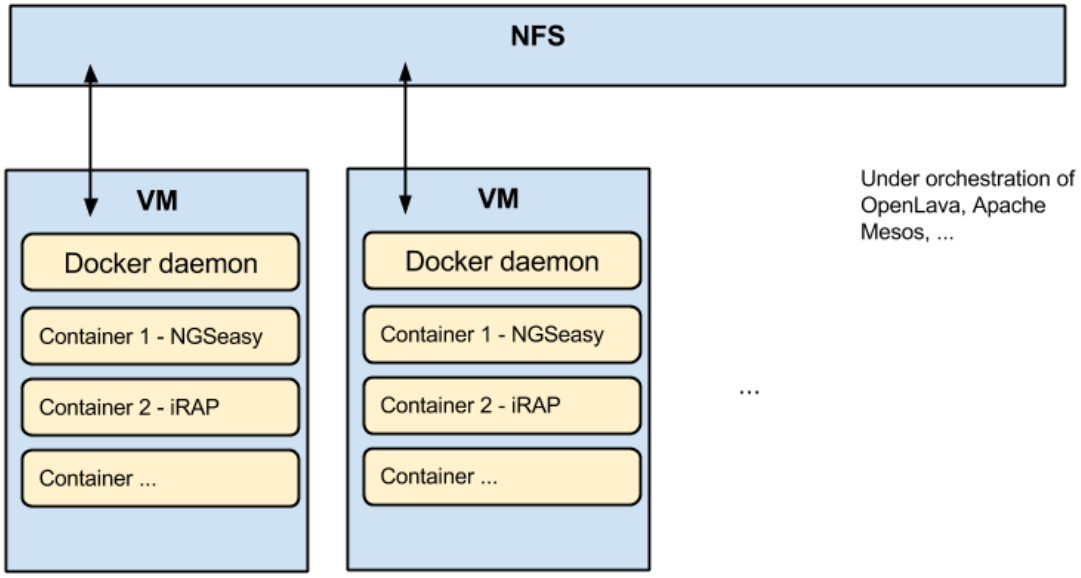


Figure 4. *Docker cluster and analysis pipelines in the "Multi-omics Research Environment" v1*

At the moment for the "Multi-omics Research Environment" v1 we are using a NFS server for data exchange: to make -omics data from iRODS available for the docker nodes to analyse them and to store analysis results. We have implemented the solution with OpenLava for workflow management amongst docker nodes. However, we are in the process of testing

other orchestration solutions since analysis pipelines are using different approaches to the distribution of tasks and jobs.

We have created a private docker repository with images of the analysis pipelines within the "Multi-omics Research Environment". We are planning to store private or manually build docker images there in case somebody needs it, e.g. manually build iRAP pipeline or NGSeasy pipeline components (see Figure 4). The dockerized pipelines will be added to the repositories upon users' request.

4. R ENVIRONMENT - R CLOUD

4.1.R Cloud concept

R Cloud is an R processing framework, scalable and distributed for exposure of R/Bioconductor packages to Java applications. It allows applications to perform R analysis on any biological data using numerous packages from Bioconductor and CRAN repositories. It facilitates creation and management of parallel computing clusters and allows applications to perform computational tasks in parallel on the cluster. The more detailed description of R Cloud can be found in deliverable D13.1.

