igsr$_{archive}$

Release 0.1.0

Sep 13, 2023

Contents:

1 Dependencies **1**

2 Installation **3**

3 Usage **5**
- 3.1 Settings file . . . 5
- 3.2 Load files . . . 6
  - 3.2.1 Errors . . . 7
- 3.3 Delete files . . . 7
- 3.4 Archive files . . . 8
  - 3.4.1 Prerequisites . . . 8
  - 3.4.2 Errors . . . 9
- 3.5 Dearchive files . . . 9
- 3.6 Move files . . . 10

4 Indices and tables **13**

CHAPTER 1

Dependencies

You will need to have the following in your system:

- Python (>=3.6)
- curl (https://en.wikipedia.org/wiki/CURL)
- A MYSQL server hosting a database with the RESEQTRACK schema. This database is basically used for tracking the files produced in IGSR.

CHAPTER 2

Installation

This codebase requires Python (3.6.0 or later) and is used to, among other things, interact programmatically with the FIle REplication (FIRE) archive implemented in the EMBL-EBI.

To install this project:

```
pip install igsr-archive
```

And you are ready to go!

CHAPTER 3

Usage

3.1 Settings file

All scripts mentioned in this document require a `settings.ini` file containing basic configuration parameters. Below is the template of one of these configuration files:

```
[mysql_conn]
host = mysql-g1kdcc-public.ebi.ac.uk
user = g1krw
port = 4197
[fire]
root_endpoint = https://hh.fire-test.sdo.ebi.ac.uk/fire
user = g1k-test-ernesto
version = v1.1
[ftp]
staging_mount=/nfs/1000g-work/G1K/archive_staging
ftp_mount=/nfs/1000g-archive/vol1
[file_type_rules]
fastq = TEST_FASTQ
txt = TEST_TXT
```

Where the `[mysql_conn]` section contains the parameters for connecting the MYSQL server hosting a database created with the RESEQTRACK schema and the `[fire]` section contains the FIRE API connection details. If you do not already have a FIRE username and password, you will first need to contact `fire@ebi.ac.uk` as these are required to connect the FIRE API. The `[ftp]` section contains the details about the staging area directory (see below why this area is important) and also the directory where the FTP server is mounted. The `[file_type_rules]` section is optional, it is used to assing a certain type to each file depending on its extension. This type is an arbitrary string used to describe each of the files being loaded in the `RESEQTRACK` MYSQL database. In the `settings.ini` file shown above, an example file named `test.fastq` will have the `TEST_FASTQ` type, while a file named `test.txt` will have the `TEST_TXT` type.

Note: FIRE provides a testing and a production environment. Each will differ in the `user`, `root_endpoint` and `password` used for connecting the API. Modify `settings.ini` depending on the environment you want to use.

3.2 Load files

This section describes how to load a certain file/s in the RESEQTRACK database. For this, we need to use the script named load_files.py as follows:

1) Load a single file

Use the -f/--file option like this:

```
load_files.py --settings settings.ini --file /path/to/file.txt --type TEST_F --dbname
↪$DBNAME --pwd $PWD
```

- --type is an arbitrary string describing the file that will be loaded in the database. i.e. FASTQ or CRAM. If this option is not specified then the file type will be set depending on the parameters in the [file_type_rules] section of settings.ini.
- --dbname is the name of the RESEQTRACK MYSQL database
- --pwd is the password for connecting the MYSQL server

By default, the script will perform a dry run and the file will not be loaded into the database. You need to run load_files.py with the option --dry False to load it.

Note: The md5 checksum for the file will be automatically calculated. Also, the script will create a new entry in the file table of the RESEQTRACK database with this md5 checksum.

