**Running Optimiser user guide**

**Purpose**

This guide details how to use optimiser to suggest optimum peak finding parameters for SIEVE preprocessed data sets.

**Prerequisites**

1. Python 2.7 minimum with specific libraries installed as per Python installation guide.
2. PeakFilter Input file(s) - LC/MS data that has been pre-processed with SIEVE and prepared as per PeakFilter\_Input\_file\_preparation\_guide and stored as a comma separated file(s) (csv)
3. A published list of around 100 lipids expected to be found in the samples in the ionisation mode.

**Manual target file preparation**

Optimiser needs a target file to compare and score PeakFilter iterations against. The target file should consist of at least 50 diverse, manually curated peaks described by their retention time (RT), *m/z*, and intensity. The more peaks in this target file the more accurate the parameter set suggested by Optimiser. The process for generating the target is as follows:

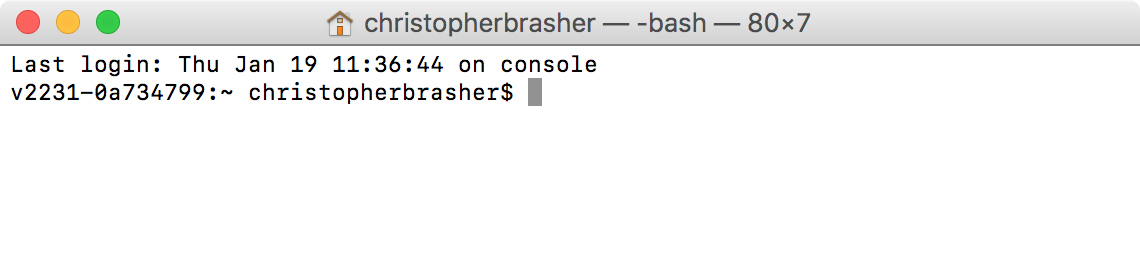
1. Choose a replicate to use as the target replicate. This should be representative of the entire data set. This can be determined by using Xcalibur Qual Browser to view total ion counts (TICs). Suitable replicate candidates will have a peak retention time, TIC profile and an intensity similar to the mean.
2. Using a published list of lipids manually locate each one in Qual Browser. Where peak quality is good refer to the PeakFilter input file and find the group of frames that make up the peak using a small RT and *m/z* tolerance (to be judged on an individual basis)
3. Sum the intensity of the frames making up the peak and assign the RT and m/z as that of the most intense constituent frame. This RT, *m/z* and intensity are 1 target.
4. Repeat this process on varying intensity peaks across a broad RT and m/z.

**Parameter range setting**

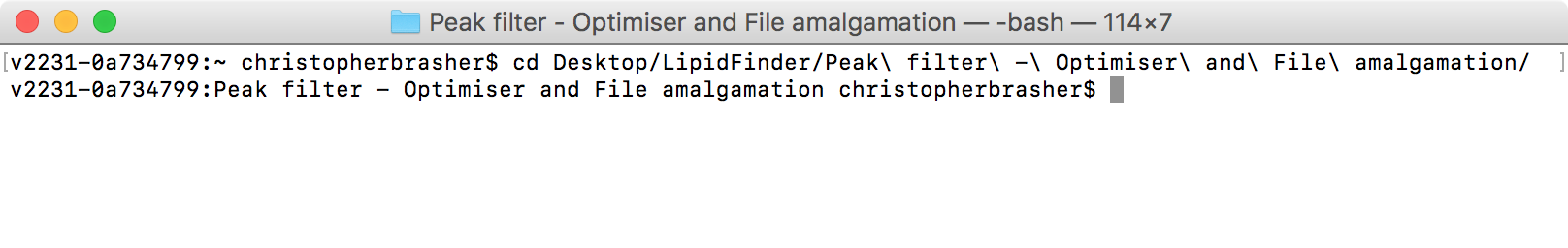
For each of the 4 tunable parameters (‘*peakAdjacentFrameMaxRT*’, ‘*peakMinFoldCutOff*’, ‘*peakMaxRTWidth*’ and ‘*mzSizeErrorPPM’*) a lowest value, highest value and increment are stored in the file *hc\_parameters\_dimensions.csv*, this file may be edited by the user. However, it must be possible for the value of ‘*peakMaxRTWidth*’ to be greater than 3 times ‘*peakAdjacentFrameMaxRT*’ or Optimiser will suggest the first, random parameter set.