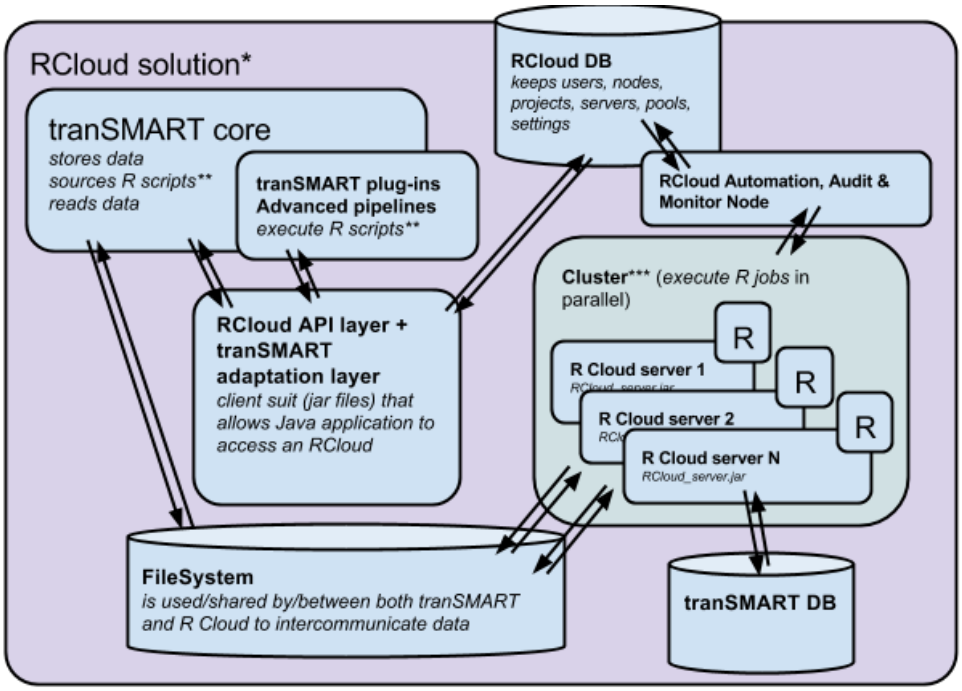


Figure 5. *R Cloud and tranSMART integration in the "Multi-omics Research Environment" v1*

We are using the R Cloud component of "Multi-omics Research Environment" for R development to combine together biostatisticians' and bioinformaticians' efforts in analysis of clinical and multi-omics data. R Cloud allows us to run heavy R jobs in parallel using a cluster of virtual machines under orchestration of OpenLava. Integration of tranSMART and RCloud allows us to accelerate R computing runs by tranSMART immensely (see Figure 5). RCloud will serve as backend computation engine for methods and datasets designed for parallel processing.

R Cloud is in development with plans to be open-sourced together with the tranSMART integration component in the next few months.

4.2.R Cloud in "Multi-omics Research Environment"

R Cloud Workbench is part of the R Cloud framework. It represents a graphical user interface to the R Cloud framework and allows users (computer scientists, statisticians, biologists) to develop, test and execute their analysis and visualisation methods on the data that will be co-located with the processing power.

