





















































Genome-Phenome Archive, 2017a) which supports XML formats. The repository metadata can be browsed publicly, with each study assigned an accession number. Datasets can be downloaded using the EGA Download Client, with authenticated user login. The archive has grown to over 1700 TB in size, with over 2302 studies (Lappalainen et al., 2015).

### **2.4.3 GSA (Genome Sequence Archive)**

The Beijing Institute of Genomics (BIG) developed the Genome Sequence Archive (GSA) to be the core resource of its BIG Data Center (BIG Data Center Members, 2016) and to meet the demands of Chinese researchers who faced bottlenecks in data transfer from INSDC (International Nucleotide Sequence Database Collaboration) databases. The archive stores both raw sequence data and metadata (Wang et al., 2017). The GSA implements the International Nucleotide Sequence Database Collaboration (INSDC) data standards and structures. Data is classified into BioProject, BioSample, Experiment and Run objects as seen in Figure 2.1. The BioProject object is assigned an accession number and contains project metadata such as description, funding information, submitter organization and publication(s). The BioSample object is also given an accession number and contains metadata relating to sequencing type and methods. The Run object has a unique accession number and contains the sequence data relating to the experiment.

Storage of various file formats such as FASTQ, BAM and VCF are supported, with over 200 TB of sequence data archived from 39 institutions in 2016 (Wang et al., 2017). Researchers wishing to submit data to the archive can do so via an input wizard for metadata and FTP (file-transfer protocol) for sequence data. Data stored in the archive can be made publicly available or private by submitters. A web based interface is provided for users to browse the archive, with search functionality to query multiple databases.

### **2.4.4 MIAME**

The BBSRC (Biotechnology and Biological Sciences Research Council) suggests in their data management plan that “data should be accompanied by the contextual information or documentation (metadata) needed to provide a secondary user with any necessary details on the origin or manipulation

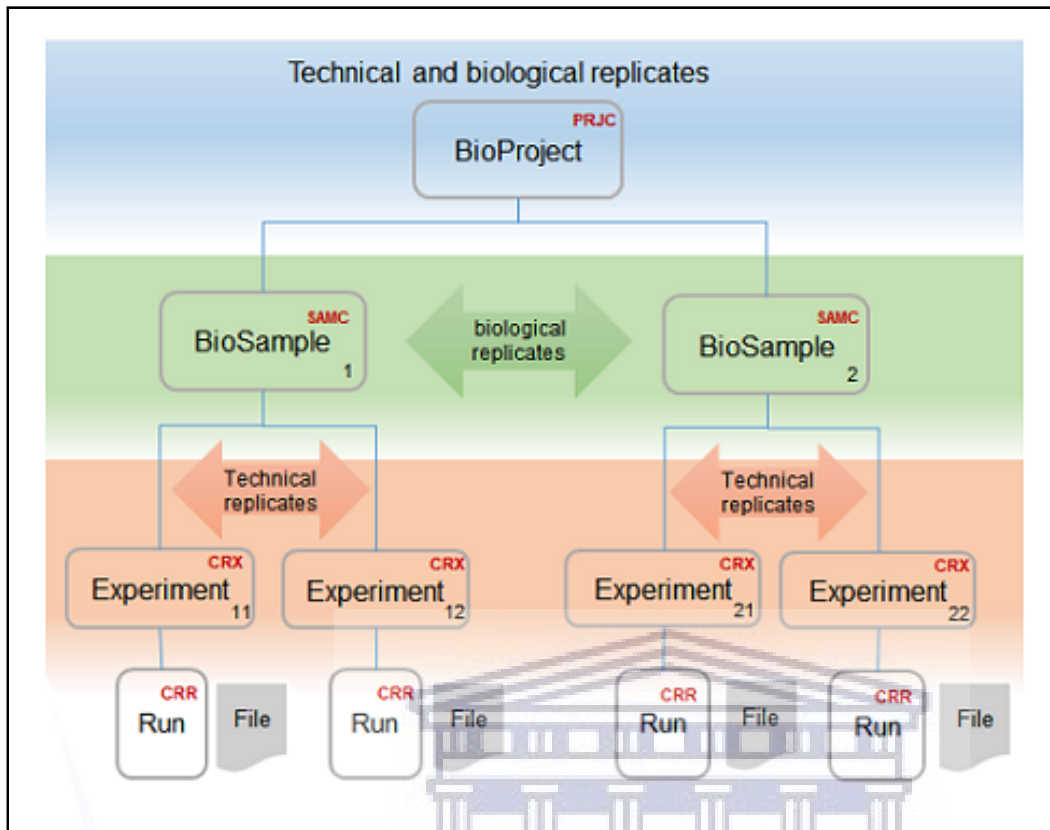


Figure 2.1: Organization of metadata objects in the GSA (Genome Sequence Archive, 2019)

of the data in order to prevent any misuse, misinterpretation or confusion. Where standards for metadata exist, it is expected that these should be adhered to” (Biotechnology and Biological Sciences Research Council, 2019). The heterogeneous nature of genomics research has meant many competing and overlapping metadata standards. Their ultimate purpose is to ensure data generated can be verified, analysed and interpreted by the broader scientific community. The Minimum Information about a Microarray Experiment (MIAME) was developed by Brazma et al. (2001) at the Functional Genomics Data Society. The standard describes the need “to enable the interpretation of the results of the experiment unambiguously and potentially to reproduce the experiment” (Brazma et al., 2001). Many journals and funding agencies now require authors working on microarray-based transcriptomics experiments to comply with the MIAME standard. The standard itself is composed of six core elements listed in Table 2.1, with Figure 2.2 providing a schematic view of their relationships.

Table 2.1: Components of the MIAME standard (Brazma et al., 2001)

Element	Description
Raw Data	Data extracted from the imaging files (CEL or GPR) relating to Hybridisation
Normalized Data	The final normalized data (Gene Expression Data Matrix)
Sample Annotation	Experimental factors and values
Experimental Design	Defined relationship between sample and data
Array Annotation	Gene Identifiers and coordinates
Lab Protocols	Experimental and data protocols used in the laboratory

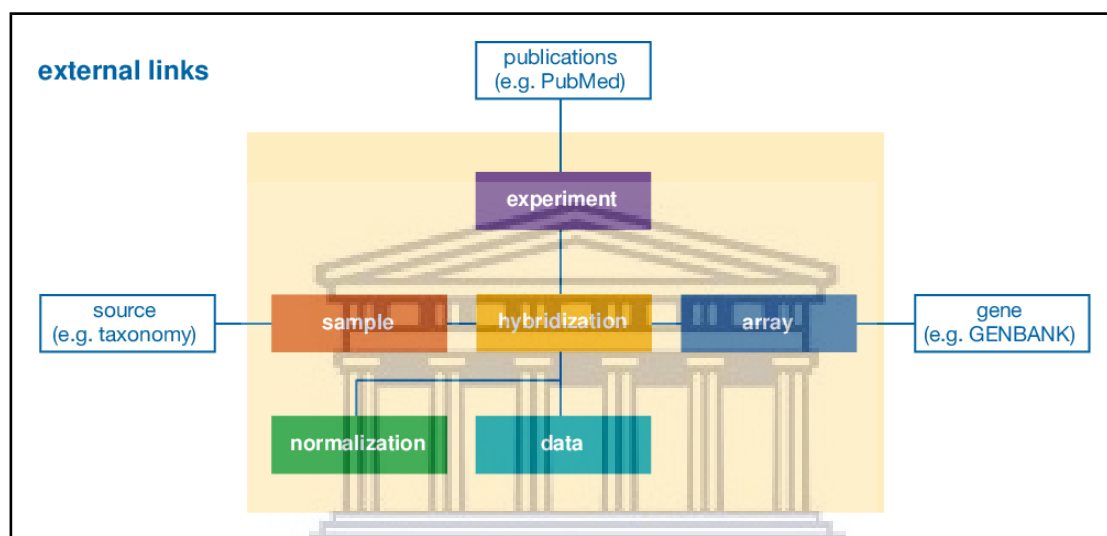


Figure 2.2: Schematic representation of the six data types captured in the MIAME standard (Brazma et al., 2001)

The MIAME standard requires data to be presented using the MAGE-TAB (Rayner et al., 2006) (spreadsheet) format. Tools have been developed by data standards communities to aid researchers in annotating their data. Annotare (Shankar et al., 2010) is a tool which supports researchers to construct MIAME-compliant annotation files based on the MAGE-TAB format.

### 2.4.5 Tab-Based Formats

The use of tab based formats has also gained traction amongst scientific researchers. Originating from the Study Data Tabulation Model (SDTM) (Wood and Ginter, 2008) it was designed for the electronic submission of clinical study data. MAGE-TAB (supporting microarray data) and ISA-TAB (González-Beltrán et al., 2012)(supporting numerous experiments such as high throughput screening, mass spectrometry and gel electrophoresis) are two formats used by researchers to collect and manage metadata. The advantages over XML (Extensible Markup Language), relate to an increase in human readability and familiarization with spreadsheet formats. Tab-based formats, through the use of multiple tools such as isaConverter (isatools, 2019), have the capability to be converted to XML. The MAGE-TAB format implements the MIAME requirements by referencing the core components listed in Table 1. MAGE-TAB is comprised of three files: The Investigation Description Format (IDF) file which contains information pertaining to the researcher, experiment descriptions and bibliographic references. The Array Design Format (ADF) file contains the assignment of sequences to positions and the Sample and Data Relationship Format (SDRF) file describes the mapping of samples to object data which is contained in the raw and processed data files, represented in ASCII or binary format.

The ISA-TAB framework is based on three main entities: Investigation, Study and Assay (Sansone et al., 2006), with the aim of structuring metadata and describing the relationship between samples and data. The format like MAGE-TAB, can be built using spreadsheets or programmatically with the aim to communicate information. The use of ontologies and controlled vocabularies is not mandated and left to the discretion of those implementing the framework. The format itself, is comprised of four components: An Investigation file, Study file, Assay file and Data file. The investigation file contains declarative information which is referenced in other files and links several study files to an investigation. The study file contains contextual information on the assays and their references. The Assay file references information on the assay (expression, protocols) and the Data file which can consist of raw, normalized or processed data. The main difference between the ISA-TAB framework and MAGE-TAB, lies in ISA-TAB's ability to complement existing biomedical

formats such as the Study Data Tabulation Model (Sansone et al., 2008).

## 2.5 FAIR Data

To alleviate the present state of affairs regarding metadata, guiding principles were proposed by Wilkinson et al. (2016) to improve data reusability and discovery. The FAIR principles are a set of guiding principles in order to make data Findable, Accessible, Interoperable and Reusable (Wilkinson et al., 2016). These principles not only apply to data but to workflows, algorithms and tools. The principles have been implemented in various degree in data repositories such as Dataverse, UniProt and FAIRDOM. Where these repositories fall short in their “FAIRness”, is in the lack of integration and harmonization of data. The integration of data resources in workflows is increasing in demand, as well as the need for data provenance. The use of multiple repositories is leading to less integration and discovery, whilst exacerbating the issue of data reproducibility and reusability. These principles are detailed in Table 2.2.

The principles can also be adopted by researchers when sharing their own research data; the context being distinguished between the sharing of raw data within their research group and results with the wider public and research community. The FAIR principles can be implemented through basic guidelines. For example to be Findable, data should at least include versioning. Interoperable and Reusable, meaning the data collection process should be noted and metadata described using clear unambiguous semantics which can be adopted from community standards. There are multiple options to make data Accessible, such as P2P networks and HTTP servers . It should be noted that the FAIR principles are not a standard or implementation solution. They are used as a common measurable denominator to evaluate whether the data in your institution are findable, accessible, interoperable and reusable, providing guidance to data management stakeholders.

