HUTER

The Human Uterus Cell Atlas

Project Acronym:	**HUTER**
Project Full Name:	**Human Uterus Cell Atlas**
Call identifier:	H2020-SC1-2019-Single-Stage-RTD
Topic:	SC1-BHC-31-2019
Grant Agreement No:	874867
Start date of Project:	01/01/2020
Project Duration	2,5 years (30 months)
Document due date:	28/02/2021
Submission Date	01/02/2021
Leader of this report:	BAHIA
Deliverable no:	7.2
Deliverable name:	Visual system & digitalization software
Dissemination level:	**Public**

Version History

Version	Date	Details
1.0	01/02/2022	Deliverable completed.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 874867.

The opinions expressed in this document reflect only the author's view and in no way reflect the European Commission's opinions. The European Commission is not responsible for any use that may be made of the information it contains.

DELIVERABLE 7.2

Visual system & Digitalization software

Table of Contents

List of Acronyms

API	Application programming interface	**DICOM**	Digital Imaging and Communications in Medicine	**RNA**	Ribonucleic acid
AWS	Amazon Web Services	**DNA**	Deoxyribonucleic acid	**scRNAseq**	Single cell RNA sequencing
BAHIA	Bahia Software SLU	**eCRF**	Electronic Case Report Form	**smFISH**	Single-molecule RNA FISH
CLI	Command line interface	**GDC**	Genome Data Commons	**TCGA**	The Cancer Genome Atlas
CRF	Case report form	**HCA**	Human Cell Atlas	**TMA**	Tissue microarray
D2.1	Deliverable 2.1	**HUTER**	Human Uterus Cell Atlas	**TSV**	Tab-separated values
D2.4	Deliverable 2.4	**IDE**	integrated development environment	**UTC**	Coordinated Universal Time
D2.5	Deliverable 2.5	**IT**	Information technology	**WDL**	Workflow description language
D7.1	Deliverable 7.1	**MS**	Microsoft	**WP7**	Work Package 7
D7.2	Deliverable 7.2	**NEMA**	ANSI-accredited Standards Developing Organization		
DCP	Data Coordination Platform	**PACS**	Picture Archiving and Communication System		

1. PURPOSE OF THIS DOCUMENT

The development of the activities regarding Deliverable D7.2 which is defined as “Visual system & Digitalization software” has been successfully completed. The purpose of this document is to provide an overview about the work performed in the framework of this Deliverable as part of the WP7 (entitled “Platform integration”).

The activities performed in the context of this deliverable are divided in two main axes: the visual interface of the platform and the digitalization software developed. The visual interface of the platform is mainly formed of visual interface of open tools that have been integrated in the platform as commented in previous deliverables as well as own developments. The digitalization software includes specific libraries designed with the aim of transforming research images generated in HUTER into the digital imaging standard (DICOM).

The Visual System and Digitalization software presented herein were developed in the framework of the WP7 as part of the Task 7.2 and 7.3 that BAHIA leads.

1.1. Related documents

Documents linked to past actions already delivered:

HUTER_WP2_D2.2_Deployment_of_HUTER_cloud_infrastructure

HUTER_WP2_D2.3_Beta_version_of_data_access_tools

HUTER_WP7_D7.1_Final_design_of_HUTER_platform_architecture

Documents linked to future actions to be delivered:

HUTER_WP2_D2.4_Data_access_tools_and_DICOM_visualization_tool

HUTER_WP2_D2.5_DICOM implementation

2. STATUS OF DELIVERABLE

The deployment of the Visual System of the HUTER Platform as well as the Digitalization Software for research images have been completed. Therefore, we consider the deliverable to be completed. However, we will continue monitoring and implementing potential improvements and refinements if needed until the project is completed.

3. GENERAL INTRODUCTION

The Human Uterus Cell Atlas project (HUTER) is focused on creating the cellular reference map of the human uterus. For this purpose, HUTER researchers will generate vast amounts of molecular and imaging data from cell sequencing technologies and uterus cells of women samples. In this context, BAHIA leads the development of the HUTER advanced digital platform in order to help researchers in several hard tasks of this project: gathering their huge amount of data, performing high throughput analysis over cell sequencing data, saving and sharing clinical and experimental metadata, sharing results among partners, visualizing advanced microscopy images, transforming closed format imaging files into an open medical standard format (DICOM), visualizing and downloading gathered data and so forth as well as other functionalities related to the HUTER management.

As introduced, the HUTER Platform has been designed to gather different kind of data, molecular (such as single cell transcriptomics, epigenomics, genomics, imaging and so forth) and clinical or biological metadata. In this context, BAHIA has been working with HUTER bioinformaticians and researchers in order to analyse their research process and data workflow. This work has let BAHIA know information to identify the best tools (detailed in D2.3 and D2.4) and the best visual interface to fulfil research needs and to provide a good user experience for experts. The visual system of the HUTER Platform is integrated by all the visual interfaces of the different tools integrated in the platform. Some proofs of different visual interfaces integrated in the HUTER Platform are provided on this report. Some of these proofs will include not only functional brief descriptions of the tools currently integrated in the platform (widely detailed in deliverables D2.3 and D2.4) but also screenshots that allow the reader to see the visual system design working properly over the HUTER Platform.

Related to the visualization of research images such as tissue microarray, immunofluorescence images or smFISH, it will be possible thanks to the integration of an advanced DICOM viewer developed and integrated in the context of D2.4. This viewer will be compatible with DICOM images because HUTER has included the extension and support of the open DICOM standard, already proved in the medical image field, to the advanced research images as one of its objectives to foster standards. The state-of-the-art research equipment that generates these kinds of images sometimes include their own non-open output format as default which hampers sharing data and results among researchers even from the same institution due to format incompatibilities. The adoption of DICOM standard will help not only to overcome these incompatibility barriers, but also to facilitate the compatibility and adoption of these images in hospitals in the future. It is important to highlight that DICOM is currently far and away the standard of choice for exchanging medical imaging data within hospitals and even nascent technologies like digital pathology are quickly converging on it. This foundation means that almost any clinical applications will be expected to consume and operate on DICOM images in the future. In this context, the Digitalization Software developed that includes technical libraries required to transform HUTER images into DICOM (to be viewable by the DICOM viewer) is detailed in this deliverable. The technical implementation of DICOM standard with the support of HUTER images will be detailed in D2.5.

Collectively, the integration of the visual system and digitalization software in the platform will simplify the visualization of advanced data. These integrations allow to reduce the dimensionality of complex data (for instance scRNAseq matrix) hosted in the platform and therefore the workload for users. Finally, thanks to the digitalization software, the HUTER Platform is already capable to transform HUTER images into medical image standard (DICOM) to visualize them over the DICOM viewer later (DICOM viewer expected in D2.4).

4. VISUAL SYSTEM

4.1. Introduction

HUTER Platform is a Cloud Platform that was designed to provide a complete suite of tools for data analysis and processing in biological research environments. Its main aim is easing the research and collaboration along the assays.

With this intention, the current platform supplies tools for digital information store, treatment and sharing for researchers, principally bioinformaticians. Specially, data management tools are focused on the reduction of the information complexity by addressing various factors like the amount of stored data or the categorization of data files among other features. For instance, bioinformaticians are released from knowing what kind of data is contained in every single file because HUTER Platform indexes all the files and link that indexation to the concerning sample and its traits. So, these tools offer access to the information in a functional way, this means asking for what would like to be studied more than which kind of file contains that information. The resulting platform tends to help researchers in focusing on what kind of information they liked to study and reducing the IT solutions complexity.

The visual system of the HUTER Platform comprises all the visualization layer which is formed of the user interfaces of different access tools integrated in the platform. In other words, this is the component that users of the platform can see and interact with it. For instance, HUTER Platform allows user to storage an unlimited amount of data in a shared environment what could lead users into a data location issue. For tackling this situation, HUTER Platform delivers a visual tool for data browsing avoiding any aspect of the technological solution behind the data storage.

Herein, the user interfaces of tools devoted to visualizing and interact with OMICs data whose integration constitutes the main part of the visual system will be detailed. Taking into account that most of the tools for research data are openly available and with the aim of contribute to open science and innovation, some of the existent cutting-edge tools were integrated with other services of the HUTER Platform, like data processing, storage or file indexation. Those applications like Cellxgene are devoted to reducing the cognitive workload data in visualization and outcomes extraction.

All this integrative effort resulted with a set of tools which forms the visual system for data management and exploitation of the HUTER Platform.

4.2. HUTER CLI

Before explaining how HUTER Tools provides a visual system for data exploitation, it is important to bear in mind that the platform architecture described in Deliverable D2.1 and D7.1. In essence, HUTER Platform concentrates the logic of the data treatment in a service layer deployed on the cloud platform. Then, users are allowed to access and process data by executing these services for a later visualization with devoted applications like Cellxgene. Hence, so as to interact with services, users find the HUTER tools suite at their disposal.

One of these tools is the HUTER CLI application, which is an easy-to-use software that will allow users to consume almost every service of the HUTER Platform while it assures the integrity and confidentiality of the accessed data. Among its functionalities, HUTER CLI will let to upload and download any file in the storage system of the HUTER Platform (AWS S3) as well as their treatment through various processes. For instance, the information contained in uploaded files can be indexed in the HUTER Platform database for data linking or analysing through users developed workflows. Further section *4.2.7 Workflow execution for analysis and processing* will detail how to execute data analysis and transformation in the platform.

Coming back to HUTER CLI interface, it is based on a text interface so users can interact with the platform by typing commands in their device command-line, receiving responses as well. After consultations with HUTER bioinformaticians, an agreement was reached for using a CLI application as the best interface for this project because the followings reasons:

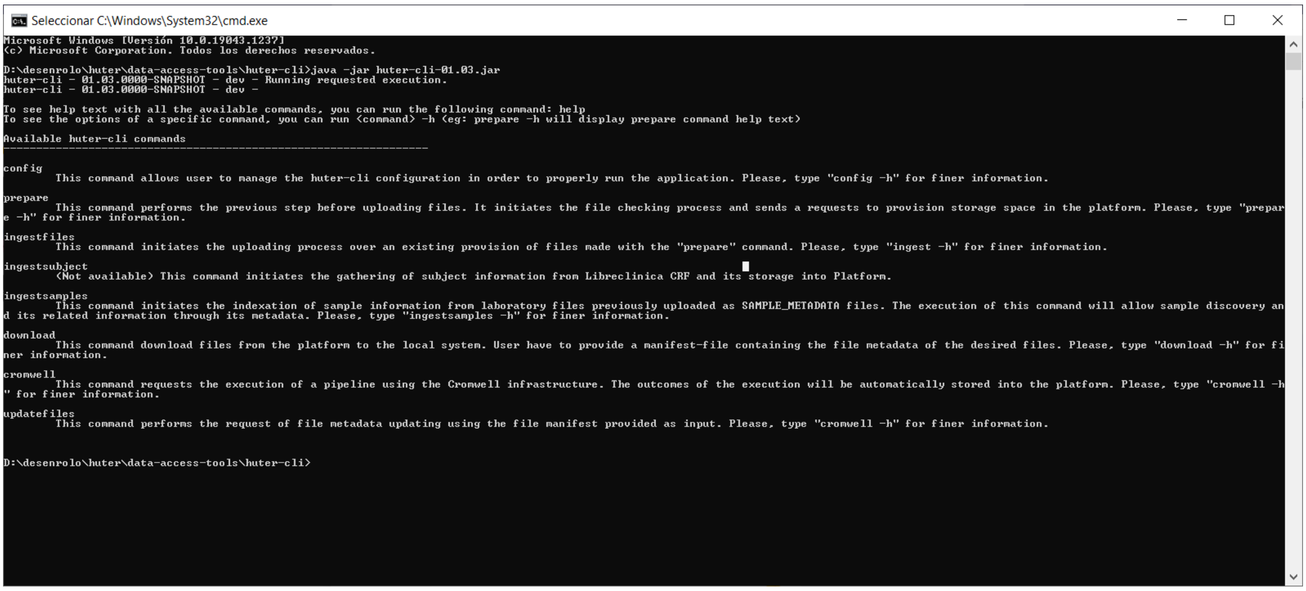
- Easy adoption: bioinformaticians will be the main target users of the HUTER Platform. They are used to work with tools like R or Linux, therefore their workflows are already adapted to command line interfaces.
- Less technical requirements: the CLI needs less system resources thus it is faster than GUI in limited resource environments.
- Automatic functions: Command Line Interfaces (CLI) are usually provided to middle or high technical users because this sort of software allows its easy integration in custom processes CLI. Regarding to HUTER Project bioinformaticians, HUTER CLI will allow them to perform several tasks automatically in scripts that contain the required commands.

As the aim of this document is to explain how HUTER Platform provides a visual tool for data management, next sections will be devoted to show how users can access to HUTER data and services through this tool. For a better understanding of the HUTER CLI design, technical requirements and functions, the document D2.4 will be delivered.

4.2.1. HUTER CLI interaction

HUTER CLI application provides users the capacity to exploit functionalities running into de HUTER Platform as cloud services. This means that users can take advantage of the high power of a cloud platform for data analysis and storage just sending requests to those HUTER services.

HUTER CLI eases the invocation of the HUTER Platform Services through a set of commands that can be complemented with additional parameters to customize the service execution. Commands are keywords that users can type to generate that request of execution to a specific service. Adding arguments to any command, the request can be configured in the desired way, for instance, to point out which files should be analyzed.



```
Microsoft Windows [Versión 10.0.19043.1237]
(c) Microsoft Corporation. Todos los derechos reservados.

D:\desenrolo\huter\data-access-tools\huter-cli>java -jar huter-cli-01.03.jar
huter-cli - 01.03.0000-SNAPSHOT - dev - Running requested execution.
huter-cli - 01.03.0000-SNAPSHOT - dev -

To see help text with all the available commands, you can run the following command: help
To see the options of a specific command, you can run <command> -h (eg: prepare -h will display prepare command help text)

Available huter-cli commands
-----------------------------------------------------------------

config
        This command allows user to manage the huter-cli configuration in order to properly run the application. Please, type "config -h" for finer information.

prepare
        This command performs the previous step before uploading files. It initiates the file checking process and sends a requests to provision storage space in the platform. Please, type "prepar
e -h" for finer information.

ingestfiles
        This command initiates the uploading process over an existing provision of files made with the "prepare" command. Please, type "ingest -h" for finer information.

ingestsubject
        (Not available) This command initiates the gathering of subject information from Libreclinica CRF and its storage into Platform.

ingestsamples
        This command initiates the indexation of sample information from laboratory files previously uploaded as SAMPLE_METADATA files. The execution of this command will allow sample discovery an
d its related information through its metadata. Please, type "ingestsamples -h" for finer information.

download
        This command download files from the platform to the local system. User have to provide a manifest-file containing the file metadata of the desired files. Please, type "download -h" for fi
ner information.

cromwell
        This command requests the execution of a pipeline using the Cromwell infrastructure. The outcomes of the execution will be automatically stored into the platform. Please, type "cromwell -h
" for finer information.

updatefiles
        This command performs the request of file metadata updating using the file manifest provided as input. Please, type "cromwell -h" for finer information.


D:\desenrolo\huter\data-access-tools\huter-cli>
```

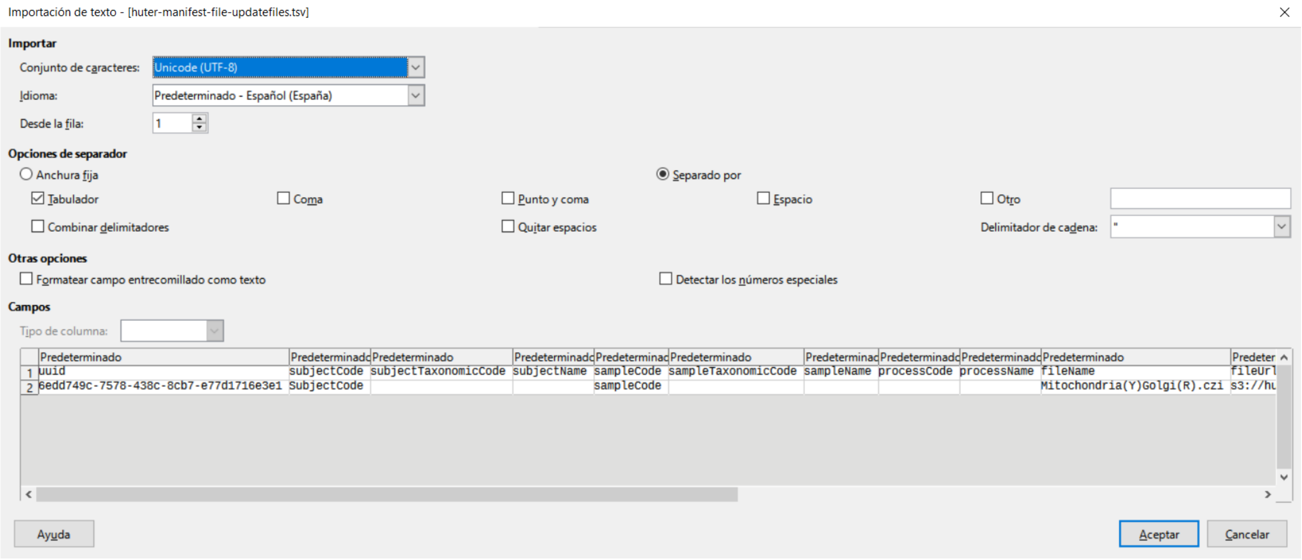
Figure 1 - huter-cli help

HUTER CLI is expected to be used by technical staff such as bioinformaticians, so it follows the well-known CLI application guidelines for interaction. For instance, it displays help information about how to use any command, including the accepted arguments and their expected values.

The set of available commands and their functionality can be seen at *Figure 1 - huter-cli help*. If a better explanation is required, every command provides a "-h" option that displays detailed information of how to use it, their parameters and the expected values. Furthermore, every command execution HUTER CLI will display information when a significant point was reached in other to give feedback and process tracking. Moreover, additional data about the HUTER CLI version, environment and current user are displayed. If a finest detail of executed steps is required, HUTER CLI creates and feeds a log file with more accurate information and detailed errors, if they happen.

4.2.1.1. Previous considerations: Manifest file

Some features of the HUTER CLI require to point at stored files in the platform. However, introducing a list of long file names in a command line will not be usable. Besides, for technical and security reasons, users are not allowed to know the location of files. Instead of that, HUTER Platform has deployed the Data browser tool for locating the stored files among other functions. It lets users to find desired files in the platform and generate a file with the selected files information. After conversations with HUTER bioinformaticians, manifest file format was selected as a flexible and simple structure for large amount of data sharing. Manifest file is a TSV file so it can be processed by common application such as MS Excel or LibreOffice Calc, see *Figure 2 - TSV file in LibreCalc (Libre Office Suite)*, allowing users to perform data calculations or value assignments to a large volume of registries. However, those actions will not affect to the TSV format what is simple and cross-platform understandable.



Importación de texto - [huter-manifest-file-updatefiles.tsv]

Importar

Conjunto de caracteres: Unicode (UTF-8)

Idioma: Predeterminado - Español (España)

Desde la fila: 1

Opciones de separador

○ Anchura fija ● Separado por

☑ Tabulador ☐ Coma ☐ Punto y coma ☐ Espacio ☐ Otro

☐ Combinar delimitadores ☐ Quitar espacios Delimitador de cadena: "

Otras opciones

☐ Formatear campo entrecomillado como texto ☐ Detectar los números especiales

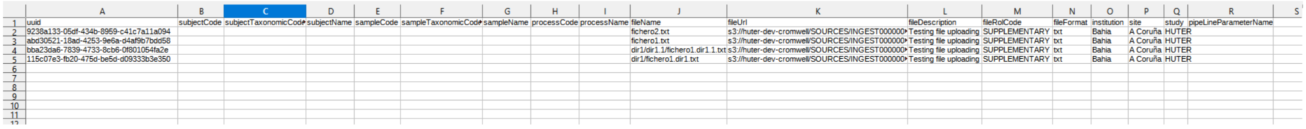
Campos

Tipo de columna:

	Predeterminado	Predeterminado	Predeterminado	Predeterminado	Predeterminado	Predeterminado	Predeterminado	Predeterminado	Predeterminado	Predeterminado	Predeter
1	uuid	subjectCode	subjectTaxonomicCode	subjectName	sampleCode	sampleTaxonomicCode	sampleName	processCode	processName	fileName	fileUrl
2	6edd749c-7578-438c-8cb7-e77d1716e3e1	SubjectCode			sampleCode					Mitochondria(Y)Golgi(R).czi	s3://hu

Figure 2 - TSV file in LibreCalc (Libre Office Suite)

An example of a manifest file that contains the location in the AWS S3 of four files registered in the platform is shown at *Figure 3 - Manifest file example*.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	uuid	subjectCode	subjectTaxonomicCod▸	subjectName	sampleCode	sampleTaxonomicCod▸	sampleName	processCode	processName	fileName	fileUrl	fileDescription	fileRolCode	fileFormat	institution	site	study	pipeLineParameterName	
2	9238a133-05df-434b-8959-c41c7a11a094									fichero2.txt	s3://huter-dev-cromwell/SOURCES/INGEST000000▸	Testing file uploading	SUPPLEMENTARY	txt	Bahia	A Coruña	HUTER		
3	abd30521-18ad-4253-9e6a-d4af9b7bdd58									fichero1.txt	s3://huter-dev-cromwell/SOURCES/INGEST000000▸	Testing file uploading	SUPPLEMENTARY	txt	Bahia	A Coruña	HUTER		
4	bba23da6-7839-4733-8cb6-0f801054fa2e									dir1/dir1.1/fichero1.dir1.1.txt	s3://huter-dev-cromwell/SOURCES/INGEST000000▸	Testing file uploading	SUPPLEMENTARY	txt	Bahia	A Coruña	HUTER		
5	115c07e3-fb20-475d-be5d-d09333b3e350									dir1/fichero1.dir1.txt	s3://huter-dev-cromwell/SOURCES/INGEST000000▸	Testing file uploading	SUPPLEMENTARY	txt	Bahia	A Coruña	HUTER		
6																			
7																			
8																			
9																			
10																			
11																			

Figure 3 - Manifest file example

Manifest file means a master file for many functionalities of the platform because it is a flexible way for communicating large list of files to the platform service through the tool suite.

In this regard, HUTER Platform follows a similar approach as other relevant research data repositories that are using manifest file. A good example is the well-known Genome Data Commons (GDC) which hosts The Cancer Genome Atlas (TCGA) data.

4.2.1.2. huter-cli help

The very first step for any user running the HUTER CLI should be the execution of the help command that can also be run by providing no command. The outcome of this command can be seen at *Figure 1 - huter-cli help*. The displayed information is the first aid for users who like to interact with HUTER Platform through the HUTER CLI.

4.2.2. Setting up the tool

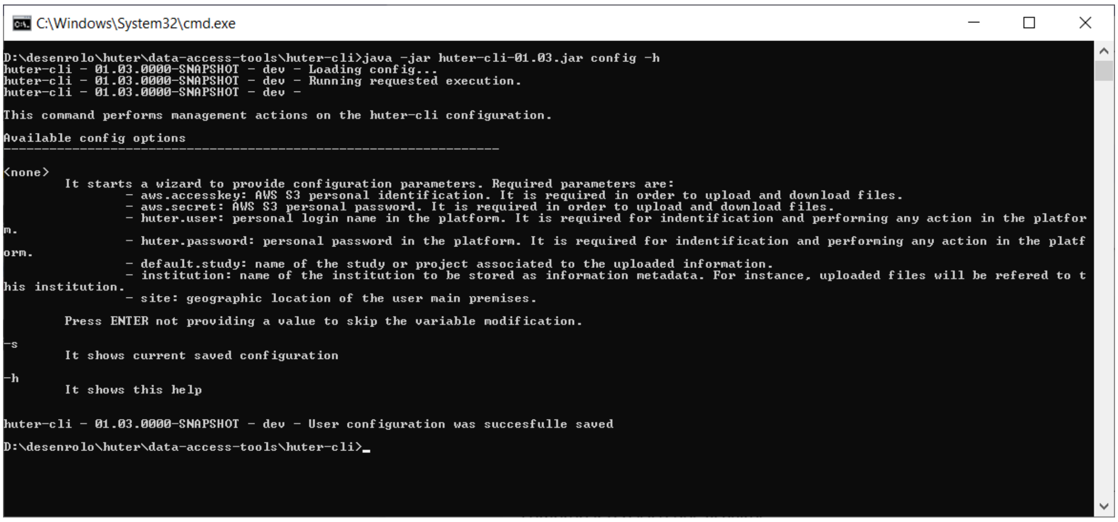
Regarding the aim of the platform, one of the requirements was the data tracking, linking and reliability. This premise involves security features for platform access and the registration of the data source such as institution and user. Therefore, the HUTER CLI was designed to simplify this repetitive and tedious task of providing user’s data when they require any service in the platform.

Deliverable D2.4 will deepen the use of registration data; in the meantime, it is enough to know that this data is required by the tool and how to provide it.

4.2.2.1. huter-cli config

The “config” command offers a set of functions for configuring and customizing the tools. The provided parameters are used to build a proper communication with the HUTER Platform services, through a secure channel while data tracking is assured.

By running the help parameter, the tool will display a more detailed explanation about how to use this command. *Figure 4 - huter-cli config -h* shows an example of the command execution.



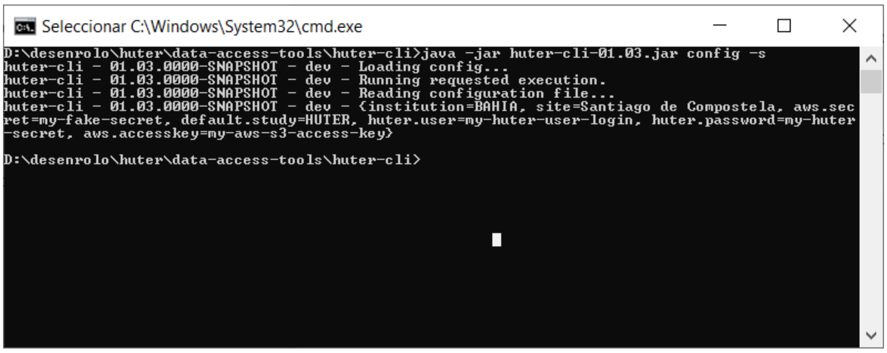
```
D:\desenrolo\huter\data-access-tools\huter-cli>java -jar huter-cli-01.03.jar config -h
huter-cli - 01.03.0000-SNAPSHOT - dev - Loading config...
huter-cli - 01.03.0000-SNAPSHOT - dev - Running requested execution.
huter-cli - 01.03.0000-SNAPSHOT - dev -

This command performs management actions on the huter-cli configuration.

Available config options
-----------------------------------------------------------------

<none>
        It starts a wizard to provide configuration parameters. Required parameters are:
                - aws.accesskey: AWS S3 personal identification. It is required in order to upload and download files.
                - aws.secret: AWS S3 personal password. It is required in order to upload and download files.
                - huter.user: personal login name in the platform. It is required for indentification and performing any action in the platfor
m.
                - huter.password: personal password in the platform. It is required for indentification and performing any action in the platf
orm.
                - default.study: name of the study or project associated to the uploaded information.
                - institution: name of the institution to be stored as information metadata. For instance, uploaded files will be refered to t
his institution.
                - site: geographic location of the user main premises.

        Press ENTER not providing a value to skip the variable modification.

-s
        It shows current saved configuration

-h
        It shows this help

huter-cli - 01.03.0000-SNAPSHOT - dev - User configuration was succesfulle saved

D:\desenrolo\huter\data-access-tools\huter-cli>_
```

Figure 4 - huter-cli config -h

For managing the configuration of the HUTER CLI, this command let users to provide values for the next options:

-h: it shows the help information.

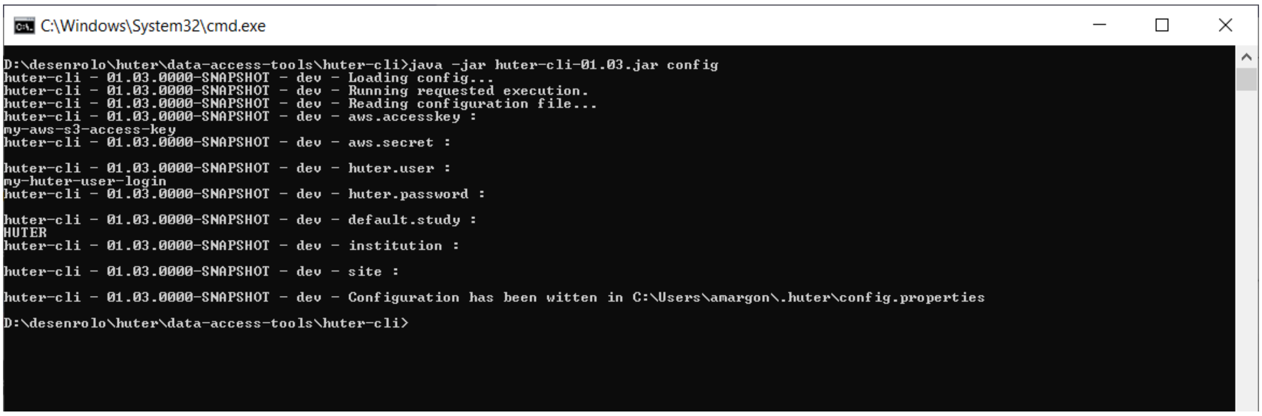
-s: it shows the current configuration for the HUTER CLI as it shows *Figure 5 - huter-cli config -s*.



```
D:\desenrolo\huter\data-access-tools\huter-cli>java -jar huter-cli-01.03.jar config -s
huter-cli - 01.03.0000-SNAPSHOT - dev - Loading config...
huter-cli - 01.03.0000-SNAPSHOT - dev - Running requested execution.
huter-cli - 01.03.0000-SNAPSHOT - dev - Reading configuration file...
huter-cli - 01.03.0000-SNAPSHOT - dev - {institution=BAHIA, site=Santiago de Compostela, aws.sec
ret=my-fake-secret, default.study=HUTER, huter.user=my-huter-user-login, huter.password=my-huter
-secret, aws.accesskey=my-aws-s3-access-key}

D:\desenrolo\huter\data-access-tools\huter-cli>
```

Figure 5 - huter-cli config -s

<none>: HUTER CLI will guide users through the configuration process. Users will be asked for every configuration variable and tracking information required by the HUTER Platform. This process is the same for a first configuration or an update, so if no value is provided for any variable its modification will be skipped. For instance, at *Figure 6 - huter-cli config* secrets were already provided so their modification was skipped by not providing a new value.


C:\Windows\System32\cmd.exe

```
D:\desenrolo\huter\data-access-tools\huter-cli>java -jar huter-cli-01.03.jar config
huter-cli - 01.03.0000-SNAPSHOT - dev - Loading config...
huter-cli - 01.03.0000-SNAPSHOT - dev - Running requested execution.
huter-cli - 01.03.0000-SNAPSHOT - dev - Reading configuration file...
huter-cli - 01.03.0000-SNAPSHOT - dev - aws.accesskey :
my-aws-s3-access-key
huter-cli - 01.03.0000-SNAPSHOT - dev - aws.secret :

huter-cli - 01.03.0000-SNAPSHOT - dev - huter.user :
my-huter-user-login
huter-cli - 01.03.0000-SNAPSHOT - dev - huter.password :

huter-cli - 01.03.0000-SNAPSHOT - dev - default.study :
HUTER
huter-cli - 01.03.0000-SNAPSHOT - dev - institution :

huter-cli - 01.03.0000-SNAPSHOT - dev - site :

huter-cli - 01.03.0000-SNAPSHOT - dev - Configuration has been witten in C:\Users\amargon\.huter\config.properties

D:\desenrolo\huter\data-access-tools\huter-cli>
```

Figure 6 - huter-cli config

4.2.3. File uploading

File storage is one of the main functionalities of the HUTER Platform. However, this not only means the deployment of a cloud storage service but also a proper file tracking and location.

Regarding section *4.2.2 Setting up the tool*, HUTER CLI also provides tracking feature for stored data. For taking advantage of those already configured values, HUTER CLI provides a functionality for file uploading and its metadata indexation, including tracking key values. Later, this file metadata could be exploited by *Data browser* for file location as well as other HUTER Platform tools, functionalities, and services.

Coming back to the file uploading process, it will be executed in two separated steps, so users have to execute two actions through the HUTER CLI to get uploaded the desired files. Although an upload could be done just in one step, this previous "arrange" action will allow users to get an early file registration. Furthermore, dividing upload in two steps, the platform can automatically provision resources before the file uploading.

4.2.3.1. huter-cli prepare

First file upload step is the execution of the "prepare" command. This command lets users to arrange the upload of files which are in a specific local folder. An upload to the HUTER Platform not only means that files will be stored in a cloud space, but their metadata will be also indexed in the central HUTER database for future exploitation on section *4.3 Data browser* or other tools. Therefore, this functionality will eventually supply identifiers and the storage location to the "prepared" files for their further upload.

Regarding how user can execute this functionality, like any other command, option "prepare -h" will show all the available options that user can run as can be seen at *Figure 7 - huter-cli prepare -h*.