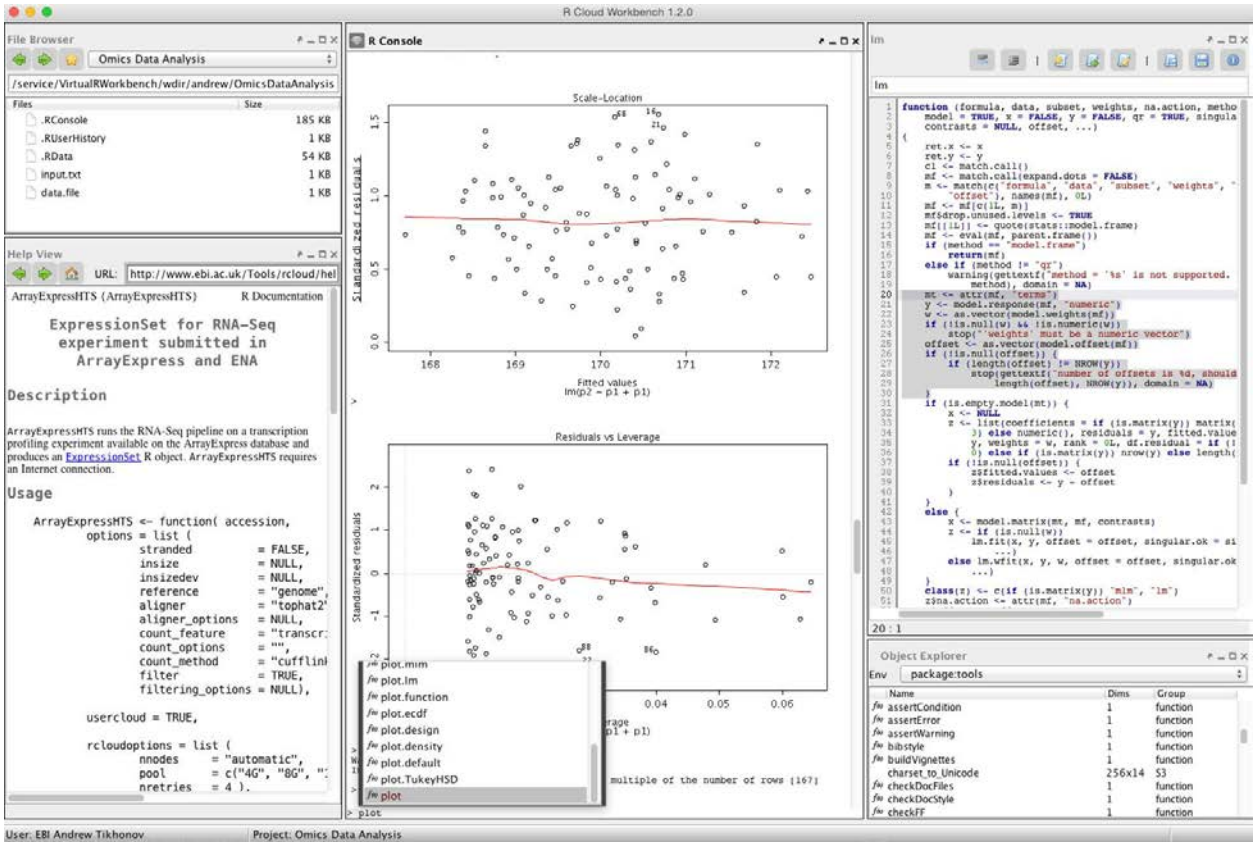


Figure 6. *R Cloud Workbench for the "Multi-omics Research Environment" v1*

The workbench is a fully-fledged development and analysis application that supports user profiles, user projects, simple data sharing, R analysis tools, package development, creation of parallel computing clusters and execution of pipelines.