## 2.6 Data Sharing

The field of genomics is becoming increasingly attractive for Sub-Saharan African countries to relieve the burden of both communicable and non-commu-

Table 2.2: Summary of FAIR guiding principles (Wilkinson et al., 2016)

Principle	Description
To be Findable:	<p>F1. (meta)data are assigned a globally unique and persistent identifier</p> <p>F2. data are described with rich metadata (defined by R1 below)</p> <p>F3. metadata clearly and explicitly include the identifier of the data it describes</p> <p>F4. (meta)data are registered or indexed in a searchable resource</p>
To be Accessible:	<p>A1. (meta)data are retrievable by their identifier using a standardized communications protocol</p> <p>A1.1 the protocol is open, free, and universally implementable</p> <p>A1.2 the protocol allows for an authentication and authorization procedure, where necessary</p> <p>A2. metadata are accessible, even when the data are no longer available</p>
To be Interoperable:	<p>I1. (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.</p> <p>I2. (meta)data use vocabularies that follow FAIR principles</p> <p>I3. (meta)data include qualified references to other (meta)data</p>
To be Reusable:	<p>R1. meta(data) are richly described with a plurality of accurate and relevant attributes</p> <p>R1.1. (meta)data are released with a clear and accessible data usage license</p> <p>R1.2. (meta)data are associated with detailed provenance</p> <p>R1.3. (meta)data meet domain-relevant community standards</p>

nicable diseases on their populations. There are also the added benefits of localized cost-effective innovations and improvements in health care systems. There is however a large disparity in the location of research being conducted, for example Adedokun et al. (2016) in a study of research publications in Sub-Saharan Africa from 2004 to 2013 found that South Africa had produced three times more publications than any other sub-Saharan African country during its period. This has been attributed to the South African commitment to funding and improving its biotechnology infrastructure. Other institutions are struggling to produce high-quality research due to power supply, internet connectivity and literature access. The concept of generating large genomic datasets remains novel to many African institutions due to these limitations, with many choosing to outsource their sequencing or download data from a publicly available dataset abroad. The transferring of datasets remains a tedious task for many institutions on the African continent. Internet connections are slow and intermittent due to poor local infrastructure, low investment in public electrical supply and in some cases adverse weather conditions. The lack of the necessary computing capacity in skills and hardware has led to institutions physically transporting their data to locations which have suitable infrastructure for storage and processing. This approach carries security risks as well as possible ethical and legal issues regarding the storage location and data access.

The ultimate goal for scientific data management is to network related information systems and to secure vulnerable data. The way forward for 'open science' and to enhance an eScience culture (Deus et al., 2008) is to further develop funded data repositories which provide storage infrastructure for both data and researcher analysis with curation to achieve long term data preservation. There are great benefits to be had in providing easy access to raw and primary data in the reuse of data for answering new scientific questions, meta studies, theory comparison and revealing of scientific misconduct (Fanelli, 2009). Researcher collaboration is an essential part of scientific research, however young researchers have not received mentorship in open science and smaller institutions being wary of publishing data in public archives (Gewin, 2016).

Many African researchers are familiar with the ardent process of sharing

and storage of biological samples, which should be done in accordance with the legislative requirements of their respective countries. This is partially due to the ‘helicopter science’ and ‘sample safari’s’ of the 1990s and due to national government awareness in the value of these samples. This in turn establishes genomic sovereignty and national security. This approach, while ample for the specific time period, did not foresee the future in the generation of large scale genomic data produced by African researchers. As of 2014, only Cameroon, Ethiopia and Tanzania incorporated ethical guidelines that focus on data and sample sharing (de Vries et al., 2014). Data sharing is a common requirement for funding agencies who wish to maximize the use of data generated to receive a greater ‘return on investment’. These returns can take the form of research validation and in future hypothesis studies. Despite the clear benefits, African researchers remain fearful of sharing their data. Concerns mainly preside over recognition and financial rewards. The latter raising concerns of an emerging stigma that African researchers are ‘mere data collectors’ being unable to make significant contributions to the scientific community. The lack in infrastructure and specialist skills has meant the turnover for African Research is lagging behind the rest of the world, with research groups abroad making use of data generation, data archiving and data processing to quickly generate publications. The concerns of data release and secondary analysis use are highlighted by a study on the role of ethical metadata in genomics research in Africa (de Vries et al., 2014). Several data repositories have taken on the responsibility of forming Data Access Committees (DACs) made up of one or multiple individuals who have the mandate to release data to external requestors based on consent and/or national research ethics.

## **2.7 Storage and Data Management**

### **2.7.1 Cloud Storage**

Large quantities of genomic data being generated are often classified as unstructured data. Unstructured data normally refers to flat-files which lack metadata detailing information on their content and provenance. One such method of handling such data is object based management, whereby attributes

are assigned to the data files making them available for query. Object storage was conceived in the 1990s by Gibson et al. (1997) to increase scalability and meet bandwidth demands in distributed networks. Object storage is of particular use to the genomics community due its support for large quantities of files and modifiable metadata. Another benefit of object storage is the ability to import external data from different sources in various formats and to publish them using a common framework integrated with existing relational data. This enables data elements to be associated in a way that supports user-defined analysis. The development of object storage is still ongoing in many projects including Google File System, GlusterFS, Swift and Amazon AWS S3.

Data is a fundamental asset for bioinformaticians in analysis and knowledge discovery. Thus bioinformatics cloud services are highly dependent on data storage. Data as a Service (DaaS) through the use of object storage solutions has enabled data to be accessed from anywhere at any time and has reduced the traditional costs of purchase and maintenance of computing infrastructure (Dai et al., 2012). Amazon AWS has provided a central repository for public biological datasets and archives such as GenBank, Ensembl, Unigene and datasets from other scientific research fields such as astronomy and chemistry.

Big data storage and analysis can be achieved by placing data in the cloud as DaaS and moving code to be executed in the cloud. Presently only a small proportion of biological data is accessible in the cloud with AWS providing GenBank, Ensembl and 1000 Genomes. The vast majority is still stored in traditional biological datasets. Subsequently, workflows executed in cloud environments are limited.

### 2.7.2 Data Management Systems

There are several approaches to managing biological data. A spreadsheet based approach utilizes programs such as Microsoft Excel and Openoffice Calc to capture and store sample metadata. These programs are familiar to researchers with laboratory backgrounds. A naive high performance computing (HPC) approach is typically used to store data files, where user groups are given rights and access to protect their data. As no 'real' database is used,

this approach lacks measures ensuring data consistency between users (Arita, 2008). This makes it unsuitable for the management of experimental data.

### 2.7.2.1 iRods

The Integrated Rule-Oriented Data System (iRods), is an open source data management system which enables user access, management and sharing of data across multiple storage systems whilst maintaining redundancy and security (Hedges et al., 2009). The middleware was developed to meet the increasing demand for digital creation in projects generating large quantities of data. The approach taken by iRods to automate curation through the use of policies or ‘iRules’, which specify actions to be taken, given a set of conditions. The iRules Engine enables the implementation of application specific processing, which can support specialized metadata management (Hedges et al., 2009). A prominent component of iRods, is its metadata management function. iRods enables metadata to be added to stored files using a (*key, value, unit*) triplet format. This metadata can be searched using SQL-like queries using the command line (iCommands) or programmatically through client API’s such as the Python iRods Client (iRods Consortium, 2018). The iRods middleware has been used in both genomic metadata (Nieroda et al., 2019) and data (Chiang et al., 2011) projects.

### 2.7.2.2 OneData

OneData is a global data management system that enables distributed data storage resources to be accessed in a transparent way (Wrzeszcz et al., 2017). Researchers are becoming increasingly reliant on distributed access to large datasets. These datasets are generated by various institutions and utilize multiple storage technologies and infrastructure. OneData primarily focuses on user authentication and authorization for data by provisioning for secure sharing and flexible metadata. It builds upon existing data management solutions such as Dropbox or local Ceph instances by decentralizing management; simplifying deployment and reducing access and authorization complexity. OneData has been deployed as e-infrastructure by EUDAT and EGI to support the West-Life project which provides computation and data management services to structural biologists (Morris et al., 2019).

## 2.8 Conclusion

The above chapter references literature highlighting the potential for the development of a Genome Archive on the African continent. The motivation which can be ascertained in the large quantities of data being generated by research groups requiring storage and curation. An opportunity was recognized in leveraging the use of ontologies of data standards to improve metadata collection and curation. This also presents opportunities for study verification and amalgamation. The FAIR principles can be used to guide policy and management to improve discoverability. The next chapter will discuss the methods and materials of the thesis.



# Chapter 3

## Design

### 3.1 Introduction

In the previous chapter, the potential development of an *African Genome Archive* was identified. This chapter will outline the domain of the thesis and the required functionality of the proposed software. A systematic approach is taken whereby the problem is modularized and relevant details are established. The scope of this project follows the conceptual design of a genome archive and the implementation of a working prototype.

### 3.2 Project Requirements

This thesis aims to meet the specific research questions outlined in chapter one in demonstrating a proof-of-concept solution. The *African Genome Archive* is the name given to the software product produced by this thesis, which expands upon existing research lab storage infrastructure. The software was primarily written in the Python programming language, with a web based front-end to enable the user to upload their project files and to browse metadata stored in the archive. iRods was utilized as middleware; communicating between the web application and backend to manage metadata and file storage.

### 3.3 System Context

Figure 3.1 depicts the Genome Archive software which interfaces with a pre-existing storage environment. A web interface is used to elicit key metadata from the submitter and to provide options for storage location and categories. When the user uploads their project file and accompanied metadata, the genome archive software manages the metadata for each sample and stores the accompanied files in the relevant storage location. Users are then able to browse and query the various projects uploaded.

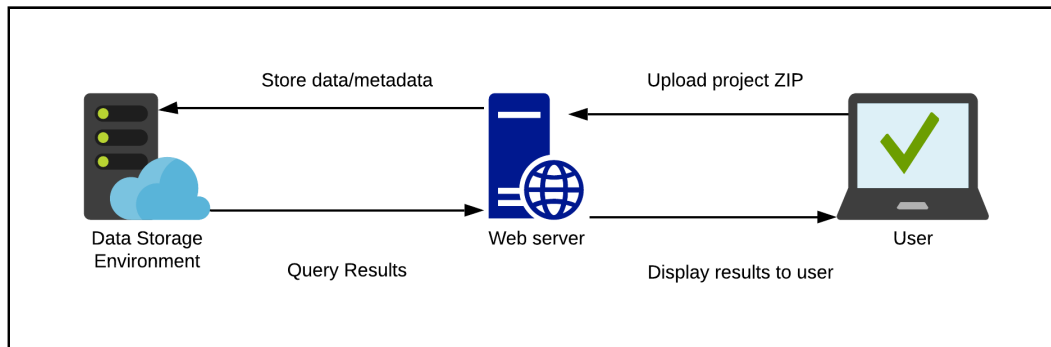


Figure 3.1: Model describing Genome Archive use cases

### 3.4 Use Cases

This section outlines example use cases for the proposed genome archive.

#### Storing metadata and data files in an organized manner

The increase in the number of datasets and dataset size has resulted in researchers struggling to retrieve datasets or files for analysis with the accompanying metadata. In this particular case a researcher may upload their dataset and metadata to a storage environment which hosts the data. The data project which is uploaded links the individual files and metadata which is then stored in a location chosen by the researcher. Subsequent to the completion of the upload, the researcher is then able to access and browse the project files and metadata, potentially resulting in greater data awareness.

#### Browse and query metadata and data stored in the archive

In another instance, a researcher may be looking for datasets to compare results or to evaluate work flows. It is also possible that a researcher may be looking for a collaboration partner who is working with a similar dataset. The metadata and data storage solution offered by this project enables researchers to query datasets and metadata, using relevant search terms which could aid data discovery.

### 3.5 Conceptual Model

The components of the software architecture are detailed in the conceptual model. The following section presents the system model of the *African Genome Archive*. Figure 3.2 depicts the user interaction with the archive system and

its components. The model also showcases the system responses based on the user instructions.

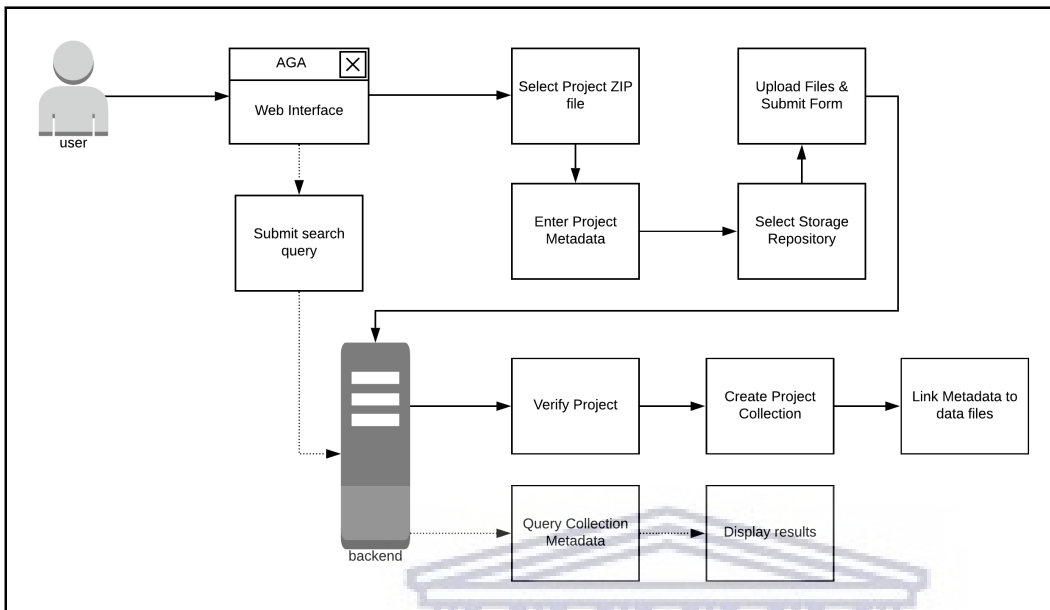


Figure 3.2: User interaction model for the African Genome Archive

### 3.5.1 User

The user interfaces the *African Genome Archive* using the provided web interface. Initially the user is taken to the home screen, where they are presented the options of uploading a project or submitting a search query (browsing the repository). The user can select to upload a project, where they are presented with a form to input high-level project metadata and to select the project file to be uploaded. The user is also able to submit a search query using the search box and to browse projects and files under their specific categories.