2) Load a list of files

You can provide the script with a list of files (one file per line) to load. For this, use the -l/--list_file option:

```
load_files.py --settings settings.ini --list_file file_list.txt --type TEST_F --
↪dbname $DBNAME --pwd $PWD
```

- --type is an arbitrary string describing each of the files that will be loaded in the database. i.e. FASTQ or CRAM. If this option is not specified then the file type will be set depending on the parameters specified in the [file_type_rules] section of settings.ini.
- --dbname is the name of the MYSQL RESEQTRACK database
- --pwd is the password for connecting the MYSQL server

By default, the script will perform a dry run and the files will not be loaded into the database. You need to run load_files.py with the option --dry False to load them.

Note: The md5 checksum will be calculated for each file and these md5 checksums will be loaded in the file table of the database

3) Load a list of files with pre-calculated md5 checksums

Use the --md5_file option with a file with the following format:

```
<md5>  <path_to_file>
```

Each of the lines in the file will contain the pre-calculated md5 checksum and the path to the file to be loaded. An example command line using this option is:

```
load_files.py --settings settings.ini --md5_file file_list.txt --type TEST_F --dbname
↪$DBNAME --pwd $PWD
```

- --type is an arbitrary string describing each of the files that will be loaded in the database. i.e. FASTQ or CRAM. If this option is not specified then the file type will be set depending on the parameters in the [file_type_rules] section of settings.ini.

- `--dbname` is the name of the MYSQL `RESEQTRACK` database
- `--pwd` is the password for connecting the MYSQL server

By default, the script will perform a dry run and the files will not be loaded into the database. You need to run `load_files.py` with the option `--dry False` to load them.

3.2.1 Errors

- When you are trying to load a file in the database you can get the following error:

```
AssertionError: A file with the name '$FILE' already exists in the DB. You need
 to change name '$FILE' so it is unique.
```

This error indicates that there is already a file entry in the database with the same basename or path. You can deactivate this check by passing the option `--unique False`

3.3 Delete files

The script to remove an entry from the `file` table of the `RESEQTRACK` database is `delete_files.py`.

1) Delete a single file

You can use it as follows:

```
delete_files.py --settings settings.ini -f /path/to/file.txt --dbname $DBNAME --pwd
↪$PWD
```

- `-f /path/to/file.txt` is the path to the file to be deleted
- `--dbname` is the name of the MYSQL `RESEQTRACK` database
- `--pwd` is the password for connecting the MYSQL server

By default, the script will perform a dry run and the file will not be removed from the database. You need to run `delete_files.py` with the option `--dry False` to remove it.

2) Remove a list of files

You can provide the script with a list of files (one file per line) to remove. For this, use the `-l`/`--list_file` option:

```
delete_files.py --settings settings.ini --list_file file_list.txt --dbname $DBNAME --
↪pwd $PWD
```

- `--list_file file_list.txt` file containing the file paths to remove
- `--dbname` is the name of the MYSQL `RESEQTRACK` database
- `--pwd` is the password for connecting the MYSQL server

By default, the script will perform a dry run and the files will not be removed from the database. You need to run `delete_files.py` with the option `--dry False` to remove them.

3.4 Archive files

The script to interact with the FIle REplication (FIRE) archive is named `archive_files.py`. This script can be used to archive files in the public area of the IGSR FTP site. Once a certain file is archived using this script, it will be accessible from our IGSR public FTP site (http://ftp.1000genomes.ebi.ac.uk/vol1/).

3.4.1 Prerequisites

- The file/s to be archived in the FTP area need to be tracked in the `file` table of the `RESEQTRACK` database. For this, you need to load them first using the `load_files.py` script explained in the previous section
- The file/s to be archived in the FTP need to be placed in the staging area of our filesystem (`/nfs/1000g-work/G1K/archive_staging`). To modify this area, change the `staging_mount` parameter from the `[ftp]` section in the `settings.ini` file.

Note 1: The path of the file that is placed in the staging area will be duplicated in the FTP area. So for example, if we want to archive `test.txt` so it can accessed from `http://ftp.1000genomes.ebi.ac.uk/vol1/test_dir/subtest_dir/test.txt`, we need to put `test.txt` in `/nfs/1000g-work/G1K/archive_staging/test_dir/subtest_dir/`.