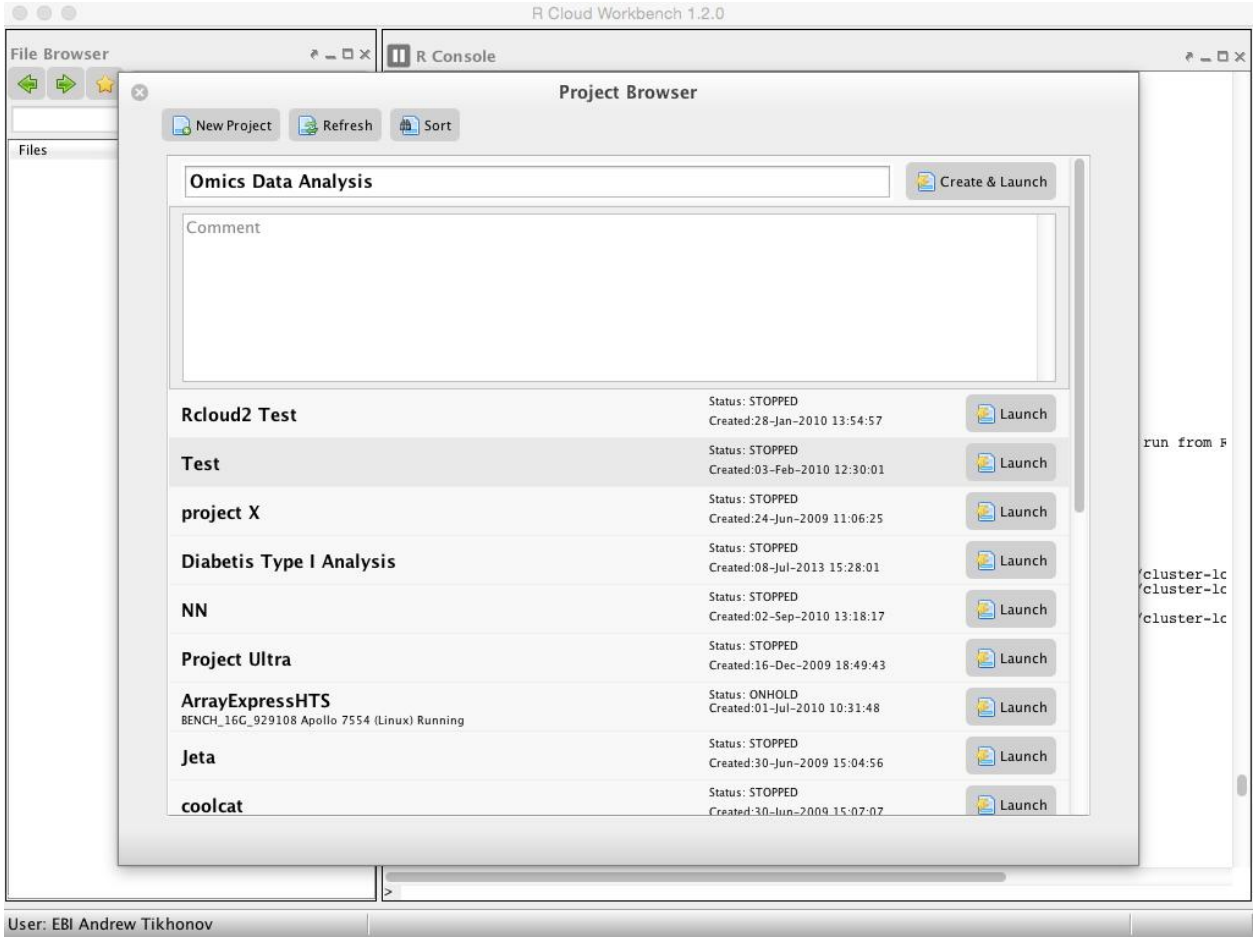


Figure 7. *Project Browser in the R Cloud Workbench*

A Separate environment created for each user keeps research work and results safe. Support for projects in the workbench offers researchers to structure their work. Projects are executed on the R Cloud servers. The R Cloud infrastructure in the background keeps track of the R server resources and replenishes them as needed according to the defined policies. When work is completed, the project state along with console output and R objects is saved and can be restored later when the project is loaded again. Researchers can disconnect from the project, close the workbench, open the workbench at a different research centre, connect to the project and show the results to colleagues and collaborators or continue the work. The project will be running in the R Cloud part of Embassy Cloud at EBI.

