Locus Reference Genomic XML Parser

Overview

The program is a stand-alone script that parses Locus Reference Genomic (LRG) XML files to extract information relating to the gene. It can either take the latest XML files from the LRG website or from a local folder.

The source files can be found at https://github.com/Graeme-Smith/UoM\_LRG\_Parser

The following information is extracted from an input LRG file and outputted into a .txt file:

HGNC gene name

LRG version

HGNC ID

Sequence source

LRG Assemblies

e.g. GRCh37.p13

end: [genomic co-ordinate]

start: [genomic co-ordinate]

Differences with reference Seq: GRCh38.p7

---------------

[Type of change]: lrg\_sequence > other\_sequence

start: [LRG\_co-ordinate]

end: [LRG\_co-ordinate]

Exon details:

---------------

[Exon number]: LRG start - LRG end

The output can either be returned to the console or saved as a .txt file.

Repository

Location: https://github.com/Graeme-Smith/UoM\_LRG\_Parser.git

Contents;

|  |  |
| --- | --- |
| Filename | Description |
| /output | Folder for storing .txt and .BED files that have been output by the program |
| .gitignore | Contains a list of files and folders to be ignored by git |
| Documentation.docx | This file |
| LRG\_1.xml | LRG XML file used when running the test script |
| main.py | Main script for performing program functionality |
| readme.txt | Description of program functionality and usage instructions |
| test.py | Test script for performing unit testing on main script |

Usage

The program is run from the command line. To run the program, first navigate to the directory to which the repository has been pulled and type:

python3

The arguments highlighted in red affect how the program responds. The table below describes how these arguments can be used:

|  |  |  |
| --- | --- | --- |
| Argument | Description | Example of command |
| -h, ‘--help’ | Returns the help message to inform the user of how to use the program | python3 lrgParser.py -h |
| '-l', '--local' | Takes a copy of the lrg file from the local directory instead of from the LRG FTP site. Assumes file is using the same naming convention as the LRG website, i.e. LRG\_{user input}.xml. If the file cannot be found it is downloaded from the internet instead. | python3 lrgParser.py -l LRG\_1.xml |
| '-g', '-hgnc' | Import LRG files for conversion into a BED file as per provided HGNC Ids. | python3 lrgParser.py -g EGFR |
| '-x’, '-xref' | Import LRG files for conversion into a BED file as per provided external reference, such as the gene ID number | python3 lrgParser.py -x 1956 |

Testing

A test script has been developed, test.py, which performs unit testing of the main script. The test script calls individual functions from within the main script and checks that the given output for each function is as expected using LRG\_1.xml as the test case. Running the test script allows the user to check that the main script functions are working as intended. Developers should run the test script after each modification of the main script to ensure the changes made have not affected the functionality.

To run the test suite, using the command line, navigate to the directory to which the repository has been pulled and type:

python3 tests/test.py

Appropriate messages will be output to the console to state whether all the tests have passed, or an error message will be displayed stating which test has failed.

Limitations

The program has been tested with Python version 3.6.7 and LRG version 1.9 (current version) only.