Note 2: If you want to modify a file that is already archived in the FTP, use the option `--update_existing True`. The file/s that will replace the archived file/s need to be placed in the staging area but it is not necessary to pre-load them in the `RESEQTRACK` database.

Important: Once the file has been correctly archived in the FTP, it will be removed from the staging area.

1) Archive a single file

Use the `-f`/`--file` option like this:

```
archive_files.py --settings settings.ini -f /nfs/1000g-work/G1K/archive_staging/file.
↪txt --dbname $DBNAME
--firepwd $FIREPWD --dbpwd $DBPWD
```

- `-f`/`--file` is the path to the file that will be archived. It needs to exist in the `file` table of the `RESEQTRACK` database
- `--dbname` is the name of the MYSQL `RESEQTRACK` database
- `--firepwd` is the password for connecting the FIRE API
- `--dbpwd` is the password for connecting the MYSQL server

By default, the script will perform a dry run and the file will not be archived in the FTP. You need to run `archive_files.py` with the option `--dry False` to archive it.

Note: Use the `--type` option if you want to update the `type` column from the `file` table of the `RESEQTRACK` database for the archived file. If you do not specify a type then it will preserve the type that was present previously.

2) Archive a list of files

You can provide the script with a list of files (one file per line) to archive. For this, use the `-l`/`--list_file` option:

```
archive_files.py --settings settings.ini --list_file file_list.txt --dbname $DBNAME --
↪firepwd $FIREPWD --dbpwd $DBPWD
```

- `--list_file file_list.txt` file containing the list of file paths to archive
- `--dbname` is the name of the MYSQL `RESEQTRACK` database
- `--firepwd` is the password for connecting the FIRE API

- --dbpwd is the password for connecting the MYSQL server

By default, the script will perform a dry run and the files will not be archived in the FTP. You need to run archive_files.py with the option --dry False to archive them.

Note: Use the --type option if you want to update the type column from the file table of the RESEQTRACK database for the archived files. If you do not specify a type then it will preserve the type that was present previously.

Note: Use the --update_existing option. Set it to True, if you want to update a file that is already archived in the FTP with a file still in the staging area

3.4.2 Errors

- When you are trying to archive a certain file in FIRE you can get the following:

```
AssertionError: File entry with path /path/to/test.txt does not exist in the DB.␣
↪You need to load it first in order
to proceed
```

This means that /path/to/test.txt is not tracked in the RESEQTRACK database, you need to load it first using the load_files.py script

3.5 Dearchive files

The script to de-archive (i.e. remove) a file or a list of files from our public FTP area is called dearchive_files.py. This script will download the file to be de-archived to a desired location before de-archiving from FIRE and will delete the entry from the file table in the RESEQTRACK database.

1) De-archive a single file

Enter the following command:

```
dearchive_files.py --settings settings.ini --file /nfs/1000g-archive/vol1/path/file --
↪md5check False --directory /dir/to/put/file --dbname $DBNAME \
--firepwd $FIREPWD --dbpwd $DBPWD
```

- --file is the path to the file to be de-archived. /nfs/1000g-archive/vol1 is the directory containing the IGSR FTP in our filesystem. This directory can be changed by modifying the ftp_mount parameter from the ftp section in the settings.ini file
- --md5check is the way to check if md5sum of downloaded file and FIRE object matches before dearchiving from FIRE, default is set to True, change to False if no check is needed
- --directory is the directory used to store the file to be de-archived
- --dbname is the name of the MYSQL RESEQTRACK database
- --firepwd is the password for connecting the FIRE API
- --dbpwd is the password for connecting the MYSQL server

By default, the script will perform a dry run and the file will not be de-archived from the FTP. You need to run dearchive_files.py with the option --dry False to de-archive it.