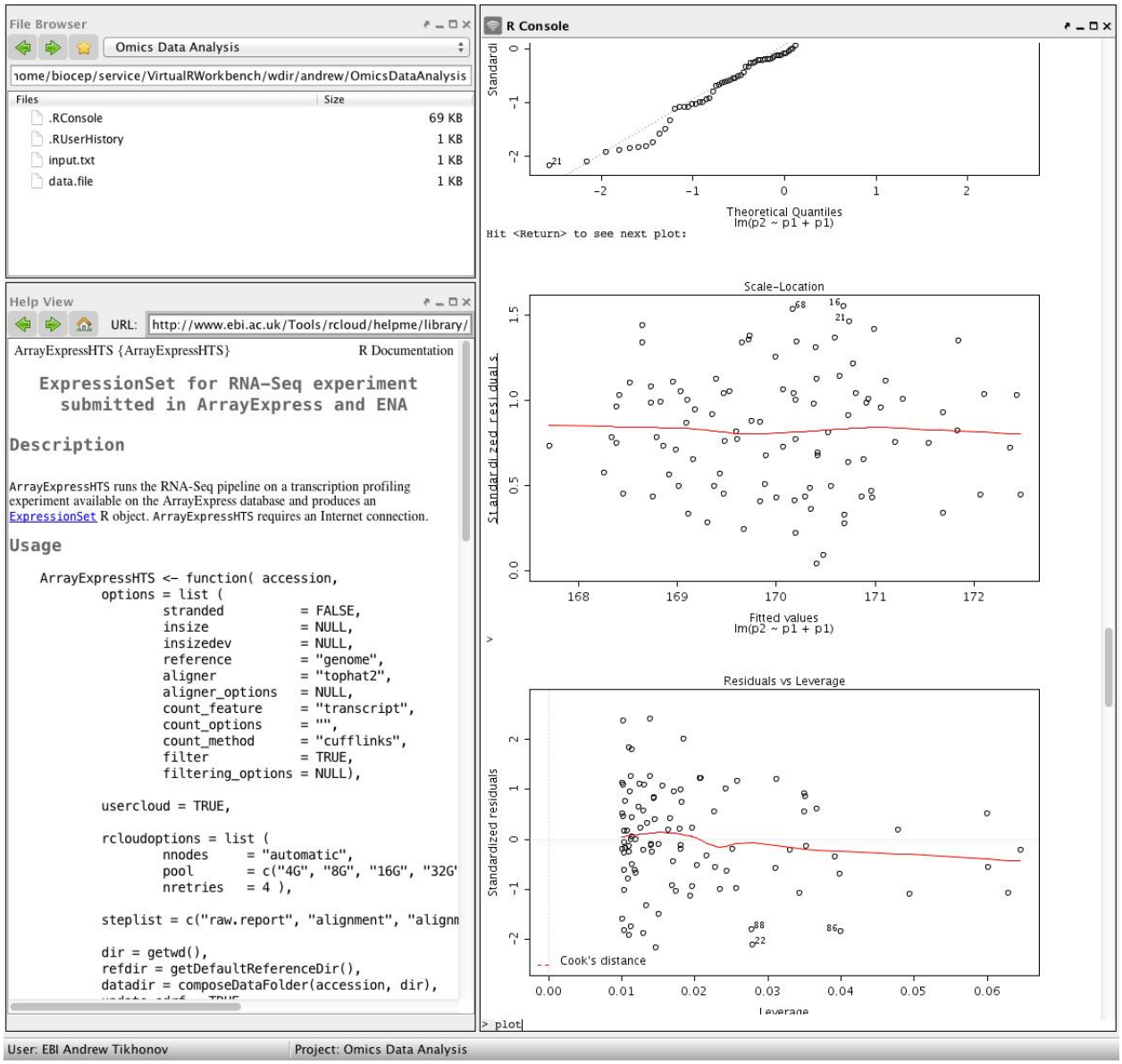


Figure 8. *R Cloud Workbench console, File Browser and Help View (option 1)*

R Cloud workbench supports code completion in the console and displays help pages for the commands and packages that are loaded in the environment. Thus, developers and researchers can always find a suggestion and a help page for a tricky function from within the workbench, which helps productivity.

The Analysis command console collects and displays analysis output as well as images. Images can be saved as png and jpg files and the console output can be saved as a research log into a .pdf file.

The File browser in the workbench supports drag & drop and allows researchers to easily upload and download necessary input and result files. The file structure is displayed in the natural form of a file tree with file size and creation date information available and shown as

well. There are additional features available like quick file preview, file and folder management, development and file sharing options.