### 3.5.2 African Genome Archive System

The Genome Archive system is hosted on a web server which interfaces with a backend storage environment. The backend system handles the majority of the work in running the genome archive, engaging both the user and the storage environment in translating instructions. The archive system verifies project files that are uploaded and links individual file metadata. It is also responsible for handling user metadata queries, where results are displayed to the user via the web interface.

### 3.6 Architectural Detail

Figure 3.3 expands on the conceptual model shown in Figure 3.2, to give a comprehensive description of the components of the *African Genome Archive* system.

Users of the *African Genome Archive* need to authenticate the credentials with the iRods backend in order to be granted access. These credentials would be obtained from their host repository (local institution). For the pur-

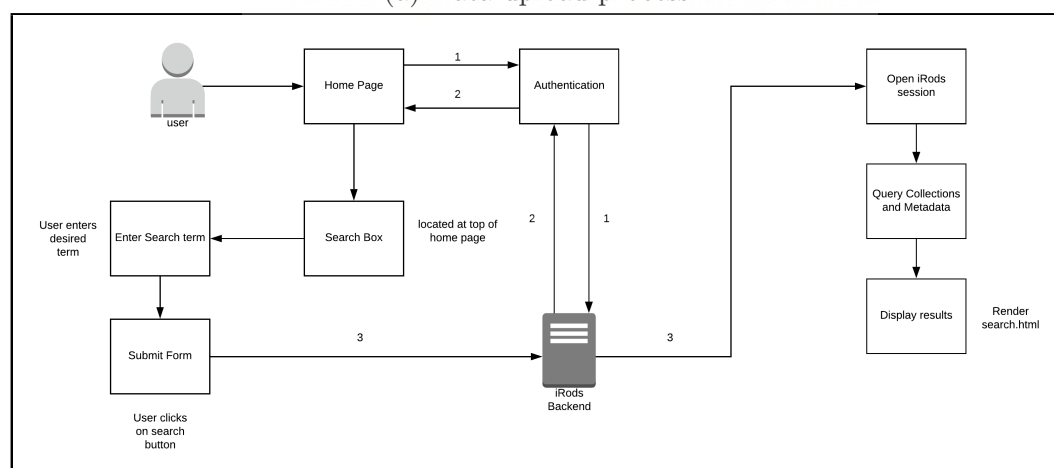
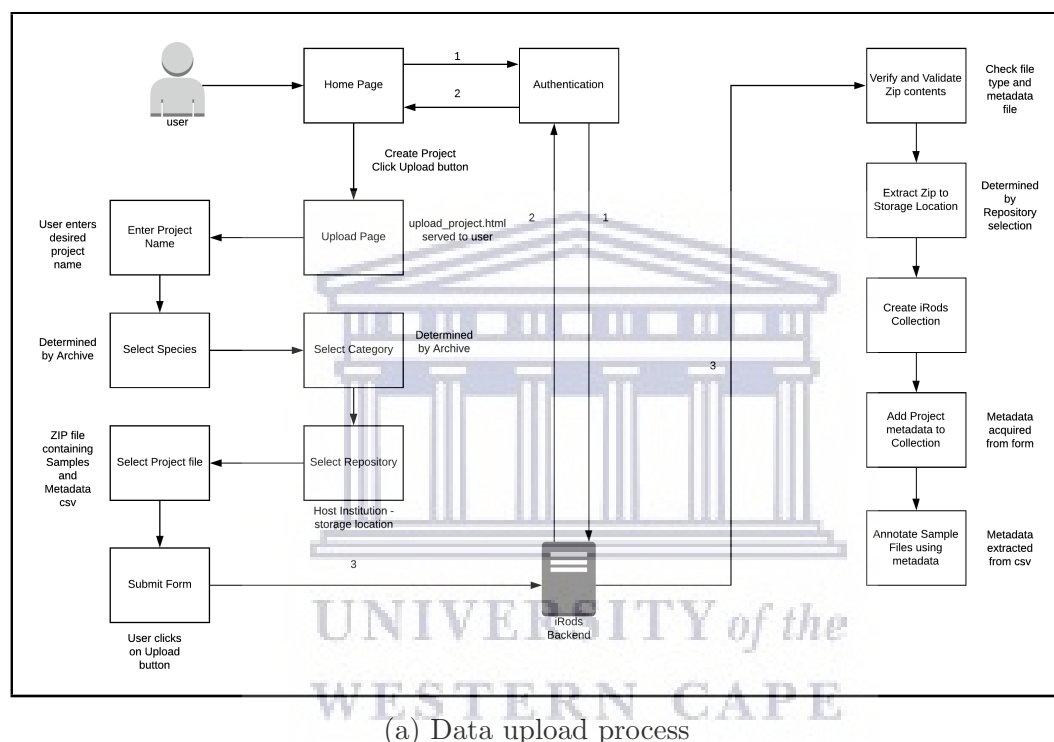


Figure 3.3: Detailed archive architecture diagram

pose of this thesis, dummy login details are assumed and the user is immediately granted access. Once on the home page, the user is presented with the option of uploading a project. The Upload Project page presents the user with a mandatory form that requires completion.

The user can then submit this form and the project file to the archive system, where the ZIP file is validated and the project name checked for uniqueness. The ZIP file contents which contain the metadata and sample files are then extracted to the chosen repository storage location. An iRods Collection is then made linking the form metadata submitted by the user to the project and the metafile contained in the project file to the individual sample files. Once this is completed the user is then able to browse and download the sample files and metadata.

The Home Page also contains a search box where the user is able to input search terms to query metadata stored in the archive system. When the search button is pressed an iRods query is initiated on the collections stored and the sample file metadata. The results of the query are then displayed to the user.

### 3.7 Backend Architecture



Figure 3.4: Backend model for the African Genome Archive

Figure 3.4 gives an overview of the backend architecture which contains the server, middleware, database and server-side framework components of the genome archive. These components form the software stack. The server

component provides the shared resources such as network connectivity, file storage, security and metadata database. The database type implemented in storing the metadata relating to the sample files is a NoSQL database. This database is interfaced with the iRods middleware component which controls data being sent between the server, database and the Flask framework. The Flask framework, which is based on the Python programming language, is the development platform of the archive web application.

### **3.8 Technical Functionality Requirements**

The overarching goal of the *African Genome Archive* software is to provide researchers with an accessible user interface to store their data and metadata. The archive should also facilitate the discovery of datasets amongst researchers. The functionality requirements defined below aid in meeting the research objectives defined in Chapter One.

**Users are able to upload data and metadata without the need for technical expertise**

Project submission and upload by the user should be processed automatically by the system, which handles metadata annotation.

**Users are able to browse datasets uploaded by other users**

The user should be presented with datasets and metadata submitted by other users which they are granted access to and able to download.

**User friendly web interface**

The user should be presented with an easy to use, clear and readable web interface which abstracts commands and technical details.

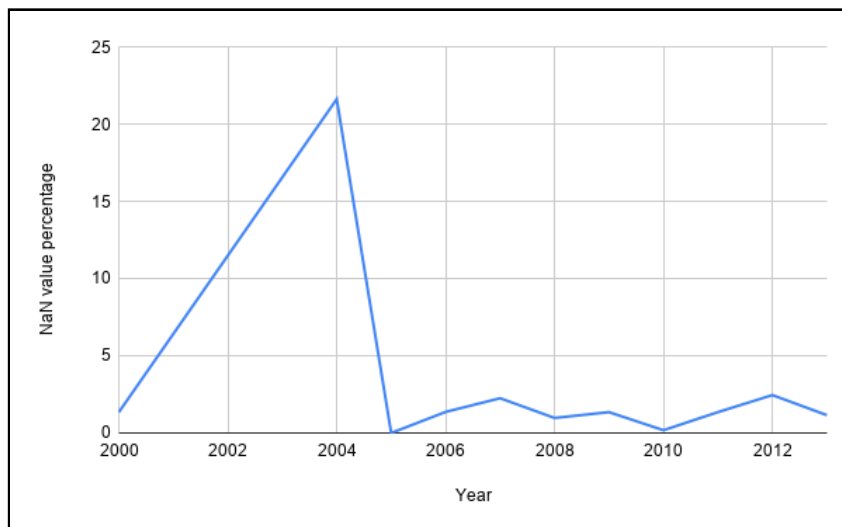
**Users are able to access and adapt a pre-existing metadata template**

The application should provide and support a metadata template which users can utilize for their datasets, with some fields being mandatory.

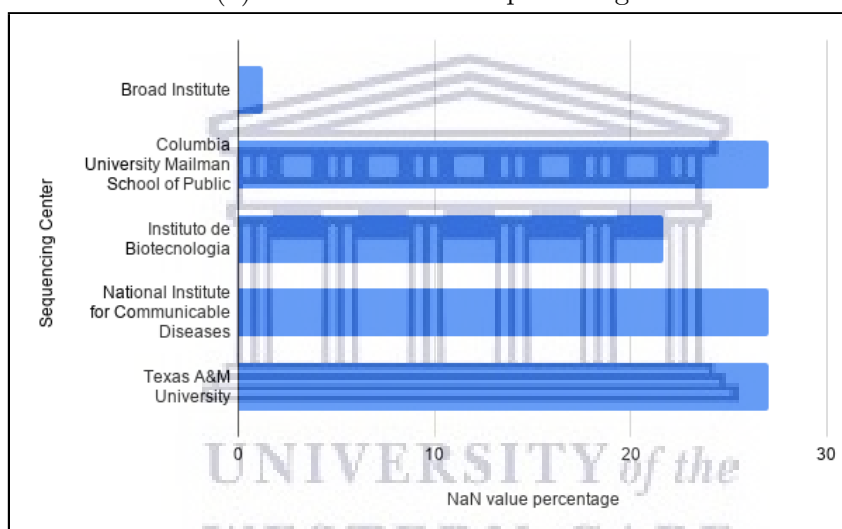
### 3.9 PATRIC Ontology Exploratory Analysis

Prior to the development of the software, a survey was conducted of South African *Mycobacterium tuberculosis* data hosted at PATRIC (Wattam et al., 2013) (Pathosystems Resource Integration Center). This was done in order to determine the requirements of the metadata schema to be used by the Genome Archive. The PATRIC website provides numerous pathogen datasets from multiple geographic locations and analysis tools to support biomedical research on bacterial infectious diseases. The dataset obtained contained 998 South African *Mycobacterium tuberculosis* samples and was exported into a comma separated value (CSV) format. The exploratory analysis was completed using the Python Pandas package and Jupyter notebooks.

The dataset included 53 column headings which were of type 64-bit integer, Object (String) and 64-bit float. 15 columns were found to have no data relating to them. These included Latitude, Longitude, Body Sample Site, Type Strain and Culture Collection. The dataset was then checked for metadata inconsistencies. Fields such as 'Isolation Source' was found to contain varying string values such as 'sputum', 'Bodily fluid', 'pleura', 'Excreted bodily substance', 'NaN', 'Pus', 'Sputum', 'Clinical Isolate', 'Likely Sputum', 'tuberculosis patients during a TB epidemic', 'patient' and 'clinical isolate'. A wide variety of date formats were also used to define the sample collection date. This included YYYY, DD/MM/YY, MM/DD/YY, DD-Month-YYYY, ISO 8601 format and year ranges e.g. 1900/2013. More surprisingly is the discrepancy seen in the 'Host Gender' field where inconsistencies can be found in the definition of male and female. The values obtained for this field are: 'Female', 'Male', 'NaN', 'male'. The 'Host Age' field reported integer values but also contained the string value 'Adult'. The variation in metadata consistency is potentially related to the eight different sequencing centers which produced the sample metadata. Not a Number (NaN) refers to null values in the dataset. The bar chart depicted in Figure 3.5(a) shows the percentage of NaN values in the dataset on an annual basis. Figure 3.5(b) depicts the average percentage of NaN values produced by the Sequencing Centres on an annual basis.



