FAIR Sequencing Data Repository based on iRODS

Felipe O. Gutierrez, Paul De Geest, Aldo Jongejan, Sjoerd Repping, J.T. van den Berg, Antoine H.C. van Kampen, Sílvia D. Olabarriaga
Academic Medical Center of the University of Amsterdam - Amsterdam, NL
{f.oliveiragutierrez, p.f.degeest, a.jongejan, s.repping, j.t.vandenberg, a.h.vankampen s.d.olabarriaga}@amc.uva.nl

Diogo F.C. Patrão
A. C. Camargo Cancer Center - São Paulo, Brazil
djogopatrao@gmail.com

ABSTRACT

Research data management (RDM) and the FAIR principles (Findable, Accessible, Interoperable, Reusable) are widely promoted as basis for a shared research data infrastructure. Nevertheless, researchers involved in next generation sequencing (NGS) still lack adequate RDM solutions. The NGS metadata is generally not stored together with the raw NGS data, but kept by individual researchers in separate files. This situation complicates RDM practice. Moreover, the (meta)data does often not meet the FAIR principles [6]. Consequently, a central FAIR-compliant repository is highly desirable to support NGS related research. We have selected iRODS (Rule-Oriented Data management systems) [3] as a basis for implementing a sequencing data repository because it allows storing both data and metadata together. iRODS serves as scalable middleware to access different storage facilities in a centralized and virtualized way, and supports different types of clients. This repository will be part of an ecosystem of RDM solutions that cover complementary phases of the research data life cycle in our organization (Academic Medical Center of the University of Amsterdam). We selected Virtuoso [5] to enrich the metadata from iRODS to enable the management of a triplestore for linked data. The metadata in the *iCat* (iRODS' metadata catalogue) and the ontology in Virtuoso are kept synchronized by enforcement of strict data manipulation policies. We have implemented a prototype to preserve raw sequencing data for one research group. Three iRODS client interfaces are used for different purposes: *Davrods* [4] for data and metadata ingestion, data retrieval; *Metalnx-web* [7] for administration, data curation, and repository browsing; and *iCommands* [2] for all tasks by advanced users. Different user profiles are defined (principal investigator, data curator, repository administrator), with different access rights. New data is ingested by copying raw sequence files and the corresponding metadata file (a *sample sheet*) to the *landing* collection on iRODS. An iRODS rule is triggered by the sample sheet file, which extracts the metadata and registers it to the iCAT as *AVU* (Attribute, Value and Unit). Ontology files are registered into Virtuoso. The sequence files are copied to the *persistent* collection and are made uniquely identifiable based on metadata. All the steps are recorded into a report file that enables monitoring and tracking of progress and faults. Here we describe the design and implementation of the prototype, and discuss the first assessment results. Initial results indicate that the proposed solution is acceptable and fits the researchers workflow well.

Keywords

Next Generation Sequencing, Research Data Management, iRODS, FAIR principles, genomics data, ontology.

INTRODUCTION

As part of modern biomedical research many types of (large) data sets are produced. OMICS experiments - and in particular next generation sequencing (NGS) - are no exception. Biomedical research involving NGS data generally comprises collaborations between clinical departments, laboratories, and data analysis groups that have different cultures and procedures for working with data. With the growth of data, it has become challenging to keep track of data, processes and outcomes of research over long periods of time and across the collaborating units. Adequate

iRODS UGM 2017 June 13-15, 2017, Utrecht, Netherlands

management of research data has become paramount to guaranteeing efficiency and integrity of research, as well as to enable future data reuse and exploitation. Important initiatives concerning OMICS data management are supported by, for example, the BBMRI and ELIXIR programs which are active across Europe. Various of these initiatives promote and highlight the FAIR principles for research data [6]. According to these principles, research data should be Findable, Accessible, Interoperable and Reusable, which are properties that demand great care in terms of collection, annotation and archival. Nevertheless, few solutions exist for NGS data management that accomodate these principles. In practice, often there is no central repository for sequencing data, and each individual researcher is responsible for the storage and backup of their data and metadata. Moreover, NGS related metadata is generally not generated automatically, requiring manual annotation by the researchers. And finally, metadata is typically not stored together with the NGS data.

