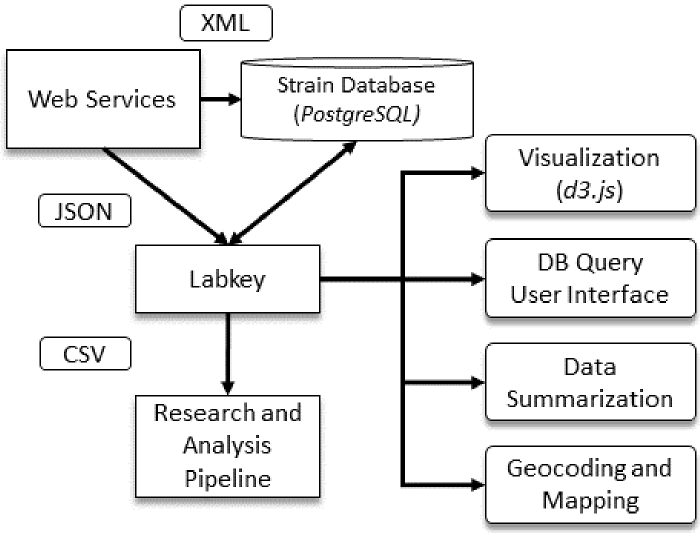
System architecture



This toolkit is built on the LabKey data platform, an open-source data management platform designed for biological data, and uses a PostgreSQL database designed for capture of metadata useful for disease outbreak investigations. The toolkit also employs D3.js, R, and open-source MITRE geospatial tools.

Here, we provide a Java program that parses xml files from NCBI describing data from the BioSample and BioProject databases. We also provide a LabKey module that, when placed into LabKey's `External Modules` folder will allow projects to be created that enable a user to interact with SRA data from within LabKey.

System requirements

Hardware requirements

- Linux server

Software requirements

- Java (version 1.7.0_02)
- Python (version 2.7.3)
- Shell (bash)
- Awk (version 3.1.8)
- JavaScript
- wget (version 1.13.4)
- PostgreSQL (version 9.1)

Note: The system was tested under the software with version provided in parentheses. Other versions may or may not work.

Project organization

`JAVA`

There are 26 java files and 3 jar files

This code is used to create and update a Posrgres Database with the XML metadata downloaded from NCBI BioSample and NCBI BioProject.

The names of 26 java files required for the project

- BioSampleParser.java
- BooleanColumn.java
- CharColumn.java
- Check_Host.java
- Collection_Owner_Table.java
- Collection_Table.java
- Cross_Reference_Table.java
- GenericTableColumn.java
- Human_Host_Table.java
- IntegerColumn.java
- Non_Human_Host_Table.java
- NumericColumn.java
- Owner_Table.java
- Project_Publication_Table.java
- Project_Sample_Table.java
- Project_Table.java
- Sample_Table.java
- Study_Method_Table.java
- Submitter_Table.java
- TableColumn.java
- TableColumnTypeException.java
- TableRow.java
- TableSQL.java
- TextColumn.java
- TimestampColumn.java
- VarcharColumn.java

The names of 3 jar files required for the project

- commons-lang3-3.1.jar
- xom-1.2.8.jar
- postgresql-9.1-902.jdbc4.jar

`scripts`

There are a total of 10 files in this directory

- BEGIN
- DataUpdate.sh
- DataUpload.sh
- END
- filter.awk
- geomap.tsv
- getBioSampleID.py
- getIds.awk
- mapBioSampleBioProjectIDs.py
- split_xml.awk

Installation

1) Update connectToDatabase in BioSampleParser.java to correspond to the installation of postgres that has been defined in your environment

```
connection = DriverManager.getConnection("jdbc:postgresql://SERVERNAME/DATABASE_NAME","USERNAME", "PASSWORD");
```

2) Compile the java code with the command

```
javac *.java
```

3) The resulting class files should be placed in the `biosampleparser` directory to allow for invocation

4) I. Run install.py: It creates a config.properties file. Please answer the questions with results that correspond to your environment. This is a sample run

```
Please select where the biosample will be downloaded:/path/to/biosample/download
You entered /path/to/biosample/download.  Is this correct? (Y or N)y
Please select where the bioproject will be downloaded:/path/to/bioproject/download
You entered /path/to/bioproject/download.  Is this correct? (Y or N)y
Please select where the scripts are:/path/to/scripts
You entered /path/to/scripts.  Is this correct? (Y or N)y
Please select directory where the mapping between Samples and Projects will go:/path/to/mapping
You entered /path/to/mapping.  Is this correct? (Y or N)y
Please select where the compiled JAVA/biosampleparser directory is
Should not include biosampleparser/ directory:/path/to/JAVA
You entered /path/to/JAVA.  Is this correct? (Y or N)y
Please enter the server where postgres is:localhost
You entered localhost.  Is this correct? (Y or N)y
Please enter the name of the database:db_name
You entered db_name.  Is this correct? (Y or N)y
Please enter username of the database:username
You entered username.  Is this correct? (Y or N)y
Please enter the username's password:password
You entered password.  Is this correct? (Y or N)y
```

II. After these paths have been configured, and the java code has been compiled, DataUpload.sh is the script which executes all the necessary helper programs to populate the database with the latest information from BioSample and BioProject. It calls DataDownload.sh (to download files), DataSplit.sh (to split the files to a manageable size for Java) DataMapping.sh (to map BioSample IDs to BioProject IDs) and DataUpdate.sh (to populate the posrgresql database). I.e. running ./DataUpload from the scripts/ directory should populate the database with all the information.

After these paths have been configured, and the java code has been compiled, `DataUpload.sh` is the script that executes all the necessary helper programs to populate the database with the latest information from BioSample and BioProject. `DataUpdate.sh` performs the same tasks, but does not download new XML files from BioSample or BioProject.

Connecting LabKey to Postgres

The LabKey project provides instructions that will allow you to connect LabKey to the PostgreSQL database that is generated with the provided Java program.

LabKey Module

We provide a LabKey module that is designed to work with the database generated with the provided java program. For more on modules please refer to the LabKey documentation.

Within the labkey module, located at `DataTools`, there are two webpart html files in `views`. These files, called `geodata.html` and `metadata.html`. Within these files, there are two code snippets that will need to be modified to work with the name of the schema of your database:

```
    _qwp1 = new LABKEY.QueryWebPart({renderTo       : "grid",
                                     title          : "SRA Search",
                                     schemaName     : <DATABASE_NAME>,
                                     dataRegionName : "metaDataRegion",
                                     sql            : _query.toString()
                                    });
```

and

```
    LABKEY.Query.GetData.getRawData({source  : {type      : 'sql',
                                                schemaName: <DATABASE_NAME>,
                                                sql       : _query.toString()
                                               },
                                     success : onSuccess,
                                     failure : onError
                                    });
```

Replace `<DATABASE_NAME>` with the name of your scheme, which defaults to `bioatt`.