(a) Annual NaN value percentage



(b) NaN value percentage by Institution

Figure 3.5: Exploratory analysis results

### 3.10 Conclusion

The above chapter describes the domain of the thesis and required functionality of the proposed software. The components of the suggested solution are identified and interpreted in order to develop the systematic approach to the project. The next chapter discusses the software implementation.

# Chapter 4

## Implementation

### 4.1 Introduction

The development of this project followed a systematic approach, whereby the problem is modularized and relevant details are established. This chapter will discuss the project scope involved in designing and implementing a working prototype to aid conceptualization of an *African Genome Archive*.

### 4.2 Software

The Flask micro web framework was used in conjunction with the iRods data management software to create and deploy the solution.

#### 4.2.1 Project Breakdown

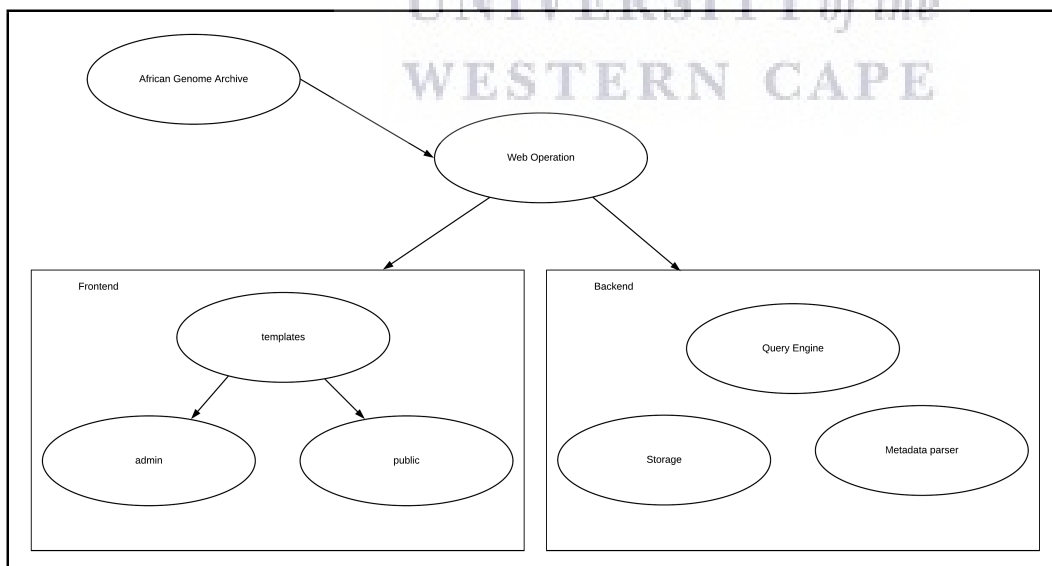


Figure 4.1: Component interaction model of the Genome Archive

Figure 4.1 provides a visual overview of the layout and interaction of components in the archive system.

## 4.3 Backend

### 4.3.1 Hardware

The Genome Archive is hosted on a virtual machine running Ubuntu 16.04, with 2GB RAM and 20GB of storage. The OpenStack Infrastructure as a Service (IaaS) platform located at the South African National Bioinformatics Institute (SANBI) was used to manage the virtual instances and connectivity.

### 4.3.2 Web Rendering

The main application file is served by initiating the Flask server on the provisioned virtual machine. Running the Flask server required the installation of *Python-pip*, *xlrd*, *simplejson* and the *python-irods client*. An ansible script, see Appendix A, was written to automatically install the dependencies once the virtual machine was deployed on the network. During the development phase the server was started using the following command:

---

```
$ python run.py
```

---

The command launches the Flask web server which runs on the IP address of the virtual machine and connects to clients using the standard HTTP port 80. When a user navigates to an endpoint, the appropriate template is built and served. This script interfaces with the *views.py* script (Listing 4.1) which manages backend and storage functionality.

### 4.3.3 Main Backend Operations

Listing 4.1: views.py script

---

```
from app import app

if __name__ == "__main__":
    app.run(host='0.0.0.0')
```

---

The *views.py* (Listing 4.1) script contains all the interaction commands and methods necessary for the application to function. The script abstracts storage operations and metadata parsing; utilizing the iRods python client to provide the web application with responses. The script also contains variables

such as *'allowed\_project\_extensions'* and *'PROJECT\_UPLOADS'* which act as definitions for the storage component. Methods pertaining to getting the overview of projects stored in the archive, getting project ID's, metadata and generating data file URLs are also included to be accessed by the user via the web interface.

The operations described below are core components of the genome archive software:

## Upload Project

The *upload\_project* method manages both *POST* and *GET* requests from the web server. If the request is a *GET*, the method renders the *upload-project.html* page which is displayed to the user. The user is then able to enter metadata and submit a project file to the archive, which is handled as a *POST* request. If a *POST* request is received, the submission form is checked for errors and the uploaded project file is acceptable for storage in the archive. The uploaded file is then moved to a storage location, where its contents are extracted. iRods collections are created for the project and its related samples using the metadata provided.

Listing 4.2: Upload project method

```
@app.route("/upload-project", methods=["GET", "POST"])
def upload_project():

    if request.method == "POST": #If project being uploaded

        if request.files:

            project = request.files["project"] #get files

            if project.filename == "": #check if project file has
                no name
                print("No filename")
                return redirect(request.url)
```

```

if allowed_project(project.filename): #make sure
    project is acceptable
    filename = secure_filename(project.filename)

project.save(os.path.join(app.config["PROJECT_UPLOADS"],
    filename))

print("Project saved") #SAVE PROJECT to Disk
print(app.config["PROJECT_UPLOADS"]+"/"+filename)

req = request.form #project metadata

with
    zipfile.ZipFile(app.config["PROJECT_UPLOADS"]+"/"+filename,"r")
    as zip_ref: #unzip file
        zip_ref.extractall(app.config["PROJECT_UPLOADS"]+"/"+
            req['projectname']+"/")

print("Project Unzipped")
irods_createCollection("/irods_1zone/home/user/"+
    req['projectname'],req) #create Project
    Collection with metadata submitted
print("Collection Created")

# create Collection for each sample in metadata
createsample_collections(app.config["PROJECT_UPLOADS"]+
    "/" + req['projectname'] + "/" , "/irods_1zone/home/user/" +
    req['projectname'] + "/" )

return redirect("/")

else:
    print("That file extension is not allowed")
    return redirect(request.url)

```

```

return render_template("public/upload_project.html",
    hide_button=False)

```

---

## View Projects

Listing 4.3 illustrates the *projects* function that handles *GET* requests which result in the rendering of the *projects.html* template, displaying the results of querying the archive by project. An iRods session is initiated using the user login and iRods zone credentials. The iRods session performs the ‘*get*’ method to request all collections from the provided zone. These results are then collated in a dictionary, with the key being the *Collection ID* and the value being the metadata relating to the project in a JSON format. The dictionary is then passed to the flask rendering template function that transcribes the data provided into a presentable format for the user to view using the *projects.html* template.

Listing 4.3: View projects method

---

```

@app.route("/projects")
def projects():
    try:
        env_file = os.environ['IRODS_ENVIRONMENT_FILE']
    except KeyError:
        env_file =
            os.path.expanduser('~/.irods/irods_environment.json')
    with iRODSSession(irods_env_file=env_file ,host='localhost',
        port=1247, user=username, password=passw,
        zone='irods_1zone') as session:

        coll = session.collections.get("/irods_1zone/home/usern")
        #get all collections

```

```

projects = dict() #make projects dict
for col in coll.subcollections:
    col2=session.collections.get(col.path) #get collection
    colmeta=col2.metadata.items() #get metadata
    metadata= irodsmetaJSON(colmeta) #convert metadata to
        JSON
    metadata['path']=app.config["PROJECT_UPLOADS"]+"/"+
    metadata['projectname']

    projects[col.id]=metadata #add to projects dict
return render_template("public/projects.html", projects=projects)

```

---

## Download sample

The *download\_sample* method, shown in Listing 4.4, is used to return the files relating to a specific sample in a project requested by the user for download. The input parameters for this method are the project and sample name. The path location for the sample is then used to retrieve the metadata which contains the file types stored for the sample and includes their location. A ZIP object is created, containing the sample files requested. The resulting ZIP file is then processed and submitted to the user as a download.

Listing 4.4: Download sample method

---

```

@app.route("/download-sample/<projectname>/<samplename>")
def download_sample(projectname,samplename):
    colpath="/irods_1zone/home/user/"+projectname+"/"+samplename

    col = irods_getCollection(colpath)
    objmeta=col.metadata.items() #get metadata
    metadata= irodsmetaJSON(objmeta) #convert metadata to JSON
    files =[metadata['BAMfilename'],metadata['VCFfilename'],
    metadata['FASTQ_r1filename'],metadata['FASTQ_r2filename']]
    filepath=app.config["PROJECT_UPLOADS"]+"/"+projectname
    dfilepath=app.config["PROJECT_DOWNLOADS"]

```

```

#create a ZipFile object
zipObj = ZipFile(dfilepath+"/"+samplename+".zip", 'w')

# Add multiple files to the zip
for sfile in files:
    if sfile != "NULL":
        zipObj.write(filepath[:]+"/"+sfile,basename(filepath[:]+"
            "/" +sfile))

# close the Zip File
zipObj.close()

try:
    return
    send_file(dfilepath[len("app/")+:"/"+samplename+".zip",
        attachment_filename=samplename+".zip")
except Exception as e:
    return str(e)

```

---

### Query Archive

Listing 4.5 illustrates the source code for implementing the query functionality which enables users to search for specific terms in the archive. Firstly an iRods session is created using the login and zone credentials provided. Inside of the iRods session a query is initiated, requesting all collection names and data objects in the archive. This query is then filtered to remove archive items which are temporarily stored due to deletion and items not containing the search term. The results are then placed into a dictionary and returned to its parent function.

---

Listing 4.5: Download sample method

```

def irods_search(term):

    try:
        env_file = os.environ['IRODS_ENVIRONMENT_FILE']

```

```

except KeyError:
    env_file =
        os.path.expanduser('~/.irods/irods_environment.json')
with iRODSSession(irods_env_file=env_file ,host='localhost',
    port=1247, user=username, password=passw,
    zone='irods_1zone') as session:

queryZone = session.query(Collection,
    CollectionMeta).filter(Criterion('=' , CollectionMeta.name,
    'repository')).filter(Criterion('like' , CollectionMeta.value,
    term))

for result in queryZone:
    if "trash" not in result[Collection.name]:
        item =dict()
        item["CollectionName"]=result[Collection.name].split('/')[ -1]
        #add value to dictionary
        item["CollectionID"]=irods_getCollection(
        result[Collection.name]).id
        item["CollectionPath"]=result[Collection.name]
        item["CollectionOriginID"]=irods_getCollection(
        result[Collection.name]).id
        results[result[Collection.name]]=item

return results

```

#### 4.3.4 Backend Storage

Figure 4.2 shows the metadata template which is completed by the researcher for each project that they upload to the archive. The template uses the Microsoft Excel spreadsheet format, which can be modified using various open source spreadsheet software suites such as Open Office.