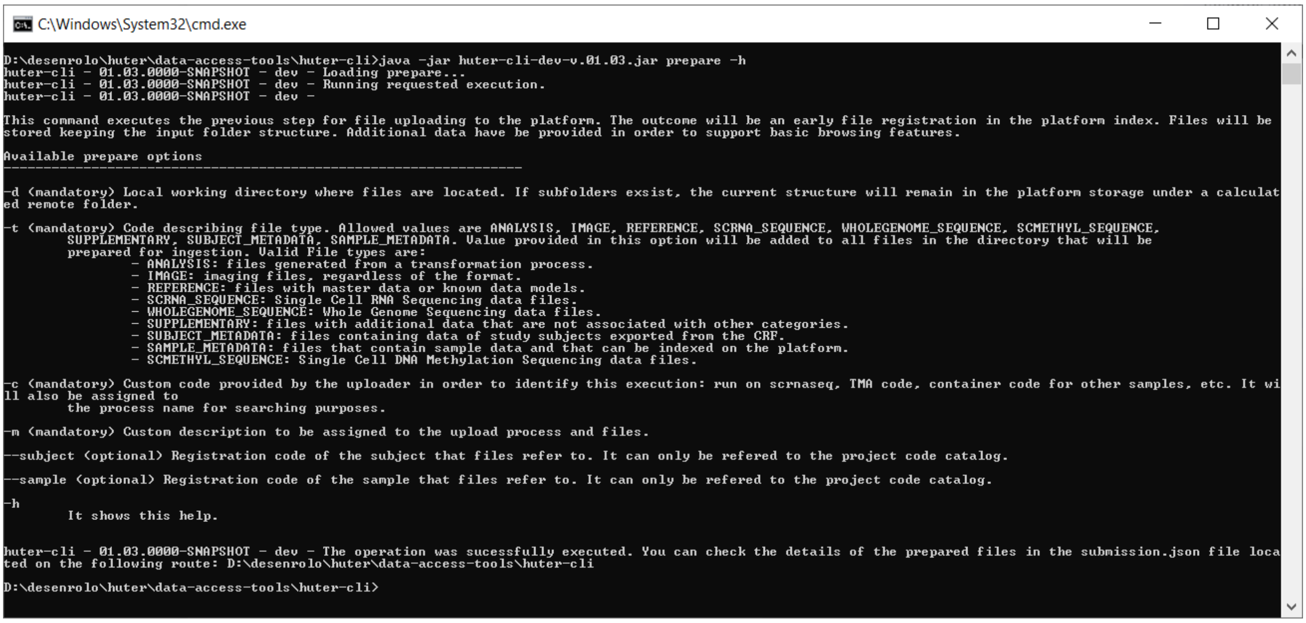
For requesting the previous registration of files in the platform, prepare command provide next options:

-h: it shows the help information to guide users.

-d: it is a mandatory option where user must provide the folder path where files to be upload are located. It will also be the path where the submission summary file will be stored after the file registration.

-t: it is also a mandatory argument where user must categorize the content of the file to be uploaded. This means that all files in the target folder have to contain the same kind of data. Values that can be provided are fully compliant with the Human Cell Atlas DCP schemas even though some new categories have been added after conversation with the HUTER project bioinformaticians. The list of available type of data is:

- ANALYSIS: type of content related to an outcome of a pipeline process. It is allowed to upload ANALYSIS files to assure data coherence if a manual upload of a file is needed.
- IMAGE: category for files containing visual data. Some HUTER partners were expected to work with SmFish, Fluorescence, Tissue Micro Array and immunohistochemistry image, so HUTER Platform offers the option to categorize them. These files will also be the input for the DICOM transformation software that will be explained later in this document.
- REFERENCE: Human Cell Atlas DCP defined this kind of data for pointing out external information used in the project for data verification.
- SCRNA_SEQUENCE: this category allows to mark files with Single Cell RNA sequencing data.
- WHOLEGENOME_SEQUENCE: this category allows to mark files with Whole genome sequencing data.
- SCMETHYL_SEQUENCE: this category allows to mark files with Single Cell DNA Methylation sequencing data.
- SUBJECT_METADATA: HUTER project uses a Case Report Form (CRF) application called LibreClinica that will be exposed in deliverable D2.4, for gathering subject information. The ingestion of that kind of data can be done by uploading exported files with the subject information so, for finding it, this category can be applied to the files.
- SAMPLE_METADATA: Sample information is expected to be upload like subject information, by using files with tabular data. So, a new category was created in order to their distinction.
- SUPPLEMENTARY: Human Cell Atlas DCP defined this category for uploading any file which type does not fit anyone else.



C:\Windows\System32\cmd.exe

```
D:\desenrolo\huter\data-access-tools\huter-cli>java -jar huter-cli-dev-v.01.03.jar prepare -h
huter-cli - 01.03.0000-SNAPSHOT - dev - Loading prepare...
huter-cli - 01.03.0000-SNAPSHOT - dev - Running requested execution.
huter-cli - 01.03.0000-SNAPSHOT - dev -

This command executes the previous step for file uploading to the platform. The outcome will be an early file registration in the platform index. Files will be
stored keeping the input folder structure. Additional data have be provided in order to support basic browsing features.

Available prepare options
-----------------------------------------------------------------

-d (mandatory) Local working directory where files are located. If subfolders exsist, the current structure will remain in the platform storage under a calculat
ed remote folder.

-t (mandatory) Code describing file type. Allowed values are ANALYSIS, IMAGE, REFERENCE, SCRNA_SEQUENCE, WHOLEGENOME_SEQUENCE, SCMETHYL_SEQUENCE,
        SUPPLEMENTARY, SUBJECT_METADATA, SAMPLE_METADATA. Value provided in this option will be added to all files in the directory that will be
        prepared for ingestion. Valid File types are:
                - ANALYSIS: files generated from a transformation process.
                - IMAGE: imaging files, regardless of the format.
                - REFERENCE: files with master data or known data models.
                - SCRNA_SEQUENCE: Single Cell RNA Sequencing data files.
                - WHOLEGENOME_SEQUENCE: Whole Genome Sequencing data files.
                - SUPPLEMENTARY: files with additional data that are not associated with other categories.
                - SUBJECT_METADATA: files containing data of study subjects exported from the CRF.
                - SAMPLE_METADATA: files that contain sample data and that can be indexed on the platform.
                - SCMETHYL_SEQUENCE: Single Cell DNA Methylation Sequencing data files.

-c (mandatory) Custom code provided by the uploader in order to identify this execution: run on scrnaseq, TMA code, container code for other samples, etc. It wi
ll also be assigned to
        the process name for searching purposes.

-m (mandatory) Custom description to be assigned to the upload process and files.

--subject (optional) Registration code of the subject that files refer to. It can only be refered to the project code catalog.

--sample (optional) Registration code of the sample that files refer to. It can only be refered to the project code catalog.

-h
        It shows this help.


huter-cli - 01.03.0000-SNAPSHOT - dev - The operation was sucessfully executed. You can check the details of the prepared files in the submission.json file loca
ted on the following route: D:\desenrolo\huter\data-access-tools\huter-cli

D:\desenrolo\huter\data-access-tools\huter-cli>
```

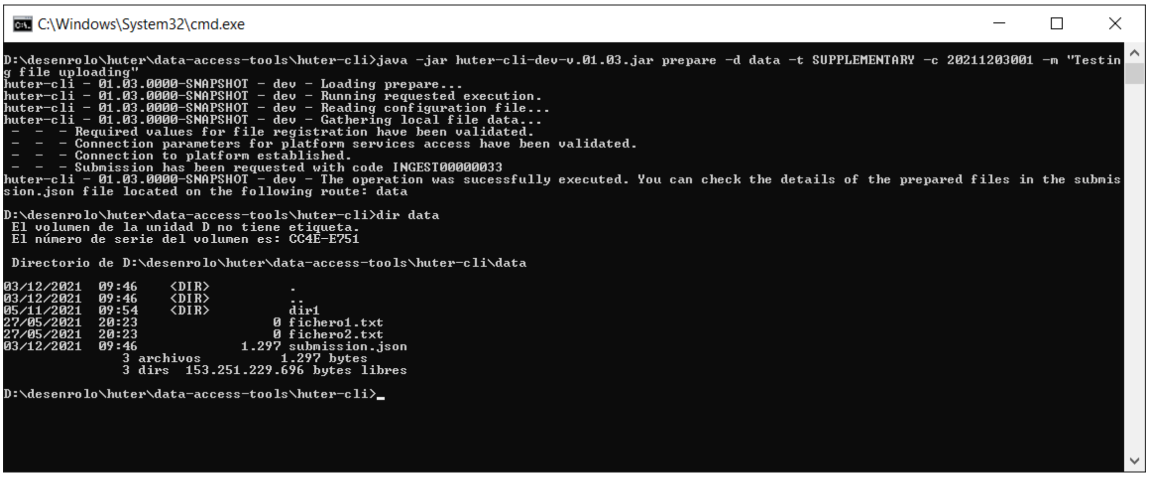
Figure 7 - huter-cli prepare -h

-c: it is a mandatory code that must be provided to track laboratory information within the platform. This value will be assigned to the upload process as and additional code.

-m: it is a mandatory description of the upload process so any human can understand which data has been stored. This description will be assigned to the process as well as the uploaded files.

--subject: it allows to assign a HUTER subject code to the elements of the upload (process and files registration) to keep track of the data.

--sample: like the subject code, this option allows to assign a HUTER sample code to the elements of the upload to keep track of the data.



C:\Windows\System32\cmd.exe

```
D:\desenrolo\huter\data-access-tools\huter-cli>java -jar huter-cli-dev-v.01.03.jar prepare -d data -t SUPPLEMENTARY -c 20211203001 -m "Testin
g file uploading"
huter-cli - 01.03.0000-SNAPSHOT - dev - Loading prepare...
huter-cli - 01.03.0000-SNAPSHOT - dev - Running requested execution.
huter-cli - 01.03.0000-SNAPSHOT - dev - Reading configuration file...
huter-cli - 01.03.0000-SNAPSHOT - dev - Gathering local file data...
 -  -  - Required values for file registration have been validated.
 -  -  - Connection parameters for platform services access have been validated.
 -  -  - Connection to platform established.
 -  -  - Submission has been requested with code INGEST00000033
huter-cli - 01.03.0000-SNAPSHOT - dev - The operation was sucessfully executed. You can check the details of the prepared files in the submis
sion.json file located on the following route: data

D:\desenrolo\huter\data-access-tools\huter-cli>dir data
 El volumen de la unidad D no tiene etiqueta.
 El número de serie del volumen es: CC4E-E751

 Directorio de D:\desenrolo\huter\data-access-tools\huter-cli\data

03/12/2021  09:46    <DIR>          .
03/12/2021  09:46    <DIR>          ..
05/11/2021  09:54    <DIR>          dir1
27/05/2021  20:23                 0 fichero1.txt
27/05/2021  20:23                 0 fichero2.txt
03/12/2021  09:46             1.297 submission.json
               3 archivos          1.297 bytes
               3 dirs  153.251.229.696 bytes libres

D:\desenrolo\huter\data-access-tools\huter-cli>_
```

Figure 8 - huter-cli prepare execution

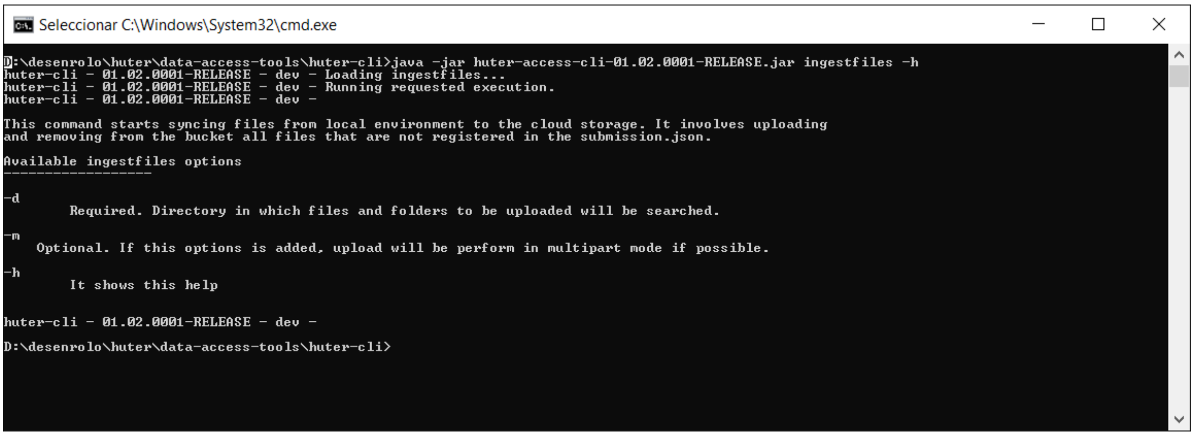
Figure 8 - huter-cli prepare execution shows the execution of the arrangement for a folder called “data” that contains test files and subfolders. During the arrangement process users are informed about the main executed steps. Finally, users are given the process identification code, in this example INGEST00000033, that can be used to find the detailed information of the upload in the data-browser (see section *4.3 Data browser* for further details). A later execution of the “dir” command displays the content of the folder including the submission.json file containing the uploading references.

4.2.3.2. huter-cli ingestfiles

The second step for file uploading process is the ingestion. This functionality relies on the previous generated file for the registration that was explained in the *4.2.3.1 huter-cli prepare* section.

The ingestion step allows copying “prepared” files from the local storage to the cloud storage of HUTER Platform through the HUTER CLI “ingestfiles” command.

File storage is one of the main functionalities of the HUTER Platform. However, this not only means the deployment of a cloud storage service but also a proper file tracking and location.



```
D:\desenrolo\huter\data-access-tools\huter-cli>java -jar huter-access-cli-01.02.0001-RELEASE.jar ingestfiles -h
huter-cli - 01.02.0001-RELEASE - dev - Loading ingestfiles...
huter-cli - 01.02.0001-RELEASE - dev - Running requested execution.
huter-cli - 01.02.0001-RELEASE - dev -

This command starts syncing files from local environment to the cloud storage. It involves uploading
and removing from the bucket all files that are not registered in the submission.json.

Available ingestfiles options
------------------

-d
        Required. Directory in which files and folders to be uploaded will be searched.

-m
    Optional. If this options is added, upload will be perform in multipart mode if possible.

-h
        It shows this help

huter-cli - 01.02.0001-RELEASE - dev -

D:\desenrolo\huter\data-access-tools\huter-cli>
```

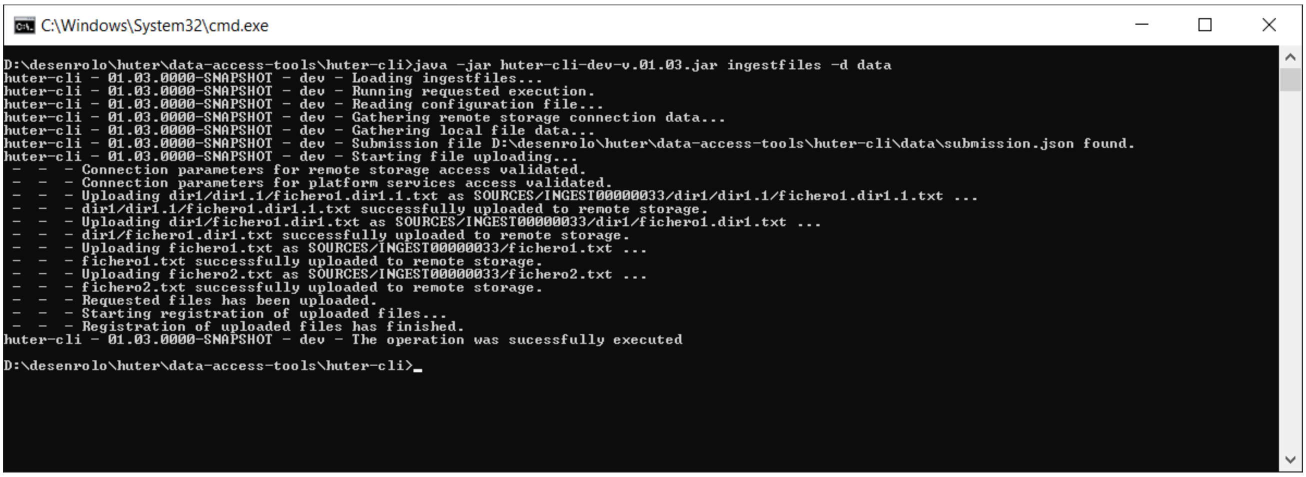
Figure 9 - huter-cli ingest-files -h

At *Figure 9 - huter-cli ingest-files -h* options for “ingestfiles” command are displayed.

-h: it shows the help information to guide users.

-d: it is the mandatory argument where user has to provide the path of previously arranged folder for ingestion. In that folder there must be a submission.json file as the outcome of a previous “prepare” command execution because it contains the given identifiers registered in the HUTER Platform for the upload process and all the files in the folder.

-m: this argument is an optional feature that will allow users to execute an uploading mode called “multipart”. This is a specific feature for AWS S3 storage that allows a file to be uploaded sending several fragments instead of a unique file block. Forthcoming Deliverable D2.4 will go in depth with this feature from a technical point of view.



```
D:\desenrolo\huter\data-access-tools\huter-cli>java -jar huter-cli-dev-v.01.03.jar ingestfiles -d data
huter-cli - 01.03.0000-SNAPSHOT - dev - Loading ingestfiles...
huter-cli - 01.03.0000-SNAPSHOT - dev - Running requested execution.
huter-cli - 01.03.0000-SNAPSHOT - dev - Reading configuration file...
huter-cli - 01.03.0000-SNAPSHOT - dev - Gathering remote storage connection data...
huter-cli - 01.03.0000-SNAPSHOT - dev - Gathering local file data...
huter-cli - 01.03.0000-SNAPSHOT - dev - Submission file D:\desenrolo\huter\data-access-tools\huter-cli\data\submission.json found.
huter-cli - 01.03.0000-SNAPSHOT - dev - Starting file uploading...
 -  -  - Connection parameters for remote storage access validated.
 -  -  - Connection parameters for platform services access validated.
 -  -  - Uploading dir1/dir1.1/fichero1.dir1.1.txt as SOURCES/INGEST00000033/dir1/dir1.1/fichero1.dir1.1.txt ...
 -  -  - dir1/dir1.1/fichero1.dir1.1.txt successfully uploaded to remote storage.
 -  -  - Uploading dir1/fichero1.dir1.txt as SOURCES/INGEST00000033/dir1/fichero1.dir1.txt ...
 -  -  - dir1/fichero1.dir1.txt successfully uploaded to remote storage.
 -  -  - Uploading fichero1.txt as SOURCES/INGEST00000033/fichero1.txt ...
 -  -  - fichero1.txt successfully uploaded to remote storage.
 -  -  - Uploading fichero2.txt as SOURCES/INGEST00000033/fichero2.txt ...
 -  -  - fichero2.txt successfully uploaded to remote storage.
 -  -  - Requested files has been uploaded.
 -  -  - Starting registration of uploaded files...
 -  -  - Registration of uploaded files has finished.
huter-cli - 01.03.0000-SNAPSHOT - dev - The operation was sucessfully executed

D:\desenrolo\huter\data-access-tools\huter-cli>_
```

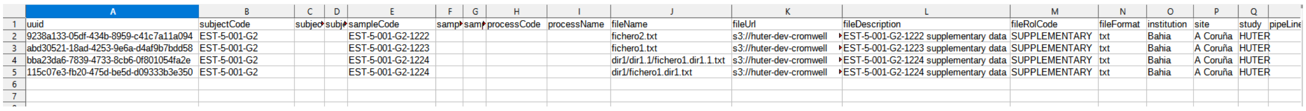
Figure 10 - huter-cli ingestfiles execution

During the upload process, user will be informed about the progress of the upload. Finally, as result of this execution, files in the “prepared” folder will be stored in the cloud storage and its previous registration in the HUTER Platform will be marked as reliable. At *Figure 10 - huter-cli ingestfiles execution* the result of an upload is displayed for example.

4.2.3.3. huter-cli updatefiles

Because of the HUTER CLI agility for file uploading, this process avoided the requirement of some tracking data like the link between files and the sample and subject which belongs to. For healing this important gap, HUTER CLI will let users to provide this information in a further and independent process. As it was introduced in section *4.2.1.1 Previous considerations: Manifest file*, manifest file is a master file that helps users during the communication between HUTER CLI and HUTER Platform services.

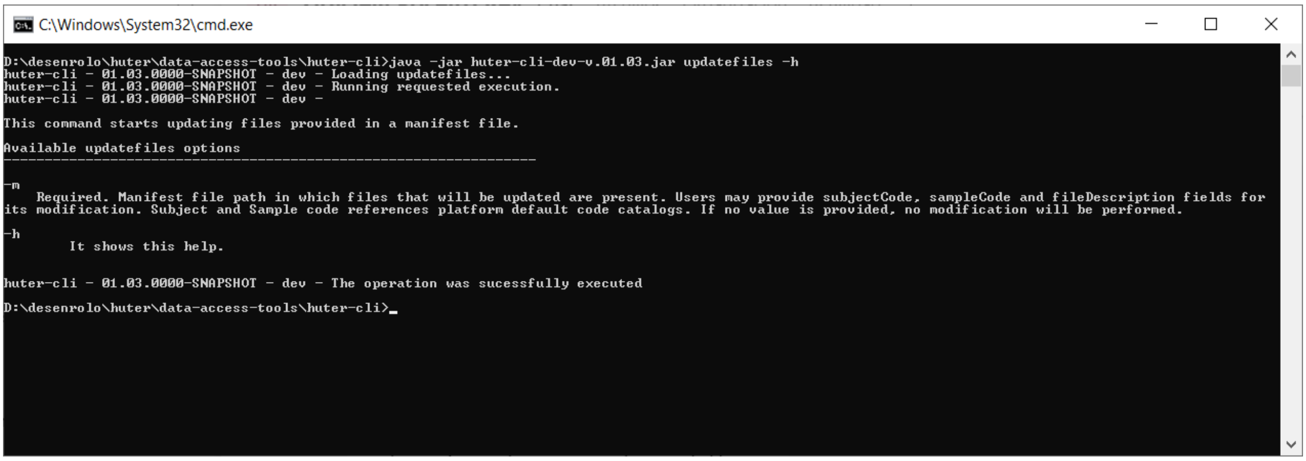
Bearing this in mind, HUTER CLI offers the “updatefiles” command that lets users to provide a modified manifest file for file metadata updating. However, just some attributes of the file metadata will be modified such as subject, sample code or a different file description that the one provided during the *huter-cli prepare* process. *Figure 11 – Updated manifest file with new file metadata* shows a manifest file where subject code, sample codes and files description were given to the contained files.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	
1	uuid	subjectCode	subjec	subj	sampleCode	samp	sam	processCode	processName	fileName	fileUrl	fileDescription	fileRolCode	fileFormat	institution	site	study	pipeLine
2	9238a133-05df-434b-8959-c41c7a11a094	EST-5-001-G2			EST-5-001-G2-1222					fichero2.txt	s3://huter-dev-cromwell	EST-5-001-G2-1222 supplementary data	SUPPLEMENTARY	txt	Bahia	A Coruña	HUTER	
3	abd30521-18ad-4253-9e6a-d4af9b7bdd58	EST-5-001-G2			EST-5-001-G2-1223					fichero1.txt	s3://huter-dev-cromwell	EST-5-001-G2-1223 supplementary data	SUPPLEMENTARY	txt	Bahia	A Coruña	HUTER	
4	bba23da6-7839-4733-8cb6-0f801054fa2e	EST-5-001-G2			EST-5-001-G2-1224					dir1/dir1.1/fichero1.dir1.1.txt	s3://huter-dev-cromwell	EST-5-001-G2-1224 supplementary data	SUPPLEMENTARY	txt	Bahia	A Coruña	HUTER	
5	115c07e3-fb20-475d-be5d-d09333b3e350	EST-5-001-G2			EST-5-001-G2-1224					dir1/fichero1.dir1.txt	s3://huter-dev-cromwell	EST-5-001-G2-1224 supplementary data	SUPPLEMENTARY	txt	Bahia	A Coruña	HUTER	
6																		
7																		

Figure 11 – Updated manifest file with new file metadata

HUTER CLI updatefiles command arguments are shown in *Figure 12 - huter-cli ingestfiles -h*:



C:\Windows\System32\cmd.exe

```
D:\desenrolo\huter\data-access-tools\huter-cli>java -jar huter-cli-dev-v.01.03.jar updatefiles -h
huter-cli - 01.03.0000-SNAPSHOT - dev - Loading updatefiles...
huter-cli - 01.03.0000-SNAPSHOT - dev - Running requested execution.
huter-cli - 01.03.0000-SNAPSHOT - dev -

This command starts updating files provided in a manifest file.

Available updatefiles options
-----------------------------------------------------------------

-m
    Required. Manifest file path in which files that will be updated are present. Users may provide subjectCode, sampleCode and fileDescription fields for
its modification. Subject and Sample code references platform default code catalogs. If no value is provided, no modification will be performed.

-h
        It shows this help.

huter-cli - 01.03.0000-SNAPSHOT - dev - The operation was sucessfully executed

D:\desenrolo\huter\data-access-tools\huter-cli>_
```

Figure 12 - huter-cli ingestfiles -h

-h: it shows the help information to guide users.

-m: it is a mandatory value with the path to the manifest file that users can generate using Data browser and then, update it. It must contain an updated definition for every files. If blank value is provided, no modification will be performed over the indexed file metadata.

4.2.4. File downloading

HUTER Plaftorm is expected to be a cloud workspace for boosting collaboration among research groups under the HUTER project and far away. Although this platform provides a complete suite of tools for data processing, visualization and storage, the ability to share information among partners is required. This idea includes the possibility of download data from HUTER Platform to partner's devices.

Regarding this requirement, HUTER CLI will be the tool aimed to perform this task.

4.2.5. huter-cli download

HUTER CLI offers a functionality for downloading files from the cloud storage to machines on partner's premises. Logically, just uploaded data can be copied from HUTER Platform to local storage so users have to provide to the file location on the cloud. Data browser (see section 4.3) is a complementary tool of the HUTER Platform that lets user to browse store information by exploiting indexed data. By using it to generate a

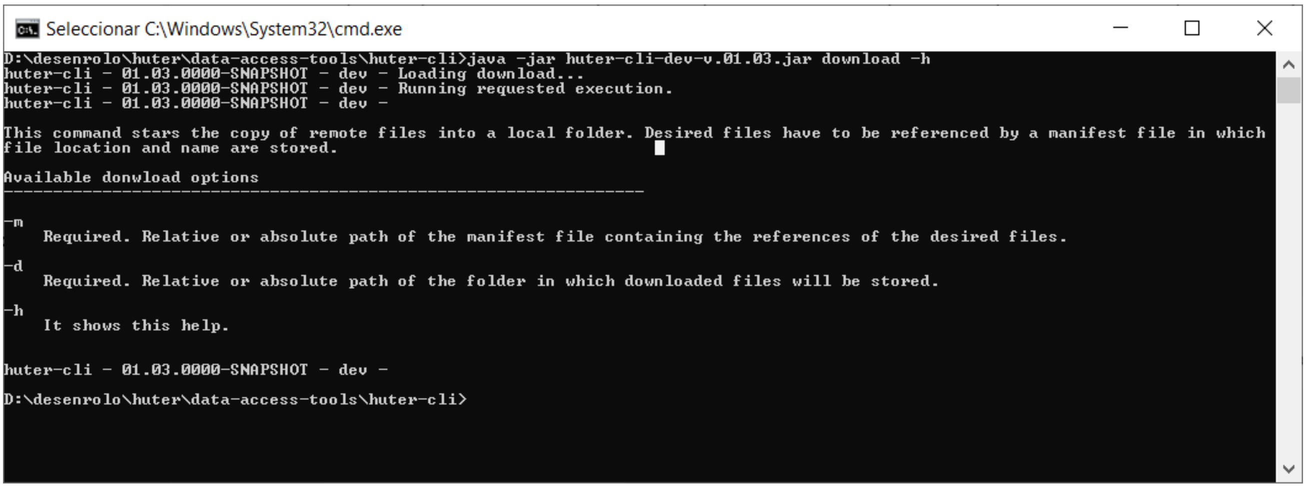
Previous considerations: Manifest file, users can get a list of files for downloading including their location and additional data.

HUTER CLI exposes the “download” command in order to take of manifest file, letting users to provide a manifest file indicating where the desired files are located. As any other command, the “-h” argument will detail the use of the command as shown in *Figure 13 - huter-cli download -h*.

-h: it shows the help information to guide users.

-m: it is a mandatory value with the path to the manifest file that users can generate using Data browser. It must contain the reference of the store files that will be downloaded.

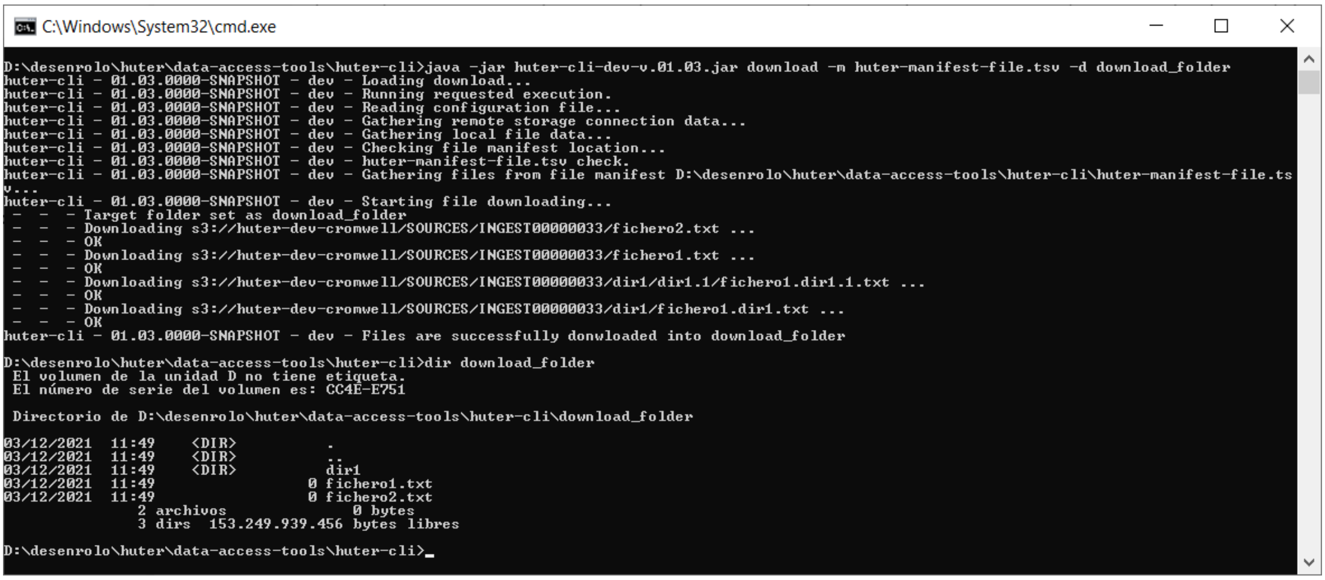
-d: it is the mandatory name of the folder where downloaded files will be stored. For keeping data organization, any element under the target folder will be stored using the same folder structure existing in the platform.



```
D:\desenrolo\huter\data-access-tools\huter-cli>java -jar huter-cli-dev-v.01.03.jar download -h
huter-cli - 01.03.0000-SNAPSHOT - dev - Loading download...
huter-cli - 01.03.0000-SNAPSHOT - dev - Running requested execution.
huter-cli - 01.03.0000-SNAPSHOT - dev -

This command stars the copy of remote files into a local folder. Desired files have to be referenced by a manifest file in which
file location and name are stored.

Available donwload options
----------------------------------------------------------------

-m
    Required. Relative or absolute path of the manifest file containing the references of the desired files.
-d
    Required. Relative or absolute path of the folder in which downloaded files will be stored.
-h
    It shows this help.

huter-cli - 01.03.0000-SNAPSHOT - dev -

D:\desenrolo\huter\data-access-tools\huter-cli>
```

Figure 13 - huter-cli download -h

The procedure to download a set of files will be simple. Once users have generated their custom manifest file, they just need to provide it to the HUTER CLI as well as the local target folder where files will be stored. For instance, using the manifest file at *Figure 3 - Manifest file example*, users will be able to execute the file downloading from the platform meanwhile the HUTER CLI will display the status of the download in real time as shown at *Figure 14 - huter-cli download execution*.



C:\Windows\System32\cmd.exe

```
D:\desenrolo\huter\data-access-tools\huter-cli>java -jar huter-cli-dev-v.01.03.jar download -m huter-manifest-file.tsv -d download_folder
huter-cli - 01.03.0000-SNAPSHOT - dev - Loading download...
huter-cli - 01.03.0000-SNAPSHOT - dev - Running requested execution.
huter-cli - 01.03.0000-SNAPSHOT - dev - Reading configuration file...
huter-cli - 01.03.0000-SNAPSHOT - dev - Gathering remote storage connection data...
huter-cli - 01.03.0000-SNAPSHOT - dev - Gathering local file data...
huter-cli - 01.03.0000-SNAPSHOT - dev - Checking file manifest location...
huter-cli - 01.03.0000-SNAPSHOT - dev - huter-manifest-file.tsv check.
huter-cli - 01.03.0000-SNAPSHOT - dev - Gathering files from file manifest D:\desenrolo\huter\data-access-tools\huter-cli\huter-manifest-file.ts
v...
huter-cli - 01.03.0000-SNAPSHOT - dev - Starting file downloading...
 -  -  - Target folder set as download_folder
 -  -  - Downloading s3://huter-dev-cromwell/SOURCES/INGEST00000033/fichero2.txt ...
 -  -  - OK
 -  -  - Downloading s3://huter-dev-cromwell/SOURCES/INGEST00000033/fichero1.txt ...
 -  -  - OK
 -  -  - Downloading s3://huter-dev-cromwell/SOURCES/INGEST00000033/dir1/dir1.1/fichero1.dir1.1.txt ...
 -  -  - OK
 -  -  - Downloading s3://huter-dev-cromwell/SOURCES/INGEST00000033/dir1/fichero1.dir1.txt ...
 -  -  - OK
huter-cli - 01.03.0000-SNAPSHOT - dev - Files are successfully donwloaded into download_folder

D:\desenrolo\huter\data-access-tools\huter-cli>dir download_folder
 El volumen de la unidad D no tiene etiqueta.
 El número de serie del volumen es: CC4E-E751

 Directorio de D:\desenrolo\huter\data-access-tools\huter-cli\download_folder

03/12/2021  11:49    <DIR>          .
03/12/2021  11:49    <DIR>          ..
03/12/2021  11:49    <DIR>          dir1
03/12/2021  11:49                 0 fichero1.txt
03/12/2021  11:49                 0 fichero2.txt
               2 archivos              0 bytes
               3 dirs  153.249.939.456 bytes libres

D:\desenrolo\huter\data-access-tools\huter-cli>_
```

Figure 14 - huter-cli download execution

4.2.6. Metadata indexation

HUTER Platform was built to manage data in research environments. In this context, metadata is a concept that refers to additional features of processed data. For instance, the metadata of genomic information is the features of the source sample and the information about the patient.

Regarding to HUTER Platform, it manages next entities: files, processes but also subjects and their samples. Therefore, file metadata means its cloud storage location, name, type, format, size and so on. Regarding samples, their metadata may be the organ zone of the sample, the preservation method or the number of cells. Subject metadata logically refers to human being traits such as age, diseases or any feature considered important in the assay context.