In our prototype we aim to develop a repository for raw NGS research data for our organization. The repository should adhere to the FAIR principles by defining standardized, minimum, interoperable metadata which needs to be preserved together with the data. iRODS [3] is used as a basis for this solution, in combination with a variety of clients to fulfill the diverse needs of the different user roles. This paper briefly introduces characteristics of NGS data before describing the designed solution. Preliminary results of a pilot case study are presented and discussed.

NGS RESEARCH DATA REPOSITORY

NGS data is generated by sequencing the genetic material within biological samples i.e. measuring the genetic material and thereby defining the order of its nucleotides. An NGS experiment generally comprises the following steps:

- prepared biological samples are sent to a sequencing facility;
- the sequencing facility performs one or more sequencing runs using a sequencing instrument (sequencer);
- the output of these runs are images that are converted to raw sequencing data (this data comprises several million reads). The NGS data is stored in a standard file format, e.g.: **fastq** (FASTQ format is a text-based format for storing both a biological sequence (usually nucleotide sequence) and its corresponding quality scores.).
- information (metadata) about the run(s), sample(s) and output file(s) are described in the so called **sample sheet**, which can take different formats and contain different contents across sequencing platforms and services.

NGS data of such form is being generated in our organization by multiple research groups that, at the moment, need to take care of their own data storage and management necessities. Metadata related to these data are being maintained locally by the respective researchers. Data and metadata are spread in various systems, using non-standardized protocols, making data retrieval, traceability and reuse extremely difficult. Lastly, due to being an academic hospital the (meta)data can be highly sensitive and therefore can not leave the site, excluding any external repositories from being used. In our project we aim to design a repository that can be used by various research groups in our organization which would also be a viable option for similar organizations. In order to follow the FAIR principles we aim to harmonize the format and content of the sample sheets, to represent the metadata in an ontology, and to store the metadata and data together. Note that the metadata associated with a raw NGS dataset is potentially extremely rich, describing the whole process from the biological sample acquisition through the wet lab sample preparation and finally sequencing. Collecting such detailed metadata is not a trivial task for researchers. Different research groups have their own priorities for NGS related metadata, therefore, in this project, we allow each research group to define a subset of the recommended metadata fields, in addition to the minimum of required metadata for an NGS experiment. This flexibility is intended to make the system more user friendly and increase the willingness of the researcher to provide this metadata, which comes at the expense of standardization.

REPOSITORY ARCHITECTURE

The main components of the proposed system are presented in Figure 1: iRODS as the main data and metadata storage server, Virtuoso as the ontology server, the different access points and connection protocols. Openlink

Virtuoso [5] is a hybrid database engine, allowing not only the management of relational databases but also of a *triplestore* for linked data. It provides support for all major ontology formats and allows data-access through the Virtuoso conductor web-interface.