The storage location of sample data files uploaded to the archive is shown in Figure 4.3. Projects uploaded to the archive are provided their own storage space through an iRods collection. The project metadata is then linked to

#	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	BAMfilename	VCFfilename	FASTQ_r1filename	FASTQ_r2filename	sample_id	collection_year	collection_month	collection_day	country	study_id	responsib_decode		culture_rc	culture_cc	culture_st	dna_extr	dna_stora	date_dna_r
2	SAWC123.bam	SAWC123.vcf	SAWC123_r1.fastq	SAWC123_r2.fastq	SAWC123	2012	10	5	South Africa	0311	Pete Smit	S123	42897	42921	yes	42923	Box 11, pc	42926
3	NULL	TB123.vcf	NULL	NULL	TB123	2012	8	8	South Africa	0312	Pete Smit	R-123	42809	42830	yes	42831	Box 7, pos	42926
4																		
5																		
6																		
7																		

Figure 4.2: Metadata template in spreadsheet format

this collection. Individual files that form part of the uploaded project are also given a collection. In this example the ‘SAWC123’ sample has a BAM file, FASTQ files and a VCF file linked. Figure 4.4 shows the metadata for a sample accessed by using the ‘*meta*’ command which queries the NoSQL database of the archive.

```

File Edit View Search Terminal Help
ubuntu@irods:~$ ilcs
/tempZone/home/alice:
C- /tempZone/home/alice/PeterTB
C- /tempZone/home/alice/Uganda TB Project
ubuntu@irods:~$ icd PeterTB
ubuntu@irods:~$ ilcs
/tempZone/home/alice/PeterTB:
C- /tempZone/home/alice/PeterTB/SAWC123
C- /tempZone/home/alice/PeterTB/TB123
ubuntu@irods:~$ icd SAWC123
ubuntu@irods:~$ ilcs
/tempZone/home/alice/PeterTB/SAWC123:
SAWC123.bam
SAWC123_r1.fastq
SAWC123_r2.fastq
SAWC123.vcf
ubuntu@irods:~$

```

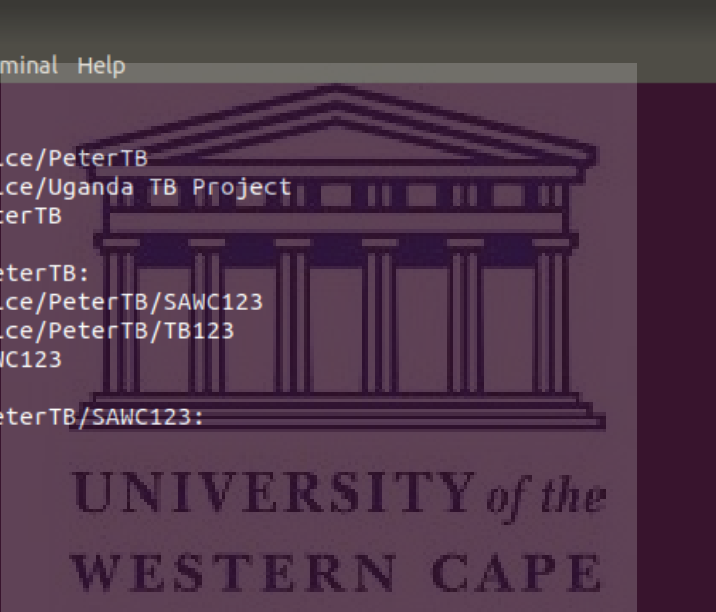


Figure 4.3: File storage location

#### 4.3.5 Installation

The source files for the *African Genome Archive* can be obtained from its Github repository using the following command on a Ubuntu operating system terminal:

---

```
$ git clone https://github.com/jamietyger/AfricanGenomeArchive.git
```

---

The installation and setup of the software was completed using the ‘Read Me’ guidelines located on the Github repository.

```

File Edit View Search Terminal Help
ubuntu@irods:~$ imeta ls -C SAWC123
AVUs defined for collection SAWC123:
attribute: accession_number
value: SRR16253647
units:
----
attribute: age
value: 66.0
units:
----
attribute: average_doc
value: 87.0
units:
----
attribute: BAMfilename
value: SAWC123.bam
units:
----
attribute: culture_complete_date
value: 42921.0
units:
----
attribute: culture_request_date
value: 42897.0
units:
----
attribute: culture_stored
value: yes
units:
----
attribute: date_dna_sent
value: 42926.0
units:
----
attribute: date_wgs_download
value: 42979.0
units:
----
attribute: decode
value: S123
units:
----
attribute: dna_extn_sucful
value: 42923.0
units:
----
attribute: dna_storage_loc
value: Box 11, pos 29
units:

```




Figure 4.4: Metadata for sample stored in the NoSQL database

## 4.4 Frontend

### 4.4.1 Web Structure

HTML was used to segment and structure the content of the archive being served by the flask server. Seven template HTML pages were created to present the various features of the archive. This included: home page, single project page, multi-project page, repository page, sample page, search page and a project upload page.

All pages were structured using the following pseudocode template:

---

```
# HTML Doc Tag
# Header Tag
# Stylesheet and javascript references
# Page Title
# NavBar---Menu links to Dashboard, Projects and Repository
  # searchBar input box with 'Search' Button
  # Links to Docs and User account
# Main Container for content
# Page footer
```

---

This pseudocode is the foundation of the '*public\_template.html*' file, which is a Jinja template containing variables and tags which can be replaced with values and manage the logic of the template. The Flask micro web framework is able to interpret and render the templates leading to modular and extensible web page design.

Every web page served by the archive software includes a navigation bar which includes the ability to collapse when the site is viewed on a smaller screen device. The items that are hidden can be viewed by clicking on the 'burger' icon in the top right hand corner which in turn opens a drop down menu containing the items.

Bulma v0.8.0, a free, open source cascading style sheet (CSS) framework was used to style the web application. The library is accessed using an external content distribution network (CDN) rather than downloaded and stored in a local directory.

#### 4.4.1.1 Web Application User Interface

The *African Genome Archive* was hosted on a virtual machine at SANBI and accessed via its assigned IP address. The index page of the archive software is depicted in Figure 4.5, featuring the navigation bar which contains links to the project pages, repository pages and search functionality.

The blue upload project button is located in the top right hand corner of the web page. Clicking on the button will direct the user to the upload project section of the website, where users presented with a form in which they need to enter metadata about the project and the storage repository relating to their institution (Figure 4.6). Once the user clicks the upload button, the project files and metadata are stored and processed in the iRods environment.

The top navigation bar contains a link to the ‘*Projects*’ section of the archive. As seen in Figure 4.7, the projects page displays all projects which have been uploaded to the archive. The user input project metadata is displayed, as well as the unique *Project ID*. Download buttons are provided to the user if they wish to download the entire project or metadata to their device.

When a user clicks on the highlighted *Project ID* for the project they wish to view, the user is then taken to the individual project page (Figure 4.8) which displays the samples relating to that project and metadata. A download button on the right hand side allows the user to download all related BAM, VCF and FASTQ files of the sample.

Figure 4.9 illustrates the sample metadata view, which is accessed by

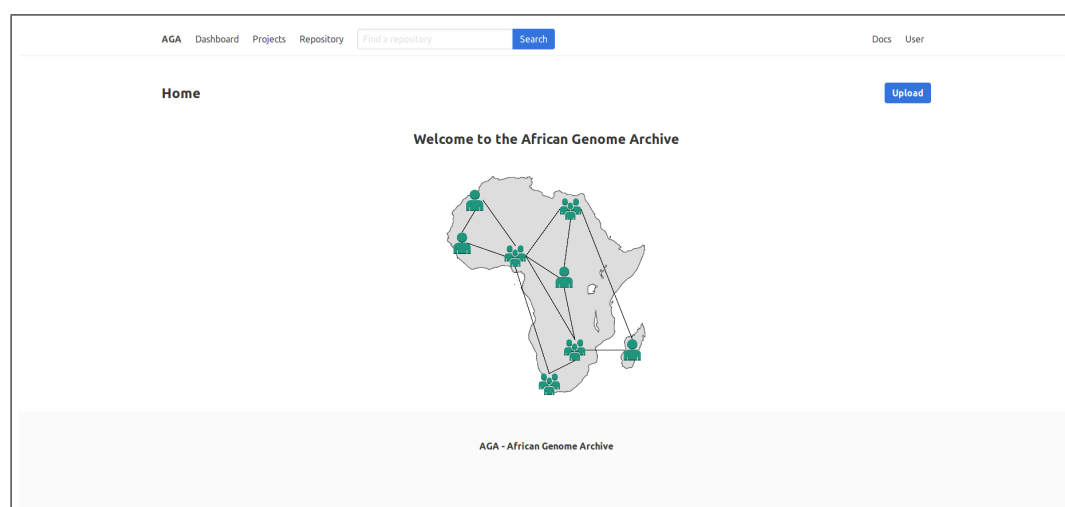


Figure 4.5: Genome Archive home page

Figure 4.6: Upload project page

Project ID	Project Name	Species	Category	Repository	Project	Metadata
10464	PeterTB	Pathogen	M.Tuberculosis	SANBI	↓	↓
10472	Uganda TB Project	Pathogen	M.Tuberculosis	Uganda	↓	↓

Figure 4.7: Browse projects page

clicking on the *Sample ID* on the project page. The entire metadata set for the sample is displayed and download buttons are generated for the associated data files. The metadata set includes non mandatory metadata fields.

The search results page (Figure 4.10) formats the keyword results which were submitted in the search box located in the navigation bar. Relevant sample files are displayed with links to their parent projects and metadata pages. Projects which are associated with the search term are also returned to the user.

AGA Dashboard Projects Repository Find a repository Search Docs User

**Uganda TB Project** Upload

Sample ID	Sample name	Sequencing Center	Accession Number	Download
10473	SAWC123	CDC	SRR16253647	Download
10479	TB123	TGEN	SRR16253648	Download

AGA - African Genome Archive

Figure 4.8: Individual project page

AGA Dashboard Projects Repository Find a repository Search Docs User

**SAWC123** Upload

Download	BAM File	VCF File	FASTQ - r1	FASTQ - r2
Download	Download	Download	Download	Download
sample_id	sample_DB	study_id	responsible_person	
SAWC123	SAWC	IS-0311	Pete Smith	
decode	culture_request_date	culture_complete_date	culture_stored	
S123	42897.0	42921.0	yes	
dna_extr_sucful	dna_storage_loc	date_dna_sent	nucleic_acid_concentration	
42923.0	Box 11, pos 29	42926.0	1354.251	
unit	purity_260_280	ug_sent	sequencing_center	
ng/ul	1.83	20.0	CDC	
date_wgs_download	accession_number	usap_run	average_doc	
42979.0	SRR16253647	42980.0	87.0	
path_res_dir	rflp_family	sitvit_family	age	
/home/mycobacteriology/Results/CDC/SAWC_Sep2017/	11.0	LAM3	66.0	
sex	isolate_date	sputum_smear	inh	
F	36224.0	3+	R	
rif	ofloxacin	etham	VCFfilename	
R	S	S	SAWC123.vcf	
FASTQ_r1filename	FASTQ_r2filename	BAMfilename	origin	
SAWC123_r1.fastq	SAWC123_r2.fastq	SAWC123.bam	/tempZone/home/alice/Uganda TB Project/	

AGA - African Genome Archive

Figure 4.9: Sample metadata view

AGA Dashboard Projects Repository Uganda Search Docs User

Browse Results - Uganda Upload

Project ID	Project Name	File ID	File Name	File Size
10472	SAWC123	10475	SAWC123.bam	0.0 Byte
10472	SAWC123	10477	SAWC123_r1.fastq	0.0 Byte
10472	SAWC123	10478	SAWC123_r2.fastq	0.0 Byte
10472	SAWC123	10476	SAWC123.vcf	0.0 Byte
10472	TB123	10480	TB123.vcf	0.0 Byte
10472	Uganda TB Project			

AGA - African Genome Archive

Figure 4.10: Search results

## 4.5 Tools

The following tools are acknowledged in their use for the completion of the software project:

- Visual Studio Code—A source-code editor developed by Microsoft,
- GitHub—An online Git repository for source control,
- Virtual Box—A free and open-source hosted hypervisor for x86 virtualization and
- Jupyter Notebooks.

## 4.6 Discussion

The Wellcome Trust Sanger Institute (WTSI) implementation of iRods as a data management system (Chiang et al., 2011) received positive feedback from users when accessing data, which may be linked to researchers having strong IT skills coming from a bioinformatics background. Presently the system is used to manage and access Binary Alignment/Map files which are then accessed by user pipelines for further analysis. Issues relating to data upload were found in performing multiple uploads (resolved by updating iRods and patch installation) and the lack of error messages providing feedback to users. Networking and hardware issues were also noted. The iRods middleware software is a mature software project with active development and a strong community. The

installation process is well documented and straightforward. The limitation of the iRods software is seen in its python client which is not updated as regularly as the central code base with very limited documentation of code, if any. Certain modules pertaining to object search functionality in the NoSQL database were inoperative, resulting in a custom workaround solution for the genome archive prototype. A metadata issue was also found in the annotation of float value data types to the relevant sample. The solution to this issue was to store all metadata values in string data type. The dependency on the python client is not ideal, given the importance and scale of the data involved. An alternative solution would be to utilize the C/C++ iRods client which has a vastly greater integration with the iRods software and code library.