Depending on the entity, its metadata can be gathered in a different way. File metadata is generated and provided by users during the File uploading process. Meanwhile sample metadata is provided by bioinformaticians in TSV file after an agreement. CRF or Case Report Forms are managed by LibreClinica for subject metadata registration. That means metadata is extremely unstable regards to their structure and format. If HUTER Platform provides a static data model it would be an obstacle for bioinformaticians research assays because metadata could not be properly indexed and exploited. In that way, HUTER Platform was designed to manage flexible data models.

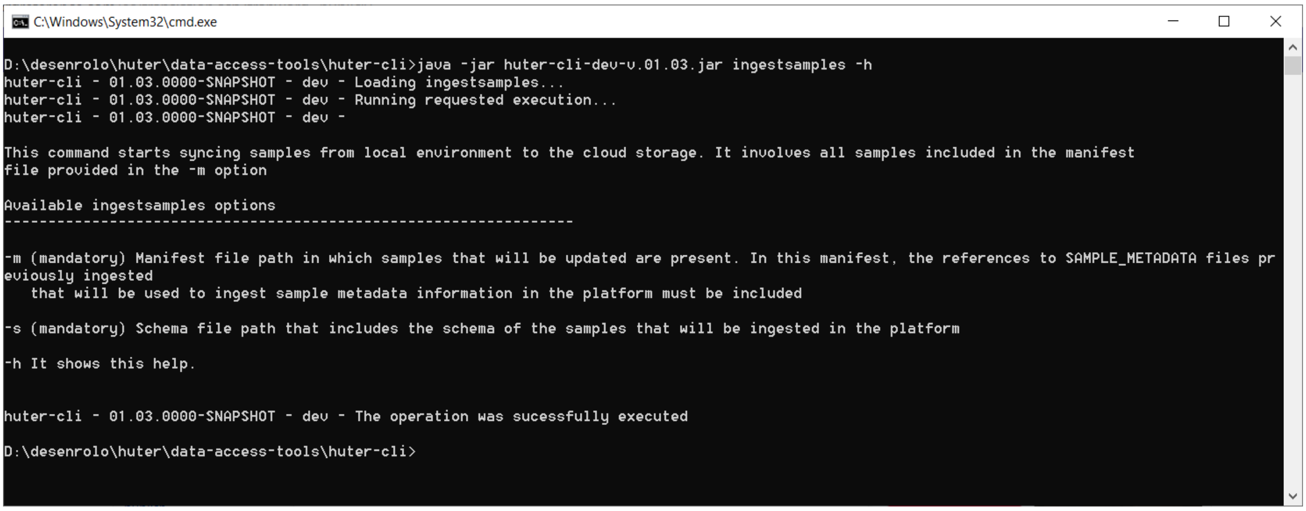
The metadata indexation can be performed by transferring data from files to the HUTER central database. In order to achieve it, a set of set processes were developed to manage different entity metadata. These processes are executed in services that can be invoked by the HUTER CLI.

4.2.6.1. huter-cli ingestsamples

One of the most important data affected for this need was the gathered and calculated sample information. For a single sample, depending on the processing method, many different variables could be required to be indexed. So, to comply with this need, HUTER Platform offers the capacity to ingest sample information in two complementary ways.

- The first one is the file upload of files categorized as SAMPLE_METADATA containing the bioinformaticians desired information for their samples in the study. Any file can be uploaded using this category and used in pipeline executions for its data processing.
- The other way to ingest and deeply exploit of sample metadata is the HUTER Platform ability to read and understand the SAMPLE_METADATA files. However, during this project, just SAMPLE_METADATA files in TSV format will be processed in order to index their metadata.

Therefore, HUTER CLI will be the tool devoted to offer both sample metadata ingestion functionalities. The first one will be performed as was exposed in section *4.2.3.1 huter-cli prepare* section, using the file type SAMPLE_METADATA. The second sample metadata ingestion is based on the first one. The command ingestsamples offers the functionality for indexing the sample metadata contained in stored files but only those within SAMPLE_METADATA category and TSV (Tabular Separate Values) format. Manifest file will also be the master file for the identification of which files should be translated into metadata for the HUTER central database.



```
D:\desenrolo\huter\data-access-tools\huter-cli>java -jar huter-cli-dev-v.01.03.jar ingestsamples -h
huter-cli - 01.03.0000-SNAPSHOT - dev - Loading ingestsamples...
huter-cli - 01.03.0000-SNAPSHOT - dev - Running requested execution...
huter-cli - 01.03.0000-SNAPSHOT - dev -

This command starts syncing samples from local environment to the cloud storage. It involves all samples included in the manifest
file provided in the -m option

Available ingestsamples options
-----------------------------------------------------------------

-m (mandatory) Manifest file path in which samples that will be updated are present. In this manifest, the references to SAMPLE_METADATA files pr
eviously ingested
   that will be used to ingest sample metadata information in the platform must be included

-s (mandatory) Schema file path that includes the schema of the samples that will be ingested in the platform

-h It shows this help.


huter-cli - 01.03.0000-SNAPSHOT - dev - The operation was sucessfully executed

D:\desenrolo\huter\data-access-tools\huter-cli>
```

Figure 15 - huter-cli ingestsamples -h

Regarding to the user interaction, as well as other commands, "ingestsamples" command provides a set of arguments for delivering its functionality:

-h: it shows the help information to guide users.

-m: it is path to the generated manifest file containing a set of sample metadata files. This value is mandatory because is the way for locating files in the platform.

-s: it is the path to the ingestion schema that addresses ingestion process through the sample metadata translation and mapping.

For example, to run a sample metadata indexation the full command to be executed is:

```
java -jar huter-cli.jar -m sampledata-files.tsv -s ingest-schema.tsv
```

The complete functionality will be explained in D2.4 deliverable, meanwhile for this document understanding, it is just needed to know the aim functionality and its use by the HUTER Platform.

4.2.6.2. huter-cli ingestsubjects

HUTER Platform was also designed to index subject information such as age, diseases and so on, that is required for an accurate omics data analysis. Even though this functionality was included under the scope of the HUTER Platform and is potentially compatible, the overwhelming effort needed for a useful subject data indexation made us to focus on sample metadata ingestion which also collects the main subject variables for HUTER researchers' analyses. This does not mean that subject data cannot be ingested but it could be addressed if future requirements.

4.2.7. Workflow execution for analysis and processing

The HUTER Platform provides a powerful capability for data processing. This feature allows users to run analysis workflows or even transformation algorithms over stored data in order to generate new files with new data. Of course, users can develop their own pipelines for data processing whatever the software they could need.

For delivering this feature, HUTER Platform relies on Cromwell application. This is a Workflow Management System geared towards scientific workflows that can be defined using WDL scripting format. It was created by Broad Institute as an open source software (see https://cromwell.readthedocs.io/ or https://github.com/broadinstitute/cromwell for detailed information).

Cromwell can be deployed on several platforms to take advantage of their processing engine. For instance, HUTER Platform uses AWS infrastructure to Cromwell and its workflow executions. Because the Cromwell integration inside the AWS infrastructure is a probed solution it can provided a high throughput for pipeline executions. Deliverable D7.1 argued about the reasons for choosing this tool to run data processing workflows.

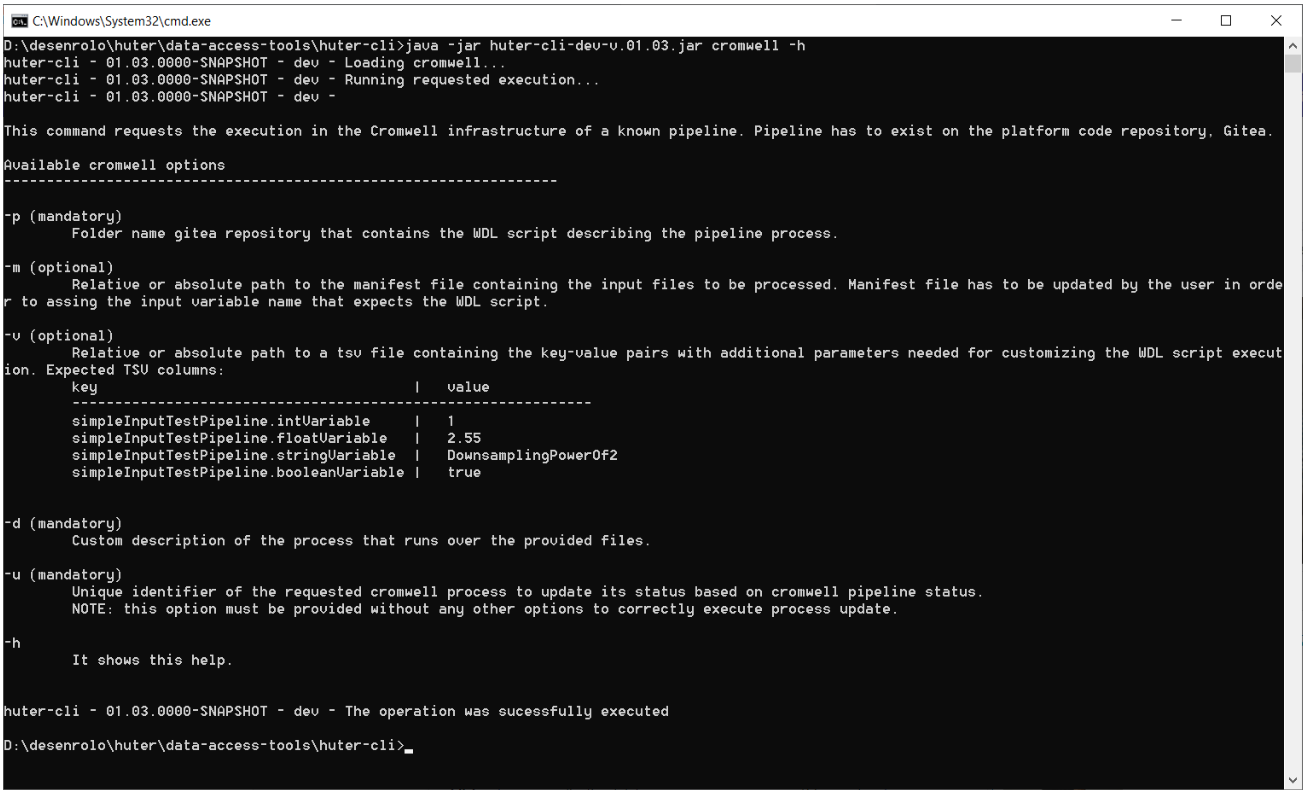
Regarding the aim of this document, next we will introduce how users can run their workflows on the HUTER Platform. However, previous considerations must be taken into account to easily understand the delivered interface.

4.2.8. Previous considerations

1. Workflow development. Deliverable D2.2 introduced the use of a tool called Gitea for source code management over a git implementation. This tool provides the capability of store source code files, including workflow ones, meanwhile keeps track of any change. One of the aims of this tools is to support bioinformaticians during the workflow development as well as any other Cromwell requirement such as docker image definitions.
2. Workflow definition. Cromwell supports various script formats that describe the workflow for data processing. This processing can be split in several tasks and each of them can have their own software requirements. For instance, a simple workflow could be the transformation of omics data into a count matrix. This process could hypothetically be split in 2 steps: the first one, the creation of the count matrix and the last one, the transformation of the output files into a different files format. So, the whole process can be easily defined into a WDL script with two tasks and this script will be stored into the gitea tool.
3. Runtime environment. Even when AWS was introduced as the infrastructure for running workflows, this is just a slight idea of how things run. AWS provides Cromwell a scalable backend composed of many runtime environments following to Cromwell definition. These environments are building as docker instances that users must defined within the required software. For instance, according to the previous example, it would be necessary a docker with the cellranger software and its dependencies for count matrix building. An additional docker could be defined just for file format transformation within the desired software. In order to define these runtimes, users have to describe them in the standard “docker files” and store them in the gitea tool. HUTER Platform is configured to detect any docker file changes and build the associated docker image automatically.
4. Cromwell API. Cromwell works as a manager, this means that it received execution requests and delegates the execution in a specific machine. For requesting an execution, Cromwell provides two interfaces: a CLI and a service API. HUTER Platform exploits its API to deploy a complete service that allows users to run workflows in Cromwell over the stored data and ingest the outcomes. The whole process is tracked by registering the WDL execution, the inputs, outputs and who made the request.

4.2.9. huter-cli cromwell (request)

According to the previous considerations, huter-cli will be the tool engaged with the request of workflow executions. Although the first proposed approach was the *Web manager for pipeline execution,* which may be seen in section *4.7* of the document, HUTER bioinformaticians suggested that a CLI tool could be more flexible and easier to adapt for their daily work. So, for bringing both ideas closer, HUTER Platform centralize the Cromwell execution request in a remote service that can be invoked by client tools. Deliverable 2.4 will go in depth about the services functionalities, including this service.



```
D:\desenrolo\huter\data-access-tools\huter-cli>java -jar huter-cli-dev-v.01.03.jar cromwell -h
huter-cli - 01.03.0000-SNAPSHOT - dev - Loading cromwell...
huter-cli - 01.03.0000-SNAPSHOT - dev - Running requested execution...
huter-cli - 01.03.0000-SNAPSHOT - dev -

This command requests the execution in the Cromwell infrastructure of a known pipeline. Pipeline has to exist on the platform code repository, Gitea.

Available cromwell options
-----------------------------------------------------------------

-p (mandatory)
        Folder name gitea repository that contains the WDL script describing the pipeline process.

-m (optional)
        Relative or absolute path to the manifest file containing the input files to be processed. Manifest file has to be updated by the user in orde
r to assing the input variable name that expects the WDL script.

-v (optional)
        Relative or absolute path to a tsv file containing the key-value pairs with additional parameters needed for customizing the WDL script execut
ion. Expected TSV columns:
        key                                    |   value
        -------------------------------------------------------------
        simpleInputTestPipeline.intVariable     |   1
        simpleInputTestPipeline.floatVariable   |   2.55
        simpleInputTestPipeline.stringVariable  |   DownsamplingPowerOf2
        simpleInputTestPipeline.booleanVariable |   true


-d (mandatory)
        Custom description of the process that runs over the provided files.

-u (mandatory)
        Unique identifier of the requested cromwell process to update its status based on cromwell pipeline status.
        NOTE: this option must be provided without any other options to correctly execute process update.

-h
        It shows this help.


huter-cli - 01.03.0000-SNAPSHOT - dev - The operation was sucessfully executed

D:\desenrolo\huter\data-access-tools\huter-cli>_
```

Figure 16 - huter-cli cromwell -h

Figure 16 - huter-cli cromwell -h shows all the options for requesting and querying pipeline executions through the pipeline manager service. This kind of interaction method allows users to exploit a high-performance processing infrastructure from humble devices, just by typing a simple command.

The available parameters are:

-h: it shows the help information to guide users.

-p: it is the name of the WDL to be run. The HUTER Platform allows users to store their WDL in a source code management application. Thanks to that, for running a WDL is just necessary to provide the WDL name, so the HUTER services can find the required workflow definition.

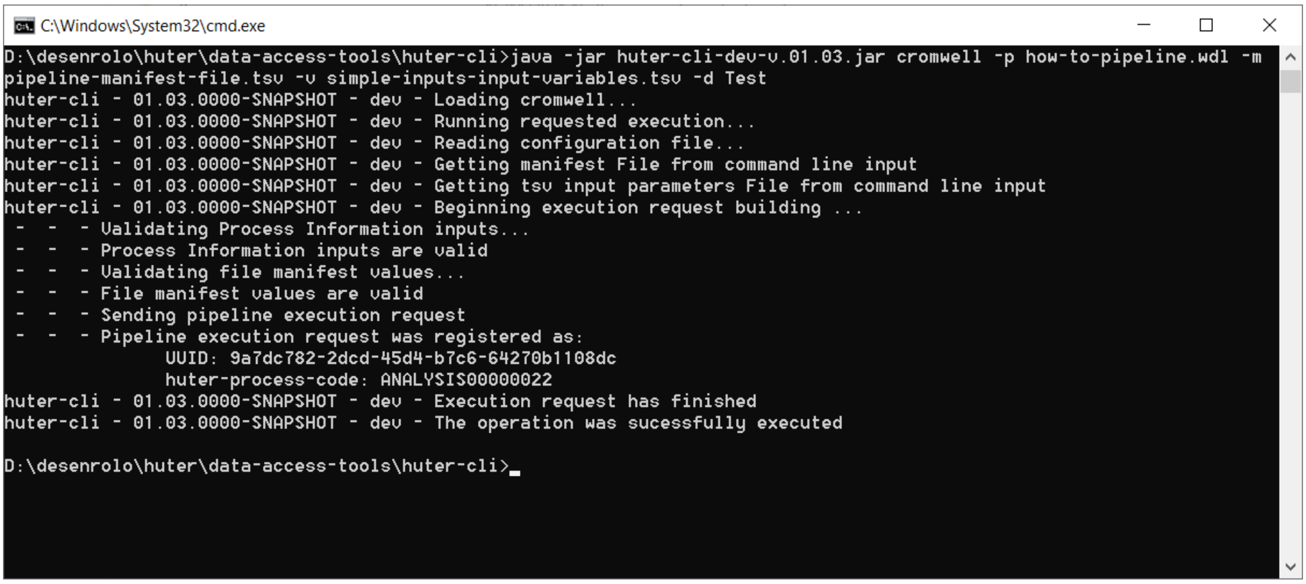
-m: it is the path to the generated manifest file containing a set of files that will be used in the workflow. This file has to be modified by the user to identify in which workflow parameter each file is referred. For instance, if a workflow WDL receives two files as parameters (param1 and param2), HUTER user should identify in the manifest file column “pipeLineParameterName”, which file is the param1 and which one the param2. File arrays (list of them) are also supported by assigning the same parameter name to all the desired files.

-v: it is the path to the values files. Workflows allows to be configured by providing which files will be processed. But also, they can be configured by other types of parameters like numbers, Strings, and so on. To provide this values, huter-cli lets users to provide a simple TSV containing key-value pairs with the parameter names and their values.

-d: it is a description of the required execution in order to provide information to other users about the reason to run the workflow or the expected values.

-u: this option should be used alone because it is related to the huter-cli cromwell (check) functionality. The provided value should be the UUID of the HUTER process in order to verify the current status of the process.

Using this Cromwell command and its options, users can request the execution of a workflow providing the wdl name and input parameters. Inputs can be files and/or some values that modifies the workflow behaviour. Nevertheless, all these pipeline parameters have to be defined in the WDL.

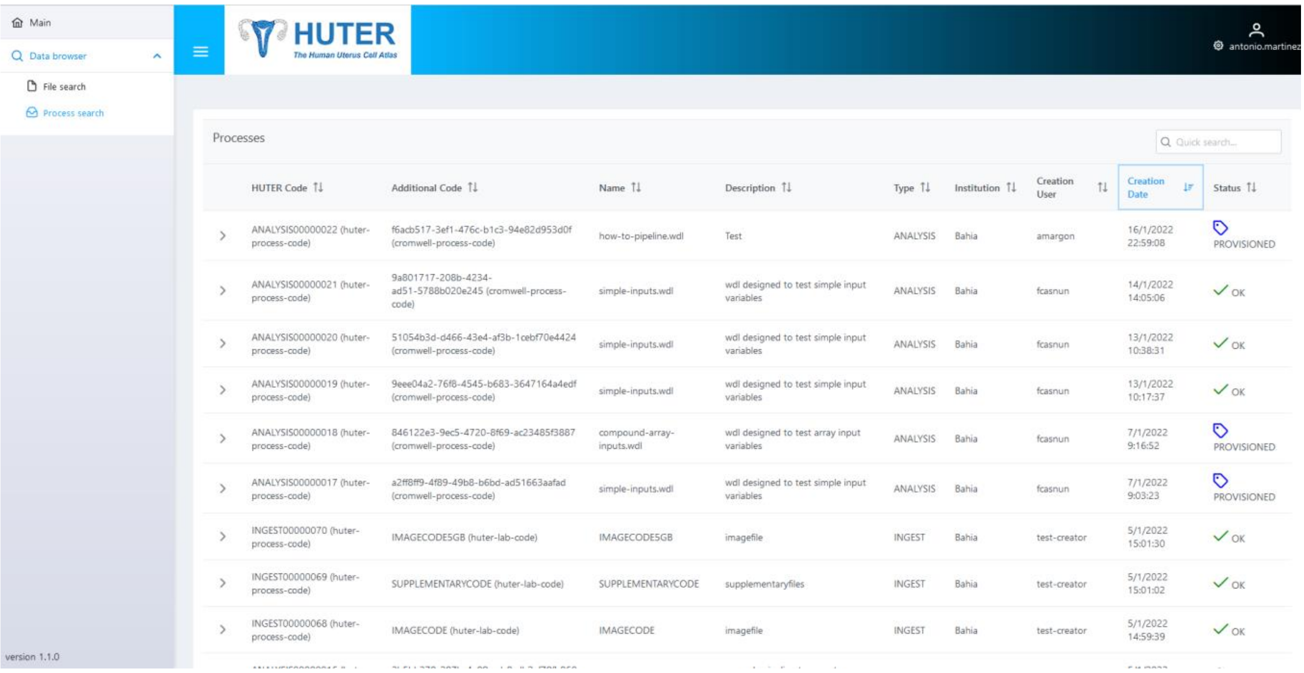


```
D:\desenrolo\huter\data-access-tools\huter-cli>java -jar huter-cli-dev-v.01.03.jar cromwell -p how-to-pipeline.wdl -m
pipeline-manifest-file.tsv -v simple-inputs-input-variables.tsv -d Test
huter-cli - 01.03.0000-SNAPSHOT - dev - Loading cromwell...
huter-cli - 01.03.0000-SNAPSHOT - dev - Running requested execution...
huter-cli - 01.03.0000-SNAPSHOT - dev - Reading configuration file...
huter-cli - 01.03.0000-SNAPSHOT - dev - Getting manifest File from command line input
huter-cli - 01.03.0000-SNAPSHOT - dev - Getting tsv input parameters File from command line input
huter-cli - 01.03.0000-SNAPSHOT - dev - Beginning execution request building ...
 -  -  - Validating Process Information inputs...
 -  -  - Process Information inputs are valid
 -  -  - Validating file manifest values...
 -  -  - File manifest values are valid
 -  -  - Sending pipeline execution request
 -  -  - Pipeline execution request was registered as:
               UUID: 9a7dc782-2dcd-45d4-b7c6-64270b1108dc
               huter-process-code: ANALYSIS00000022
huter-cli - 01.03.0000-SNAPSHOT - dev - Execution request has finished
huter-cli - 01.03.0000-SNAPSHOT - dev - The operation was sucessfully executed

D:\desenrolo\huter\data-access-tools\huter-cli>_
```

Figure 17 - huter-cli cromwell request execution

For instance, *Figure 17 - huter-cli cromwell request execution* shows how to request the execution of a pipeline for testing Cromwell parametrization. The given parameters are the manifest file that includes the files to be processed by the pipeline and the file with the parameter values. When the execution request is accepted, the user is informed about the process code in order to later check its status. Also, at section *4.3.2 Process search*, users can look for their request executions. However, at current time, the only way to update the process status is to execute the *huter-cli cromwell (check)* option.

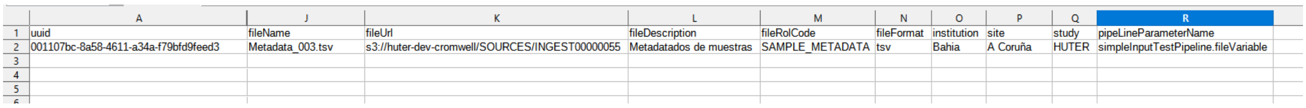
The request execution is visible in the Data Browser as shown at *Figure 18 - Pipeline execution registration as PROVISIONED*. The status of an accepted execution request is PROVISIONED because Cromwell will plan its execution as soon as possible. While a status update is not requested, the current status will remain.



Processes

HUTER Code	Additional Code	Name	Description	Type	Institution	Creation User	Creation Date	Status
ANALYSIS00000022 (huter-process-code)	f6acb517-3ef1-476c-b1c3-94e82d953d0f (cromwell-process-code)	how-to-pipeline.wdl	Test	ANALYSIS	Bahia	amargon	16/1/2022 22:59:08	PROVISIONED
ANALYSIS00000021 (huter-process-code)	9a801717-208b-4234-ad51-5788b020e245 (cromwell-process-code)	simple-inputs.wdl	wdl designed to test simple input variables	ANALYSIS	Bahia	fcasnun	14/1/2022 14:05:06	✓ OK
ANALYSIS00000020 (huter-process-code)	51054b3d-d466-43e4-af3b-1cebf70e4424 (cromwell-process-code)	simple-inputs.wdl	wdl designed to test simple input variables	ANALYSIS	Bahia	fcasnun	13/1/2022 10:38:31	✓ OK
ANALYSIS00000019 (huter-process-code)	9eee04a2-76f8-4545-b683-3647164a4edf (cromwell-process-code)	simple-inputs.wdl	wdl designed to test simple input variables	ANALYSIS	Bahia	fcasnun	13/1/2022 10:17:37	✓ OK
ANALYSIS00000018 (huter-process-code)	846122e3-9ec5-4720-8f69-ac23485f3887 (cromwell-process-code)	compound-array-inputs.wdl	wdl designed to test array input variables	ANALYSIS	Bahia	fcasnun	7/1/2022 9:16:52	PROVISIONED
ANALYSIS00000017 (huter-process-code)	a2ff8ff9-4f89-49b8-b6bd-ad51663aafad (cromwell-process-code)	simple-inputs.wdl	wdl designed to test simple input variables	ANALYSIS	Bahia	fcasnun	7/1/2022 9:03:23	PROVISIONED
INGEST00000070 (huter-process-code)	IMAGECODE5GB (huter-lab-code)	IMAGECODE5GB	imagefile	INGEST	Bahia	test-creator	5/1/2022 15:01:30	✓ OK
INGEST00000069 (huter-process-code)	SUPPLEMENTARYCODE (huter-lab-code)	SUPPLEMENTARYCODE	supplementaryfiles	INGEST	Bahia	test-creator	5/1/2022 15:01:02	✓ OK
INGEST00000068 (huter-process-code)	IMAGECODE (huter-lab-code)	IMAGECODE	imagefile	INGEST	Bahia	test-creator	5/1/2022 14:59:39	✓ OK

Figure 18 - Pipeline execution registration as PROVISIONED

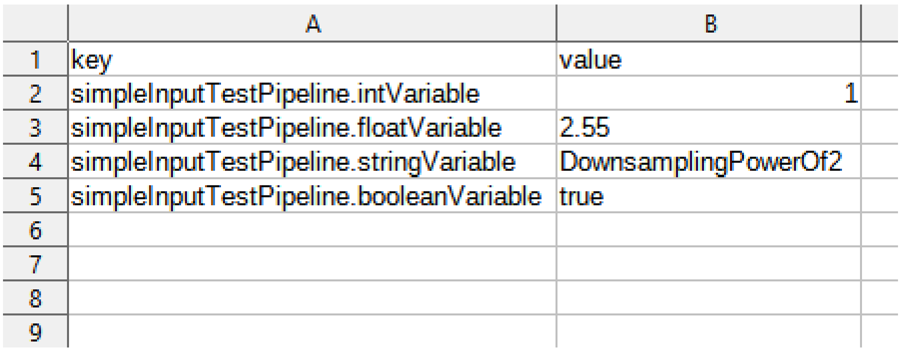


	A	J	K	L	M	N	O	P	Q	R
1	uuid	fileName	fileUrl	fileDescription	fileRolCode	fileFormat	institution	site	study	pipeLineParameterName
2	001107bc-8a58-4611-a34a-f79bfd9feed3	Metadata_003.tsv	s3://huter-dev-cromwell/SOURCES/INGEST00000055	Metadatados de muestras	SAMPLE_METADATA	tsv	Bahia	A Coruña	HUTER	simpleInputTestPipeline.fileVariable
3										
4										
5										

Figure 19 - huter-cli cromwell manifest file with input file parameters

At Figure 19 - huter-cli cromwell manifest file with input file parameters, the required modification for matching file registries and wdl parameters can be seen. With this simple action, users can define how input files are identified in the wdl by a parameter name. Furthermore, if WDL requires a list of input files, users can assign the same parameter name to every single file in the manifest file that has to be part of the input list.

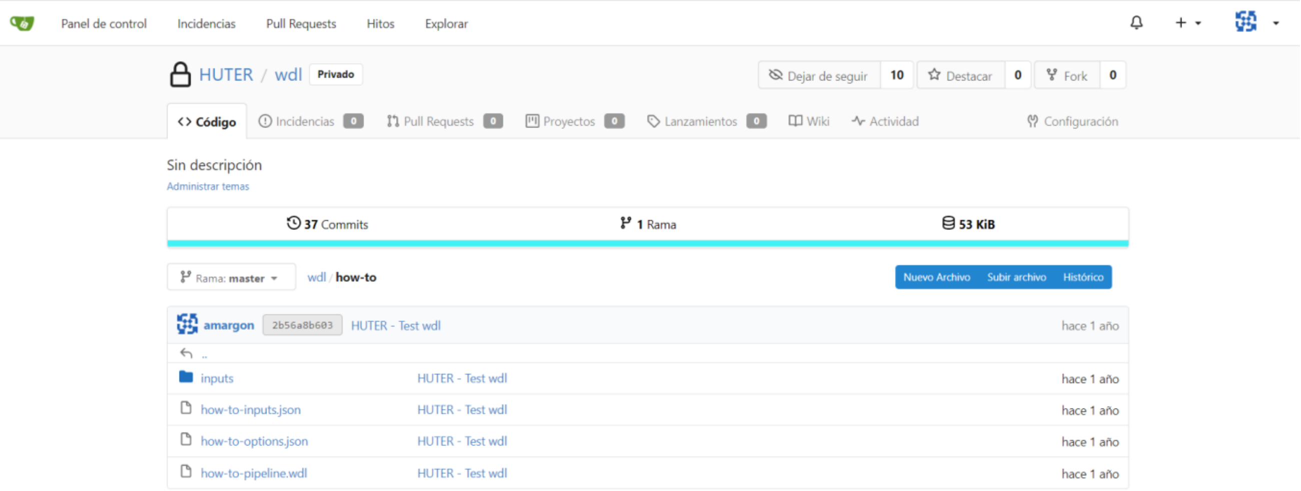
According to user's experience, this straightforward method is very useful for simple WDL and it can be evolved to exploit all the options provided by the WDL format.



	A	B
1	key	value
2	simpleInputTestPipeline.intVariable	1
3	simpleInputTestPipeline.floatVariable	2.55
4	simpleInputTestPipeline.stringVariable	DownsamplingPowerOf2
5	simpleInputTestPipeline.booleanVariable	true
6		
7		
8		
9		

Figure 20 - huter-cli cromwell, example of parameter file

With regards to the parametrization of other variables in the WDL, users can provide a simple tsv file that follows the approach of the manifest file. As shown at *Figure 20 - huter-cli cromwell, example of parameter file*, users can type a matrix where each tuple is a key-value pair that matches the WDL parameters. Like the input files, simple lists are supported as input parameter but complex structures such as map are not implemented yet. In a future release, this feature is expected to be accomplished in order to achieve a complete integration between our command line tool and the Cromwell API.



Sin descripción
Administrar temas

amargon 2b56a8b603	HUTER - Test wdl	hace 1 año
..		
inputs	HUTER - Test wdl	hace 1 año
how-to-inputs.json	HUTER - Test wdl	hace 1 año
how-to-options.json	HUTER - Test wdl	hace 1 año
how-to-pipeline.wdl	HUTER - Test wdl	hace 1 año

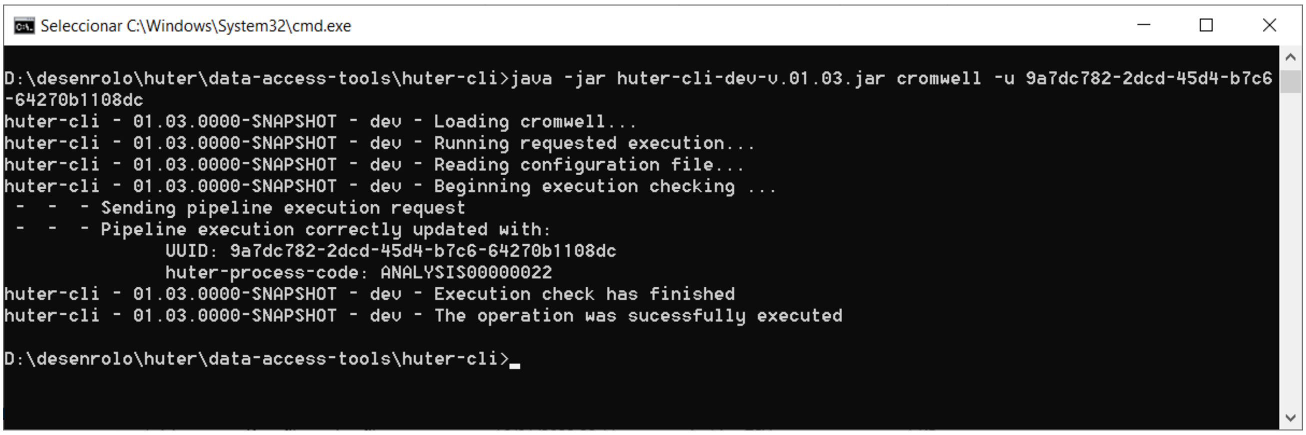
Figure 21 - gitea, test WDL

Logically, also the name of the WDL has to be provided. The huter-cli tool relies on the service to retrieve the WDL definition from the gitea service by locating it thanks to its name, as *Figure 21 - gitea, test WDL* displays. This way suffers from a slight limitation because only the last version of the WDL can be executed. However, in a next version of the tool suite users will be able exploit gitea changes tracking to provide a revision or tag code to be executed.

4.2.10. huter-cli cromwell (check)

Once the execution has been requested, users receive a list of identification codes for their workflows. These codes are the keys for asking the platform about the process status.

In this early version of platform, users have to request every status update by execution a huter-cli command as shown at *Figure 22 - huter-cli cromwell -u*.



