

Running AltAnalyze in Unix-Linux

Users have 3 options for running AltAnalyze on Unix or Linux:

- 1) Access the graphical user interface from binaries
- 2) Access the graphical user interface from Source-Code
- 3) Run AltAnalyze via command-line from Source-Code or binaries

Option 1 - Access the graphical user interface from binaries

Distributed with AltAnalyze are binaries created by the program `cx_freeze`. These have not been extensively tested on all Linux implementations except for Ubuntu, however, if compatible, they should contain all necessary dependencies to run AltAnalyze (python and Tkinter). Go to the AltAnalyze program directory and double-click on the file named "AltAnalyze". If it does not execute, there is either system incompatibility or the file does not have permissions to open. To change the permissions, find the file in a terminal window and type:

```
$ chmod 777 AltAnalyze
```

If this does not work, proceed to option 2.

Option 2 - Access the graphical user interface from Source-Code

Most users will likely want to choose this option, since most Linux configurations have Python and Tkinter. If not, the user can install these. To run, first go to the AltAnalyze program directory and find the directory "Source_code", copy or move the contents to the main AltAnalyze directory, open a terminal window, move to the AltAnalyze program directory (e.g., `cd AltAnalyze_v1release`) and type:

```
$ python AltAnalyze.py
```

This will run AltAnalyze. If you receive an error that Tkinter or PMW are not installed, you can install these from <http://www.python.org> or proceed to step 3.

Option 3 - Run AltAnalyze via command-line

Since configuring some Unix machines to have the necessary python dependencies can be problematic, users are recommended to use the command-line variables to run AltAnalyze. These options are preferred by many users, since they can be used to quickly start and run a job without interruptions, at the machine or remotely. To run AltAnalyze via the command-line, open a terminal (at the machine or remotely), move to the AltAnalyze program directory and enter the appropriate flags. First, you will need to download the appropriate species database. To download databases for human enter:

```
$ python AltAnalyze.py --species Hs --update Official --version  
EnsMart65 --platform RNASeq
```

To analyze CEL files using default parameters enter the following into the command-line:

```
$ python AltAnalyze.py --species Hs --arraytype exon --celdir  
"/home/hESC_exon/" --output "/home/hESC_exon" --expname "hESC-  
differentiation"
```

More sophisticated commands can also be used.

```
$ python AltAnalyze.py --species Hs --arraytype exon --celdir  
"/home/hESC_exon/" --output "/home/hESC_exon" --expname "hESC-  
differentiation" --runGOElite no --dabgp 0.01--rawexp 100 --  
avgallss yes --noxhyb yes --analyzeAllGroups "all groups" --  
GEcutoff 4 --probetype core --altp 0.01 --altmethod FIRMA --  
altscore 8 --exportnormexp yes --runMiDAS no --ASfilter yes --  
mirmethod "two or more" --calcNip yes
```

If running using the binary, which will not require any dependencies be installed, call:

```
./AltAnalyze instead of python AltAnalyze.py
```

For details, see: <http://www.altanalyze.org/help.htm#commandline> and

<http://code.google.com/p/altanalyze/wiki/CommandLineMode>

<http://code.google.com/p/altanalyze/wiki/StandAloneDependencies>