2) De-archive a list of files

You can provide the script with a list of files (one file per line) to de-archive. For this, use the -l/--list_file option:

```
dearchive_files.py --settings settings.ini --list_file file_list.txt --md5check False␣
↪--directory /dir/to/put/file --dbname $DBNAME \
--firepwd $FIREPWD --dbpwd $DBPWD
```

- `--list_file` is the list of files to de-archive
- `--md5check` is the way to check if md5sum of downloaded file and FIRE object matches before dearchiving from FIRE, default is set to True, change to False if no check is needed
- `--directory` is the directory used to store the files to de-archive
- `--dbname` is the name of the MYSQL `RESEQTRACK` database
- `--firepwd` is the password for connecting the FIRE API
- `--dbpwd` is the password for connecting the MYSQL server

By default, the script will perform a dry run and the files will not be de-archived from the FTP area. You need to run `dearchive_files.py` with the option `--dry False` to de-archive them.

3.6 Move files

The script to move a file/s from one directory in the public IGSR FTP area to a different directory is `move_files.py`. This script will also update the entry in the file table from the `RESEQTRACK` database with the updated path.

1) Move a single file

Use the `--origin` and `--dest` options like this:

```
move_files.py --settings settings.ini --origin /nfs/1000g-archive/vol1/dir1/test.txt -
↪-dest /nfs/1000g-archive/vol1/dir2/test.txt \
 --dbname $DBNAME --firepwd $FIREPWD --dbpwd $DBPWD
```

- `--origin` is the current path for the file to move
- `--dest` is the final path for the moved file
- `--dbname` is the name of the MYSQL `RESEQTRACK` database
- `--firepwd` is the password for connecting the FIRE API
- `--dbpwd` is the password for connecting the MYSQL server

By default, the script will perform a dry run and the file will not be moved. You need to run `move_files.py` with the option `--dry False` to move it.

2) Move a list of files

You can provide the script with a list of files to move. This can be done by creating a 2-columns file with the following format:

```
<origin>\t<dest>
```

Where, for each line, the first column is the current path in the FTP filesystem of the file to move, and the second column is the path to which the file will move.

The script is run by doing:

```
move_files.py --settings settings.ini --list_file file_list.txt --dbname $DBNAME --
↪firepwd $FIREPWD --dbpwd $DBPWD
```

By default, the script will perform a dry run and the files will not be moved. You need to run `move_files.py` with the option `--dry False` to move them.

3) Move the contents of an entire directory in the public IGSR FTP area

Use the `--src_dir` and `--tg_dir` options like this:

3.1) Move only the files in `--src_dir` without moving the files in subdirectories

To move the files that are located in `--src_dir` without moving the files within any of the `--src_dir` subdirectories you need to run the script doing:

```
move_files.py --settings settings.ini --src_dir "/nfs/1000g-archive/vol1/dir1/*" --tg_
↪dir /nfs/1000g-archive/vol1/dir2/ \
 --dbname $DBNAME --firepwd $FIREPWD --dbpwd $DBPWD
```

- `--src_dir` is the directory in the FTP area containing the files to be moved. You can use the wildcard to define the pattern for searching the files, i.e. `--src_dir "*.txt"`. Note the double quotes.
- `--tg_dir` is the directory in the FTP area where the files specified by `--src_dir` will be moved.

3.2) Move the files in `--src_dir` and any of the files in its subdirectories

To move the files that are located in `--src_dir` and any of the files within any of the `--src_dir` subdirectories, you need to run the script doing:

```
move_files.py --settings settings.ini --src_dir "/nfs/1000g-archive/vol1/dir1/**/*" --
↪tg_dir /nfs/1000g-archive/vol1/dir2/ \
--dbname $DBNAME --firepwd $FIREPWD --dbpwd $DBPWD
```

Note the double asterisk, which indicates any subdirectory included in the parent directory `/nfs/1000g-archive/vol1/dir1/`

By default, the script will perform a dry run and the files will not be moved. You need to run `move_files.py` with the option `--dry False` to move them.

CHAPTER 4

Indices and tables

- genindex
- search