FFAT Explanatory document.

This document is a guide to the suite of Python scripts developed for analysis of proteome text files as part of research work directed by Prof Tim Levine at UCL.

These scripts were written by John Slee (contact john.slee@btinternet.com) .

I am an amateur working by myself. I make no guarantees to the quality of the code. If you think there are better and faster ways of doing things, you may well be correct.

I do very little error checking in the code. This is because I am in control of what goes in, and in the event of a failure I can go and track it down. If this were to be used in a multi-user environment with files being brought in from all over, then error handling and data quality checking would probably need improvement.

Neither do I guarantee the results in all circumstances. If you use these script you should check the results are what you would expect.

Dev environment and notes

I used Jupyter Notebook on an Apple iMac for development. It was originally written in a version of Python2 but upgraded to Python 3.

In order to help my debugging and record keeping I developed a logging system which (optionally) enabled a log file to be written of progress, file names, versions, etc. You may find it easier to strip those parts out and replace with your own version of logging. (small note. I found instances of amendments to files not producing the results I expected, and generally this was because the old version was still in memory. I got in the habit of incrementing the version number everytime I did a change so I could check whether the new version had been picked up and used).

The scripts go to specific paths on my iMac to retrieve and to write files. I believe all these directory names are in variables at the top of files and hence should be straightforward for you to replace with names of your own directories.

The structure of the scripts is as follows:

Release python files:

Proteome FFAT GH22

Master file. Imports the other files, asks the user a sequence of questions to produce the desired analysis on the target file.

Proteome Analysis GH22

High level analysis. This contains a list of available analyses and the routines to run them.

Proteome Core GH22

Contains useful functions.

Contains routines to convert proteome text files into a class which the analyser routine can analyse. I received proteome files forma variety of sources and hence in a variety of formats. Each format needed its own converter. These are numbered, and in the text file I used to keep track of the proteome source files I have a number that indicates which routine is to be used to decode it. You should pay particular attention to this aspect of the work if you are using these scripts or developing your own as it is quite easy to fail to fully and accurately convert certain formats into a usable proteome. I would suggest you assume any code you have written does not properly convert the raw files until you have satisfactorily demonstrated that it does.

The routines allow you to have test files, which I generated by either writing specific pseudo-proteome segments or taking small segments of full proteomes. I would strongly recommend you have a separate directory of test files that you can optionally select in order to give you confidence about the analysis on the full proteomes. This saves run time and analysis time.

Proteome Scorer GH22.

Contains the low level scoring functions.