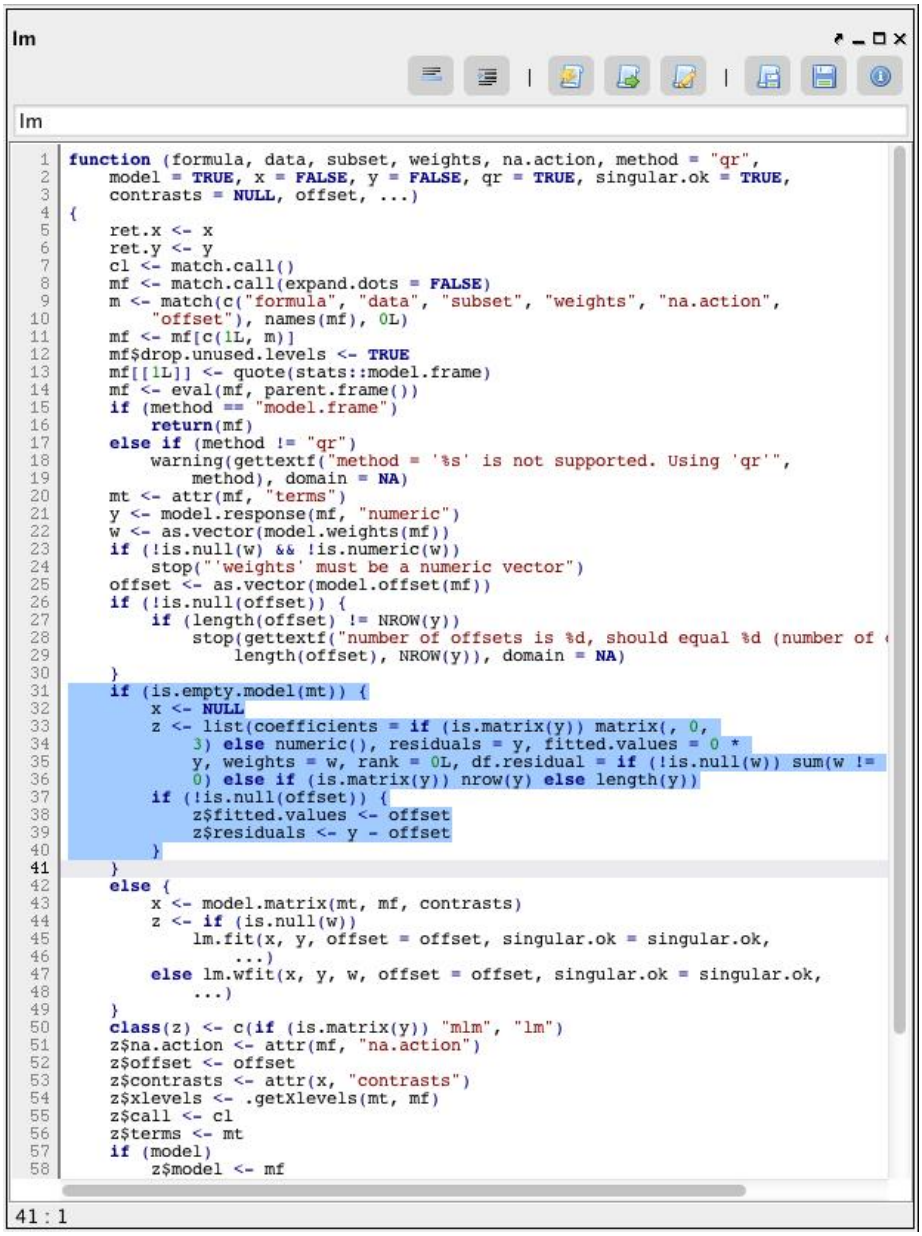


Figure 9. *R Cloud Workbench Editor*

The R code editor in the workbench supports syntax highlighting, code and file path completion, fast code jumps and navigation, invocation of help, automatic indentation and a number of other essential features a standard code editor should have. We at EBI have developed a number of packages in the editor of the R Cloud Workbench.

Works to setup the R Cloud Workbench as part of the EMIF Embassy Cloud are ongoing.

5. CLINICAL DATA - tranSMART

The open source tranSMART platform provides researchers with a single self-service web portal with access to phenotypic, 'omics, and unstructured text-based data from multiple sources, combined with search and analysis capabilities. A more detailed description of tranSMART can be found in deliverables D13.2 ("Data analysis tools for vertical projects v1"), D14.2 ("A data management solution for vertical projects, version 1") and D14.5 ("A data management solution for vertical projects, version 2").

tranSMART instance version 1.2 is installed on one of the virtual machines of Embassy Cloud. Postgres database is located on another VM.

We are planning to use tranSMART to bring together patient level data for the vertical tracks. For instance, we could automatically load -omics analysis results into tranSMART. This tranSMART instance is integrated with R Cloud that allows parallel R computation (see Figure 5).

6. EXAMPLES OF USAGE

"Multi-omics Research Environment" is in active development and test stage. We will run the further set of tests as soon as EMIF AD vertical -omics data becomes available. At the moment we are testing the environment by using publicly available data from EMBL-EBI archives (ENA, ArrayExpress).

Below are the examples of "Multi-omics Research Environment" usage. Let's imagine the following simple scenario: user would like to upload raw data files to iRODS, then he or she would like to run one of the dockerized pipelines to analyse data and finally, to download the analysis results.

1. Upload data by using iRODS client "iCommands". "iCommands" can be installed for all widely used platforms (see https://pods.iplantcollaborative.org/wiki/display/DS/Using+iCommands for more details) and data upload is done by **iput** command. First, on my local computer outside of the cloud I am running **iinit** command to store the connection details.

```
iinit
One or more fields in your iRODS environment file (.irodsEnv)
are
missing; please enter them.
Enter the host name (DNS) of the server to connect
to:193.62.52.17
Enter the port number:1257
Enter your irods user name:testrods
Enter your irods zone:emif
Those values will be added to your environment file (for use by
other i-commands) if the login succeeds.

Enter your current iRODS password: …
iput HG00109.3.M_120202_5_1.fastq.gz
iput HG00109.3.M_120202_5_2.fastq.gz
ils
/emif/home/testrods:
  HG00109.3.M_120202_5_1.fastq.gz
  HG00109.3.M_120202_5_2.fastq.gz
```

   Another way to upload data is by using web iRODS client "iDrop Web" (see Figure 10, 11). "iDrop Web" client for the EMIF iRODS is available in any browser by URL: https://193.62.52.17:8551/idrop-web2/. Accounts are created for individual users.