**Running Optimiser using Mac OSX/Linux**

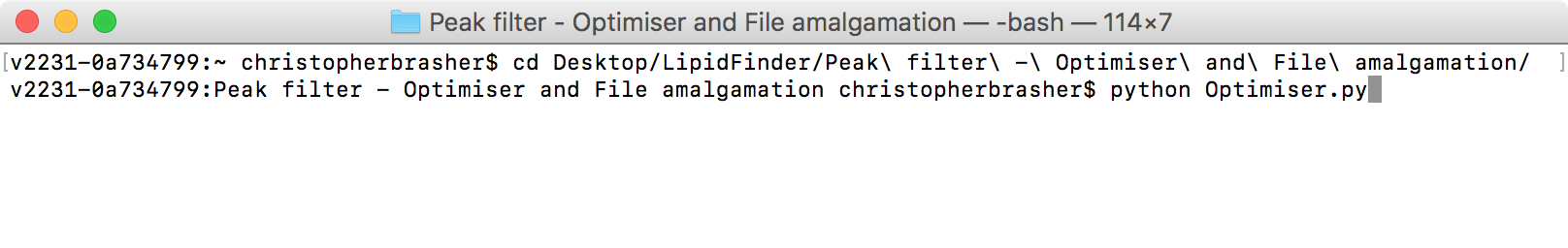
1. Open Terminal



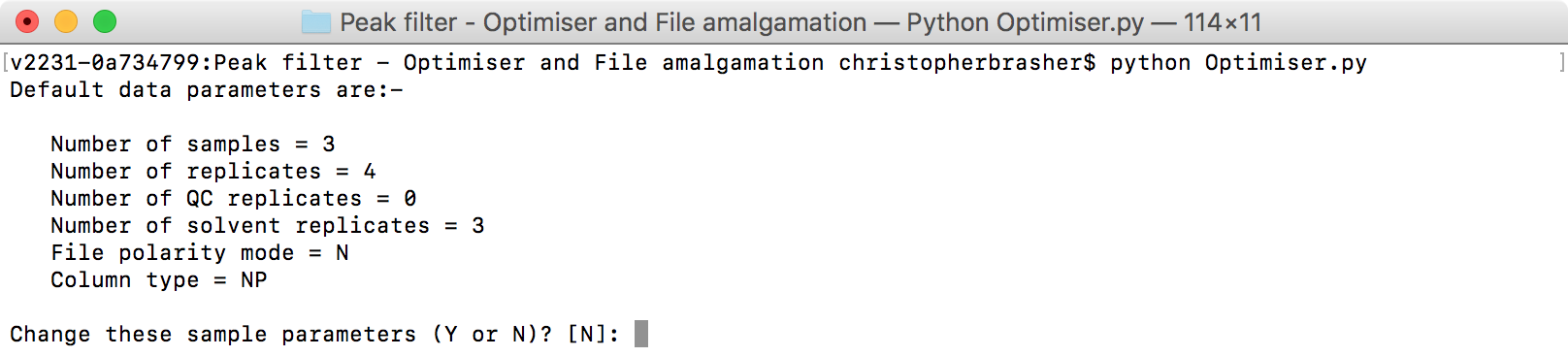
1. Navigate to the folder where the Amalgamator code is.



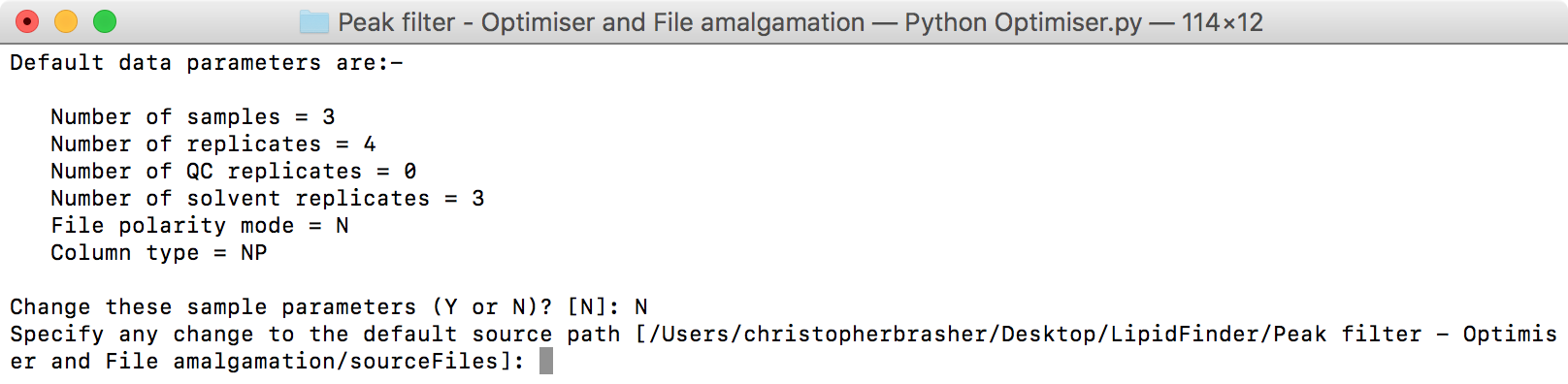
1. Type in the console – “python Optimiser.py” (this is case sensitive) – This will start the Optimiser program and import all dependant libraries.