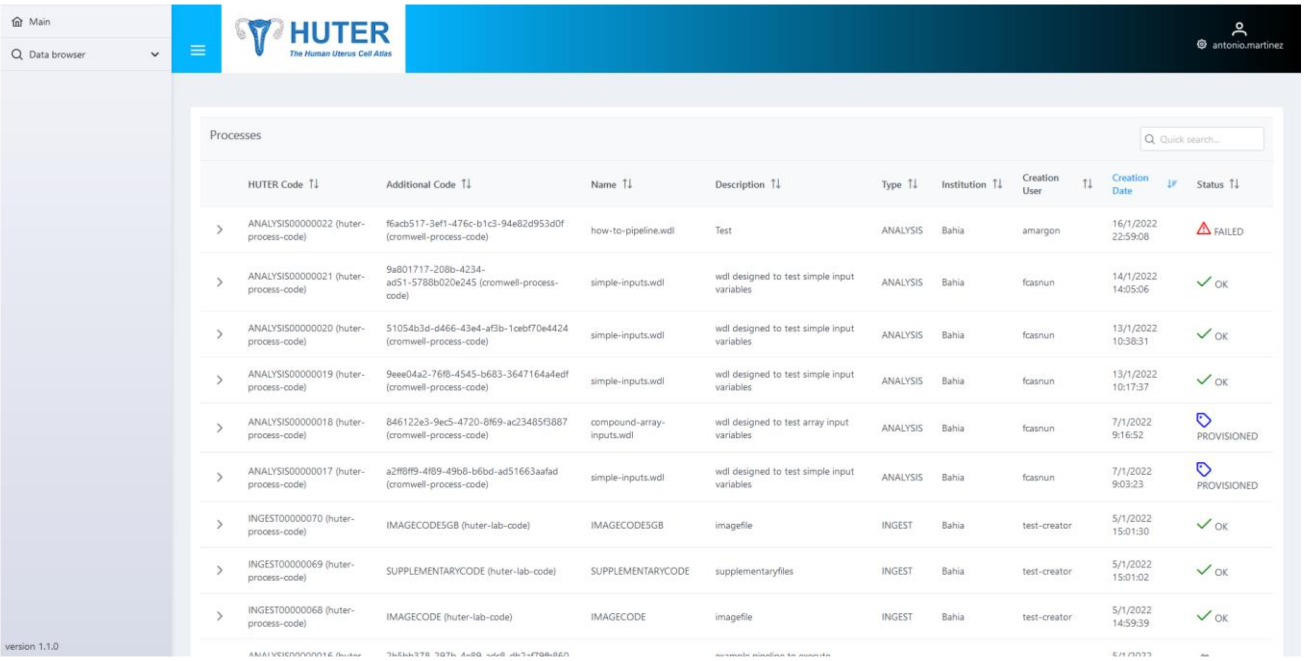
```
D:\desenrolo\huter\data-access-tools\huter-cli>java -jar huter-cli-dev-v.01.03.jar cromwell -u 9a7dc782-2dcd-45d4-b7c6
-64270b1108dc
huter-cli - 01.03.0000-SNAPSHOT - dev - Loading cromwell...
huter-cli - 01.03.0000-SNAPSHOT - dev - Running requested execution...
huter-cli - 01.03.0000-SNAPSHOT - dev - Reading configuration file...
huter-cli - 01.03.0000-SNAPSHOT - dev - Beginning execution checking ...
 -  -  - Sending pipeline execution request
 -  -  - Pipeline execution correctly updated with:
               UUID: 9a7dc782-2dcd-45d4-b7c6-64270b1108dc
               huter-process-code: ANALYSIS00000022
huter-cli - 01.03.0000-SNAPSHOT - dev - Execution check has finished
huter-cli - 01.03.0000-SNAPSHOT - dev - The operation was sucessfully executed

D:\desenrolo\huter\data-access-tools\huter-cli>_
```

Figure 22 - huter-cli cromwell -u

By performing this action, users request the pipeline manager to check process status in the cromwell infrastructure for a next updating in the HUTER Platform. Then, the process status is shown to the user through the command line but also its readiness to be retrieved by any other deployed tool.

For instance, as shown in section *4.3.2, Process search* tool in the data-browser can be used to check the stored process status. This is useful when running long processes that can last many hours, so bioinformaticians should not be forced to jot down the process code obtained in the section *4.2.9 huter-cli cromwell (request)*. At *Figure 23 - Data browser, execution request updated*, a set of executions are shown with different final statuses where users can verify if the process has properly finished or not.

Apart from checking the process status, if a process finishes and output file exists, the pipeline manager service will ingest the output file in the platform automatically. Hence, HUTER Platform provides integrated tools that provides a unified set of functionalities all inside the cloud infrastructure.



Processes

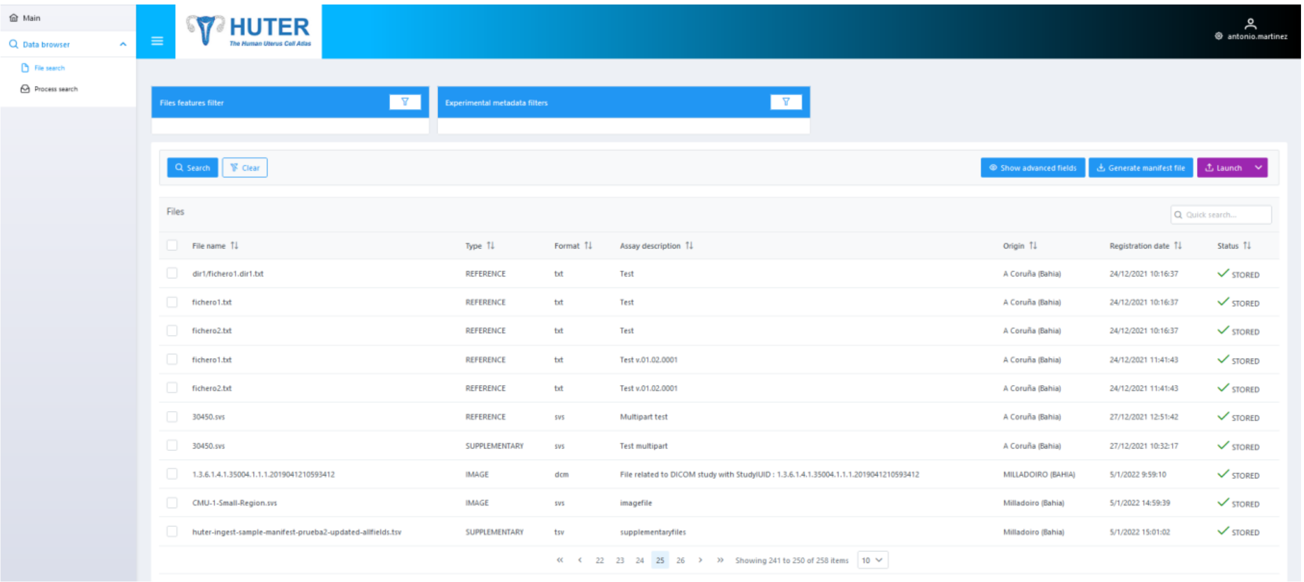
	HUTER Code	Additional Code	Name	Description	Type	Institution	Creation User	Creation Date	Status
>	ANALYSIS00000022 (huter-process-code)	f6acb517-3ef1-476c-b1c3-94e82d953d0f (cromwell-process-code)	how-to-pipeline.wdl	Test	ANALYSIS	Bahia	amargon	16/1/2022 22:59:08	FAILED
>	ANALYSIS00000021 (huter-process-code)	9a801717-208b-4234-ad51-5788b020e245 (cromwell-process-code)	simple-inputs.wdl	wdl designed to test simple input variables	ANALYSIS	Bahia	fcasnun	14/1/2022 14:05:06	✓ OK
>	ANALYSIS00000020 (huter-process-code)	51054b3d-d466-43e4-af3b-1cebf70e4424 (cromwell-process-code)	simple-inputs.wdl	wdl designed to test simple input variables	ANALYSIS	Bahia	fcasnun	13/1/2022 10:38:31	✓ OK
>	ANALYSIS00000019 (huter-process-code)	9eee04a2-76f8-4545-b683-3647164a4edf (cromwell-process-code)	simple-inputs.wdl	wdl designed to test simple input variables	ANALYSIS	Bahia	fcasnun	13/1/2022 10:17:37	✓ OK
>	ANALYSIS00000018 (huter-process-code)	846122e3-9ec5-4720-8f69-ac23485f3887 (cromwell-process-code)	compound-array-inputs.wdl	wdl designed to test array input variables	ANALYSIS	Bahia	fcasnun	7/1/2022 9:16:52	PROVISIONED
>	ANALYSIS00000017 (huter-process-code)	a2ff8ff9-4f89-49b8-b6bd-ad51663aafad (cromwell-process-code)	simple-inputs.wdl	wdl designed to test simple input variables	ANALYSIS	Bahia	fcasnun	7/1/2022 9:03:23	PROVISIONED
>	INGEST00000070 (huter-process-code)	IMAGECODE5GB (huter-lab-code)	IMAGECODE5GB	imagefile	INGEST	Bahia	test-creator	5/1/2022 15:01:30	✓ OK
>	INGEST00000069 (huter-process-code)	SUPPLEMENTARYCODE (huter-lab-code)	SUPPLEMENTARYCODE	supplementaryfiles	INGEST	Bahia	test-creator	5/1/2022 15:01:02	✓ OK
>	INGEST00000068 (huter-process-code)	IMAGECODE (huter-lab-code)	IMAGECODE	imagefile	INGEST	Bahia	test-creator	5/1/2022 14:59:39	✓ OK

Figure 23 - Data browser, execution request updated

4.3. Data browser

The HUTER Platform provides a tool for data transferring from the laboratory infrastructure to the HUTER cloud storage in AWS S3, the HUTER CLI. Additionally, and even more important, it also registers the ingested files in a central database for a later exploitation. In this context, HUTER Platform can store a large amount of data, so a visual interface is required to reduce the effort for searching not only the uploaded files, but also the executed processes on them.

Data browser is a tool that reduces the complexity of dealing with a large number of files and related data through a web application. That means users can easily find and exploit the information stored in the cloud platform. In order to ease this task, the web interface offers a flexible and adaptative communication tool between users and platform not considering their location, devices or even their user expertise.



□	File name ⇅	Type ⇅	Format ⇅	Assay description ⇅	Origin ⇅	Registration date ⇅	Status ⇅
□	dir1/fichero1.dir1.txt	REFERENCE	txt	Test	A Coruña (Bahia)	24/12/2021 10:16:37	✓ STORED
□	fichero1.txt	REFERENCE	txt	Test	A Coruña (Bahia)	24/12/2021 10:16:37	✓ STORED
□	fichero2.txt	REFERENCE	txt	Test	A Coruña (Bahia)	24/12/2021 10:16:37	✓ STORED
□	fichero1.txt	REFERENCE	txt	Test v.01.02.0001	A Coruña (Bahia)	24/12/2021 11:41:43	✓ STORED
□	fichero2.txt	REFERENCE	txt	Test v.01.02.0001	A Coruña (Bahia)	24/12/2021 11:41:43	✓ STORED
□	30450.svs	REFERENCE	svs	Multipart test	A Coruña (Bahia)	27/12/2021 12:51:42	✓ STORED
□	30450.svs	SUPPLEMENTARY	svs	Test multipart	A Coruña (Bahia)	27/12/2021 10:32:17	✓ STORED
□	1.3.6.1.4.1.35004.1.1.1.2019041210593412	IMAGE	dcm	File related to DICOM study with StudyIUID : 1.3.6.1.4.1.35004.1.1.1.2019041210593412	MILLADOIRO (BAHIA)	5/1/2022 9:59:10	✓ STORED
□	CMU-1-Small-Region.svs	IMAGE	svs	imagefile	Milladoiro (Bahia)	5/1/2022 14:59:39	✓ STORED
□	huter-ingest-sample-manifest-prueba2-updated-allfields.tsv	SUPPLEMENTARY	tsv	supplementaryfiles	Milladoiro (Bahia)	5/1/2022 15:01:02	✓ STORED

« ‹ 22 23 24 25 26 › » Showing 241 to 250 of 258 items 10

Figure 24 - Data browser - File search tool

4.3.1. File search

The main functionality of the Data browser is to provide a visual tool for search files which are stored in the platform. Hence, these files should have been previously registered using the *HUTER CLI*.

The interface of the file search functionality was designed to be simple and intuitive for the users as can be seen at *Figure 24 - Data browser - File search tool*. The web interface is composed of widgets for data filtering, a tool bar where additional functions can be launched and a paginated table where registered files are displayed.

4.3.1.1. File metadata

File metadata is displayed in a paginated table. Any column can be ordered by clicking on its name. Additionally, a Quick search can be performed by typing in the provided box what will filter the registries that contain that word in any of its fields.

Every column is displayed as text, but two columns have a different behaviour: registration date and status.

- Registration date shows the date when file was ingested in the platform. This date is location sensitive, that means, depending on the user location it will show the date in the user time zone. However, the internal date is registered as a UTC+0 date time, allowing a more adaptative system for international use.
- Status is the current situation of the file in the platform. This field displays an encoded value that allows its search but also an icon for easily understand the meaning. If further information is required, a help tag will display the description of the status by locating mouse on the icon.

  - PROVISIONED means that file was prepared (see section *4.2.3.1 huter-cli prepare*) but not uploaded yet.
  - STORED means that file is already in the platform and it is ready for its exploitation.
  - VALIDATION_FAIL will be shown if file does not match quality requirements.
  - DELETED is expected to be used if a file was stored and then, deleted.

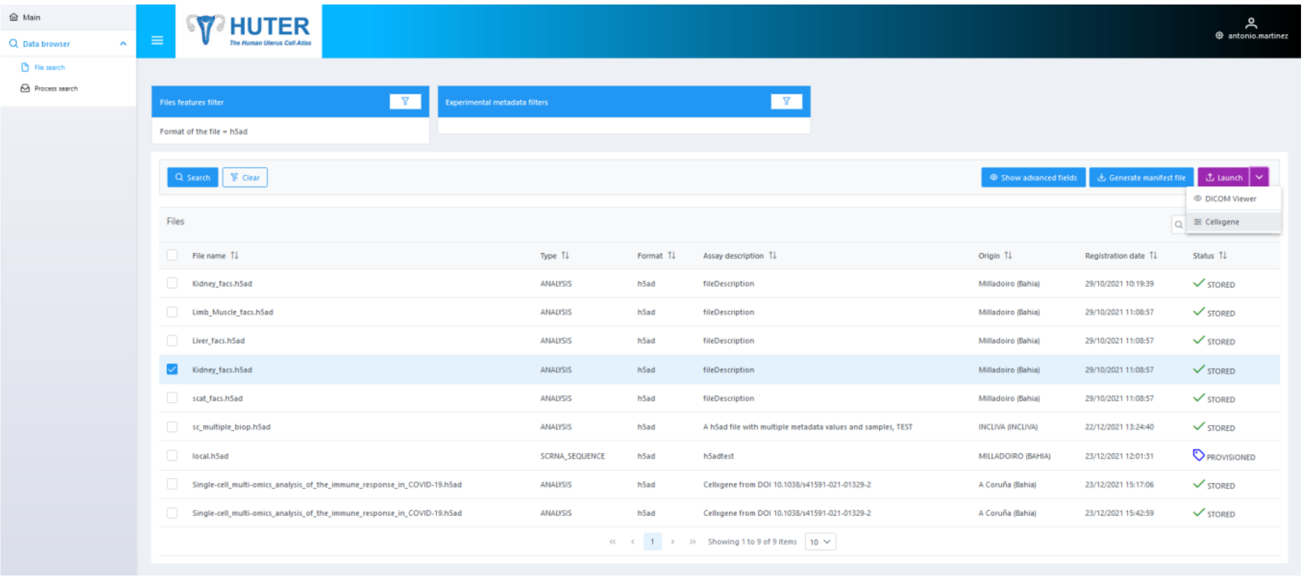
4.3.1.2. File search toolbar

Toolbar offers additional functionalities over the registered files, for instance users can load other tools in the platform over a file.

Current tools are:

- Generate manifest file. It allows users to generate a *Previous considerations*: Manifest file within the selected files. Only STORED files can be included in the manifest file.
- Show additional fields. It is a tool for changing the visualized metadata. By clicking on this button, some table columns disappear and new ones, with a more detailed data, are launched. This mode is expected to be used by bioinformaticians when they are running as shown in section *4.4.2*, because they need to load some files from AWS S3 storage to their R Studio workspace.
- Launch menu offers a list of applications that can be loaded over a file selection. Data browser is aimed to be the launcher for complementary tools over the stored files like scRNAseq matrix or DICOM images. Each application has its own requirements in order to be launched, like the file type or the number of selected files.
  - Dicom Viewer option opens the Advanced DICOM viewer to visualize the selected IMAGE file in dcm format. No other file type or IMAGE format could be loaded in this Viewer.
  - Cellxgene option loads a Cellxgene instance in web mode over the selected h5ad file. This kind of file has to be indexed as ANALYSIS type and h5ad format to assure that contains correct data.

Thanks to the launcher function, any user can visualize file data using any of the specialized tool that are deployed in the HUTER Platform. For instance, *Figure 25 - Data browser - Launching Cellxgene with a stored h5ad file* shows how an h5ad file can be opened from data browser using the Cellxgene.



Format of the file = h5ad

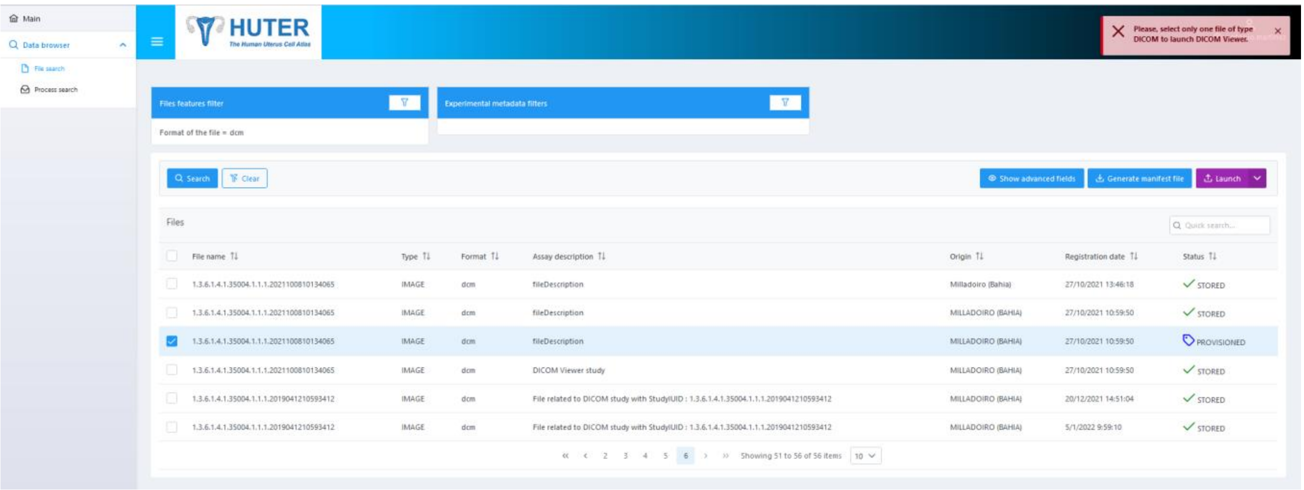
☐	File name	Type	Format	Assay description	Origin	Registration date	Status
☐	Kidney_facs.h5ad	ANALYSIS	h5ad	fileDescription	Milladoiro (Bahia)	29/10/2021 10:19:39	✓ STORED
☐	Limb_Muscle_facs.h5ad	ANALYSIS	h5ad	fileDescription	Milladoiro (Bahia)	29/10/2021 11:08:57	✓ STORED
☐	Liver_facs.h5ad	ANALYSIS	h5ad	fileDescription	Milladoiro (Bahia)	29/10/2021 11:08:57	✓ STORED
☑	Kidney_facs.h5ad	ANALYSIS	h5ad	fileDescription	Milladoiro (Bahia)	29/10/2021 11:08:57	✓ STORED
☐	scat_facs.h5ad	ANALYSIS	h5ad	fileDescription	Milladoiro (Bahia)	29/10/2021 11:08:57	✓ STORED
☐	sc_multiple_biop.h5ad	ANALYSIS	h5ad	A h5ad file with multiple metadata values and samples, TEST	INCLIVA (INCLIVA)	22/12/2021 13:24:40	✓ STORED
☐	local.h5ad	SCRNA_SEQUENCE	h5ad	h5adtest	MILLADOIRO (BAHIA)	23/12/2021 12:01:31	PROVISIONED
☐	Single-cell_multi-omics_analysis_of_the_immune_response_in_COVID-19.h5ad	ANALYSIS	h5ad	Cellxgene from DOI 10.1038/s41591-021-01329-2	A Coruña (Bahia)	23/12/2021 15:17:06	✓ STORED
☐	Single-cell_multi-omics_analysis_of_the_immune_response_in_COVID-19.h5ad	ANALYSIS	h5ad	Cellxgene from DOI 10.1038/s41591-021-01329-2	A Coruña (Bahia)	23/12/2021 15:42:59	✓ STORED

« ‹ 1 › » Showing 1 to 9 of 9 items 10

Figure 25 - Data browser - Launching Cellxgene with a stored h5ad file

Furthermore, the *Advanced DICOM viewer* can be directly loaded to visualize a specific DICOM image. These images are not stored in the AWS S3 but in a specialized imaging server called PACS. However, a reference to those images is indexed to support the viewer launching from Data Browser.

Furthermore, if a non-uploaded file is selected for its visualization, the launching functionality will warn that the file will not be loaded. *Figure 26 - Data browser - Failure when DICOM viewer is launched with non-existing file* shows how a PROVISIONED, but not uploaded, DICOM file cannot be loaded because it does not exist in the platform.



Please, select only one file of type DICOM to launch DICOM Viewer.

Format of the file = dcm

☐	File name	Type	Format	Assay description	Origin	Registration date	Status
☐	1.3.6.1.4.1.35004.1.1.1.2021100810134065	IMAGE	dcm	fileDescription	Milladoiro (Bahia)	27/10/2021 13:46:18	✓ STORED
☐	1.3.6.1.4.1.35004.1.1.1.2021100810134065	IMAGE	dcm	fileDescription	MILLADOIRO (BAHIA)	27/10/2021 10:59:50	✓ STORED
☑	1.3.6.1.4.1.35004.1.1.1.2021100810134065	IMAGE	dcm	fileDescription	MILLADOIRO (BAHIA)	27/10/2021 10:59:50	PROVISIONED
☐	1.3.6.1.4.1.35004.1.1.1.2021100810134065	IMAGE	dcm	DICOM Viewer study	MILLADOIRO (BAHIA)	27/10/2021 10:59:50	✓ STORED
☐	1.3.6.1.4.1.35004.1.1.1.2019041210593412	IMAGE	dcm	File related to DICOM study with StudyIUID : 1.3.6.1.4.1.35004.1.1.1.2019041210593412	MILLADOIRO (BAHIA)	20/12/2021 14:51:04	✓ STORED
☐	1.3.6.1.4.1.35004.1.1.1.2019041210593412	IMAGE	dcm	File related to DICOM study with StudyIUID : 1.3.6.1.4.1.35004.1.1.1.2019041210593412	MILLADOIRO (BAHIA)	5/1/2022 9:59:10	✓ STORED

« ‹ 2 3 4 5 6 › » Showing 51 to 56 of 56 items 10

Figure 26 - Data browser - Failure when DICOM viewer is launched with non-existing file

These additional deployed tools will be later explored in this document, but new ones could be integrated in the platform due to the flexible deploying solution.

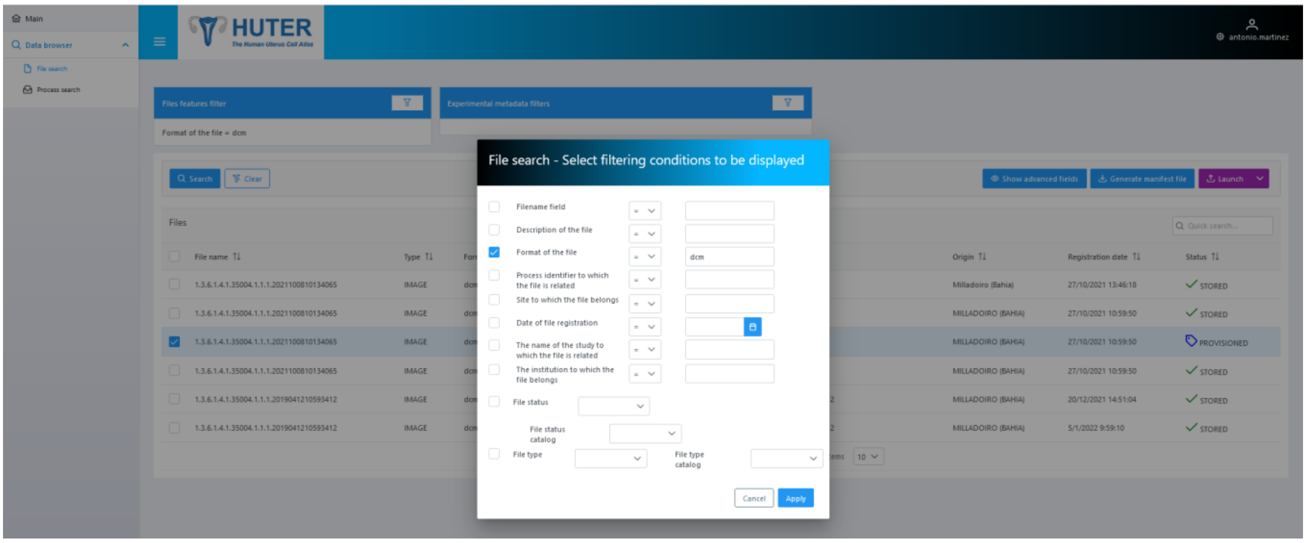
4.3.1.3. File search filters

During HUTER project, and far away, a large number of files are expected to be stored in the cloud space. However, locating them would be a problem for users if they were just able to search them by name. For easing this task, HUTER Platform has transformed this human-related issue into an opportunity for boosting data tracking (see section *4.2.6 Metadata indexation*). Data tracking provides links among registries in the platform, what includes files and entity metadata. This net of information offers to any user the ability of finding data through the metadata of its related entities. For instance, users will be able to find a file in the platform by its name but also by providing its type, its registration date and, even, the sample type from where file data was obtained.

For this reason, Data Browser exposes a web interface which allows users to select this filtering options during file browsing. Although this feature could look simply, section *4.2.6 Metadata indexation* introduced the functionality for dynamic data ingestion what mean the ability to index any traits of an entity. So, the creation of a tool for filtering data over an unknown set of features was a challenge that Data browser provides as filtering widgets.

These widgets are able to generate or destroy filtering options depending on the indexed data. Hence, users could add a new characteristic to the sample metadata and these widgets could be configured to show users a filtering option over the new traits. Adding this capacity to the ability of linking data, Data browser provides a powerful tool for file managing in a cloud environment avoiding the need of replicating a file index. So, users may stop worrying about remembering what kind of data they need.

File metadata filtering was mentioned to illustrate with an example how these widgets reduce the cognitive effort of finding files in the HUTER Platform. *Figure 27 - Data browser, file search by file metadata* shows the set of filters that can be applied for file location by looking for some metadata values.



Format of the file = dcm

File search - Select filtering conditions to be displayed

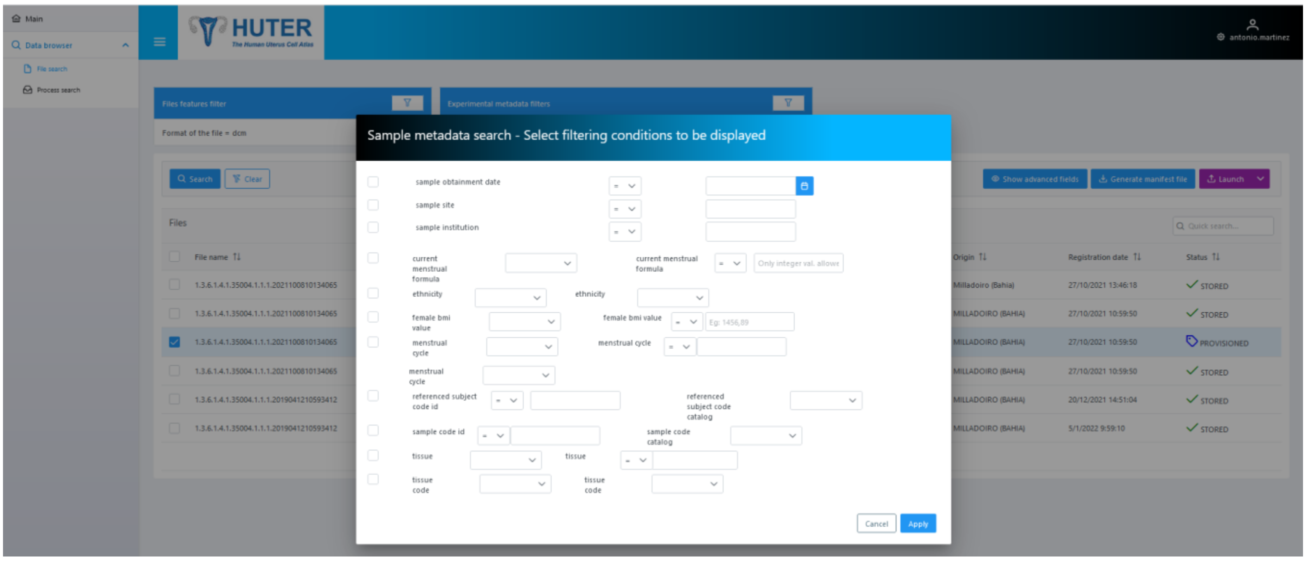
- □ Filename field =
- □ Description of the file =
- ☑ Format of the file = dcm
- □ Process identifier to which the file is related =
- □ Site to which the file belongs =
- □ Date of file registration =
- □ The name of the study to which the file is related =
- □ The institution to which the file belongs =
- □ File status
- File status catalog
- □ File type
- File type catalog

File name	Type	Origin	Registration date	Status
1.3.6.1.4.1.35004.1.1.1.2021100810134065	IMAGE	Milladoiro (Bahia)	27/10/2021 13:46:18	✓ STORED
1.3.6.1.4.1.35004.1.1.1.2021100810134065	IMAGE	MILLADOIRO (BAHIA)	27/10/2021 10:59:50	✓ STORED
☑ 1.3.6.1.4.1.35004.1.1.1.2021100810134065	IMAGE	MILLADOIRO (BAHIA)	27/10/2021 10:59:50	PROVISIONED
1.3.6.1.4.1.35004.1.1.1.2021100810134065	IMAGE	MILLADOIRO (BAHIA)	27/10/2021 10:59:50	✓ STORED
1.3.6.1.4.1.35004.1.1.1.2019041210593412	IMAGE	MILLADOIRO (BAHIA)	20/12/2021 14:51:04	✓ STORED
1.3.6.1.4.1.35004.1.1.1.2019041210593412	IMAGE	MILLADOIRO (BAHIA)	5/1/2022 9:59:10	✓ STORED

Figure 27 - Data browser, file search by file metadata

On the other hand, *Figure 28 - Data browser, file search by sample metadata shows* some examples of how file filtering by sample features can be done. Each check represents a filter that can be applied to find a sample and, from there, all files related to that sample. These filtering options are dynamically generated over some traits of the sample so it can be changed if new sample characteristics are stored. For instance, bioinformaticians could add a new sample trait and, after an easy database configuration, a new filter option will be generated without deploying a new data browser version.

Finally, additional widgets could be developed over process and subject metadata using the same approach if they were required in the future.



Format of the file = dcm

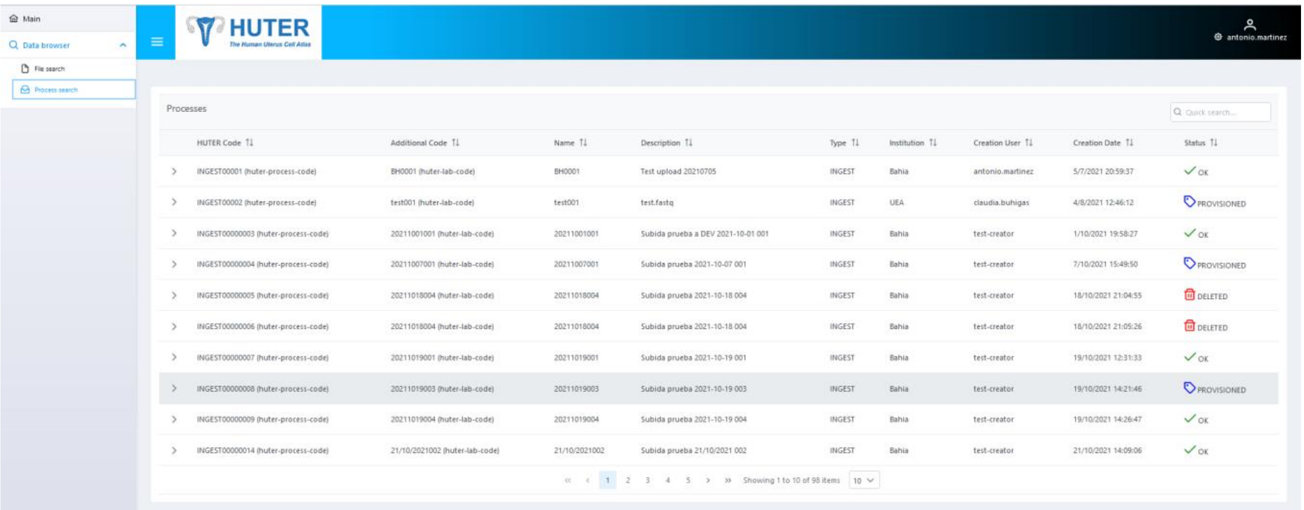
Sample metadata search - Select filtering conditions to be displayed

- □ sample obtainment date =
- □ sample site =
- □ sample institution =
- □ current menstrual formula — current menstrual formula =
- □ ethnicity — ethnicity
- □ female bmi value — female bmi value =
- □ menstrual cycle — menstrual cycle =
- menstrual cycle
- □ referenced subject code id = — referenced subject code catalog
- □ sample code id = — sample code catalog
- □ tissue — tissue =
- □ tissue code — tissue code

Figure 28 - Data browser, file search by sample metadata

4.3.2. Process search

Every user interaction within the platform is expected to be registered as a process. Using this information, platform can assure the required data tracking and additional information for user processes.