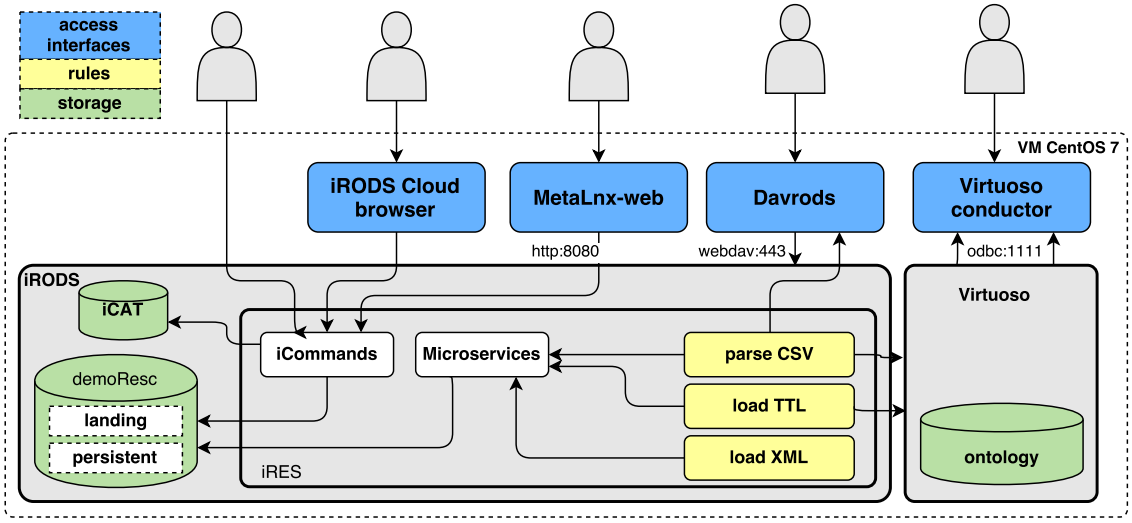

Figure 1. Main components of our FAIR NGS Data repository: iRODS for files and metadata, Virtuoso for RDF store, four types of user interfaces (blue), and rules (yellow). Data and metadata storage resources are depicted in green. A data repository is a collection with two areas called landing (for data ingestion) and persistent (for retrieval). Rules are triggered by new data on the landing area, and use microservices to create and load metadata and ontology.

Within iRODS there are currently two storage resources (the iCat and the data storage space) which are placed in the same storage pool. The data storage space is further divided into repositories that correspond to collections that are potentially owned by different research groups. Each repository has two directories, each with different access types or permissions (*landing* and *persistent*). The landing directory is temporarily used for new data and metadata ingestion and the persistent directory holds data and metadata permanently. Rules in iRODS are used for extracting metadata from files written into the landing directory, loading them into the iCAT and Virtuoso [5], and moving them to the persistent directory.

The users can directly access iRODS in three ways, depending on their needs and knowledge: iCommands [2], Davrods [4] and MetaLnx-web [7]. Davrods [4] is an Apache WebDAV interface that provide access to iRODS. It is a bridge between the WebDAV protocol and the iRODS API, implemented as an Apache HTTPD module. It leverages the Apache server implementation of the WebDAV protocol, mod_dav, for compliance with the WebDAV Class 2 standard. Through Davrods the user can ingest and retrieve data from iRODS, mounting it as a file system in his/her workstation. Therefore, the user cannot work with metadata through this client interface. Metalnx-web [7] is a web application designed to work alongside iRODS. It provides a graphical UI that can help simplify most administration, collection management, and metadata management tasks removing the need to memorize the long list of iCommands. This is the most rich interface to access iRODS, it includes functionalities to execute all iCommands using a graphical interface and also retrieve data and metadata.

FUNCTIONAL WORKFLOW

Figure 2 presents the functional workflow. It describes a generic approach for ingesting data into iRODS (through Davrods), for generating metadata into iRODS and ontology into Virtuoso (through iRODS rules), and for retrieving data and metadata.

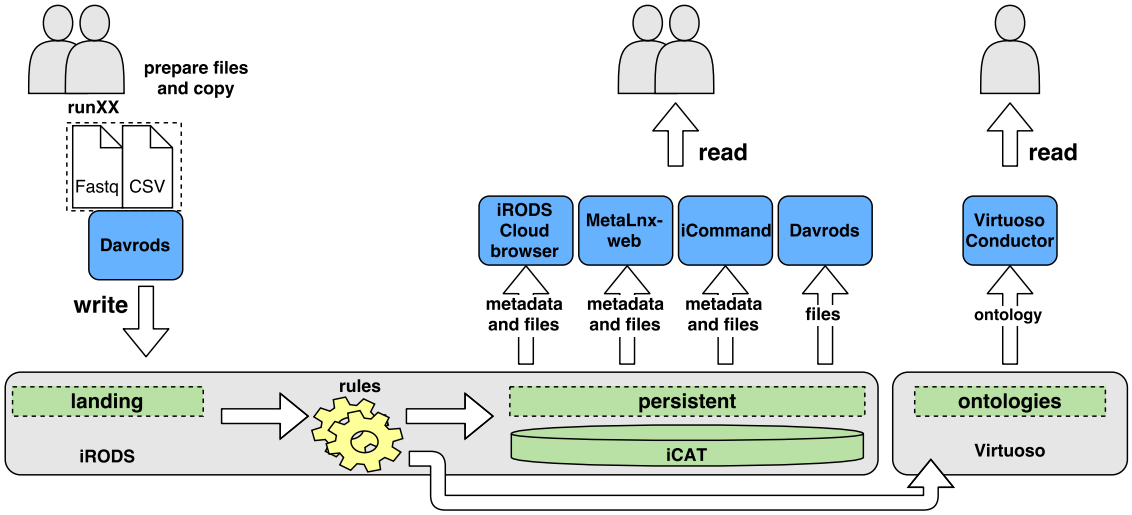

Figure 2. Functional workflow to ingest and retrieve (meta)data from iRODS. Downwards arrows indicate data/metadata ingestion flow, upwards arrows indicate retrieval flows and the horizontal arrows indicate automated processing.