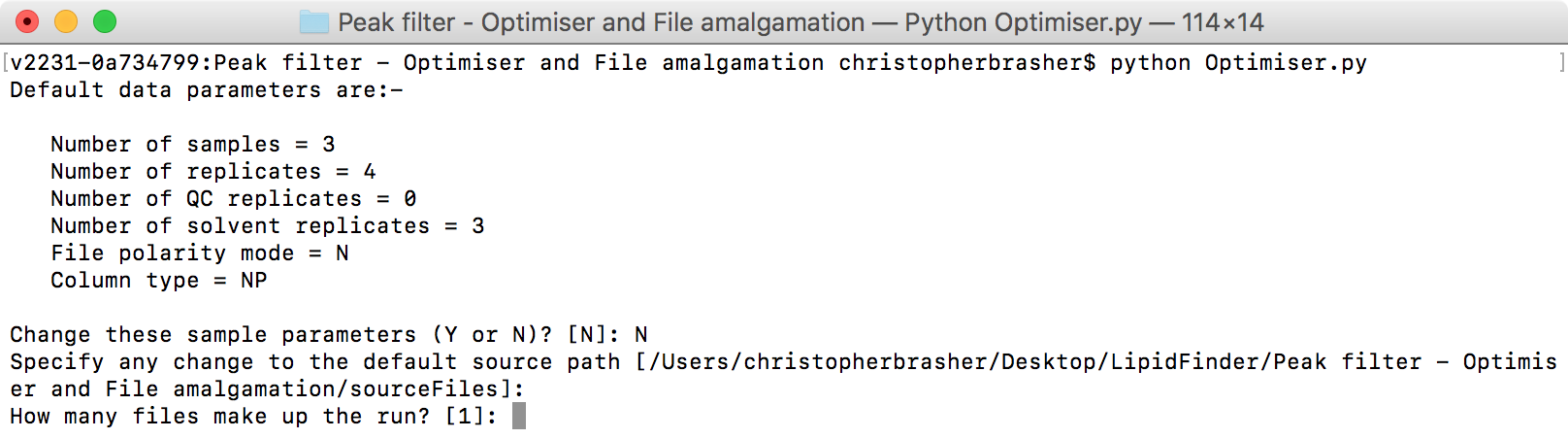
1. Optimiser will prompt for confirmation of the input file details.
   1. If these have been modified in the parameters.csv file in the PeakFilter folder before step 3 then these should be correct. However, these can be modified at this point by entering “Y” at the prompt. Each file parameter will now be prompted for individually.
   2. If the parameters are are correct accept the default ‘No’ (‘[N]’) and hit return.



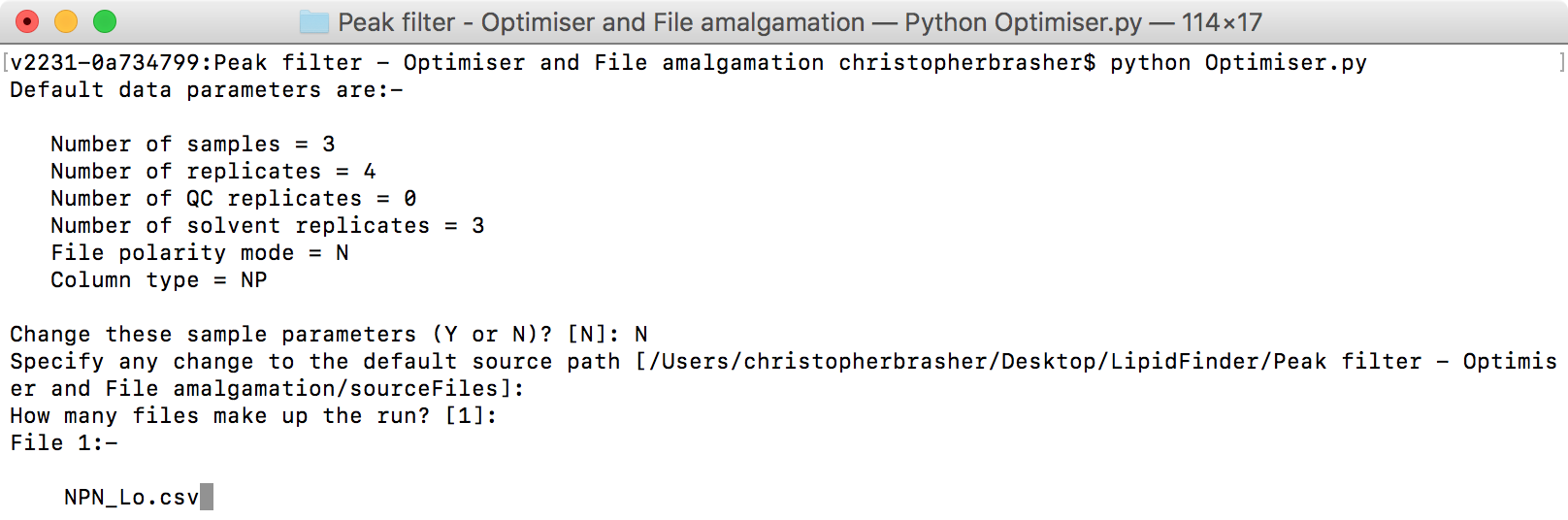
1. Optimiser input files can read from anywhere on the computer file system, however the default location is ‘PeakFilter/sourceFiles’. If there are multiple input files they must reside in the same folder. Additionally, the target file must reside in the same folder as the source file(s). To change the default path insert the full path to the input file folder here.