	HUTER Code	Additional Code	Name	Description	Type	Institution	Creation User	Creation Date	Status
>	INGEST00001 (huter-process-code)	BH0001 (huter-lab-code)	BH0001	Test upload 20210705	INGEST	Bahia	antonio.martinez	5/7/2021 20:59:37	✓ OK
>	INGEST00002 (huter-process-code)	test001 (huter-lab-code)	test001	test.fastq	INGEST	UEA	claudia.buhigas	4/8/2021 12:46:12	PROVISIONED
>	INGEST00000003 (huter-process-code)	20211001001 (huter-lab-code)	20211001001	Subida prueba a DEV 2021-10-01 001	INGEST	Bahia	test-creator	1/10/2021 19:58:27	✓ OK
>	INGEST00000004 (huter-process-code)	20211007001 (huter-lab-code)	20211007001	Subida prueba 2021-10-07 001	INGEST	Bahia	test-creator	7/10/2021 15:49:50	PROVISIONED
>	INGEST00000005 (huter-process-code)	20211018004 (huter-lab-code)	20211018004	Subida prueba 2021-10-18 004	INGEST	Bahia	test-creator	18/10/2021 21:04:55	DELETED
>	INGEST00000006 (huter-process-code)	20211018004 (huter-lab-code)	20211018004	Subida prueba 2021-10-18 004	INGEST	Bahia	test-creator	18/10/2021 21:05:26	DELETED
>	INGEST00000007 (huter-process-code)	20211019001 (huter-lab-code)	20211019001	Subida prueba 2021-10-19 001	INGEST	Bahia	test-creator	19/10/2021 12:31:33	✓ OK
>	INGEST00000008 (huter-process-code)	20211019003 (huter-lab-code)	20211019003	Subida prueba 2021-10-19 003	INGEST	Bahia	test-creator	19/10/2021 14:21:46	PROVISIONED
>	INGEST00000009 (huter-process-code)	20211019004 (huter-lab-code)	20211019004	Subida prueba 2021-10-19 004	INGEST	Bahia	test-creator	19/10/2021 14:26:47	✓ OK
>	INGEST00000014 (huter-process-code)	21/10/2021002 (huter-lab-code)	21/10/2021002	Subida prueba 21/10/2021 002	INGEST	Bahia	test-creator	21/10/2021 14:09:06	✓ OK

Showing 1 to 10 of 98 items

Figure 29 - Data browser, process search

This interface is simpler than File search one because is not expected to be intensely exploited. However, as shown at *Figure 29 - Data browser, process search*, users can search for a specific process to know its status, so it is a complementary tool for the HUTER CLI functionalities such as pipeline execution request. For instance, using this functionality, users can check if a process has correctly finished or not.

4.4. Data visualization and interactive processing tools

Along the HUTER Project, data visualization tools were expected to be developed and some efforts were made in that sense. However, the overwhelming workload for building just one cutting-edge tool could jeopardized the Platform deployment. That is why the agreed approach was to deploy existing tools that can run fully integrated within the platform services.

So, herein, the set of tools for data visualization and interactive processing will be collected. These tools are devoted to exploit data that users share through the HUTER Platform avoiding software requirements on premises. The provided set of tools was selected to be useful regarding to the stored data: Single Cell RNA Sequencing, Single Cell DNA Methylation Sequencing. However, it is possible to expand the suite by deploying new tools that can manage new types of data or file formats.

One of the most important assets of this solution is that can be accessed through a web browser. This means that users can take advantage of a powerful cloud infrastructure meanwhile they use it from anywhere and almost any kind of device.

Other plus of this solution is that data is already stored in the cloud space, so there is no need of downloading it anywhere. This is especially useful because OMICs data normally spend lots of gigabytes that do not have to be downloaded using this solution.

4.4.1. Cellxgene

The HUTER Platform integrates Cellxgene in order to provide tools to visualize, analyze and exploit scRNAseq datasets (see Figure 30). Cellxgene is an open and interactive data explorer for single-cell datasets, such as those coming from the Human Cell Atlas, designed by the Chan Zuckerberg foundation. In fact, Cellxgene is becoming in the standard visualization tool for these kind of data among the research community beyond the HCA. Its interface is based on web development techniques to enable fast visualizations of at least 1 million cells.

The interface was designed to enable biologists and computational researchers to explore their complex data in a user-friendly way. The fact of having this tool integrated in the platform will reduce the cognitive load for the experts.

For Cellxgene load, section *4.3.1.2 File search toolbar* explains how users can run it from *4.3 Data browser* over the stored files.

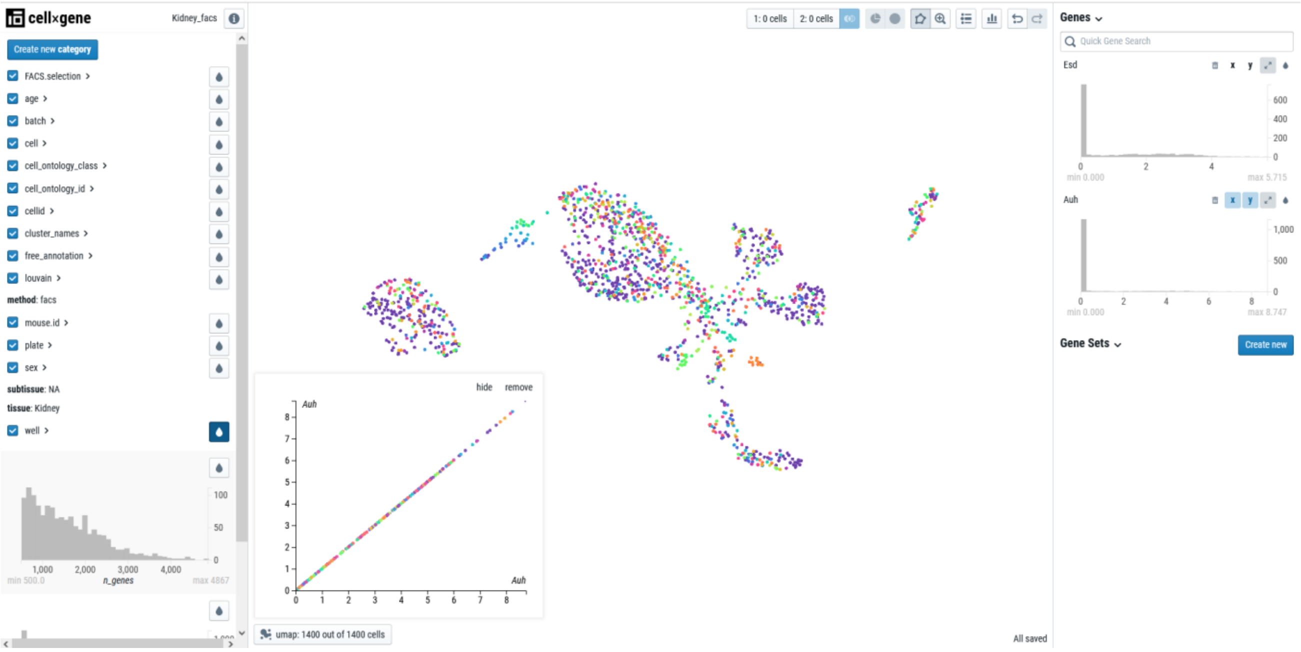


Figure 30 - scRNAseq matrix information visualized with Cellxgene

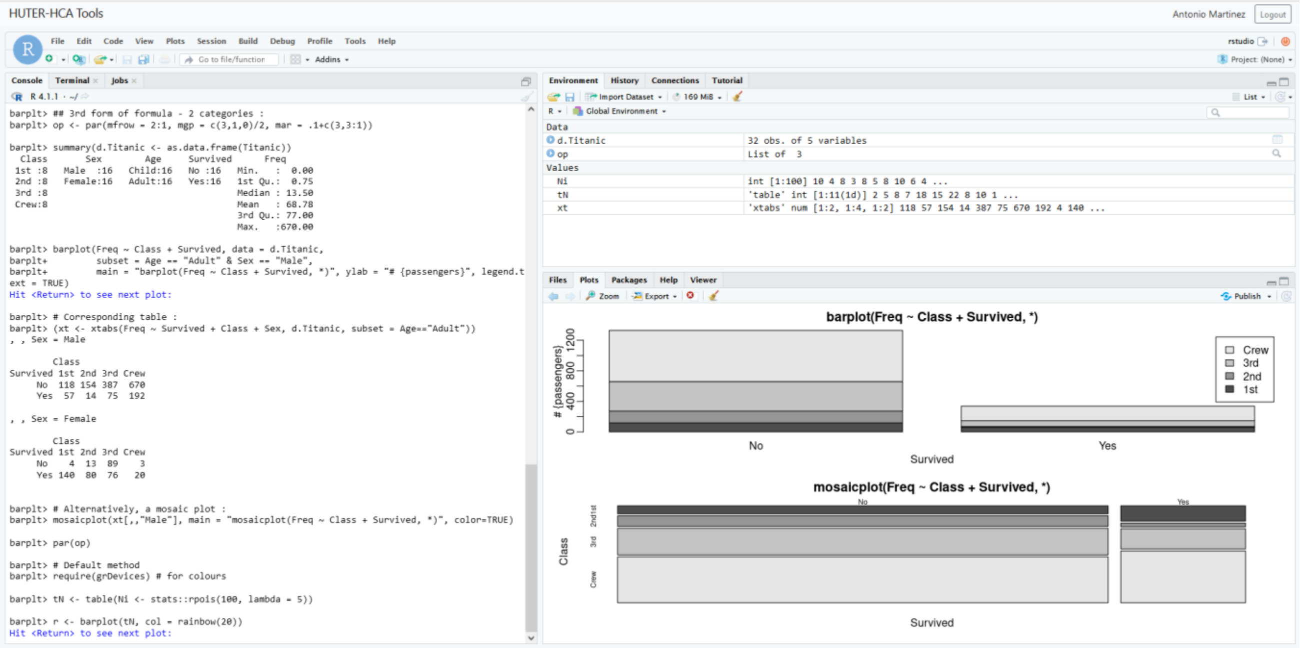
For dissemination purpose, during the final steps of the HUTER Project, a Cellxgene could be publicly available for exploring the assays outcomes.

4.4.2. R Studio

Although the platform allows heavy file processing (such as sequencing files and others) in parallel through WDL and Cromwell pipelines, researchers require to perform certain data exploration and statistical analysis over the result of that sequencing files. For these purposes, R is the most widely programming language used in bioinformatics by far. R is not only a programming language but also a free software environment for statistical computing and graphics which is supported by the R Foundation for Statistical Computing. There are a lot of features which makes it the tool most used by bioinformatics, some are below:

- R provides a high level of control that is not present in other statistical analysis software. Furthermore, it allows to perform more functions than only statistical analysis due to being a complete programming language.
- R is free and open source for which you do not require a license to use it.
- R has a broad community which provides many built packages to perform analysis.

In this context, RStudio is an integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management. This software is usually used by bioinformaticians in their computers or servers to perform any analysis using R language in an easy way. Therefore, the HUTER Platform integrates an open source RStudio Server. This provides a browser-based interface to a version of R running on the platform, bringing the power and productivity of the RStudio integrated development environment to server-based deployments of R (see Figure 31).



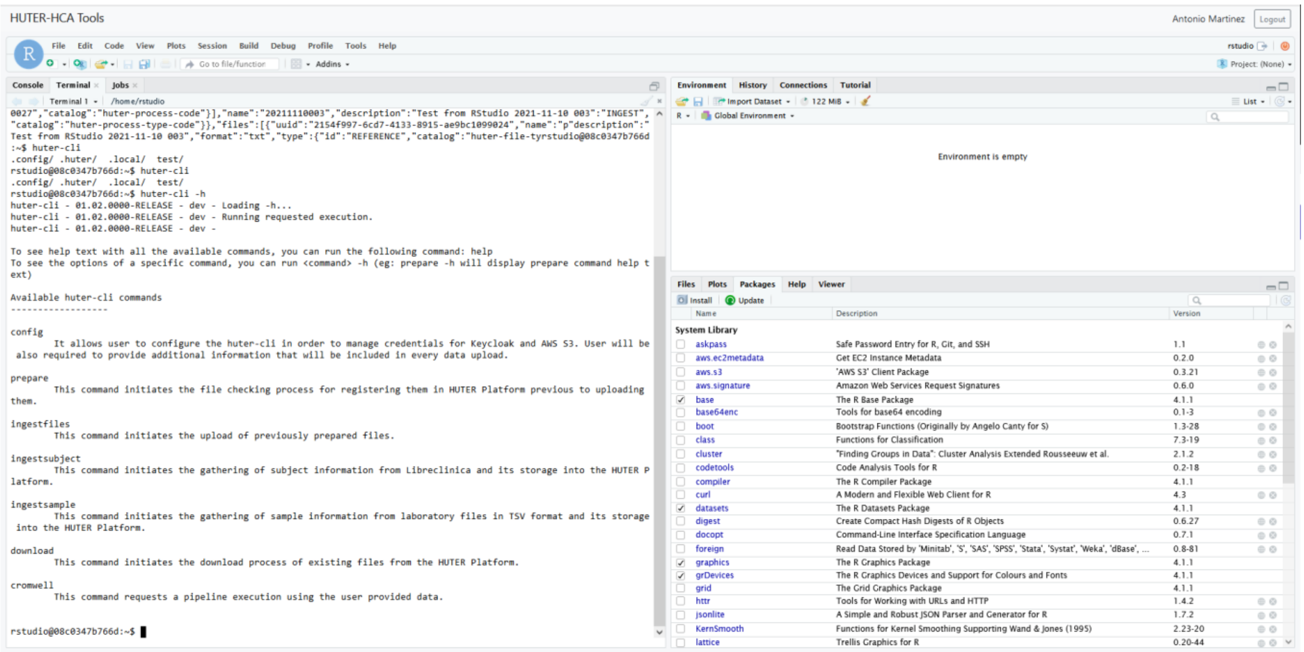
```
barplt> ## 3rd form of formula - 2 categories :
barplt> op <- par(mfrow = 2:1, mgp = c(3,1,0)/2, mar = .1+c(3,3:1))

barplt> summary(d.Titanic <- as.data.frame(Titanic))
  Class       Sex          Age       Survived       Freq
 1st :8   Male  :16   Child:16   No :16   Min.   :  0.00
 2nd :8   Female:16   Adult:16   Yes:16   1st Qu.:  0.75
 3rd :8                                   Median : 13.50
 Crew:8                                   Mean   : 68.78
                                          3rd Qu.: 77.00
                                          Max.   :670.00

barplt> barplot(Freq ~ Class + Survived, data = d.Titanic,
barplt+         subset = Age == "Adult" & Sex == "Male",
barplt+         main = "barplot(Freq ~ Class + Survived, *)", ylab = "# {passengers}", legend.t
ext = TRUE)
Hit <Return> to see next plot:

barplt> # Corresponding table :
barplt> (xt <- xtabs(Freq ~ Survived + Class + Sex, d.Titanic, subset = Age=="Adult"))
, , Sex = Male

        Class
Survived 1st 2nd 3rd Crew
     No  118 154 387  670
     Yes  57  14  75  192

, , Sex = Female

        Class
Survived 1st 2nd 3rd Crew
     No    4  13  89    3
     Yes 140  80  76   20


barplt> # Alternatively, a mosaic plot :
barplt> mosaicplot(xt[,,"Male"], main = "mosaicplot(Freq ~ Class + Survived, *)", color=TRUE)

barplt> par(op)

barplt> # Default method
barplt> require(grDevices) # for colours

barplt> tN <- table(Ni <- stats::rpois(100, lambda = 5))

barplt> r <- barplot(tN, col = rainbow(20))
Hit <Return> to see next plot:
```

```
Data
d.Titanic     32 obs. of 5 variables
op            List of  3
Values
  Ni          int [1:100] 10 4 8 3 8 5 8 10 6 4 ...
  tN          'table' int [1:11(1d)] 2 5 8 7 18 15 22 8 10 1 ...
  xt          'xtabs' num [1:2, 1:4, 1:2] 118 57 154 14 387 75 670 192 4 140 ...
```

Figure 31 - R Studio workspace and functions

R Studio Server can be accessed as seen on section *4.5 Intranet*, just for allowed users, by loading a web interface over a personal workspace. For working over stored files, users have to bring a copy of files to their workspaces using the standard AWS CLI or the aws.s3 package for R Studio, both are accessible from the R Studio Server. The list of AWS CLI functionalities is available at https://docs.aws.amazon.com/cli/latest/userguide/cli-services-s3-commands.html, whereas aws.s3 R package documents are available at https://cran.r-project.org/web/packages/aws.s3/readme/README.html. For getting the proper file location in the cloud storage users can make use of *Data browser* (see section *4.3*) for retrieving the full path of the desired files.

The user workspace is persistent, what means that any user can store data in it by their own access. If an outcome was required to be stored in the platform, it could be ingested using a huter-cli instance which is accessible from the R Studio Server console as shown at *Figure 32 - R Studio with huter-cli, aws-cli and aws.s3 R package*.



```
0027","catalog":"huter-process-code"}],"name":"20211110003","description":"Test from RStudio 2021-11-10 003":"INGEST",
"catalog":"huter-process-type-code"}},"files":[{"uuid":"2154f997-6cd7-4133-8915-ae9bc1099024","name":"p"description":"
Test from RStudio 2021-11-10 003","format":"txt","type":{"id":"REFERENCE","catalog":"huter-file-tyrstudio@08c0347b766d
:~$ huter-cli
.config/ .huter/  .local/  test/
rstudio@08c0347b766d:~$ huter-cli
.config/ .huter/  .local/  test/
rstudio@08c0347b766d:~$ huter-cli -h
huter-cli - 01.02.0000-RELEASE - dev - Loading -h...
huter-cli - 01.02.0000-RELEASE - dev - Running requested execution.
huter-cli - 01.02.0000-RELEASE - dev -

To see help text with all the available commands, you can run the following command: help
To see the options of a specific command, you can run <command> -h (eg: prepare -h will display prepare command help t
ext)

Available huter-cli commands
------------------

config
        It allows user to configure the huter-cli in order to manage credentials for Keycloak and AWS S3. User will be
 also required to provide additional information that will be included in every data upload.

prepare
        This command initiates the file checking process for registering them in HUTER Platform previous to uploading
them.

ingestfiles
        This command initiates the upload of previously prepared files.

ingestsubject
        This command initiates the gathering of subject information from Libreclinica and its storage into the HUTER P
latform.

ingestsample
        This command initiates the gathering of sample information from laboratory files in TSV format and its storage
 into the HUTER Platform.

download
        This command initiates the download process of existing files from the HUTER Platform.

cromwell
        This command requests a pipeline execution using the user provided data.


rstudio@08c0347b766d:~$ █
```

	Name	Description	Version
	System Library		
□	askpass	Safe Password Entry for R, Git, and SSH	1.1
□	aws.ec2metadata	Get EC2 Instance Metadata	0.2.0
□	aws.s3	'AWS S3' Client Package	0.3.21
□	aws.signature	Amazon Web Services Request Signatures	0.6.0
☑	base	The R Base Package	4.1.1
□	base64enc	Tools for base64 encoding	0.1-3
□	boot	Bootstrap Functions (Originally by Angelo Canty for S)	1.3-28
□	class	Functions for Classification	7.3-19
□	cluster	"Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al.	2.1.2
□	codetools	Code Analysis Tools for R	0.2-18
□	compiler	The R Compiler Package	4.1.1
□	curl	A Modern and Flexible Web Client for R	4.3
☑	datasets	The R Datasets Package	4.1.1
□	digest	Create Compact Hash Digests of R Objects	0.6.27
□	docopt	Command-Line Interface Specification Language	0.7.1
□	foreign	Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ...	0.8-81
☑	graphics	The R Graphics Package	4.1.1
☑	grDevices	The R Graphics Devices and Support for Colours and Fonts	4.1.1
□	grid	The Grid Graphics Package	4.1.1
□	httr	Tools for Working with URLs and HTTP	1.4.2
□	jsonlite	A Simple and Robust JSON Parser and Generator for R	1.7.2
□	KernSmooth	Functions for Kernel Smoothing Supporting Wand & Jones (1995)	2.23-20
□	lattice	Trellis Graphics for R	0.20-44

Figure 32 - R Studio with huter-cli, aws-cli and aws.s3 R package

4.4.3. Advanced DICOM viewer

Another tool whose interface is integrated in the visual system of the HUTER Platform is the DICOM viewer (expected to be delivered in Deliverable D2.4).

4.4.3.1. Summarized HUTER Imaging proposal

DICOM is a standard for Medical Imaging that was proposed to be extended to research environments in order to boost both fields. Bahia proposal included the transformation of private formats of immunohistochemistry, tissue microarray, multifluorescence and smFISH images to DICOM that is covered in section *5 DIGITALIZATION SOFTWARE*.

Previous named types of images have an extremely high resolution that cannot be managed as a simple image such a photographic camera picture. Moreover, other specific features must be addressed.

For the simplest image type, immunohistochemistry, DICOM defines a quite stable structure based on building a pyramidal structure. Each level of the pyramid contains the same image at different resolution, from a full resolution at the bottom to a smaller one on the top. In order to nimbly support high resolutions, DICOM defines each level structure as a grid of tiles that represents the full image at the pertinent resolution (see Figure 33*)*.

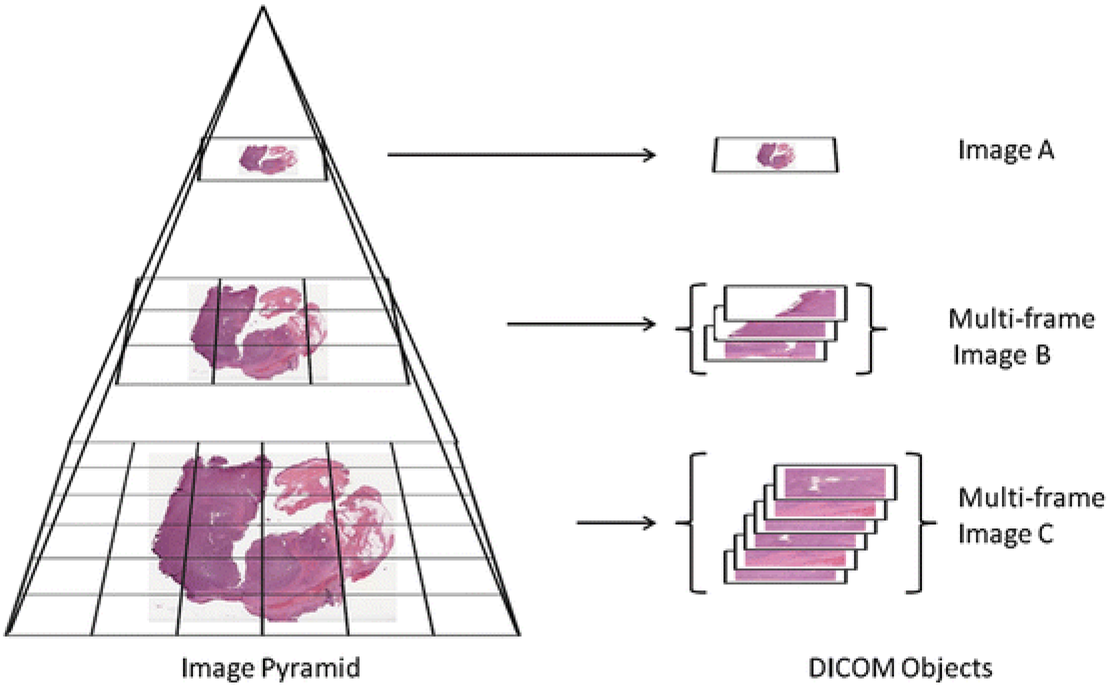


Figure 33 - DICOM pyramidal structure for high resolution imaging

DICOM offered two options for managing each tile position in the pyramid. During the HUTER project, the selected option for the HUTER project was deprecated. Regardless, the current solution keeps on demonstrating that DICOM can be applied for microscopy imaging because the pyramidal imaging structure remains unaltered on the other DICOM option.

Deliverable D2.5 will go in depth about the new DICOM structures that were proposed to DICOM NEMA Organization to support the selected image types. These structures follow the DICOM rules for image building, but they have not been implemented until now. That is why an advanced DICOM Viewer will be required to provide an application for visualizing the expected HUTER images in DICOM format. Besides, this viewer will allow processing these images by advanced functionalities.

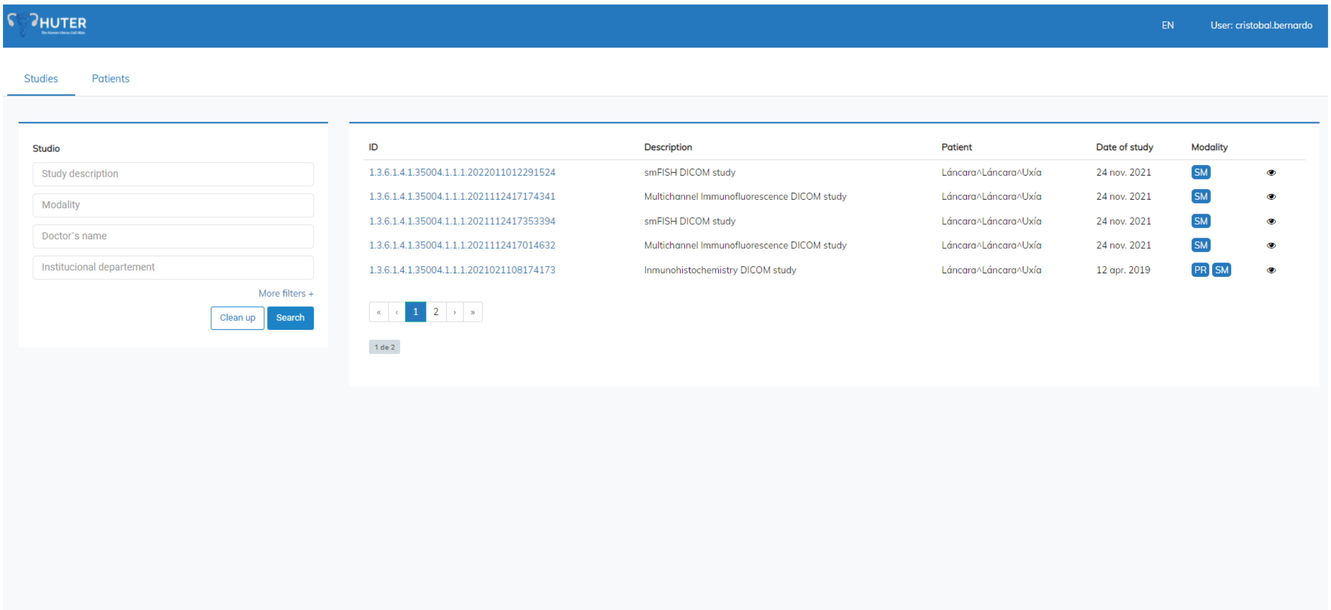
Advanced DICOM Viewer is a solution for visualizing the proposed DICOM formats. The aim of this Viewer is to provide a Proof-of-Concept tool to access new imaging formats based on DICOM keeping a minimal set of advanced tools for imaging analysis in research assays.

4.4.3.2. Advanced DICOM viewer loading

Following the guidelines of the HUTER Platform proposal about a cloud platform for data storage, exploitation and sharing, this viewer will be accessed as a web application. That means that no new software installation must be performed in order to access the transformed images.

There are two ways of loading the DICOM Viewer, both being integrated in the security system of the platform, so only allowed user could run the application.

One way is using the integrated DICOM Study Browser. For accessing this option, users can locate the application in their *Intranet* (see section *4.5*).



ID	Description	Patient	Date of study	Modality	
1.3.6.1.4.1.35004.1.1.1.2022011012291524	smFISH DICOM study	Láncara^Láncara^Uxía	24 nov. 2021	SM	👁
1.3.6.1.4.1.35004.1.1.1.2021112417174341	Multichannel Immunofluorescence DICOM study	Láncara^Láncara^Uxía	24 nov. 2021	SM	👁
1.3.6.1.4.1.35004.1.1.1.2021112417353394	smFISH DICOM study	Láncara^Láncara^Uxía	24 nov. 2021	SM	👁
1.3.6.1.4.1.35004.1.1.1.2021112417014632	Multichannel Immunofluorescence DICOM study	Láncara^Láncara^Uxía	24 nov. 2021	SM	👁
1.3.6.1.4.1.35004.1.1.1.2021021108174173	Inmunohistochemistry DICOM study	Láncara^Láncara^Uxía	12 apr. 2019	PR SM	👁

Figure 34 - Advanced DICOM viewer, study search tool

Users are allowed to browse the DICOM studies in the PACS (the imaging server) in order to locate the desired image using the tool shown at *Figure 34 - Advanced DICOM viewer, study search tool*.

On the other hand, from *Data browser*, users can select a DICOM image register to upload the application through the Launcher menu. This way will open a new instance of the Advanced DICOM Viewer loading the selection DICOM image.

4.4.3.3. Advanced DICOM viewer immunohistochemistry imaging

This section will cover how a user could explore a DICOM image of immunohistochemistry (see Figure 35). This kind of images are digitalized images of samples that are usually analyzed through a microscopy. The requirements of immunohistochemistry fit the DICOM definition introduced in section *4.4.3.1 Summarized HUTER Imaging proposal*, so the application can easily manage the simple pyramidal image by placing suitable tiles at the screen.

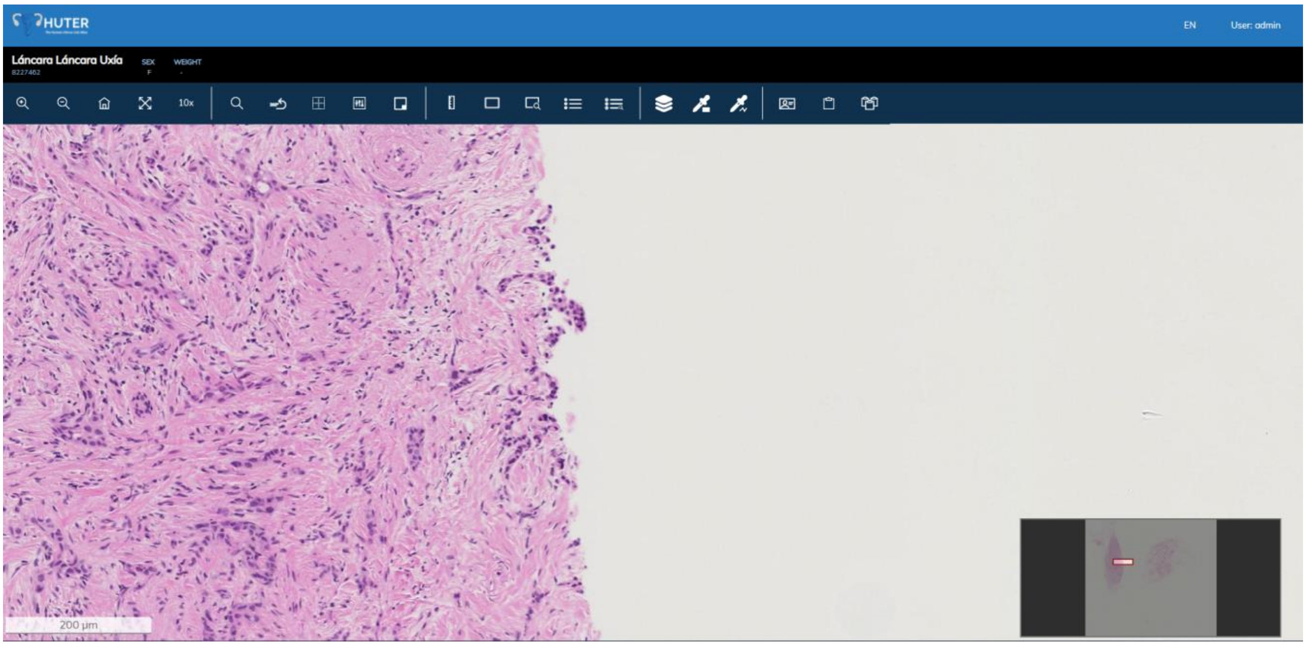


Figure 35 - Advanced DICOM Viewer, immunohistochemistry image

The navigation through this pyramidal structure is a bit complex due to the existence of 3 axis of movement: left-right, up-down, zoom in-zoom out. Each movement requires a continuous recalculation of the displayed tiles as well as retrieving them from the PACS (for further information, see Deliverable D2.4). This issue was solved in order to achieve a smooth navigation assuring good image quality (see Figure 36).