Similarities to the African Genome Archive prototype are seen in the tagging of metadata to the sample files, however the WTSI implementation maintains a separate tracking database which is queried by Perl modules. The architecture of the African Genome is comparable to the EGA system provided by EBI-EMBL, in that users are submitting data and metadata to the archive using web input tools. The metadata catalog can be browsed by the public with the data accessed only by accredited users. The African Genome Archive prototype lacks the implementation of a data access committee seen in the EGA, limiting the usage of certain datasets due to ethical considerations.

## 4.7 Conclusion

This chapter summarized the software implementation used in developing the genome archive software. The next chapter will discuss the testing methodology and system evaluation.

# Chapter 5

## Testing Methodology

### 5.1 Introduction

The *African Genome Archive* aims to provide researchers with a platform that stores and manages data relating to their research projects. The archive facilitates the use of metadata standards, which enable researchers to actively discover newly uploaded datasets. This chapter discusses the testing methodology implemented to address the issues raised by the research questions noted in Chapter 1 and discussion of results.

### 5.2 Usability Testing

The goal of the testing phase is to perform rudimentary tasks on the system which are to be compared to existing methods used by researchers. The evaluation is then utilised in identifying ‘software bugs’, assessing whether the software requirements are met and in measuring the quality of the product before release. Usability testing is a popular technique introduced in the late 1980s, used to evaluate user performance and acceptance of systems (Wichansky, 2000). In effect, to analyse how suitable a design is for its target user group. In this regard, user testing is more commonly seen in the development process of primitive prototypes. Multiple inspection methods such as heuristic evaluation (Pinelle et al., 2008) and feature inspection are used in highlighting design issues in the software and interface (Nielsen et al., 1994).

#### 5.2.1 Cognitive Walkthrough

The Cognitive Walkthrough (CW) is a usability inspection method which is used to predict the ease of use of the software by simulating the user problem-solving process in a given scenario. The approach was developed in the 1990s and has seen widespread use in websites, automatic teller machines and pro-

programming languages (Blackmon and Bainbridge, 2004). The CW method has also faced resistance from researchers who tend to find this approach to be tedious and time consuming (Wechsung, 2014). Given that an *African Genome Archive* software would be utilized by researchers with varying expertise and experience, a CW evaluation of the three core software components (Figure 5.1) is suitable.

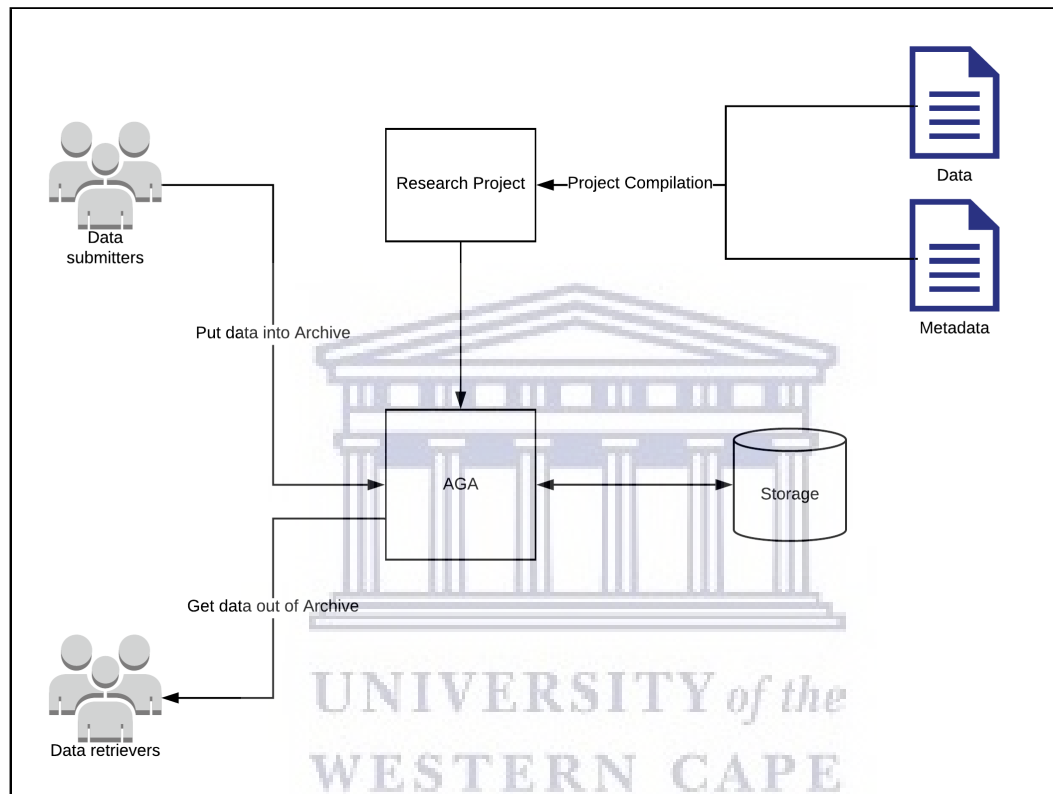


Figure 5.1: User testing experiment components

### 5.2.2 Experiment Definition

A two-phase approach described by Ghalibaf et al. (2018) was adopted when carrying out the CW evaluation. The first phase being the preparatory phase where the user scenario is outlined and relevant tasks and action sequences are established. This phase is then executed leading into the evaluation phase which identifies user highlighted issues with the software.

### 5.2.3 Preparation Phase

#### 5.2.3.1 Target User Definition

The target user definition is used to determine the expected user profile and attributes of the software users. Users of the archive are expected to be in the bioinformatics field and assuming the role of a data depositor or data retriever. Users are expected to have a basic level of computer literacy and a range in different knowledge and skill areas.

#### 5.2.3.2 Scenario Definition

The scenario definition is required to specify a representative and frequently faced user situation. A routine scenario for the genome archive is as follows:

*A researcher has a dataset with its accompanying metadata and wishes to upload, and store it on the Genome Archive. As soon as the upload is complete they wish to view their stored project in the archive and browse the sample metadata. The researcher then wishes to browse other data stored in the archive and carries out a search query on the archive. Once they have found the sample that they are looking for, they download it to their personal computer.*

#### 5.2.3.3 Action Sequence Definition:

The determined scenario is then analysed and separated into tasks which are performed in order to complete the scenario. Table 5.1 shows a list of tasks relating to the scenario defined. As shown in Table 5.2, Task 1 is comprised of multiple actions which are completed in sequence to achieve the task. The complete action sequences for tasks can be found in Appendix B.

Table 5.1: Scenario task list

Task Number	Task Description	No. of actions	User Role
1	Upload the resultant ZIP file to the archive	5	Data Depositor
2	Find the uploaded project in the archive	3	Data Retriever
3	View metadata relating to sample from uploaded project	2	Data Retriever
4	Find a project in the archive using the search functionality	5	Data Retriever
5	Download a sample file from the newly discovered project	4	Data Retriever
Total		19 actions	

Table 5.2: Task 1 action sequence

No.	Description	Type
1.1	Click on 'Upload' on the home page	User action
1.2	Display metadata input form	System response
1.3	Enter information into form	User action
1.4	Click on 'Choose file' to select Project ZIP file	User action
1.5	Click on upload	User action

## 5.2.4 Evaluation Phase

The evaluation phase analyses the user interaction with the software by examining the action sequence and tasks outlined in the preparation phase.

### 5.2.4.1 Measurement Criteria

The measuring criteria assess the ease of use of the software, by inspecting each action performed by the user. Blackmon et al. (2002) outlined in their paper, four questions to be employed when conducting a cognitive walkthrough:

1. Will the user try and achieve the right outcome?
2. Will the user notice that the correct action is available to them?
3. Will the user associate the correct action with the outcome they expect to achieve?
4. If the correct action is performed; will the user see that progress is being made towards their intended outcome?

The first question analyses whether the user has the correct conceptual model and performs the correct action at the correct time. Question 2 refers

to action being visible to the user, evaluating whether they can see what they need to do. The third question makes reference to labelling and signifiers, in that the user can recognize the action as the correct one. Question 4 references the user's understanding of the feedback provided by the software.

#### 5.2.4.2 Analysis Methodology

The defined scenario was then performed by carrying out the established tasks. A qualitative and quantitative approach was taken whereby each task in the action sequence was evaluated using the measure criteria questions, with a 'Yes' or 'No' response. A 'No' response mandated a recommendation or practical solution. The results and findings are presented in the experiment results section of this thesis.

#### 5.2.5 African Genome Archive ontology

The exploratory analysis of the PATRIC metadata ontology in Chapter 3, influenced the construction of the ontology implemented for *Mycobacterium tuberculosis* data deposited in the genome archive. Metadata fields were chosen to form the core components of a sample collected and uploaded to the system. Table 5.3 describes the key fields which are mandatory for users to submit in their sample metadata.

Table 5.3: Mandatory metadata fields

Field	Description	Type
Filename (BAM,VCF,FASTQ)	The name of the assembly file relating to the sample (Must match exactly)	String
Sample ID	Unique Identifier for sample	String
Collection Year	Year sample was collected	Integer
Collection Month	Month sample was collected	Integer
Collection Day	Day Sample was collected	Integer
Country	Name of country where the sample was collected	String

Researchers are able to include their own metadata fields if they so wish. Examples include drug resistance, accession numbers, sequencing Centre, strain and sample source.

The Genome Archive requests metadata input from the user when submitting a project. The requested metadata is listed in Table 5.4.

Table 5.4: Project submission form metadata

Field	Description	Type
Project Name	Name of project given by user	String
Species	Name of species relating to the dataset (e.g. Pathogen)	String (Predefined values)
Category	The specific sample category relating to the dataset (e.g. M.Tuberculosis)	String (Predefined values)
Repository	Storage location for the dataset	String (Predefined values)

Figure 5.2 is derived from the metadata fields mentioned in Table 5.3 and Table 5.4. The figure represents an ontological view of the *Mycobacterium tuberculosis* projects hosted in the archive and the relationship with their associated samples.

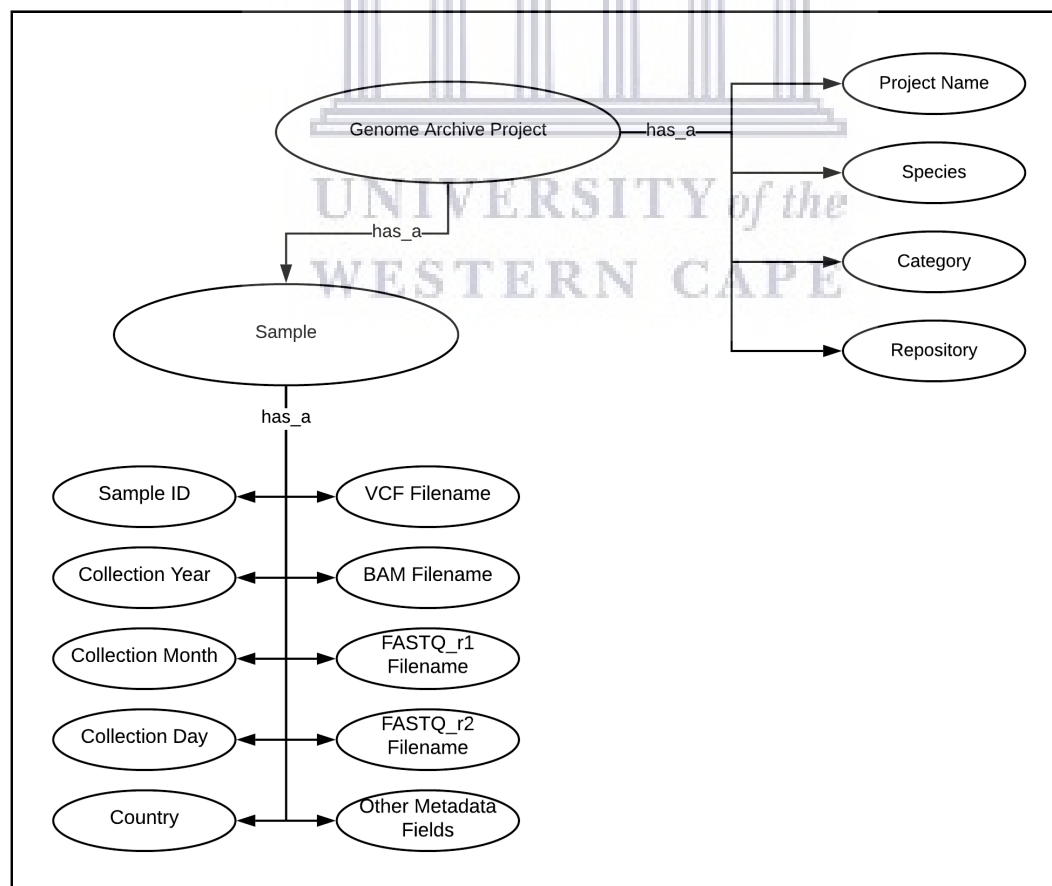


Figure 5.2: Genome Archive ontological view

### 5.2.6 Experiment Results

A total of five tasks with altogether 19 actions were performed in order to achieve the defined scenario. See Tables 5.1 and 5.2 and also Tables B.1a to B.1e in Appendix B. The evaluation methodology was executed, with 76 questions being answered, resulting in 9 issues being raised with the software.