Before ingestion to the repository, the sequencing files and the metadata need to be collected into a single directory that has the name of the sequence run that will be ingested into iRODS. This directory is represented by the name **runXX** in Figure 2, and it contains all fastq files and one sample sheet (CSV file). Once the user is logged in through a mounted Davrods client, they can copy the complete directory to the landing directory. The rules are triggered when the sample sheet, a CSV file, is copied through Davrods into a landing directory of an iRODS collection and the sample sheet is correctly interpreted. Each rule has its specific set of functions and they use several microservices from iRODS. At the end of the rule execution, all the results are moved to the persistent directory, which has read-only access and the metadata are loaded into iRODS and Virtuoso, allowing the users to query and find their data in an easy way, improving Findability (see FAIR principles).

The rules implement several steps and each step is monitored along the workflow and possible failures are reported into a *report file*. This report is stored together with the data, and it can be consulted by the user to check if the data and metadata were loaded correctly into the repository. Each line of the report file matches to one step in the workflow, indicating also the date, time, the path to the file that was being handled and the resulting status. If the rule executed correctly, the final result is *OK*; otherwise, an error message explains the specific step that went wrong. This report file is encoded in *html*, for a user friendly view of all steps and messages.

USER POLICIES AND ROLES

Different research groups need to have separated repositories for their NGS research data. We create iRODS collections, users and groups to organize them inside the storage server. One of the policies of our iRODS instance is for creating metadata in iCat and Virtuoso automatically when the researchers ingest data (see figure 1) on their own landing directories. additionally, we can define different rules for each research group. Furthermore, the two different directories at the storage space have different permission policies. On the landing directory the user has permissions to write files, and on the persistent directory the user has read-only permissions. We have enabled the SSL connection on iRODS server, so all clients that connect to it are using a secure protocol. The iRODS users have the same access rights through Metalnx-web as on iCommands. The Davrods access works with the group access, so all users inside a specific group can have access to specific *landing* and *persistent* directories.

We have defined four principal user roles for our system. The most basic user role is the *Researcher* that only has

access to the landing and persistent storage places for the groups they belong to. The files ingested into the landing directory are private to this user. The files and metadata in the persistent directory are readable by all users in the group. The *Data Steward* has full access to the user group directory they belong to, including rights to change the metadata. The *Principal Investigator* has full control of the repository, being able to include users into groups and assign roles. Finally, the *Repository Administrator* has the ability to create rules and manage all services in the system. The three first users belong mainly to iRODS, whereas the last user also has control of the server.

USE CASE EVALUATION

The use case of a single research group is currently used for evaluating the prototype environment. A repository was created for this group with corresponding collection, group and user accounts. A user was assigned by the research group and trained to use the system. The goal was to store some of the existing raw NSG data into the repository with minimum metadata. The datasets used in this evaluation covered varied conditions regarding the number and size of files. The beta user connected to the repository from a Linux desktop using a mounted Davrods drive for data ingestion and retrieval, and iCommands for metadata search. The metadata and data preparation was supported by the *Data Steward* from our team for the first datasets. The user, *Data Steward* and developer had close communication during the whole process to discuss difficulties and design improvements which were then implemented using the agile development methodology [1]. We monitored and tested the user experience along the phases, detailed in the workflow section, and summarized here:

- **Data and metadata preparation:** the user prepares a directory with NGS data files and a sample sheet (CSV file) with metadata that follows a specific format.
- **Data and metadata ingestion:** the user accesses the Davrods mount point drive to copy data and metadata to the landing directory.
- **Rule processing:** the sample sheet triggers an iRODS rule, generating a XML file with AVU metadata (Attribute, Value, Unit) to be ingested into iRODS and a RDF file to be ingested into Virtuoso.
- **Data and metadata retrieval:** Metalnx and iCommands are used to search files through metadata. Davrods can be used to retrieve data only.