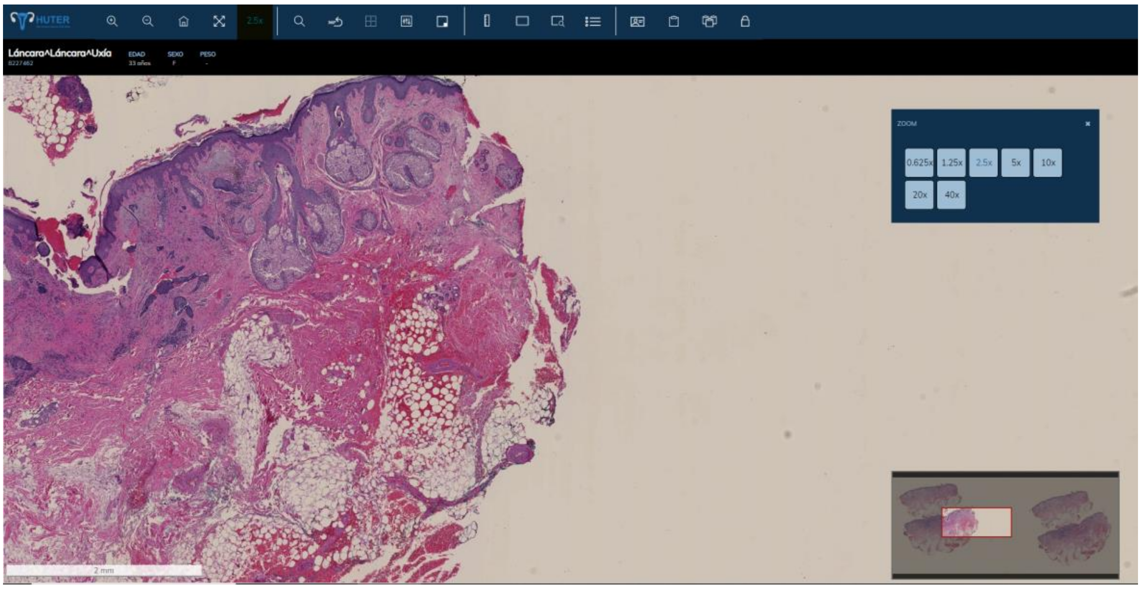


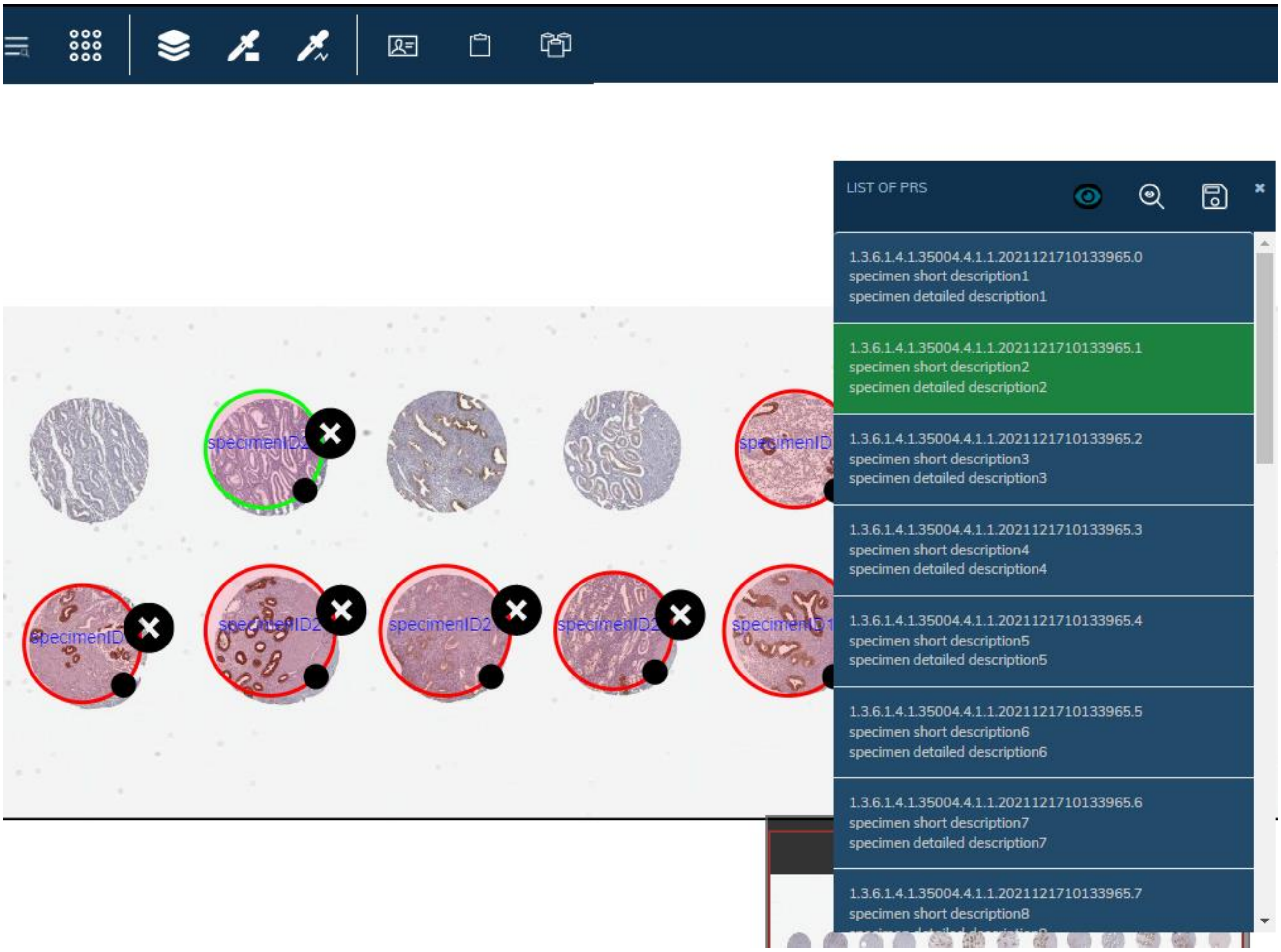
Figure 36 - Advanced DICOM Viewer tools

Apart from image navigation, viewer offers some tools for helping in visual analysis.

- Point transformation: it allows modification of the image points location providing rotation.
- Measures: it allows to calculate distances between points in the images at any zoom level.
- Colour transformation: it allows modification on colour intensity and values that could help users to identify interesting areas.
- Annotations: it allows users to add comments and mark a specific area.
- Region of interest: it allows users to create DICOM ROIs that can be processed by a CAD (Computer Aided Diagnosis).
- Navigation map: it is a widget that allows user to keep track of the visualized areas of the image.
- Metadata: it allows to display patient and sample/specimen data.

4.4.3.4. Advanced DICOM viewer Tissue Micro Array Imaging

Tissue Micro Array, also known as TMA, are images where several specimens share the same container as shown at *Figure 37 - Advanced DICOM Viewer, Tissue MicroArray imaging*.



1.3.6.1.4.1.35004.4.1.1.2021121710133965.0
specimen short description1
specimen detailed description1

1.3.6.1.4.1.35004.4.1.1.2021121710133965.1
specimen short description2
specimen detailed description2

1.3.6.1.4.1.35004.4.1.1.2021121710133965.2
specimen short description3
specimen detailed description3

1.3.6.1.4.1.35004.4.1.1.2021121710133965.3
specimen short description4
specimen detailed description4

1.3.6.1.4.1.35004.4.1.1.2021121710133965.4
specimen short description5
specimen detailed description5

1.3.6.1.4.1.35004.4.1.1.2021121710133965.5
specimen short description6
specimen detailed description6

1.3.6.1.4.1.35004.4.1.1.2021121710133965.6
specimen short description7
specimen detailed description7

1.3.6.1.4.1.35004.4.1.1.2021121710133965.7
specimen short description8

Figure 37 - Advanced DICOM Viewer, Tissue MicroArray imaging

DICOM defines a TMA as an immunohistochemistry image within several metadata registry, one for each specimen in the image. So, HUTER project proposal for TMA had to implement the DICOM definition but also provide solutions for the existing gaps.

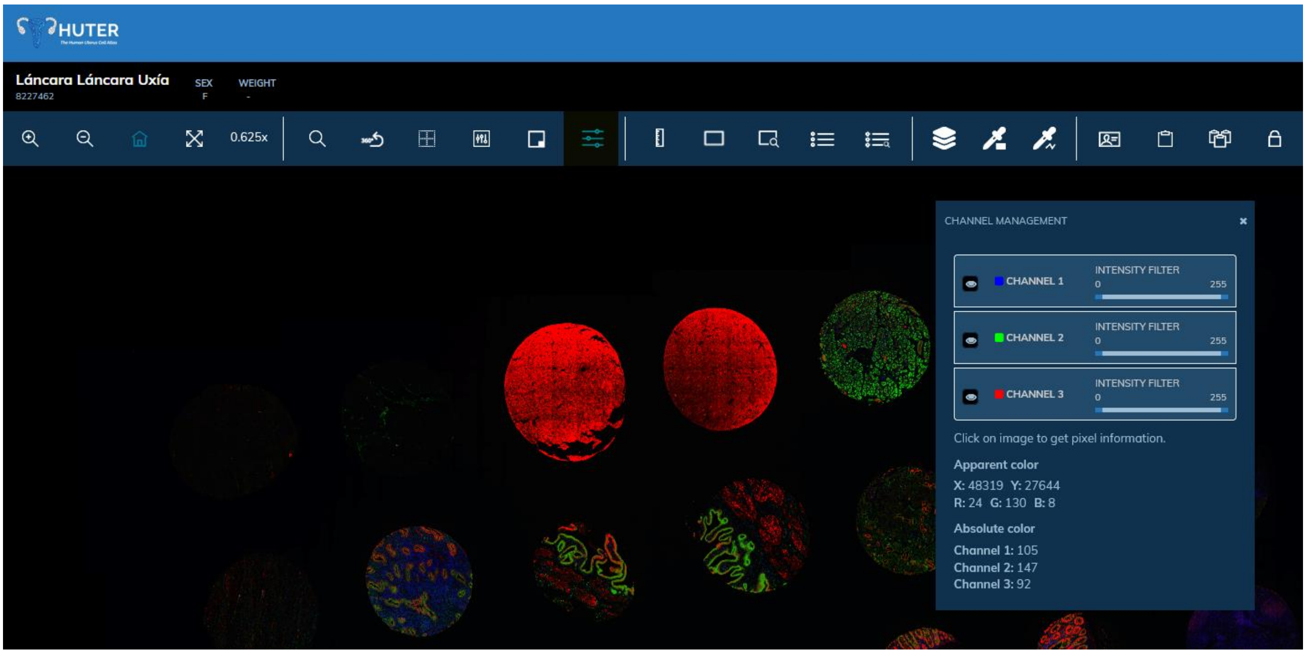
Advanced DICOM viewer is able to understand the TMA definition, based on the immunohistochemistry one but supporting multiple specimen metadata and their location. This last feature required a two-steps process. That is why *DIGITALIZATION SOFTWARE* process creates the object for data storage, but it is the user who eventually locates each specimen on the image through the viewer.

4.4.3.5. Advanced DICOM viewer multispectral imaging

Multispectral images are a special kind of image where the visual information has not been obtained using common light but a specific wavelength. This also means that each pixel value does not represents a colour but an excitation to a specific wavelength. Multispectral images commonly contain a set of images obtained using different wave lengths.

DICOM does not have any definition for this kind of images so HUTER project proposal for supporting it consists of treating multispectral images as a multi-pyramidal structure where each pyramid was the image obtained through a specific wavelength. Besides, additional fields for wavelength identification had to be included in our proposal.

Regarding Advanced DICOM viewer, in order to properly browse these images, it had to introduce an additional axis of movement: the channel or wavelength value. Thanks to this navigation functionality, viewer can display various pyramids at the same time and even overlay them (see Figure 38).



Láncara Láncara Uxía
8227462
SEX: F
WEIGHT: -

CHANNEL MANAGEMENT

	INTENSITY FILTER
🟦 CHANNEL 1	0 – 255
🟩 CHANNEL 2	0 – 255
🟥 CHANNEL 3	0 – 255

Click on image to get pixel information.

Apparent color
X: 48319 **Y:** 27644
R: 24 **G:** 130 **B:** 8

Absolute color
Channel 1: 105
Channel 2: 147
Channel 3: 92

Figure 38 - Advanced DICOM Viewer, multispectral imaging

An additional function that is implemented in the viewer is the identification of channels by colour. Because pixel values do not mean colour, users can set the desired colour for each channel. In that case, pixel value is transformed into colour intensity, so it eases the identification of the pixel in overlayed images.

4.4.3.6. Advanced DICOM viewer smFISH

Single molecule fluorescence in situ hybridization (smFISH) is a powerful technique to study gene expression in single cells due to its ability to detect and quantify individual RNA molecules. As a spatial transcriptomics technique, it is able to achieve spatially resolved RNA profiling of individual cells. Complementary to deep sequencing-based methods, smFISH provides information about the cell-to-cell variation in transcript abundance and the subcellular localization of a given RNA.

smFISH studies allow all Immunofluorescence functionalities and add a new feature to navigate across Z-Axis (see Figure 39).

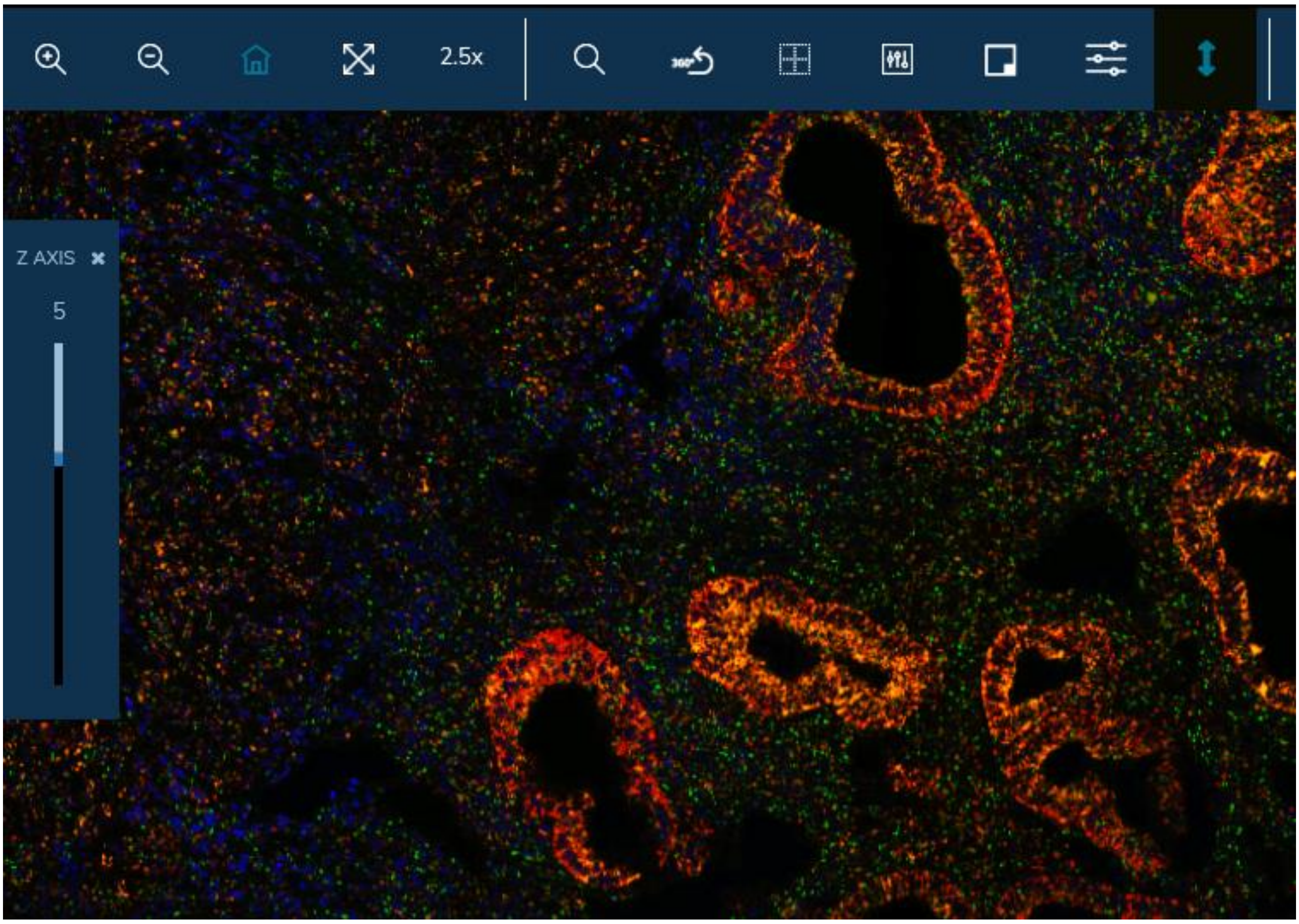


Figure 39 - Advanced DICOM Viewer, smFISH imaging

4.4.4. Genomic and variant call browser

During the first part of the project (explained in Deliverable D2.3) a set of open tools supported by the research community and potential developments were analysed, in close collaboration with HUTER bioinformaticians, in order to figure out if they should be integrated in the HUTER Platform. Among those tools were JBrowse 2 (because is an open viewer for genome files visualization) and the development of a Variant Call Browser (in case of the tool were required by HUTER researchers for variants analysis obtained by genome sequencing).

However, because the HUTER project is more focused on single cell transcriptomic experiments than in genomic studies, it was decided focusing more on the integration of other more required tools for that purpose instead of integrating JBrowse 2 or develop a Variant Call Browser (for instance the Cellxgene that provides scRNA matrices visualization with metadata, the RStudio for custom statistical analysis, the DICOM viewer for microscopy images in DICOM format, and the rest of the tools detailed before).

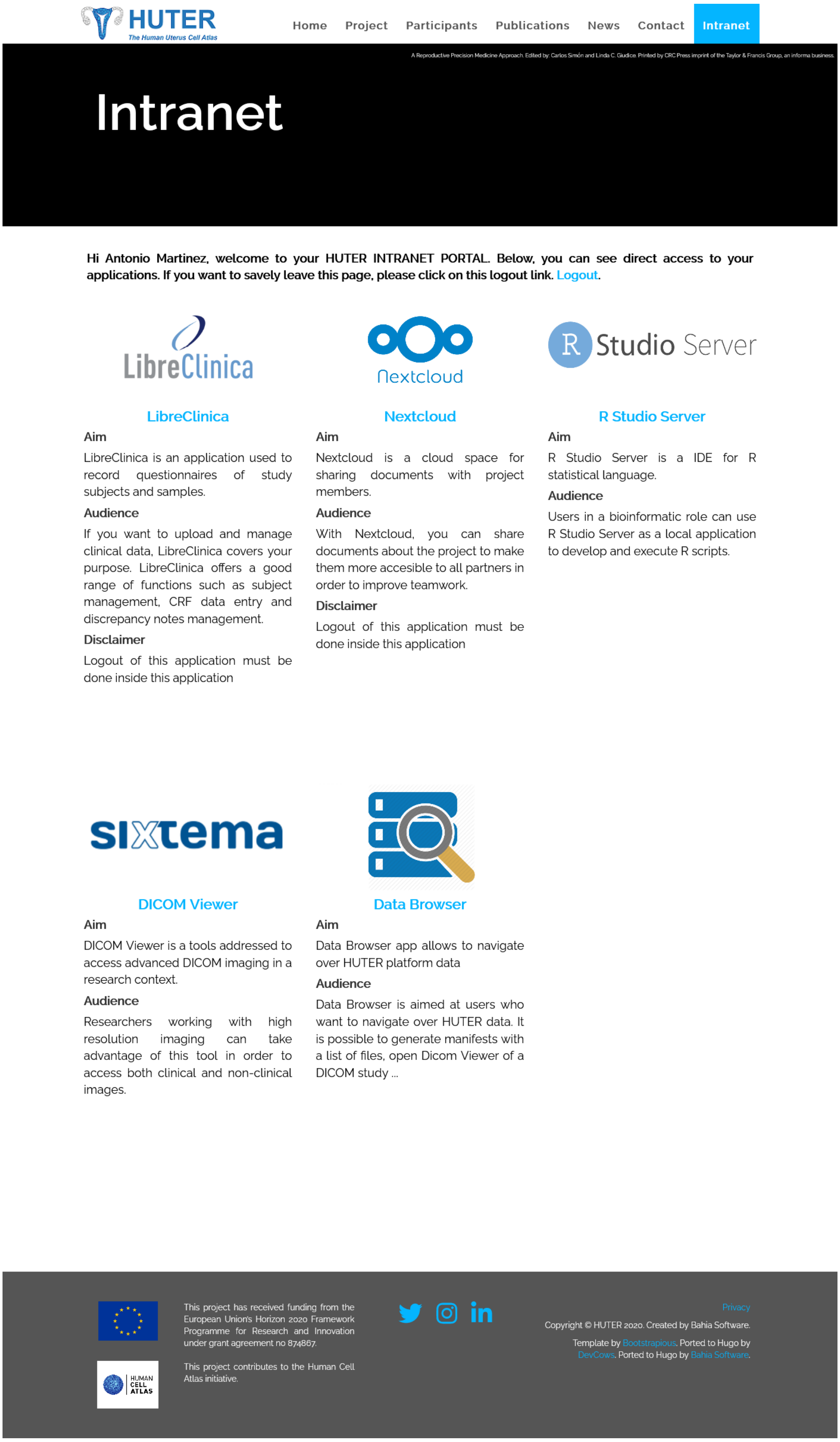
4.5. Intranet

HUTER Platform was designed as a heterogeneous environment where several applications can run independently but also integrated through services. This approach, backed by the AWS infrastructure, makes HUTER Platform a flexible environment that can easily deploy new tools for working with stored data. However, a continuous modification of the tool set could be annoying and confusing for users because they always should be up to date on the tool set.

To reduce the effort of constantly updating tool bookmarks, HUTER Platform provides a dynamic dashboard for loading applications regarding user permissions. With this user custom space all the applications that user could run will be gathered in a page as it is shown at *Figure 40 - Custom Intranet*.

For each application, apart from a link to access it, some useful information will be displayed in order to help users to identify the aim of the tool. For instance, a LibreClinica shortcut is displayed for those users in charge of registering data of subjects who are involved in the assay.

Depending on the logged user, the set of tools will be different and easily modifiable by granting permissions to a user on an application. This flexibility does not require any deployment of the Intranet app, so it boosts the promotion and publication of new applications by reducing the failure point: if users have permissions on an application, they will see it on their Intranet.



HUTER
The Human Uterus Cell Atlas
Home
Project
Participants
Publications
News
Contact
Intranet
A Reproductive Precision Medicine Approach. Edited by: Carlos Simón and Linda C. Giudice. Printed by CRC Press imprint of the Taylor & Francis Group, an informa business.

Intranet

Hi Antonio Martinez, welcome to your HUTER INTRANET PORTAL. Below, you can see direct access to your applications. If you want to savely leave this page, please click on this logout link. Logout.

LibreClinica

Aim

LibreClinica is an application used to record questionnaires of study subjects and samples.

Audience

If you want to upload and manage clinical data, LibreClinica covers your purpose. LibreClinica offers a good range of functions such as subject management, CRF data entry and discrepancy notes management.

Disclaimer

Logout of this application must be done inside this application

Nextcloud

Aim

Nextcloud is a cloud space for sharing documents with project members.

Audience

With Nextcloud, you can share documents about the project to make them more accesible to all partners in order to improve teamwork.

Disclaimer

Logout of this application must be done inside this application

R Studio Server

Aim

R Studio Server is a IDE for R statistical language.

Audience

Users in a bioinformatic role can use R Studio Server as a local application to develop and execute R scripts.

DICOM Viewer

Aim

DICOM Viewer is a tools addressed to access advanced DICOM imaging in a research context.

Audience

Researchers working with high resolution imaging can take advantage of this tool in order to access both clinical and non-clinical images.

Data Browser

Aim

Data Browser app allows to navigate over HUTER platform data

Audience

Data Browser is aimed at users who want to navigate over HUTER data. It is possible to generate manifests with a list of files, open Dicom Viewer of a DICOM study ...

This project has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under grant agreement no 874867.

This project contributes to the Human Cell Atlas initiative.

Privacy

Copyright © HUTER 2020. Created by Bahia Software.

Template by Bootstrapious. Ported to Hugo by DevCows. Ported to Hugo by Bahia Software.

Figure 40 - Custom Intranet

4.6. LibreClinica

The HUTER project requires the collection of human samples and biological and clinical data from the subjects in order to analyze all the parameters to obtain results and conclusions. Therefore, HUTER researchers need to collect the biological and clinical metadata of subjects from different countries where clinicians, researchers and collaborators are collecting this data. This is a similar approach to multicenter clinical trials, where patients of different locations and hospitals are requested to enroll in a study. All the clinical and biological data of patients that is relevant for the clinical trial is usually collected by clinicians in questionnaires called Case Report Forms. As time goes by, new IT solutions were developed to replace CRF on paper sheet by creating eCRF (Electronic Case Report Form) applications. These software eases users to perform validation and communication workflow in a more secure and flexible way.

For those reasons, the HUTER Platform was required to provide one of these tools in order to gather a set of subjects for sample extraction. The HUTER project context also encourage the use of an eCRF application because the focus group should be obtained across different countries in Europe where partners were located.

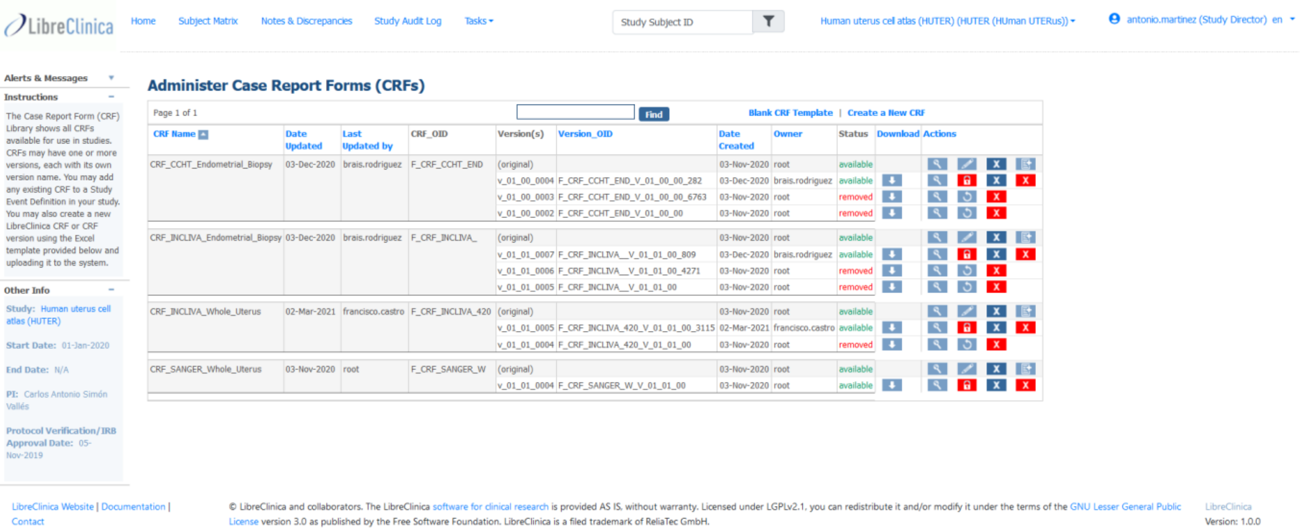
Regarding these requirements, and others that were detailed in Deliverable D2.3, BAHIA provided LibreClinica as eCRF tool integrated in the HUTER Platform. Although in Deliverable D2.4 its functionalities will be detailed, in this document they will be introduced in order to show how users can interact with this application.

Considering LibreClinica is open-source software, it is important to remark that in Deliverable D2.3 a Libreclinica interface modification was described with the aim of strengthen the usability of the application. Those modifications boosted the application with a more intuitive interface allowing users to find the desired data and not on dealing with software options.

Next, main functionalities will be described in order to refresh how this eCRF backs user’s research during the first steps of their assays.

4.6.1. Case Report Form Templates

The main advantage of using LibreClinica as an eCRF applications is the possibility to build CRF templates. Libreclinica eCRF templating allows the creation a dynamic set of question as well as their answers. Those answers can be displayed to the user as simple input fields or more complex ones with data type validation or combo list selection (see Figure 41).



The Case Report Form (CRF) Library shows all CRFs available for use in studies. CRFs may have one or more versions, each with its own version name. You may add any existing CRF to a Study Event Definition in your study. You may also create a new LibreClinica CRF or CRF version using the Excel template provided below and uploading it to the system.

Study: Human uterus cell atlas (HUTER)

Start Date: 01-Jan-2020

End Date: N/A

PI: Carlos Antonio Simón Vallés

Protocol Verification/IRB Approval Date: 05-Nov-2019

Administer Case Report Forms (CRFs)

Page 1 of 1

CRF Name	Date Updated	Last Updated by	CRF_OID	Version(s)	Version_OID	Date Created	Owner	Status	Download	Actions
CRF_CCHT_Endometrial_Biopsy	03-Dec-2020	brais.rodriguez	F_CRF_CCHT_END	(original)		03-Nov-2020	root	available		🔍 ✎ X 📝
				v_01_00_0004	F_CRF_CCHT_END_V_01_00_00_282	03-Dec-2020	brais.rodriguez	available	⬇	🔍 🔒 X X
				v_01_00_0003	F_CRF_CCHT_END_V_01_00_00_6763	03-Nov-2020	root	removed	⬇	🔍 ↺ X
				v_01_00_0002	F_CRF_CCHT_END_V_01_00_00	03-Nov-2020	root	removed	⬇	🔍 ↺ X
CRF_INCLIVA_Endometrial_Biopsy	03-Dec-2020	brais.rodriguez	F_CRF_INCLIVA_	(original)		03-Nov-2020	root	available		🔍 ✎ X 📝
				v_01_01_0007	F_CRF_INCLIVA__V_01_01_00_809	03-Dec-2020	brais.rodriguez	available	⬇	🔍 🔒 X X
				v_01_01_0006	F_CRF_INCLIVA__V_01_01_00_4271	03-Nov-2020	root	removed	⬇	🔍 ↺ X
				v_01_01_0005	F_CRF_INCLIVA__V_01_01_00	03-Nov-2020	root	removed	⬇	🔍 ↺ X
CRF_INCLIVA_Whole_Uterus	02-Mar-2021	francisco.castro	F_CRF_INCLIVA_420	(original)		03-Nov-2020	root	available		🔍 ✎ X 📝
				v_01_01_0005	F_CRF_INCLIVA_420_V_01_01_00_3115	02-Mar-2021	francisco.castro	available	⬇	🔍 🔒 X X
				v_01_01_0004	F_CRF_INCLIVA_420_V_01_01_00	03-Nov-2020	root	removed	⬇	🔍 ↺ X
CRF_SANGER_Whole_Uterus	03-Nov-2020	root	F_CRF_SANGER_W	(original)		03-Nov-2020	root	available		🔍 ✎ X 📝
				v_01_01_0004	F_CRF_SANGER_W_V_01_01_00	03-Nov-2020	root	available	⬇	🔍 🔒 X X

LibreClinica Website | Documentation | Contact

© LibreClinica and collaborators. The LibreClinica software for clinical research is provided AS IS, without warranty. Licensed under LGPLv2.1, you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License version 3.0 as published by the Free Software Foundation. LibreClinica is a filed trademark of ReliaTec GmbH.

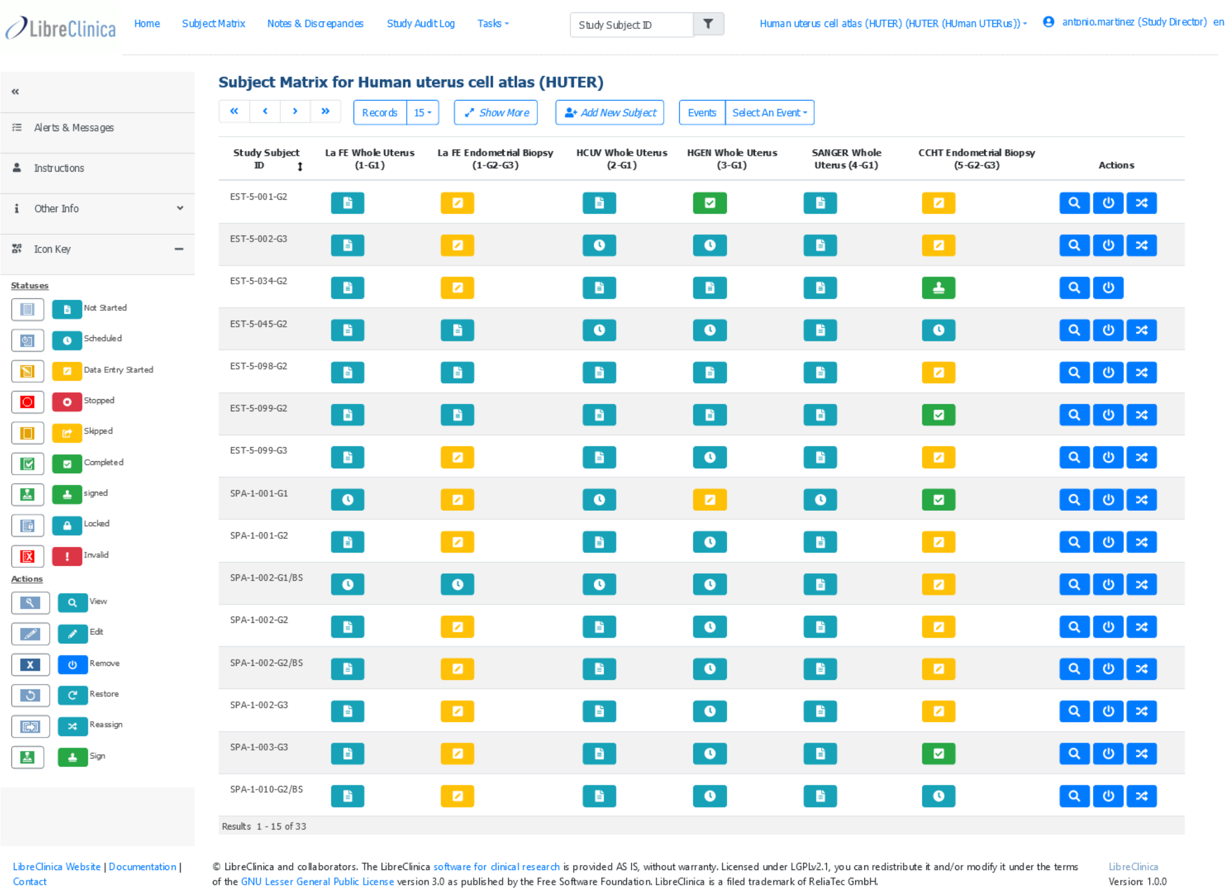
LibreClinica
Version: 1.0.0

Figure 41 - LibreClinica, CRF management

Logically, Electronic CRF are more flexible than in-paper ones because their definition can be easily modified and versioned, what reduces the effort for future assays. Even, if a CRF template is currently in use, its modification can be applied to the already filled instances. Of course, any modification has to be compliant to some data integrity policies that Libreclinica allows to configure.

4.6.2. Subject registration

CRF template are used to gather subject information in a standardized way for a later evaluation. Before that, assay candidates should be identified. So, LibreClinica provides the capacity of making lists of subjects who potentially can be rolled on the assay (see Figure 42).