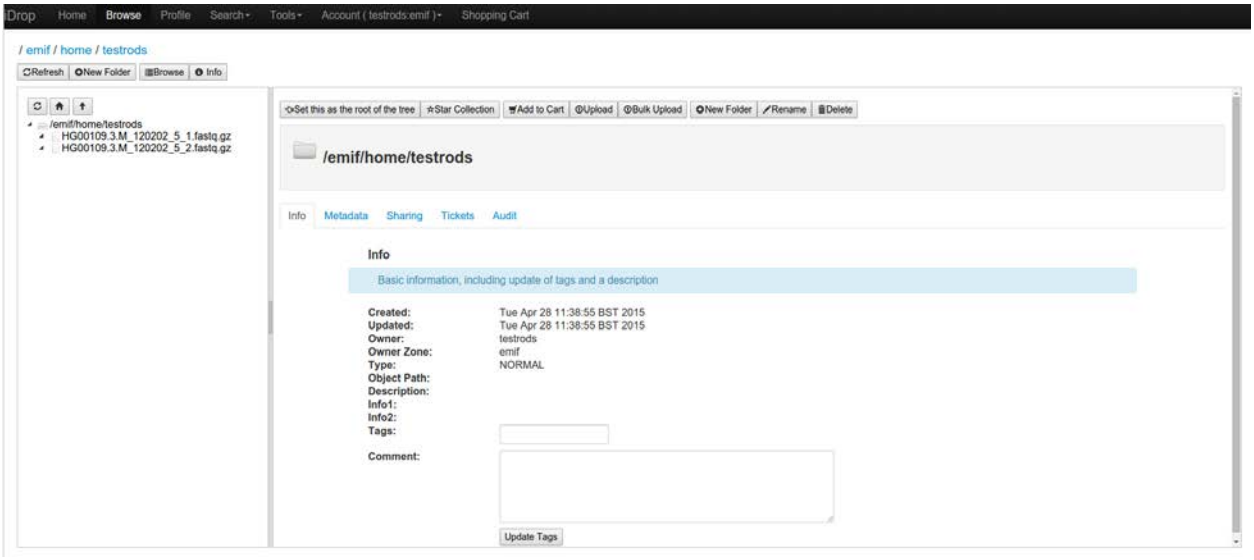


Figure 10. *Data uploading to EMIF iRODS by using iDrop Web client*

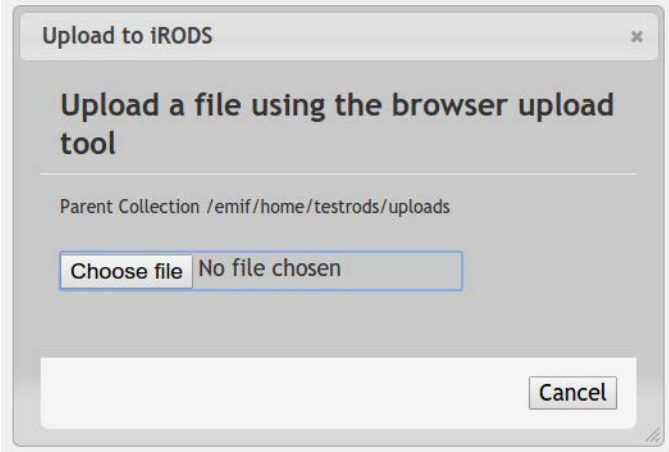


Figure 11. *Data uploading to EMIF iRODS by using iDrop Web client*

2. Now user is ready to analyse data. The first step here is to login to the cloud from local computer by using specially created portal VM (here under IP address 192.168.1.33).

```
ssh testuser@192.168.1.33
testuser@192.168.1.33's password:
...
```

The second step is to copy files from iRODS to NFS server: **ils** command lists the files available for the user to download; NFS server is mounted to the portal VM so copying process can be done directly into the directory for the data analysis (here /nfs/iRAP/); **iget** command downloads the file.

```
ils
/emif/home/testrods:
  HG00109.3.M_120202_5_1.fastq.gz
```

```
  HG00109.3.M_120202_5_2.fastq.gz

cd /nfs/iRAP/raw_data/home_sapiens
iget HG00109.3.M_120202_5_1.fastq.gz
iget HG00109.3.M_120202_5_2.fastq.gz
ls
ERR188040_1.fastq.gz
ERR188040_2.fastq.gz
ERR188231_1.fastq.gz
ERR188231_2.fastq.gz
HG00109.3.M_120202_5_1.fastq.gz
HG00109.3.M_120202_5_2.fastq.gz
```

Often analysis pipelines require a special directory structure with raw data, reference (e.g. genome sequences and annotations) data and configuration files. NFS server shared between docker VMs (VMs with docker engines) is used to meet this requirement.