We do this monitoring by asking the user directed questions regarding the following topics:

- **clarity for the user:** does the user know what to do in each case? For example, we wish to understand whether the sample sheet requirements were clear enough to enable successful data ingestion for varied datasets.
- **feasibility for the user:** is it feasible for the user to accomplish the requirements in a practical and workable manner?
- **ease of use:** is it easy for the user to use the system? ie.: mounting the landing directory, copying files, etc.
- **feedback for the user:** is the feedback coming from the system sufficient, clear, easy to find and timely?
- **robustness:** to determine if the system performs well, not only under ordinary conditions, but also under unusual conditions that stress its designers assumptions.
- **performance with different workloads:** to determine if the system performs well with more than one user accessing it through different computers, as well with different sets of files.

Below we briefly describe our preliminary results for this usability test in a qualitative manner. We found that during preparation of the data and metadata the user had only minor problems with user input in the sample sheet and file names, e.g. the user left certain mandatory fields open. The user was mostly satisfied with the process of ingesting

the data and metadata after mounting the Davrods interface on their personal computer, although it was noted that due to the amount of data files being ingested the system can become slower. Additionally, the user had some minor comments regarding the level of detail of the feedback coming from the system. In terms of rule processing the user experienced some more serious problems, e.g.: mac OS systems generate hidden files that caused errors on the system side. As for retrieving data and metadata the user experienced minor problems learning iCommands but were mostly satisfied using Metalnx-web, which implements all iCommands available.

DISCUSSION

Metadata is recognized as useful on all types of RDM system, however annotating detailed metadata takes much effort. Therefor, we opted for a simple approach that is feasible for the users. The simplicity of the current sample sheet, which is mostly generated already by the sequencing facilities, has been considered an advantage during this first evaluation. The choice for iRODS as a basis for our NGS repository has also been considered positive in this first evaluation. It proved to be flexible to accommodate various changes, in particular regarding metadata ingestion, which is expected to be a dynamic component of any RDM system that evolves along time. We expect that this technology will become better supported in our organization, leading to long term sustainability for the solution proposed here.

Last, and most important, is the usability of the system. We have set up the protoype in a way that the user doesn't need to have any iRODS knowledge by: choosing to use Davrods allowing users to drag and drop files to the system; processing the metadata automatically using iRules; using Metalnx-web as an interface for retrieving data and metadata; having a *data steward* responsible for helping successful ingestion of the data and metadata; and having a *administrator* responsible for implementing new policies through iRules. We believe that these choices make our prototype user friendly.

Based on the preliminary results of our usability tests we found that the prototype was generally a feasible solution for the management of NGS data and metadata and we could reach the goals of our prototype even though it is very simple for this first version. Additionally, we found that performance improvements of the *landing* directory could be made. Finally, since we chose to reuse existing generic clients to access the repository, we observed that they provide only limited contextual feedback. Therefore more user-friendly tools that are better customized to this application area might be necessary to address the needs of other user profiles in the future.

ACKNOWLEDGMENTS

This project is financially supported by the AMC Innovation Fund. The authors would like to thank Rudy Scholte, Lieuwe Kool, Joyce Nijkamp, Barbera van Schaik and Niek de Vries for their insightful contributions that helped shape up our RDM solution.

REFERENCES

[1] Agile methodology. `http://agilemethodology.org/`. Accessed: 2017-05-01.

[2] irods icommands. `https://docs.irods.org/4.1.10/`. Accessed: 2017-05-01.

[3] A. Rajasekar, R. Moore, C.-y. Hou, C. A. Lee, R. Marciano, A. de Torcy, M. Wan, W. Schroeder, S.-Y. Chen, L. Gilbert, et al. irods primer: integrated rule-oriented data system. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 2(1):1–143, 2010.

[4] T. Smeele and C. Smeele. Davrods - an Apache WebDAV interface to iRODS. *iRODS User Group Meeting 2016 Proceedings*, 1:41–47, Dec. 2016.

[5] Openlink Virtuoso software. `https://virtuoso.openlinksw.com/`. Accessed: 2017-05-01.

[6] M. D. Wilkinson et al. The FAIR guiding principles for scientific data management and stewardship. *Scientific data*, 3, 2016.

[7] S. Worth. An administrative and metadata ui for irods. `https://github.com/Metalnx/metalnx-web`. Accessed: 2017-05-01.