Subject Matrix for Human uterus cell atlas (HUTER)

Study Subject ID	La FE Whole Uterus (1-G1)	La FE Endometrial Biopsy (1-G2-G3)	HCUV Whole Uterus (2-G1)	HGEN Whole Uterus (3-G1)	SANGER Whole Uterus (4-G1)	CCHT Endometrial Biopsy (5-G2-G3)	Actions
EST-5-001-G2	Not Started	Data Entry Started	Not Started	Completed	Not Started	Data Entry Started	View Remove Reassign
EST-5-002-G3	Not Started	Data Entry Started	Scheduled	Scheduled	Not Started	Data Entry Started	View Remove Reassign
EST-5-034-G2	Not Started	Data Entry Started	Not Started	Not Started	Not Started	signed	View Remove
EST-5-045-G2	Not Started	Not Started	Scheduled	Scheduled	Not Started	Scheduled	View Remove Reassign
EST-5-098-G2	Not Started	Not Started	Not Started	Not Started	Not Started	Data Entry Started	View Remove Reassign
EST-5-099-G2	Not Started	Not Started	Not Started	Not Started	Not Started	Completed	View Remove Reassign
EST-5-099-G3	Not Started	Data Entry Started	Not Started	Scheduled	Not Started	Data Entry Started	View Remove Reassign
SPA-1-001-G1	Scheduled	Data Entry Started	Scheduled	Data Entry Started	Scheduled	Completed	View Remove Reassign
SPA-1-001-G2	Not Started	Data Entry Started	Not Started	Scheduled	Not Started	Data Entry Started	View Remove Reassign
SPA-1-002-G1/BS	Scheduled	Scheduled	Scheduled	Scheduled	Not Started	Data Entry Started	View Remove Reassign
SPA-1-002-G2	Not Started	Data Entry Started	Not Started	Scheduled	Not Started	Data Entry Started	View Remove Reassign
SPA-1-002-G2/BS	Not Started	Data Entry Started	Not Started	Scheduled	Not Started	Data Entry Started	View Remove Reassign
SPA-1-002-G3	Not Started	Data Entry Started	Not Started	Scheduled	Not Started	Data Entry Started	View Remove Reassign
SPA-1-003-G3	Not Started	Data Entry Started	Not Started	Scheduled	Not Started	Completed	View Remove Reassign
SPA-1-010-G2/BS	Not Started	Data Entry Started	Not Started	Scheduled	Not Started	Scheduled	View Remove Reassign

Results 1 - 15 of 33

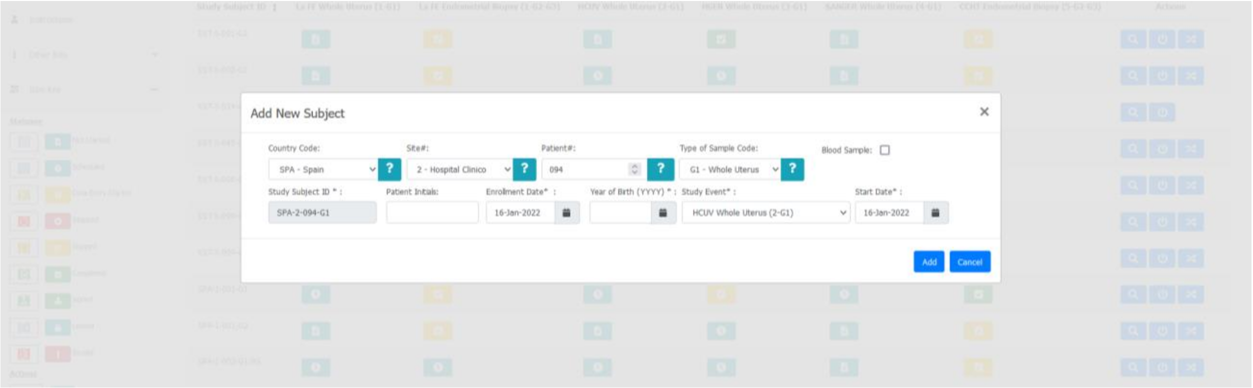
LibreClinica Website | Documentation | Contact

© LibreClinica and collaborators. The LibreClinica software for clinical research is provided AS IS, without warranty. Licensed under LGPLv2.1, you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License version 3.0 as published by the Free Software Foundation. LibreClinica is a filed trademark of ReliaTec GmbH.

LibreClinica Version: 1.0.0

Figure 42 - LibreClinica, full subject matrix

BAHIA developed an ad-hoc interface in LibreClinica to ease the assignment of an assay code for every candidate. Those codes follow the study format code proposed to the committee and must be unaltered when they are referenced. This code assures the anonymity of the candidates during the whole project execution. So, it allows researchers to access candidate's medical data without knowing them (see Figure 43).



Add New Subject

Country Code: SPA - Spain

Site#: 2 - Hospital Clinico

Patient#: 094

Type of Sample Code: G1 - Whole Uterus

Blood Sample: □

Study Subject ID * : SPA-2-094-G1

Patient Initials:

Enrollment Date* : 16-Jan-2022

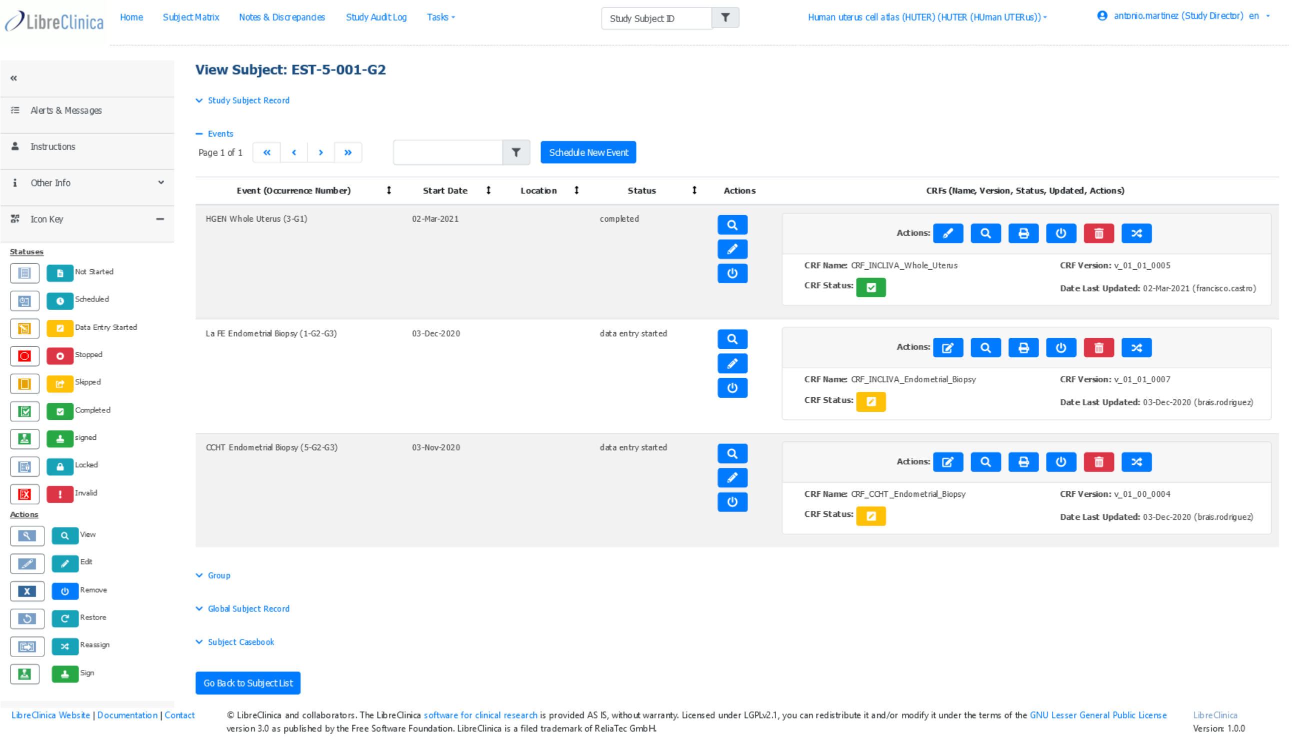
Year of Birth (YYYY) * :

Study Event* : HCUV Whole Uterus (2-G1)

Start Date* : 16-Jan-2022

Figure 43 - LibreClinica, subject code selector for HUTER Project

Although HUTER Project did not expect to repeat interview or to select a subject for more than one kind of assay, LibreClinica is able to manage the assignment of several eCRF to the same candidate. Besides, it eases the scheduling of interviews in date and time (see Figure 44).



View Subject: EST-5-001-G2

Event (Occurrence Number)	Start Date	Location	Status	Actions	CRFs (Name, Version, Status, Updated, Actions)
HGEN Whole Uterus (3-G1)	02-Mar-2021		completed		Actions: CRF Name: CRF_INCLIVA_Whole_Uterus CRF Status: ✓ CRF Version: v_01_01_0005 Date Last Updated: 02-Mar-2021 (francisco.castro)
La FE Endometrial Biopsy (1-G2-G3)	03-Dec-2020		data entry started		Actions: CRF Name: CRF_INCLIVA_Endometrial_Biopsy CRF Status: ✎ CRF Version: v_01_01_0007 Date Last Updated: 03-Dec-2020 (brais.rodriguez)
CCHT Endometrial Biopsy (5-G2-G3)	03-Nov-2020		data entry started		Actions: CRF Name: CRF_CCHT_Endometrial_Biopsy CRF Status: ✎ CRF Version: v_01_00_0004 Date Last Updated: 03-Dec-2020 (brais.rodriguez)

© LibreClinica and collaborators. The LibreClinica software for clinical research is provided AS IS, without warranty. Licensed under LGPLv2.1, you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License version 3.0 as published by the Free Software Foundation. LibreClinica is a filed trademark of ReliaTec GmbH.

LibreClinica
Version: 1.0.0

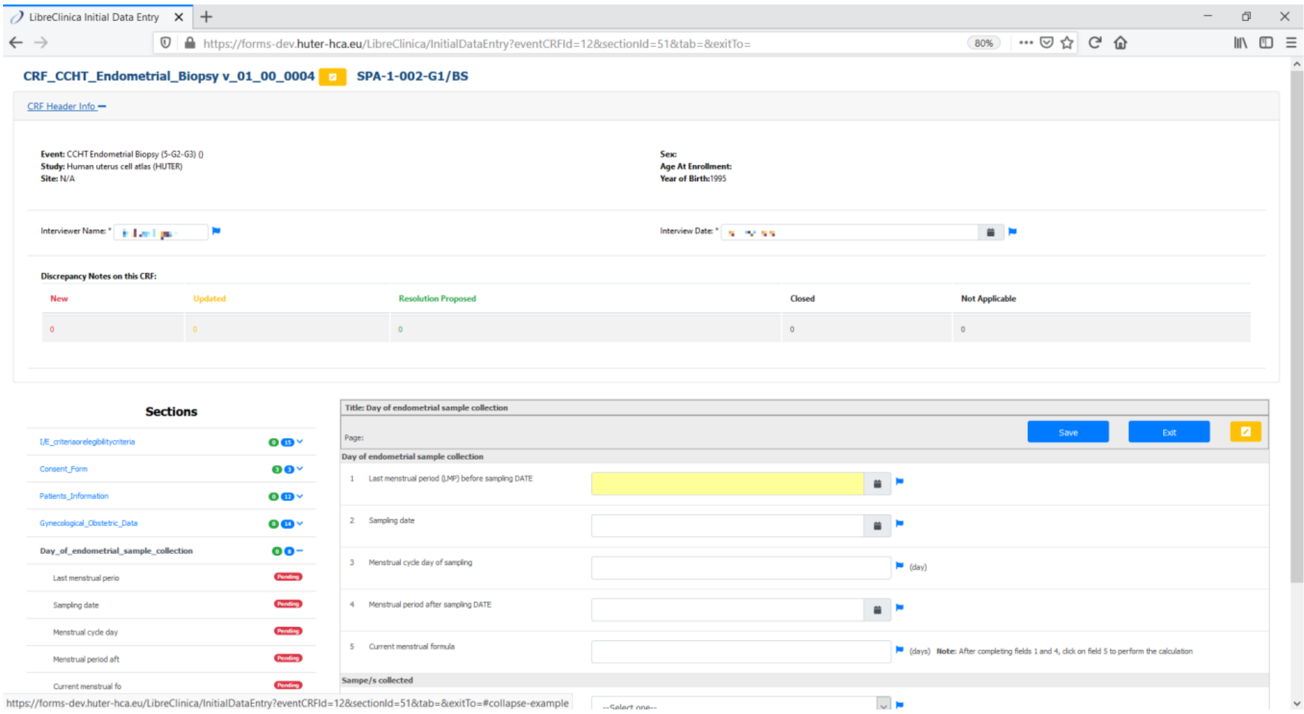
Figure 44 - LibreClinica, subject appointment list

4.6.3. eCRF filling

Once candidates are registered and given an appointment, they are ready to be interviewed. Interviewers can browser or search for an appointment through the main page of the LibreClinica. *Figure 42 - LibreClinica, full subject matrix* shows a subject matrix or table where candidates and CRF are displayed.

Icons are coloured following a legend that was renewed from the original one. By clicking on a button of a non-interviewed candidate and CRF, users can access to the list of appointments of that candidate as Figure *44 - LibreClinica, subject appointment list* shows. HUTER projects expects just one interview per candidate but LibreClinica offers the assignment of several.

Every appointment can be independently managed, so entering one of them a dynamic form with several sheets will be displayed. *Figure 45 - LirbeClinica - Electronic Case Report interface* shows how users can gathered subject data even being informed by validation errors.



LibreClinica Initial Data Entry
https://forms-dev.huter-hca.eu/LibreClinica/InitialDataEntry?eventCRFId=12&sectionId=51&tab=&exitTo=
80%

CRF_CCHT_Endometrial_Biopsy v_01_00_0004 SPA-1-002-G1/BS

CRF Header Info

Event: CCHT Endometrial Biopsy (5-G2-G3) ()
Study: Human uterus cell atlas (HUTER)
Site: N/A

Sex:
Age At Enrollment:
Year of Birth: 1995

Interviewer Name: * [redacted]

Interview Date: * [redacted]

Discrepancy Notes on this CRF:

New	Updated	Resolution Proposed	Closed	Not Applicable
0	0	0	0	0

Sections

I/E_criteriaorelegibilitycriteria	0 15
Consent_Form	3 3
Patients_Information	0 12
Gynecological_Obstetric_Data	0 14
Day_of_endometrial_sample_collection	0 5
Last menstrual perio	Pending
Sampling date	Pending
Menstrual cycle day	Pending
Menstrual period aft	Pending
Current menstrual fo	Pending

Title: Day of endometrial sample collection

Page:

Day of endometrial sample collection

1 Last menstrual period (LMP) before sampling DATE

2 Sampling date

3 Menstrual cycle day of sampling (day)

4 Menstrual period after sampling DATE

5 Current menstrual formula (days) **Note:** After completing fields 1 and 4, click on field 5 to perform the calculation

Sampe/s collected

--Select one--

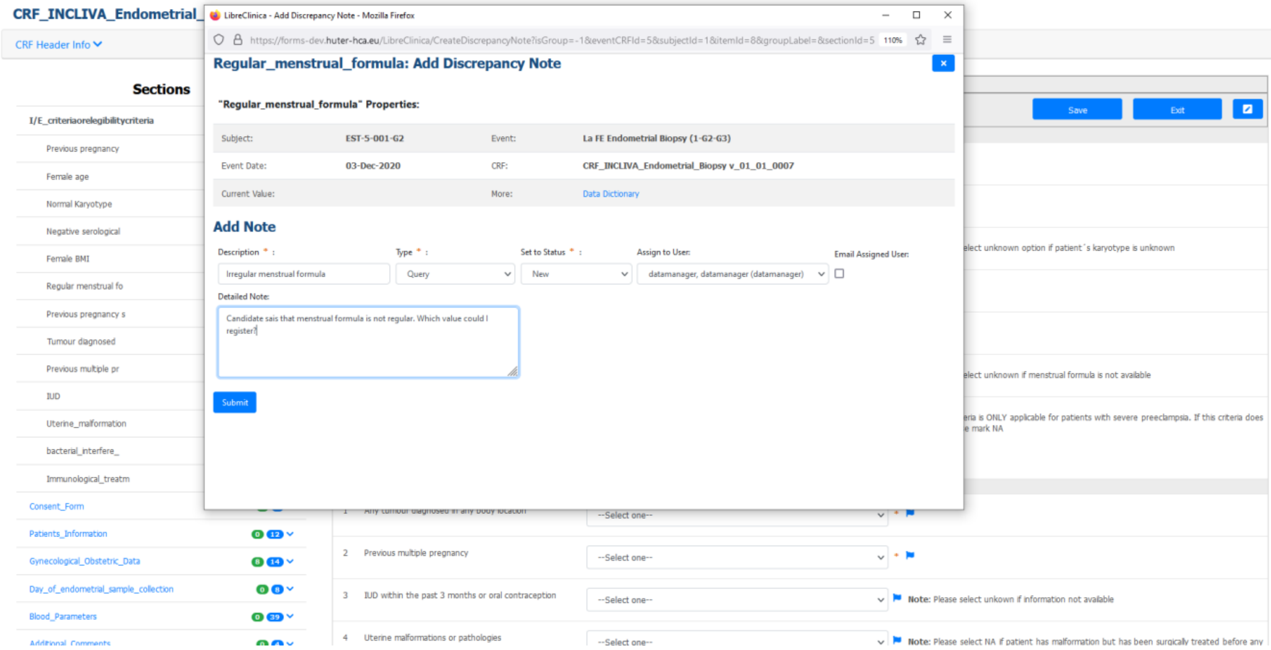
https://forms-dev.huter-hca.eu/LibreClinica/InitialDataEntry?eventCRFId=12&sectionId=51&tab=&exitTo=#collapse-example

Figure 45 - LirbeClinica - Electronic Case Report interface

Please, for further information about LibreClinica, see Deliverables D2.2 and the D2.3.

4.6.4. Data validation and collaborative review

Figure 45 - LirbeClinica - Electronic Case Report interface introduced the possibility of validating data. However, sometimes the interviewer can be sure about the introduced data. To not block the interview, LibreClinica allows users to start a workflow for discrepancy resolution. Interview can save the candidate answer and create and annotation by clicking the red flag and providing some data as shown at *Figure 46 - LibreClinica, discrepancy note creation*.



https://forms-dev.huter-hca.eu/LibreClinica/CreateDiscrepancyNote?isGroup=-1&eventCRFId=5&subjectId=1&itemId=8&groupLabel=&sectionId=5

Regular_menstrual_formula: Add Discrepancy Note

"Regular_menstrual_formula" Properties:

Subject:	EST-5-001-G2	Event:	La FE Endometrial Biopsy (1-G2-G3)
Event Date:	03-Dec-2020	CRF:	CRF_INCLIVA_Endometrial_Biopsy v_01_01_0007
Current Value:		More:	Data Dictionary

Add Note

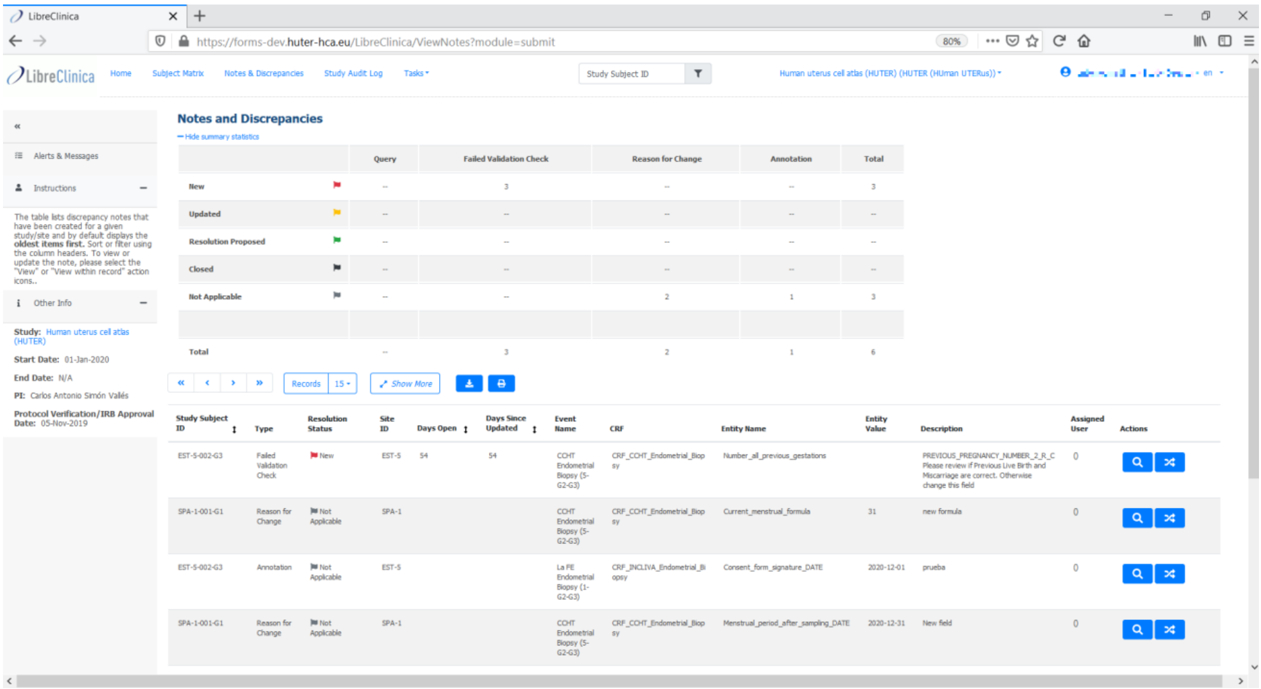
Description * : Irregular menstrual formula
Type * : Query
Set to Status * : New
Assign to User: datamanager, datamanager (datamanager)
Email Assigned User: □
Detailed Note: Candidate sais that menstrual formula is not regular. Which value could I register?

elect unknown option if patient´s karyotype is unknown
elect unknown if menstrual formula is not available
eria is ONLY applicable for patients with severe preeclampsia. If this criteria does e mark NA

1	Any tumour diagnosed in any body location	--Select one--	
2	Previous multiple pregnancy	--Select one--	
3	IUD within the past 3 months or oral contraception	--Select one--	Note: Please select unkown if information not available
4	Uterine malformations or pathologies	--Select one--	Note: Please select NA if patient has malformation but has been surgically treated before any

Figure 46 - LibreClinica, discrepancy note creation

Once the CRF is saved, the person in charge of the assay is now ready to see that discrepancy note and solve. Perhaps validation rule was incorrect, and the inserted value is right, or a clarifying answer can be sent to the interviewer for data curation. This communication can be performed from the Discrepancy Notes interface displayed at *Figure 47 - LibreClinica, discrepancy notes management*.



https://forms-dev.huter-hca.eu/LibreClinica/ViewNotes?module=submit

The table lists discrepancy notes that have been created for a given study/site and by default displays the **oldest items first.** Sort or filter using the column headers. To view or update the note, please select the "View" or "View within record" action icons..

Study: Human uterus cell atlas (HUTER)
Start Date: 01-Jan-2020
End Date: N/A
PI: Carlos Antonio Simón Vallés
Protocol Verification/IRB Approval Date: 05-Nov-2019

Notes and Discrepancies

Hide summary statistics

		Query	Failed Validation Check	Reason for Change	Annotation	Total
New	🟥	--	3	--	--	3
Updated	🟨	--	--	--	--	--
Resolution Proposed	🟩	--	--	--	--	--
Closed	■	--	--	--	--	--
Not Applicable	■	--	--	2	1	3
Total		--	3	2	1	6

Study Subject ID	Type	Resolution Status	Site ID	Days Open	Days Since Updated	Event Name	CRF	Entity Name	Entity Value	Description	Assigned User	Actions
EST-5-002-G3	Failed Validation Check	New	EST-5	54	54	CCHT Endometrial Biopsy (5-G2-G3)	CRF_CCHT_Endometrial_Biopsy	Number_all_previous_gestations		PREVIOUS_PREGNANCY_NUMBER_2_R_C Please review if Previous Live Birth and Miscarriage are correct. Otherwise change this field	0	
SPA-1-001-G1	Reason for Change	Not Applicable	SPA-1			CCHT Endometrial Biopsy (5-G2-G3)	CRF_CCHT_Endometrial_Biopsy	Current_menstrual_formula	31	new formula	0	
EST-5-002-G3	Annotation	Not Applicable	EST-5			La FE Endometrial Biopsy (1-G2-G3)	CRF_INCLIVA_Endometrial_Biopsy	Consent_form_signature_DATE	2020-12-01	prueba	0	
SPA-1-001-G1	Reason for Change	Not Applicable	SPA-1			CCHT Endometrial Biopsy (5-G2-G3)	CRF_CCHT_Endometrial_Biopsy	Menstrual_period_after_sampling_DATE	2020-12-31	New field	0	

Figure 47 - LibreClinica, discrepancy notes management

Regardless, LibreClinica assures data integrity and reliability by providing to tools:

1. The creation of rules over the data to assure right values, ranges or even coherence between different answers.
2. An asynchronous communication system to health data during or after the interviews.

4.6.5. Data access and export

General subject data can be accessed from their registry by searching them in the Task menu as shown at *Figure 48 - LibreClinica, data export*.

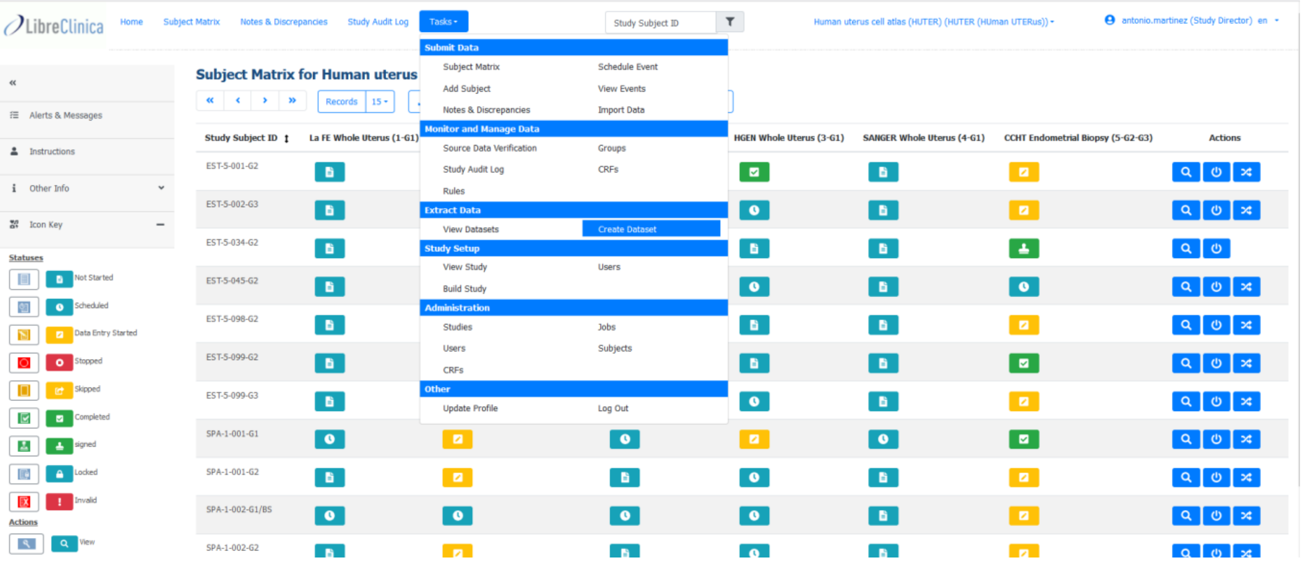
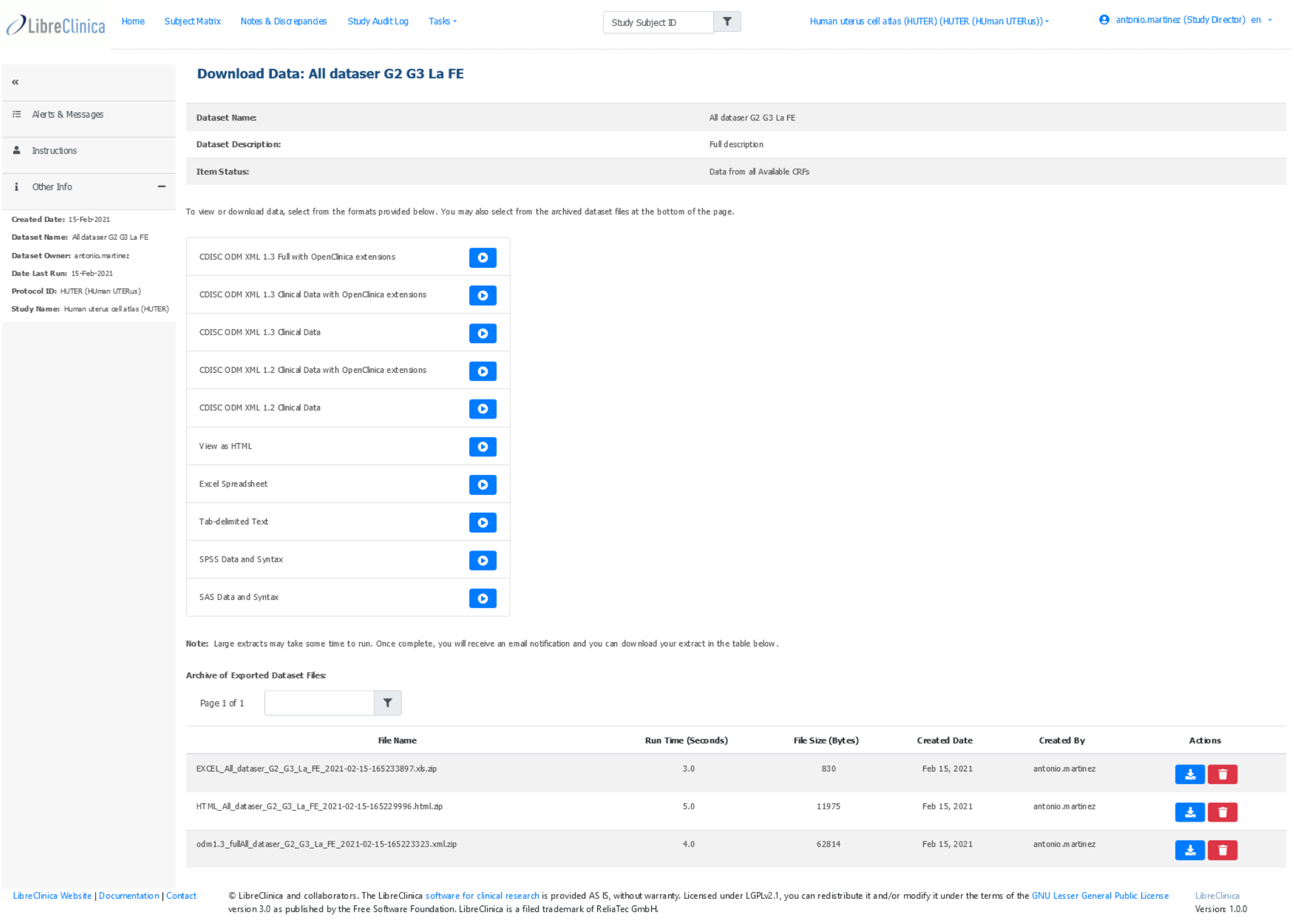


Figure 48 - LibreClinica, data export

Furthermore, the subject matrix allows to access to any subject CRF appointment where more specific subject data is registered. From this page, users can view the data in subject CRF.

According to the researchers needs, subject data is not only needed to checked candidate suitability but also for exploiting it during omics data analysis. That is why CRF answers should be exported to a simple data format in order to process them in bioinformatic pipelines.

Original LibreClinica application already supported the exportation of full or partial CRF data. In HUTER Project scope, a useful file format for processing is tsv but other standard ones are also available as shown at *Figure 49 - LibreClinica, data export available formats*.



Created Date: 15-Feb-2021
Dataset Name: All dataser G2 G3 La FE
Dataset Owner: antonio.martinez
Date Last Run: 15-Feb-2021
Protocol ID: HUTER (HUman UTERus)
Study Name: Human uterus cell atlas (HUTER)

Download Data: All dataser G2 G3 La FE

Dataset Name:	All dataser G2 G3 La FE
Dataset Description:	Full description
Item Status:	Data from all Available CRFs

To view or download data, select from the formats provided below. You may also select from the archived dataset files at the bottom of the page.

- CDISC ODM XML 1.3 Full with OpenClinica extensions
- CDISC ODM XML 1.3 Clinical Data with OpenClinica extensions
- CDISC ODM XML 1.3 Clinical Data
- CDISC ODM XML 1.2 Clinical Data with OpenClinica extensions
- CDISC ODM XML 1.2 Clinical Data
- View as HTML
- Excel Spreadsheet
- Tab-delimited Text
- SPSS Data and Syntax
- SAS Data and Syntax

Note: Large extracts may take some time to run. Once complete, you will receive an email notification and you can download your extract in the table below.

Archive of Exported Dataset Files:

Page 1 of 1

File Name	Run Time (Seconds)	File Size (Bytes)	Created Date	Created By	Actions
EXCEL_All_dataser_G2_G3_La_FE_2021-02-15-165233897.xls.zip	3.0	830	Feb 15, 2021	antonio.martinez	
HTML_All_dataser_G2_G3_La_FE_2021-02-15-165229996.html.zip	5.0	11975	Feb 15, 2021	antonio.martinez	
odm1.3_fullAll_dataser_G2_G3_La_FE_2021-02-15-165223323.xml.zip	4.0	62814	Feb 15, 2021	antonio.martinez	

LibreClinica Website | Documentation | Contact

© LibreClinica and collaborators. The LibreClinica software for clinical research is provided AS IS, without warranty. Licensed under LGPLv2.1, you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License version 3.0 as published by the Free Software Foundation. LibreClinica is a filed trademark of ReliaTec GmbH.

LibreClinica
Version: 1.0.0

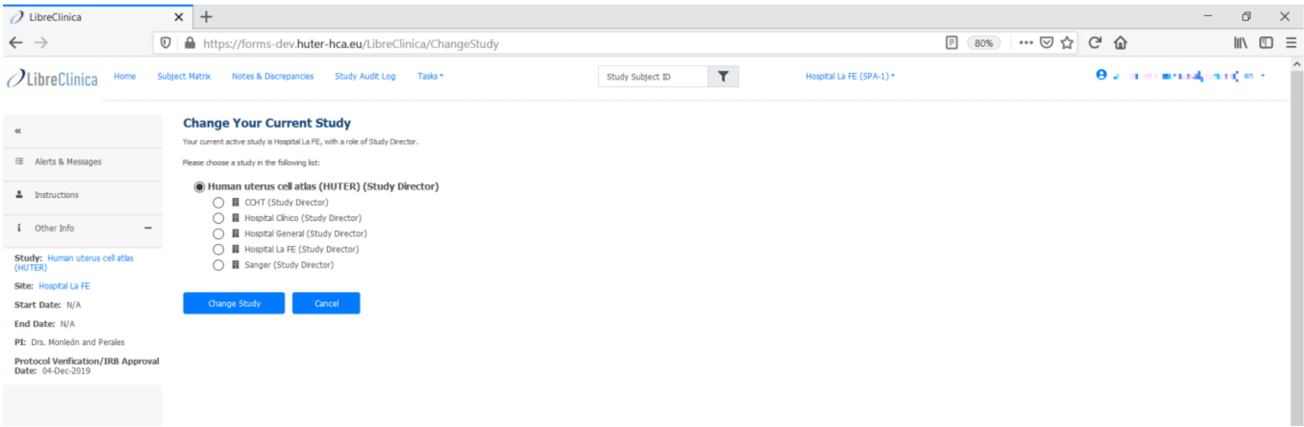
Figure 49 - LibreClinica, data export available formats

4.6.6. Secure interaction

The Libreclinica deployment also provided security features over data that are required for sensible information management. Of course, only allowed users could access to LibreClinica and that was achieved by extending LibreClinica to be integrated with the user management tool of the HUTER Platform. So, the access control is managed just by one tool that works as a Single Sign On for any tools in the platform: Keycloak.