#### 5.2.6.1 Quantitative Results

Table 5.5 shows the number of identified problems which were highlighted using the measurement criteria questions and their percentages.

Table 5.5: Issues found in relation to criteria questions

Issues	Question 1	Question 2	Question 3	Question 4
Number of issues found	0	0	2	7
Percentage (%)	0	0	22	78

Table 5.6 depicts the percentage of problems detected in each task, with their percentages.

Table 5.6: Issues found in relation to tasks

Issues	Task 1	Task 2	Task 3	Task 4	Task 5
Number of issues found	2	2	1	1	3
Percentage (%)	22	22	11	11	33

#### 5.2.6.2 Qualitative Results

The qualitative findings of the evaluation are seen in Table 5.7, with a description of the issue and a suggested solution provided. Duplicate issues were removed.

### 5.2.7 Discussion

The CW method of evaluating software usability enabled a systematic approach to assess the suitability of the *African Genome Archive* system. The methodology does not consider the validity and accuracy of data and metadata stored, however, it does present the usefulness of the software to users. Software issues are highlighted to guide further development and research of the software. Question 4 presented the highest percentage of issues highlighted with the software. The main source of which is the lack of user feedback when

Table 5.7: Results of qualitative evaluation

<b>Problem Description</b>	<b>Recommendation</b>
Lack of user understanding of input boxes	Implement Tooltips
No user feedback when project uploaded	Redirect user to status page displaying upload status
Project page metadata unclear	Improved labelling with descriptions
Sample page metadata unclear	Provide detailed explanation and meanings of metadata terms
Download not labeled on project page	Add label
User download feedback not clear	Add status message when user downloads file

executing a task. Question 3 highlighted issues with users failing to recognize actions which are caused by inadequate labelling and metadata descriptions. The result of these problems are increased user dissatisfaction and reluctance to use the system. Questions 1 and 2 noted no issues which suggests that the core conceptual model used is consistent and that the actions available to the user are clearly visible. The majority of issues seen were in relation to Task 5, which lacked metadata explanation, labelling and user feedback. The reasoning for the large number of issues can be attributed to the large number of actions required to complete the task and task complexity. Due to the fact that several issues were raised by the CW methodology, it can be suggested that further evaluation methods be used in order to complement the existing evaluation and to form a comparison of issues raised.

### 5.3 Conclusion

This chapter summarized the experiment design and testing methodology used in evaluating the genome archive software. The next chapter will discuss the summary and future work of the thesis.

# Chapter 6

## Summary and Future Improvements

### 6.1 Introduction

The results of this project have laid the foundation for further investigation into the development of multi-site distributive data storage for bioinformatics researchers. In the continued expansion of research projects and data growth on the African continent, it is evident that developing an archive platform will aid researcher efforts in delivering their scientific analysis and data discovery. Recent projects such as the H3A Africa Data Archive (Parker et al., 2019), have been built to sustain human genomic and phenotypic data generated by their projects and aid data submission to the European Genome-Phenome Archive.

### 6.2 Summary

The *African Genome Archive* project was aimed at providing a solution to the ever escalating issue of research data management. The issues alluded to in Chapter 1, highlighted a need for effective data management for small research groups and the implementation of metadata standards. A proof-of-concept archive platform was designed and developed as an aid for researcher metadata and data storage. The genome archive prototype was deployed on an existing private researcher environment which provided an opportunity to evaluate its value in a functioning research data workflow.

The observations made in this thesis show that the development of an *African Genome Archive* is possible from a technical implementation. iRods was successfully implemented in brokering data and metadata for researchers. Nevertheless, more consensus is needed amongst the research community in terms of metadata standards. This is a continued effort and will improve with increased networking and collaboration amongst institutions. On the

data packaging process, users were capable of preparing their allocated data and metadata for upload to the archive. There is room for specific tools to be developed to aid this process, however that would be accompanied with increased maintenance.

The archive featured a simple to use web interface which in turn made it easier for researchers to store and link data to metadata. Researcher use of the software was consistently in line with the proposed conceptual model and users actions required being visible. This was aided by the use of a graphical user interface and labelling of actions. Issues regarding the use of the correct action and more prominently the lack of user feedback in certain circumstances limited user progress towards completing the task.

The *African Genome Archive* project achieved the research goals set out in Chapter 1, in creating an iRods based storage platform which supported *Mycobacterium tuberculosis* data and metadata. A technical issue is raised by the dependence on using iRods as a middleware component in its long term sustainability. Due to the open source nature of iRods, its development and support is heavily reliant on its community assistance and growth. The web front end and search functionality were successfully implemented, and the software made available to potential users on its GitHub repository.

The project has demonstrated that the creation of an *African Genome Archive* is feasible, in allowing researchers to deposit data and metadata in an environment that makes data discovery possible. It enables researchers to be less reliant on private data storage, allowing data to be shared and used in continuous research efforts. The increased development of community data standards can further aid this process and research study amalgamation projects on the continent.

### **6.3 Future Improvements**

With the current research environment tending towards distributed cloud architectures for analysis workflows, various improvements can be made to the Genome Archive system:

### **Utilize Cloud Storage Environments**

This would enable researchers to leverage cloud based data storage solutions where localized storage is insufficient.

### **Security Limitations**

Presently, the archive platform only supports open access pathogen data which does not have any ethical restrictions. In the future, should human genetic data be stored on the archive, increased attention to data security and access would be required. A solution would be to develop an access control mechanism whereby privileges are granted by a committee overseeing data access and ethical clearance submissions. Encryption could also be used in securing sensitive data.

### **Data Retrieval Interface**

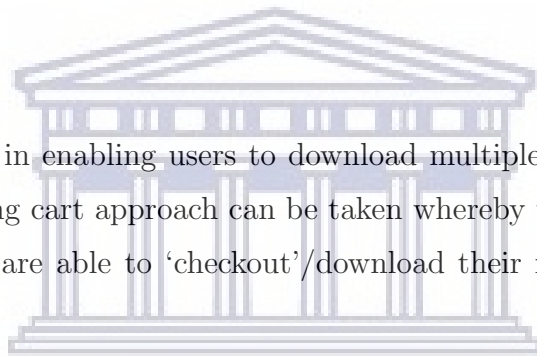
Improvements can be made in enabling users to download multiple datasets at the same time. A shopping cart approach can be taken whereby users add datasets to their 'cart' and are able to 'checkout'/download their requested files.

### **Metadata standards development**

Collaboration input from various research groups would be beneficial to the establishment of minimal data standards. Greater community consensus is required before an implementation can be considered. Future versions of the Archive software will look to extend collaboration with existing metadata standards communities such as The Open Biological and Biomedical Ontologies (OBO) Foundry to avoid development in isolation. Detailed descriptions and explanations of metadata terms can also be included when interacting with the user interface.

### **Software Integration**

There is potential to integrate the archive for use with data analysis platforms such as the Galaxy Project, where data from the archive is utilized in multiple workflows.



### **User Feedback Interface**

The inclusion of system status prompts to inform the user of their progress or task completion.



UNIVERSITY *of the*  
WESTERN CAPE

# Appendix A

## Ansible Tasks

Listing A.1: Ansible Tasks - main.yml

---

```
---

- name: get irods apt key
  apt_key:
    url: https://packages.irods.org/irods-signing-key.asc
    state : present
- name: add irods apt repository
  apt_repository:
    repo: "deb [arc=amd64] https://packages.irods.org/apt/ {{
        ansible_distribution_release }} main"
    filename: renci-irods.list
- name: install irods
  apt:
    name: "{{item}}"
    state: present
    update_cache: yes
  loop:
    - irods-server
    - irods-database-plugin-postgres
- name: correct host name
  block:
    - name: run hostname command
      shell: "hostname {{ ansible_hostname.split('.') [0]
          }}.sanbi.ac.za"
      changed_when: false
```



```
- name: change hostname file
  lineinfile:
    path: /etc/hostname
    regexp: '.*'
    line: "{{ ansible_hostname.split('.') [0] }}.sanbi.ac.za"

- name: correct hosts file
  template:
    src: "{{role_path}}/files/hosts.j2"
    dest: /etc/hosts
    owner: root
    group: root
    mode: 0644

- name: deploy config files
  template:
    src: "{{role_path}}/files/setup_input.txt.j2"
    dest: "/home/ubuntu/setup_input.txt"

- name: check if irods is setup
  stat:
    path: /home/ubuntu/complete
  register: do_install

- name: irods post install
  shell: "python /var/lib/irods/scripts/setup_irods.py <
        setup_input.txt"
  register: failed
  failed_when: false
  when: not do_install.stat.exists

- set_fact:
  failed:
    rc: 1
  when: do_install.stat.exists
```



```
- copy:  
  content: ""  
  dest: "/home/ubuntu/complete"  
  force: no  
  when: "failed.rc == 0"
```

---



UNIVERSITY *of the*  
WESTERN CAPE

# Appendix B

## Action Sequences for Tasks

Table B.1: Action sequence tables for scenario tasks

(a) Task 1 action sequence

No.	Description	Type
1.1	Click on 'Upload' on the home page	User action
1.2	Display metadata input form	System response
1.3	Enter information into form	User action
1.4	Click on 'Choose file' to select Project ZIP file	User action
1.5	Click on upload	User action

(b) Task 2 action sequence

No.	Description	Type
2.1	Click on Projects in navigation bar	User action
2.2	Click on Project ID number	User action
2.3	Display Project page	System response

(c) Task 3 action sequence

No.	Description	Type
3.1	Click on Sample ID number	User action
3.2	Display Sample metadata page	System response

(d) Task 4 action sequence

No.	Description	Type
4.1	Type search term in search box	User action
4.2	Click on search button	User action
4.3	Display search results	System response
4.4	Click on Project ID	User action
4.5	Display Project page	System response

(e) Task 5 action sequence

No.	Description	Type
5.1	Click on Sample ID	User action
5.2	Display sample metadata page	System response
5.3	Click on download button for sample	User action
5.4	Serve file to user	System response

# Bibliography

Adedokun, B. O., Olopade, C. O., and Olopade, O. I. (2016). Building local capacity for genomics research in Africa: recommendations from analysis of publications in Sub-Saharan Africa from 2004 to 2013. *Global Health Action*, 9(1):31026.

Arita, M. (2008). A pitfall of wiki solution for biological databases. *Briefings in Bioinformatics*, 10(3):295–296.

Benson, D. A., Cavanaugh, M., Clark, K., Karsch-Mizrachi, I., Ostell, J., Pruitt, K. D., and Sayers, E. W. (2018). GenBank. *Nucleic Acids Research*, 46(D1):D41–D47.

Bernstein, F. C., Koetzle, T. F., Williams, G. J., Meyer Jr, E. F., Brice, M. D., Rodgers, J. R., Kennard, O., Shimanouchi, T., and Tasumi, M. (1977). The Protein Data Bank: a computer-based archival file for macromolecular structures. *Journal of Molecular Biology*, 112(3):535–542.

BIG Data Center Members (2016). The big data center: from deposition to integration to translation. *Nucleic Acids Research*, 45(D1):D18–D24.

Bilofsky, H. S. and Christian, B. (1988). The GenBank<sup>®</sup> genetic sequence data bank. *Nucleic Acids Research*, 16(5):1861–1863.

Biotechnology and Biological Sciences Research Council (2019). *BBSRC Template*. BBSRC, UK.

Blackmon, M. and Bainbridge, W. (2004). Cognitive walkthrough. *Encyclopedia of Human-Computer Interaction*, 2:104–107.

Blackmon, M. H., Polson, P. G., Kitajima, M., and Lewis, C. (2002). Cognitive walkthrough for the web. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 463–470.

Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., Stoeckert, C., Aach, J., Ansorge, W., Ball, C. A., Causton, H. C., Gaasterland, T., Glenisson, P., Holstege, F. C., Kim, I. F., Markowitz, V., Matese,

- J. C., Parkinson, H., Robinson, A., Sarkans, U., Schulze-Kremer, S., Stewart, J., Taylor, R., Vilo, J., and Vingron, M. (2001). Minimum information about a microarray experiment (MIAME)—toward standards for microarray data. *Nature Genetics*, 29(4):365–371.
- Caplan, P. (2003). *Metadata Fundamentals for All Librarians*. American Library Association.
- Chiang, G.-T., Clapham, P., Qi, G., Sale, K., and Coates, G. (2011). Implementing a genomic data management system using iRods in the Wellcome Trust Sanger Institute. *BMC Bioinformatics*, 12(361):1–8.
- Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387.
- Cohen, S. M. (2000). Aristotle’s metaphysics. *Stanford Encyclopedia of Philosophy*.
- Côté, R. G., Jones, P., Apweiler, R., and Hermjakob, H. (2006). The ontology lookup service, a lightweight cross-platform tool for controlled vocabulary queries. *BMC Bioinformatics*, 7(97):1–7.
- CrowdFlower (2017). *2017 Data Scientist Report*. Figure Eight Inc.
- Dai, L., Gao, X., Guo, Y., Xiao, J., and Zhang, Z. (2012). Bioinformatics clouds for big data manipulation. *Biology Direct*, 7(1):43.
- de Vries, J., Williams, T. N., Bojang, K., Kwiatkowski, D. P., Fitzpatrick, R., and Parker, M. (2014). Knowing who to trust: exploring the role of ‘ethical metadata’ in mediating risk of harm in collaborative genomics research in Africa. *BMC Medical Ethics*, 15(1):62.
- Deus, H. F., Stanislaus, R., Veiga, D. F., Behrens, C., Wistuba, I. I., Minna, J. D., Garner, H. R., Swisher, S. G., Roth, J. A., Correa, A. M., et al. (2008). A semantic web management model for integrative biomedical informatics. *PloS one*, 3(8):e2946.
- Dugan, V. G., Emrich, S. J., Giraldo-Calderón, G. I., Harb, O. S., Newman, R. M., Pickett, B. E., Schriml, L. M., Stockwell, T. B., Stoeckert Jr,

- C. J., Sullivan, D. E., et al. (2014). Standardized metadata for human pathogen/vector genomic sequences. *PloS one*, 9(6):e99979.
- Edwards, P. N., Mayernik, M. S., Batcheller, A. L., Bowker, G. C., and Borgman, C. L. (2011). Science friction: Data, metadata, and collaboration. *Social Studies of Science*, 41(5):667–690.
- European Genome-Phenome Archive (2017a). EGA Rest API. <https://ega-archive.org/submission/programmatic-submissions>.
- European Genome-Phenome Archive (2017b). EGACryptor. [https://www.ebi.ac.uk/ega/submission/tools/EGA\\_webin\\_data\\_uploader](https://www.ebi.ac.uk/ega/submission/tools/EGA_webin_data_uploader).
- Fanelli, D. (2009). How many scientists fabricate and falsify research? A systematic review and meta-analysis of survey data. *PloS one*, 4(5):e5738.
- Gene Ontology Consortium (2004). The gene ontology (go) database and informatics resource. *Nucleic Acids Research*, 32(suppl.1):D258–D261.
- Genome Sequence Archive (2019). Documents for GSA. <http://gsa.big.ac.cn/documents>.
- Gewin, V. (2016). Data sharing: An open mind on open data. *Nature*, 529(7584):117–119.
- Ghalibaf, A. K., Jangi, M., Habibi, M. R. M., Zangouei, S., and Khajouei, R. (2018). Usability evaluation of obstetrics and gynecology information system using cognitive walkthrough method. *Electronic Physician*, 10(4):6682.
- Gibson, G. A., Nagle, D. F., Amiri, K., Chang, F. W., Feinberg, E. M., Gobioff, H., Lee, C., Ozceri, B., Riedel, E., Rochberg, D., et al. (1997). File server scaling with network-attached secure disks. In *ACM SIGMETRICS Performance Evaluation Review*, volume 25, pages 272–284.
- Gingeras, T. R. and Roberts, R. J. (1980). Steps toward computer analysis of nucleotide sequences. *Science*, 209(4463):1322–1328.
- González-Beltrán, A., Maguire, E., Rocca-Serra, P., and Sansone, S.-A. (2012). The open source ISA software suite and its international user com-

- munity: knowledge management of experimental data. *EMBnet Journal*, 18(B):35–37.
- Greenberg, J. (2003). Metadata and the world wide web. *Encyclopedia of Library and Information Science*, 3:1876–1888.
- Greenberg, J. (2005). Understanding metadata and metadata schemes. *Cataloging and Classification Quarterly*, 40(3-4):17–36.
- Hedges, M., Blanke, T., and Hasan, A. (2009). Rule-based curation and preservation of data: A data grid approach using iRods. *Future Generation Computer Systems*, 25(4):446–452.
- Heery, R. and Anderson, S. (2005). *Digital Repositories Review*. Joint Information Systems Committee.
- Hey, T. and Trefethen, A. (2003). The data deluge: An e-science perspective. *Grid Computing: Making the Global Infrastructure a Reality*, pages 809–824.
- Hey, T. and Trefethen, A. E. (2005). Cyberinfrastructure for e-science. *Science*, 308(5723):817–821.
- Huang, H. and Qin, J. (2013). Understanding metadata functional requirements in genome curation work. *Proceedings of the American Society for Information Science and Technology*, 50(1):1–4.
- iRods Consortium (2018). iRods python client. <https://github.com/irods/python-irodsclient>.
- isatools (2019). MAGE to ISA converter. <https://isa-tools.org/tag/isaconverter/index.html>.
- Jones, M. B., Berkley, C., Bojilova, J., and Schildhauer, M. (2001). Managing scientific metadata. *IEEE Internet Computing*, 5(5):59–68.
- Kless, D., Milton, S., and Kazmierczak, E. (2012). Relationships and relata in ontologies and thesauri: Differences and similarities. *Applied Ontology*, 7(4):401–428.

- Lappalainen, I., Almeida-King, J., Kumanduri, V., Senf, A., Spalding, J. D., Saunders, G., Kandasamy, J., Caccamo, M., Leinonen, R., Vaughan, B., Laurent, T., Rowland, F., Marin-Garcia, P., Barker, J., Jokinen, P., Torres, A., de Argila, J., Llobet, O., Medina, I., Puy, M., Alberich, M., de la Torre, S., Navarro, A., Paschall, J., and Flicek, P. (2015). The European Genome-Phenome Archive of human data consented for biomedical research. *Nature Genetics*, 47(7):692–695.
- Lawrence, B., Lowry, R., Miller, P., Snaith, H., and Woolf, A. (2008). Information in environmental data grids. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1890):1003–1014.
- Levine, D. M., Dutta, N. K., Eckels, J., Scanga, C., Stein, C., Mehra, S., Kaushal, D., Karakousis, P. C., and Salamon, H. (2015). A tuberculosis ontology for host systems biology. *Tuberculosis*, 95(5):570–574.
- Morris, C., Andreetto, P., Banci, L., Bonvin, A. M., Chojnowski, G., del Cano, L., Carazo, J. M., Conesa, P., Daenke, S., Damaskos, G., et al. (2019). West-life: A virtual research environment for structural biology. *Journal of Structural Biology*: X, 1:100006.
- Nature Cell Biology (2008). Standardizing data. *Nature Cell Biology*, 10(10):1123–1124.
- Nielsen, J., Mack, R. L., et al. (1994). *Usability Inspection Methods*, volume 1. Wiley New York.
- Nieroda, L., Maas, L., Thiebes, S., Lang, U., Sunyaev, A., Achter, V., and Peifer, M. (2019). iRods metadata management for a cancer genome analysis workflow. *BMC Bioinformatics*, 20(1):29.
- Noy, N. F., Shah, N. H., Whetzel, P. L., Dai, B., Dorf, M., Griffith, N., Jonquet, C., Rubin, D. L., Storey, M.-A., Chute, C. G., et al. (2009). Bioportal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Research*, 37(suppl\_2):W170–W173.

- Parker, Z., Maslamoney, S., Meintjes, A., Botha, G., Panji, S., Hazelhurst, S., and Mulder, N. (2019). Building infrastructure for African human genomic data management. *Data Science Journal*, 18(1).
- Pinelle, D., Wong, N., and Stach, T. (2008). Heuristic evaluation for games: usability principles for video game design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1453–1462.
- Rayner, T. F., Rocca-Serra, P., Spellman, P. T., Causton, H. C., Farne, A., Holloway, E., Irizarry, R. A., Liu, J., Maier, D. S., Miller, M., et al. (2006). A simple spreadsheet-based, miame-supportive format for microarray data: Mage-tab. *BMC Bioinformatics*, 7(1):489.
- Robbins, R. J. (1996). Bioinformatics: Essential infrastructure for global biology1. *Journal of Computational Biology*, 3(3):465–478.
- Sansone, S.-A., Rocca-Serra, P., Brandizi, M., Brazma, A., Field, D., Fostel, J., Garrow, A. G., Gilbert, J., Goodsaid, F., Hardy, N., et al. (2008). The first RSBI (ISA-TAB) workshop: “Can a simple format work for complex studies?”. *OMICS A Journal of Integrative Biology*, 12(2):143–149.
- Sansone, S.-A., Rocca-Serra, P., Tong, W., Fostel, J., Morrison, N., Jones, A. R., and RSBI Members (2006). A strategy capitalizing on synergies: the reporting structure for biological investigation (RSBI) working group. *OMICS A Journal of Integrative Biology*, 10(2):164–171.
- Schuler, G. D., Epstein, J. A., Ohkawa, H., and Kans, J. A. (1996). Entrez: Molecular biology database and retrieval system. *Methods in Enzymology*, 266:141–162.
- Schulz, S., Kumar, A., and Bittner, T. (2006). Biomedical ontologies: what part-of is and isn’t. *Journal of Biomedical Informatics*, 39(3):350–361.
- Shankar, R., Parkinson, H., Burdett, T., Hastings, E., Liu, J., Miller, M., Srinivasa, R., White, J., Brazma, A., Sherlock, G., Stoeckert Jr., C. J., and Ball, C. A. (2010). Annotare—a tool for annotating high-throughput biomedical investigations and resulting data. *Bioinformatics*, 26(19):2470–2471.

- Sirugo, G., Williams, S. M., and Tishkoff, S. A. (2019). The missing diversity in human genetic studies. *Cell*, 177(1):26–31.
- Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldberg, L. J., Eilbeck, K., Ireland, A., Mungall, C. J., the OBI Consortium, Leontis, N., Rocca-Serra, P., Ruttenberg, A., Sansone, S.-A., Scheuermann, R. H., Shah, N., Whetzel, P. L., , and Lewis, S. (2007). The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology*, 25(11):1251.
- Stephens, Z. D., Lee, S. Y., Faghri, F., Campbell, R. H., Zhai, C., Efron, M. J., Iyer, R., Schatz, M. C., Sinha, S., and Robinson, G. E. (2015). Big data: Astronomical or Genomical? *PLoS biology*, 13(7):e1002195.
- Wang, Y., Song, F., Zhu, J., Zhang, S., Yang, Y., Chen, T., Tang, B., Dong, L., Ding, N., Zhang, Q., et al. (2017). Gsa: genome sequence archive. *Genomics, Proteomics and Bioinformatics*, 15(1):14–18.
- Warner, J. L., Jain, S. K., and Levy, M. A. (2016). Integrating cancer genomic data into electronic health records. *Genome Medicine*, 8(1):113.
- Wattam, A. R., Abraham, D., Dalay, O., Disz, T. L., Driscoll, T., Gabbard, J. L., Gillespie, J. J., Gough, R., Hix, D., Kenyon, R., et al. (2013). Patric, the bacterial bioinformatics database and analysis resource. *Nucleic Acids Research*, 42(D1):D581–D591.
- Wechsung, I. (2014). An evaluation framework for multimodal interaction. *T-Labs Series in Telecommunication Services*, 10:978–3.
- Wichansky, A. M. (2000). Usability testing in 2000 and beyond. *Ergonomics*, 43(7):998–1006.
- Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., et al. (2016). The fair guiding principles for scientific data management and stewardship. *Scientific Data*, 3.

- Wood, F. and Ginter, T. (2008). Evolution and implementation of the CDISC study data tabulation model (SDTM). *Pharmaceutical Programming*, 1(1):20–27.
- Wrzeszcz, M., Opiola, L., Zemek, K., Kryza, B., Dutka, L., Słota, R., and Kitowski, J. (2017). Effective and scalable data access control in onedata large scale distributed virtual file system. *Procedia Computer Science*, 108:445–454.