Example of directory structure for the iRAP pipeline is presented on Figure 12, notably files from the raw_data subdirectory are created by using **iget** command.

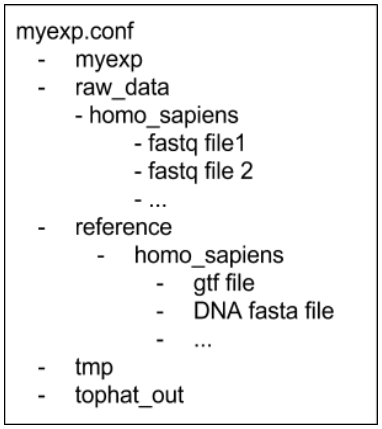


Figure 12. *iRAP directories on NFS server*

3. Run dockerized iRAP transcriptomic pipeline using NFS server for data input/output (see Figure 13).
4. iRAP results of the differential expression analysis are saved on the NFS server. We can place the results back to iRODS and then download them from the environment to the outside world. From portal VM:

   ```
   iput iRAP_results.tar.gz
   ```

   From local computer:

   ```
   iget iRAP_results.tar.gz
   ```

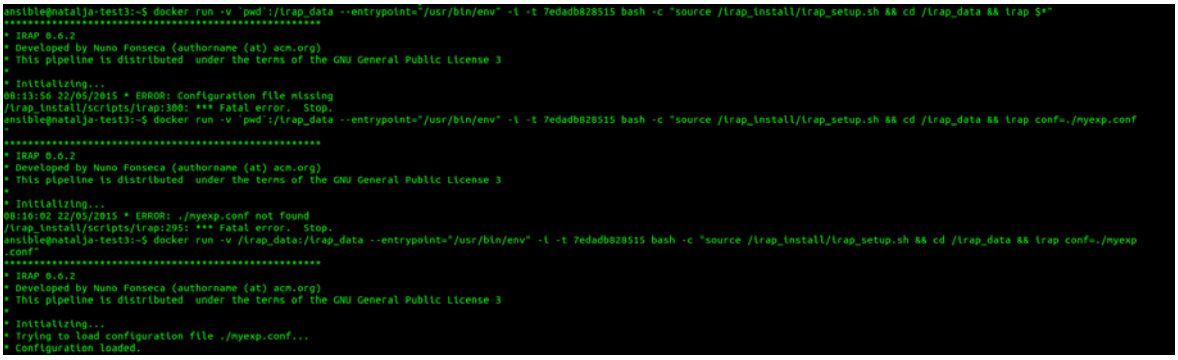


Figure 13. *iRAP is running on docker cluster of "Multi-omics Research Environment"*

7. NEXT STEPS

We will continue our work with "Multi-omics Research Environment". Our next steps include:

- **Testing** of "Multi-omics Research Environment" **with publicly available datasets** (e.g. AddNeuroMed).
- **Collecting and analysing multi-omics data from 1000 samples AD cohort** by using "Multi-omics Research Environment"; in this process the users of the environment will be representatives from AD modalities (metabolomics, proteomics, genomics).
- **Gathering of EMIF AD users experiences** with the "Multi-omics Research Environment".
- **Modification of "Multi-omics Research Environment"** which includes also possibility to delete exiting components or add new ones based on gathered EMIF users experience.

These modifications and final tuning will lead us to the three remaining deliverables: D13.4 (Data analysis tools for vertical projects v2), D13.5 (Data analysis and visualisation tools, including workflows for linkage with omics data v2) and eventually to the D13.6 (Final suite of modules and tools for data analysis, visualisation and linkage of EHR data with omics data). Timelines here are appropriate expected delivery dates: month 36 to month 60 accordingly.

Our choice of technologies and cloud platform (VMware) is not final except for the usage of ansible scripts for the environment administration and installation. Ansible scripts allow to transfer the environment from one technology to another. VMWare cloud platform has been chosen for practical reasons; this platform was available on Embassy Cloud when the project started.

At the moment we are testing "Multi-omics Research Environment" portability and scalability by transferring it from VMWare cloud platform onto OpenStack cloud platform. Our plans include also testing of "Multi-omics Research Environment" on Amazon Web Services and Google Cloud platforms through possible collaborations (e.g. with Harvard Medical School who are using AWS already) and/or through getting tenancy on those platforms.