Apart from user management, LibreClinica also provides user role separation. This means that every user has specific grants allowing them to view or modify the minimum required data. This separation of grants assures the data integrity and the proper access to it. For instance, it was introduced the interviewer role and the person in charge role, who is responsible of defining the data gathering protocol implemented in LibreClinica.

Besides, Libreclinica also provides the capacity of group data by different scopes. The highest one is always mandatory and is called “study”. For that scope, HUTER project was selected as the context in which data will be grouped, that means candidates will be registered as HUTER project subjects.



LibreClinica
https://forms-dev.huter-hca.eu/LibreClinica/ChangeStudy

Study: Human uterus cell atlas (HUTER)
Site: Hospital La FE
Start Date: N/A
End Date: N/A
PI: Drs. Monleón and Perales
Protocol Verification/IRB Approval Date: 04-Dec-2019

Change Your Current Study

Your current active study is Hospital La FE, with a role of Study Director.

Please choose a study in the following list:

● **Human uterus cell atlas (HUTER) (Study Director)**
- ○ CCHT (Study Director)
- ○ Hospital Clinico (Study Director)
- ○ Hospital General (Study Director)
- ○ Hospital La FE (Study Director)
- ○ Sanger (Study Director)

Figure 50 - LibreClinica, HUTER Project and Sites

The next available scope is the Site what means a specific place inside the project. This feature allows researchers to delimit the data visibility, so it was applied to the subject registration place. *Figure 50 - LibreClinica, HUTER Project and Sites* shows the different partner's location where subject were registered. Thanks to the site scope, interviewers from one of those sites could not see subject data from other sites.

4.7. Web manager for pipeline execution

In Deliverable D2.2, a tool called Web manager for pipeline execution was proposed for requesting Cromwell executions. However, after some discussions with the bioinformaticians, it was replaced by a new option in the huter-cli, see section 4.2.7 Workflow execution for analysis and processing.

Two main reasons addressed that decision:

1. The complexity of building a completely useful web application for Cromwell executions integrated in the platform.
2. The bioinformatician's preferences for a CLI tool that allows them to integrate the tool in custom scripts.

However, the software design allows the evolution of the platform by delivering a huter-cli evolution to boost its functionalities as well as the deployment of this web interface with the same purpose. This is possible because tools were designed only as data providers and user interface. However, the core of the execution logic is deployed in a service that can be invoked by users from everywhere.

5. DIGITALIZATION SOFTWARE

5.1. Introduction

State-of-the-art research equipment sometimes include their own non-open output format as default which hampers sharing data and results among researchers even from the same institution due to format incompatibilities. This incompatibility issue coupled with the fact that the scientific sector does not usually have well-stablished, defined and adopted standard formats and protocols, raise the need of extending an open standard definition for these output data in order to enable data sharing and compatibility not only among HUTER partners but also among different projects that contribute with data to HCA. In this line, HUTER has included the extension and support of the open DICOM (Digital Imaging and Communication On Medicine) standard already proved in the medical image field to the HUTER advanced research images (as one of its objectives). The use of this standard already guarantees the interoperability and communication among different equipment in hospitals that are manufactured from different companies. Therefore, the implementation of the HUTER imaging types into DICOM standard could foster the transfer of these advanced microscopes and images to hospitals in the future. After a wide analysis in close collaboration with the HUTER partners, the HUTER image types selected to be adapted to DICOM are tissue microarray, immunofluorescence imaging and smFISH. An implementation of these image types to DICOM standard is being developed in the context of Deliverable D2.5 and they will be discussed with the DICOM secretariat, which is the institution who leads the changes and improvements in the standard. However, the aim of this section is to describe the digitalization software developed to transform the HUTER image types into DICOM standard format. This tool demonstrates that the use of the DICOM standard for research images is possible. Of course, the DICOM images will be visualizable through the DICOM viewer that will be delivered in Deliverable D2.4.

From physical samples to digital images of them, imaging is a powerful tool in research assays because it contains lots of information that can be exploited through several techniques: physician’s analysis, segmentation algorithms, artificial view and so on. Research requirements are demanding for technological providers because researchers need the most possible detailed information obtained from samples to study their traits. That is why research imaging is so demanding, because it is expected to provide new sample information. Once an assay publishes its results, sometimes the outcomes are transferred to the Clinical environment as a model or simplification that allows facultative staff to focus on a small set of traits. Regarding this simplification, sometimes Clinical imaging suffers from a lack of information medical imaging format is only focus on some known techniques. Furthermore, research laboratories normally work with private imaging software for their image acquisition processes because it is linked and optimized for a specific imaging hardware and private digital imaging formats.

With the aim of getting closer clinical and research environments, BAHIA proposed the use of medical imaging standards adapted to research environments under the HUTER Project scope. The BAHIA proposal was to perform a proof of concept by applying DICOM imaging to digital image formats usually generated only in research laboratories such as multispectral or smFISH.

Some partners in the HUTER project works with imaging data so that is why BAHIA proposed the transformation of their image formats to DICOM. Due to DICOM has a definition for laboratory imaging, it was taken as a starting point to develop new definitions for our partner's image types: Tissue Micro Array, Multispectral and smFISH.

When talking about image types, we are talking about images obtained using a specific technique that could affect to the sample preparation, the acquisition device or both. However, the resulting digital image is stored in a file using a particular file format. For instance, Tissue Micro Array is an image type containing several samples in the same image, but the file format could be svs, tiff, scn or anyone else depending on the acquisition device, usually private ones. For that reason, the scope of this proposal required to narrow down the target set of imaging file formats to be addressed for the DICOM transformation. So, eventually, HUTER partners agreed to work over:

- Tissue Micro Array images in SVS format from Aperio devices.
- Mulispectral images in CZI format from Zeiss devices.
- smFISH images as a set TIFF files plus an index in XML from Perkin Elmer devices.

5.2. DICOM in a brief

DICOM is an imaging standard for medical purposes managed by NEMA consortium. Its use is spread all over the world and in medical areas such as radiology, laboratory or cardiology. One of the most powerful features of DICOM imaging standard is that it defines structures not only for imaging but also related data and, even also, how to communicate the information among systems.

To summarize, so as to achieve the definition of several structures devoted to specific image types, DICOM builds those definitions by joining smaller data structures. For instance, there is a structure for storing patient information or a sample preparation procedure in a laboratory. These structures belong to already existing DICOM definitions for radiology or laboratory imaging. However, it can also be combined to create new definitions for image types which are not supported yet. Hence, the proposal of BAHIA is to take advantage of these structure to combine them and evolve the required ones.

NEMA consortium is constantly publishing and updating DICOM definitions in order to fit the medical requirement for imaging data. To organize those papers, they publish supplements which are addressed to specific imaging environments and issues. For instance, for laboratory images DICOM has published various supplements such as supplement 122 treating how to store sample data information or supplement 145, which is essential in HUTER project scope. Supplement 145 covers how to build DICOM imaging structures for very-high resolution images like Immunohistochemistry ones generated in pathologic anatomy laboratories. Therefore, BAHIA considered this DICOM supplement of interest for building a compliant solution for research imaging.

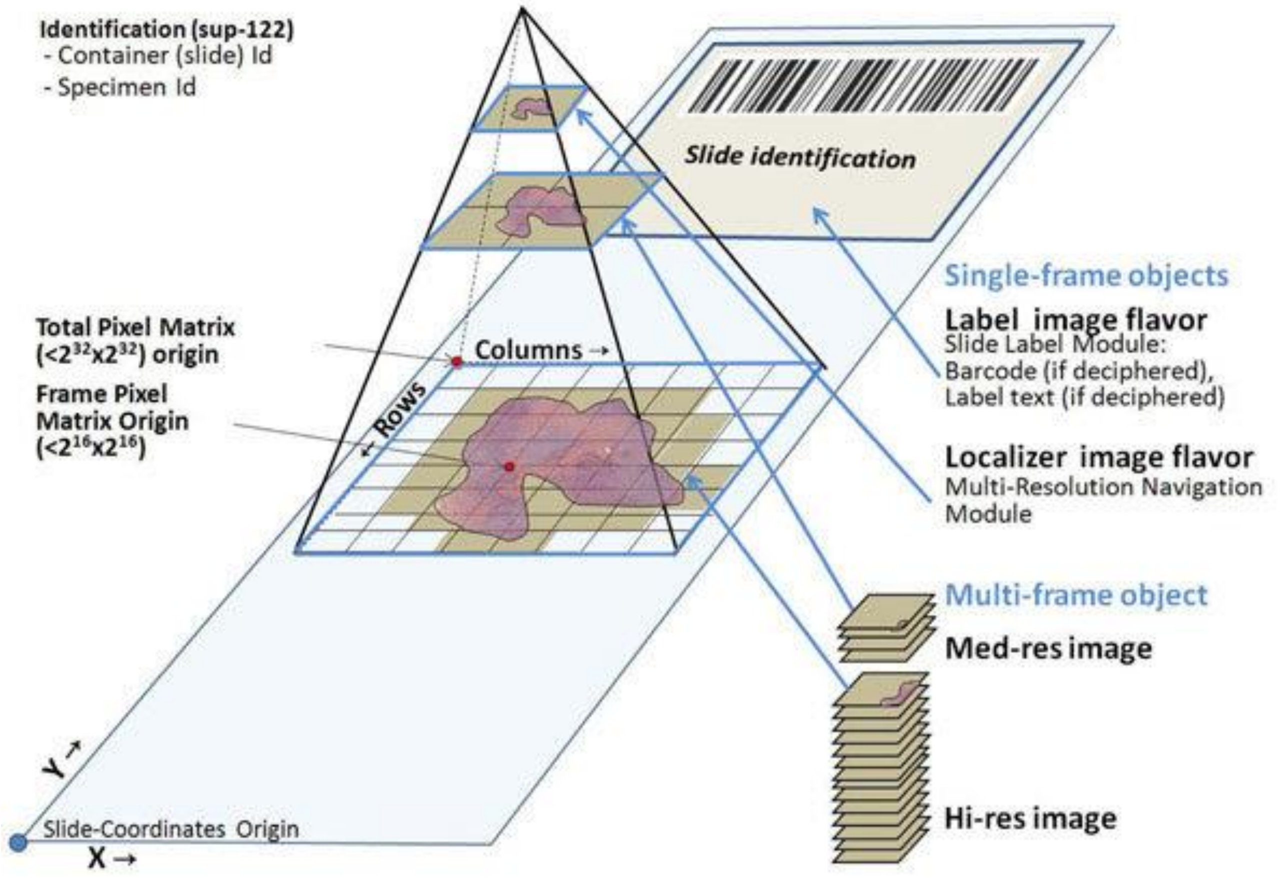


Figure 51 - DICOM laboratory imaging structure

DICOM Supplement 145 cope with high resolution images in the order of 900 million of pixels for a 20x magnification or 3600 Mpixels for 40x ones. For navigating through such amount of information, DICOM defines that these images should be divided into small resolution tiles, in the order of the 1 Mpixel which means 1024 x 1024 pixels. Thanks to that approach, a portion of the image could be displayed by joining a small set of neighbour tiles.

Even though little tiles are easy to handle, thousands of them are not easy to be displayed at the same time by a common computer. Hence, DICOM solves that issue by defining a pyramidal image composed by layers. Each level in the pyramid contains a smaller and tiled copy of the original image. Thanks to this solution, users can firstly access to an overview of the image and make zoom by diving into the pyramid levels. *Figure 51 - DICOM laboratory imaging structure* shows the pyramidal structure and clarifies how the navigation from the top to the bottom can be done in order to get a more detailed portion of the sample image. Consequently, this approach allows to manage oversized images by dividing them into smaller pieces.

Even when research images are expected to be more complex and detailed than medical ones, BAHIA relies on the DICOM supplement 145 definition to support the previously referred kinds of images. It is also important to keep in mind that the main target of the developed software is to proof that DICOM can support research imaging structures, so the adaptation of the biological sample metadata was not covered in this evolution.

5.3. Imaging scope

Regarding to the previous introduction to DICOM standard and how it deals with medical imaging, next it will be shortly exposed why BAHIA has developed a tool for research imaging transformation into DICOM format.

As far as this project was concerned, two years ago (2020) the DICOM definition was not able to fully support the addressed image types: Tissue Micro Array, Multispectral and smFISH. Before continuing, it is required to talk about the special features of these images and why DICOM was not ready to support them. The common factor of all the image types is that they usually result into high resolutions images which can be addressed by applying the solution proposed by BAHIA in the context of HUTER project.

Because the proposal to improve DICOM support for these images will be fully explained in the future Deliverable D2.5, herein we include a brief introduction to the proposal.

5.3.1. Tissue Micro Array

TMA is a common technique used in Pathological Anatomy laboratories because it supports the analysis of several samples in the same container. This means that a digital image of a TMA is, in essence, an immunohistochemistry one which is currently supported by DICOM. Albeit the crucial difference is the need of identifying and locating all the samples in the image by broadening the DICOM definition described in the supplement 122.

So, the BAHIA proposal is to transform this image into a pyramidal image following the supplement 145 like the immunohistochemistry ones. For specimen location, BAHIA proposed the creation of additional DICOM objects called Presentation State, or PR. PR objects allow to enclose a section of the image and assign them a specimen identifier, so any DICOM client such as seen in section *4.4.3 Advanced DICOM viewer* can display and locate specimens in the TMA slide.

5.3.2. Multispectral imaging

Images taken by applying multispectral techniques results in special data structures composed by a set of pictures of the same sample. These techniques are based on preparing the sample with a reagent and taking images of it under different wave-length lights. The reagent will be in a specific way under each wavelength along the different parts of the sample so captured images will display different and complementary information.

Unlike immunohistochemistry and TMA, multispectral imaging was not defined in the DICOM standard. However, there were discussions inside DICOM groups about how to store wavelength information in the metadata, which is an important feature for these images. Those discussions were considered by BAHIA to describe a DICOM structure for multispectral imaging. Once the wavelength storage was partially solved by following the DICOM recommendations, the next point to be addressed were the support of several images in the same DICOM object.

BAHIA designed a multi-pyramidal image from DICOM Supplement 145 where each pyramid contains the information of a wavelength image. Likewise, high resolution images are supported as well as a new axis of information was included: the wavelength channel.

5.3.3. smFISH

Single Molecule Fluorescence In Situ Hybridization (smFISH) is an extremely complex technique that generates oversized images, from few to hundreds Gigabytes. It shares a common trait with the multispectral images: the existence of wavelength channels. However, it also includes information in the z-axis which means that images provide various focus for the same point.

This sort of images were brand new ones when HUTER Project started in terms of standardization and in the medical context so there was scarce information about its technical characteristics. Following the solution applied to *5.3.2 Multispectral imaging*, BAHIA designed a structure to support the transformation of smFISH images into multi-pyramidal images that support high resolution, wavelength information and z-planes. Each trait is represented by a new pyramid in the images that *4.4.3 Advanced DICOM viewer* can display and manage independently or all together.

5.4. Java library

The BAHIA proposal is composed of a theorical paper for DICOM evolution that will be explained in Deliverable D2.5 and its implementation as a Proof of Concept for its feasibility testing. In this section, solved issues during the DICOM transformation tool building will be explained.

The first point to introduce is the set of technologies for developing the dicomization software and why they were selected.

Java was selected as the programming language because it offers a flexible and mature development framework that can be run on multiplatform execution environments. For instance, it can be used to develop from local application to cloud services. However, some drawbacks of Java have to be fixed to achieve an efficient transformation tool for imaging. Java does not provide the fastest runtime but, on the other hand, it is easily deployable on scalable platforms which provides a high throughput for data processing. This means Java offers a reliable parallel processing even when each running process could not be the fastest one.

Furthermore, Java supports lots of libraries for building some of the features required by the DICOM transformation tool such as DICOM structures building. DCM4CHEE java library is used to create and join those small DICOM components that allows to store the proposed DICOM definitions for research imaging. That is, the tiled pyramidal image as well as the minimum metadata for image accessing.

5.4.1. Diagram of components

Next section will introduce a diagram wherein library components are displayed in order to understand how BAHIA has designed the dicomization software. *Figure 52 - Dicomization software components* shows the structure of the main library and its components. Every component has a unique responsibility so they can be configured to run together. By orchestrating some specific components, the dicomization tool can adapt the transformation to read the input image format, write the proper DICOM result format, set the pyramid features and the pixel compression or the provide metadata to be injected.

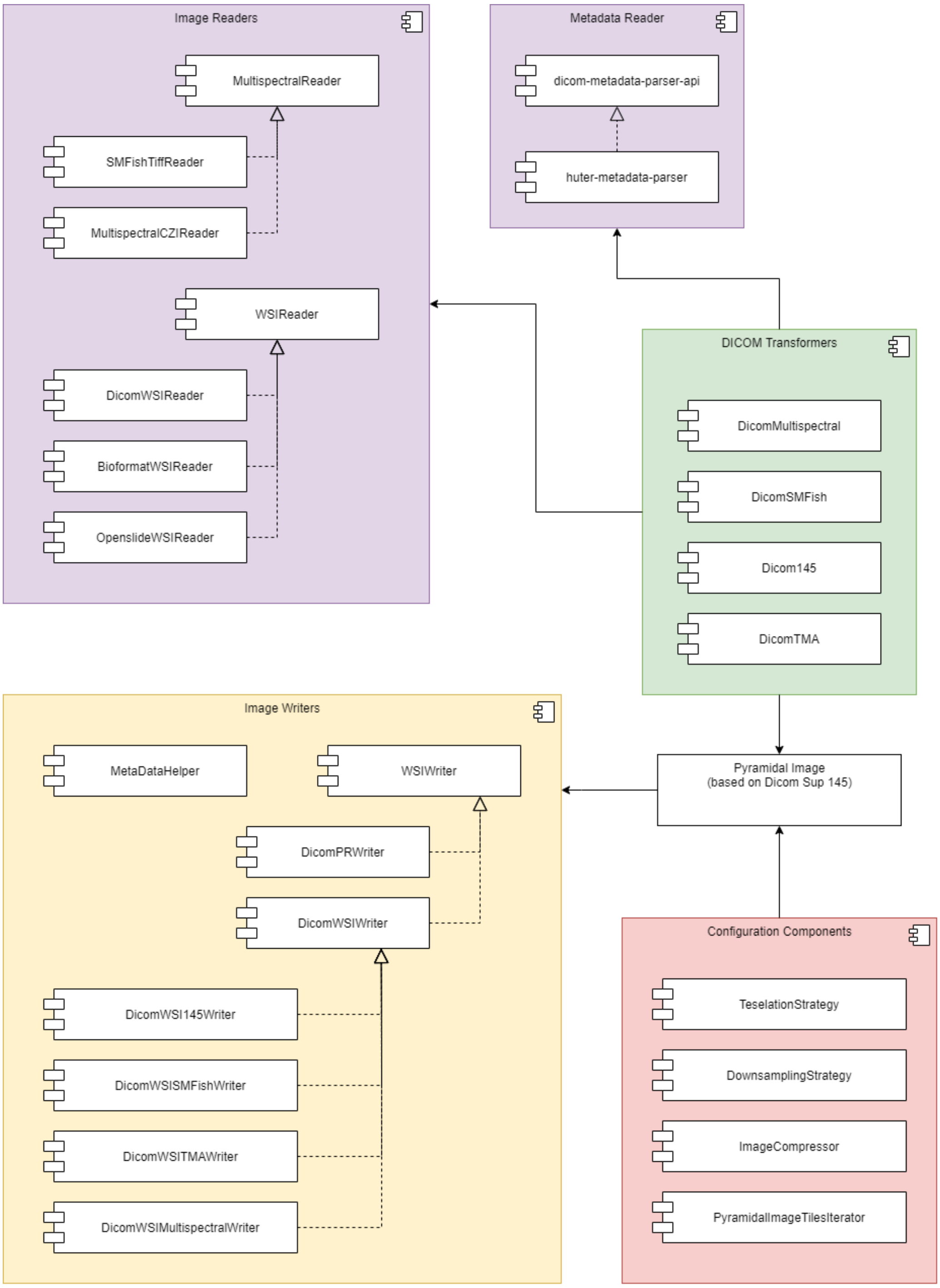


Figure 52 - Dicomization software components

Logically, not all the components can run together. For instance, dicomization tool will not be able to generate a Multispectral image from a Tissue Micro Array one.

In order to better understand the composition of the transformation library, components at *Figure 52 - Dicomization software components* have been grouped by modules following their functionality.

5.4.1.1. *Image readers*

It groups the developed components for accessing images. They are based on the Open Source libraries exposed at section *5.4.2 Reading private imaging formats*, so they take advantage of these libraries for getting the portion of images that will shape each tiled level of the pyramid.

5.4.1.2. *Metadata reader*

It just contains the component that allows the dicomization library to get a set of user-defined data for its later injection in the resulting DICOM image. The provided metadata is expected to be in a well-known DICOM XML format because only exists one implementation. However, thanks to the existence of dicom-metadata-parser-api, other input file format could be developed while they follow the API definition.

5.4.1.3. *Configuration components*

This is a set of algorithms that can run together in order to achieve a specific pyramid structure or process execution.

- TeselationStrategy includes a set of options that allows users to configure the shape and size of the DICOM image tiles. For instances, users can define rectangular tiles with a fixed size or square tiles with a regular size calculated by the library to fit the original image dimension.
- DownsamplingStrategy offers a set of algorithms that defines how many pyramid levels will be created and their resolution. For instance, users can configure each level in the pyramid to be the half of the level below or follow an exponential size reduction. However, this does not mean that tiles size will be changed. To keep an accurate visualization, instead of changing the tile size in downsampled levels the number of tiles is reduced to fit the current level dimensions.
- ImageCompressor is a sort of image format converter that helps readers to provide an input tile in the required format for storage. The reason is that some private images can use image formats not supported by BAHIA visualization tool. Even when DICOM allows image tiles to be stored in well-known formats such as JPEG, TIFF or JPEG2000; it was required to transform some input formats like JPEG 16bit to a more suitable JPEG 8bits that can be displayed on web browser. Besides, other converters could be developed to reach new requirements in further evolutions.
- PyramidalImageTilesIterator. This configuration option does not affect to the image structure but to the transformation process. It can active different versions of the transformations process according to the order of pyramidal tiles generation.

5.4.1.4. *Image Writers*

This set of components was developed to generate files following the pyramidal definition depending on the required image type. For instance, there is at least one writer for each of the image types considered in the BAHIA proposal. Furthermore, a set of MetadataWriter components were developed to adapt the specific metadata to the resulting DICOM image. For instance, unlikely TMA image, multispectral ones must contain the wavelength information in their metadata as well as smFISH contains the reference of z-plane tiles that no other images need.

5.4.1.5. *DicomTransformers*

They are the core of the dicomization processes. Each one of these components are devoted for orchestrating the execution of the rest of configured components. Therefore, BAHIA developed a DICOM transformer for each resulting image type: immunohistochemistry, TMA, multispectral and smFISH. Logically, each transformer can properly run with the suitable components from the other modules.

Dicomization library can be invoked from a command line interface or integrated in any java application. For HUTER project, the command line solution is compliant with the Cromwell executions that will be explained at section *5.6 Dicomization management*.

5.4.2. Reading private imaging formats

One of the reasons for proposing the use of DICOM standard in research environments was the overwhelming expansion of privative imaging solutions in laboratories. These solutions usually consist of hardware that generates digital images using advance techniques and a commercial software for visualization and analysis. Some vendors use digital imaging format that are not understandable by any other visualization software but theirs. If any researcher would like to lend or borrow images among the community, they would not be likely to visualize the images because the proprietary format. So, in order to transform these private formats into DICOM, the first obstacle to deal with has been the possibility to read the original images.

There are some programming libraries for imaging access that are released as Open Source software. This means that they can be of free use by following their soft licenses. Open Source licenses usually restrict their use to non-commercial activities like the HUTER Project, so there is no problem in integrating them into this tool.

Going ahead, the options used to build the components in the Image Readers module were the next libraries:

- Openslide (https://openslide.org/). This is a C library that provides a simple interface to read whole-slide images like TMA or immunohistochemistry ones. It also provides a Java binding which was integrated in the BAHIA dicomization tool as a configurable option for image reading. One of the pluses of Openslide is its speed, providing a high reading rate. On the other hand, the set of suitable image types for reading is quite short regarding to Bioformats option.

- Bioformats (https://www.openmicroscopy.org/bio-formats/). This a library from the OME project which chases the standardization of a common imaging format for laboratories. Despite BAHIA imaging approach, Bioformats is not a rival but an allied. Bioformats software can be used by the BAHIA dicomization tool to read images of several format types. That is why it was integrated as the main image reading option in our tool.

5.5. DICOM storage

DICOM was introduced not only as an image format but also as an imaging communication standard. That is because DICOM defines how images can be retrieved, saved or queried when they are stored in a service compliant with the DICOM definition.

Storage services for medical imaging are usually called PACS (Picture Archive and Communication System). That name clarifies that the deployment of a unified service for imaging storage is a common infrastructure in Clinical environments. PACSs offers not only storage space but an interface for easily locate images by clinical data such as HER number, study number and so on. If the PACS follows DICOM rules, this means that the communication interface could be exploited by the IT community using a well-known protocol.

It is important to mention to this point that the Advanced DICOM viewer exploits the DICOM interface to be able to show the transformed images. Thanks to following a standard interface, this viewer could visualize other DICOM images (for research imaging) whatever the DICOM PACS were deployed.

So, the deployment of a PACS for DICOM imaging storage is supported by the clinical experience, unlike filesystem storage. That is the reason why a DICOM PACS was deployed in the HUTER project scope as the solution for imaging storage. Nevertheless, the common AWS S3 storage solution is used to store the original images from the research labs in order to be the source for their transformation to DICOM.

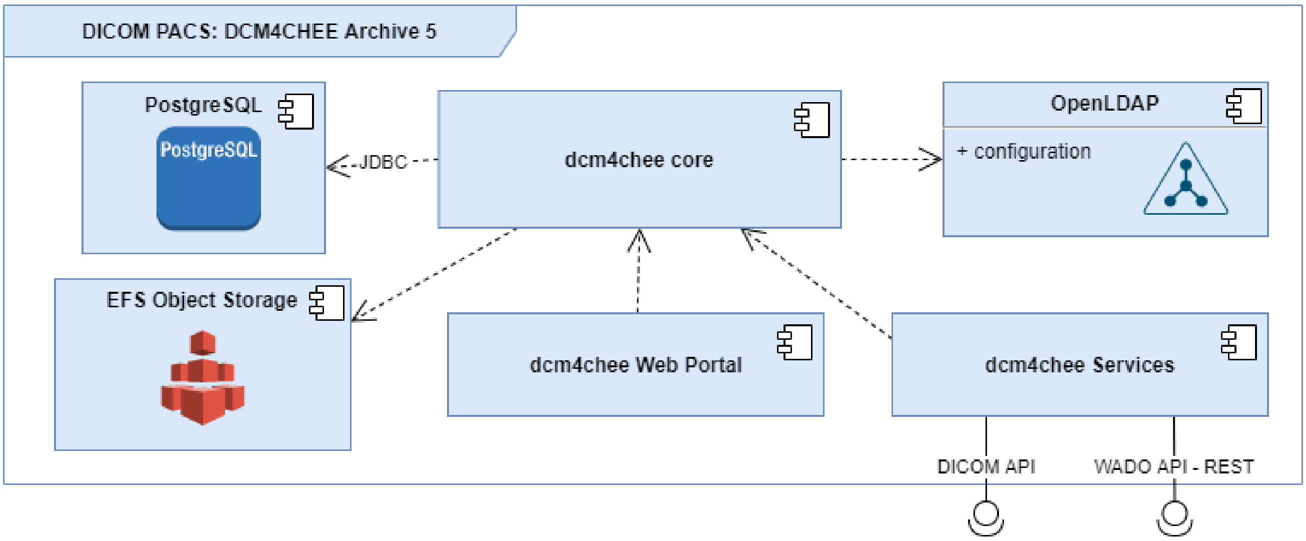


Figure 53 - PACS for DICOM imaging storage

The deployed DICOM PACS solution is the Open Source application called dcm4chee that was introduced at Deliverable 2.3. Figure 53 - PACS for DICOM imaging storage shows the deployed infrastructure for DICOM imaging communication from storage to any DICOM viewer such as in section *4.4.3 Advanced DICOM viewer*.

5.6. Dicomization management

So far, the dicomization tool was introduced as a set of independent tools under the HUTER Platform. However, the BAHIA proposal for the HUTER Platform is to provide a unified cloud platform where tools can be integrated to share their outcomes.

The way to execute heavy processes in the platform while keeping the data in the cloud was introduced in previous deliverables, the Cromwell infrastructure, and shown at *Figure 54 - Processing infrastructure*. Hence, BAHIA developed a workflow using the Cromwell tools to manage the dicomization of the stored images in the AWS S3 service.

Users can request the dicomization by using the huter-cli command exposed in section *4.2.7 Workflow execution for analysis and processing* for running the delivered WDL.

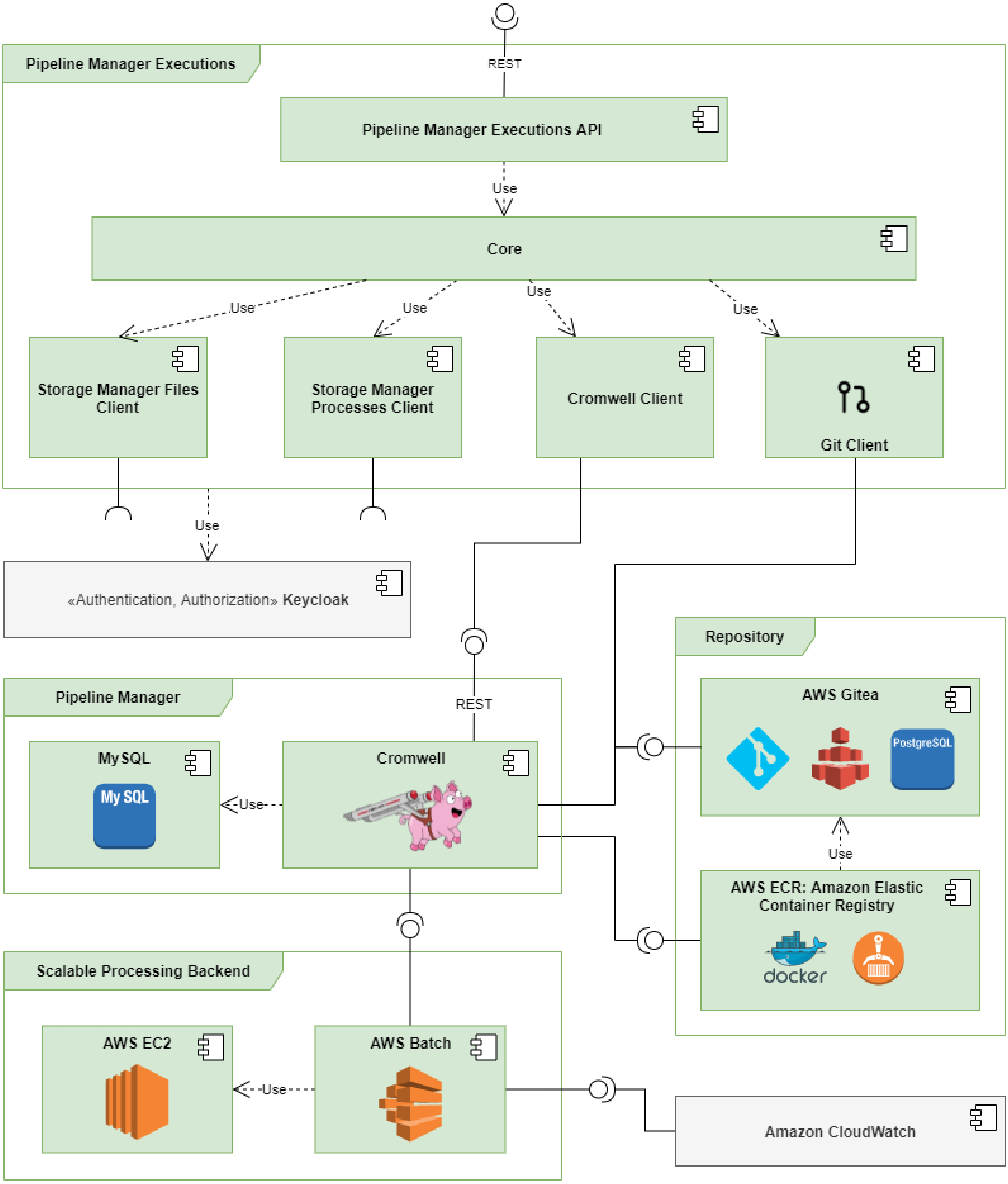

REST
Pipeline Manager Executions
Pipeline Manager Executions API
Use
Core
Use
Use
Use
Use
Storage Manager Files Client
Storage Manager Processes Client
Cromwell Client
Git Client
Use
«Authentication, Authorization» Keycloak
Repository
AWS Gitea
PostgreSQL
Use
AWS ECR: Amazon Elastic Container Registry
docker
Pipeline Manager
REST
MySQL
My SQL
Use
Cromwell
Scalable Processing Backend
AWS EC2
Use
AWS Batch
Amazon CloudWatch

Figure 54 - Processing infrastructure

Besides, a docker image definition was created to automatically build a docker container image which was able to run the dicomization tool over the stored images.