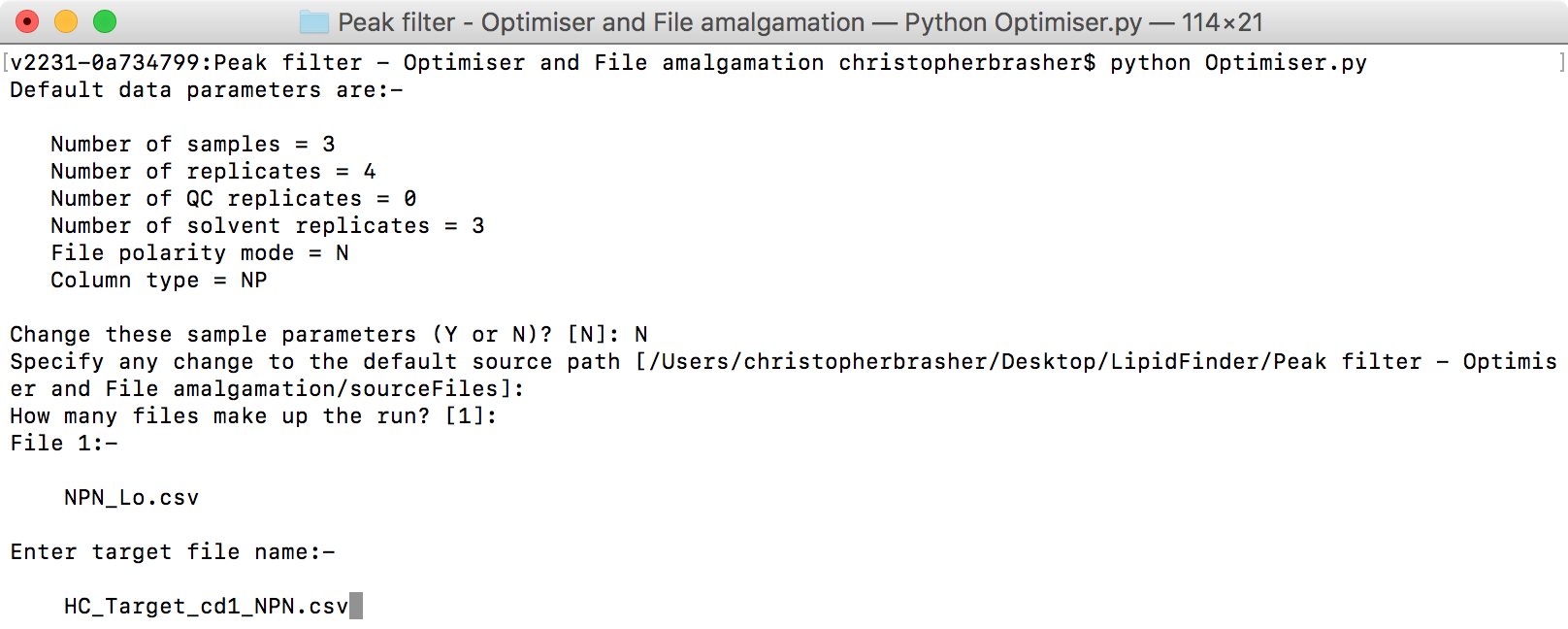
1. Occasionally, there may be more than one input file if the SIEVE runs have been split by retention time ranges to ease the processing requirements on the computer. At this point the default 1 file input file can be changed to any number of files provided the rules described in the ‘PeakFilter input file preparation guide’ are followed. These can be found in the ‘File formatting considerations’ section.



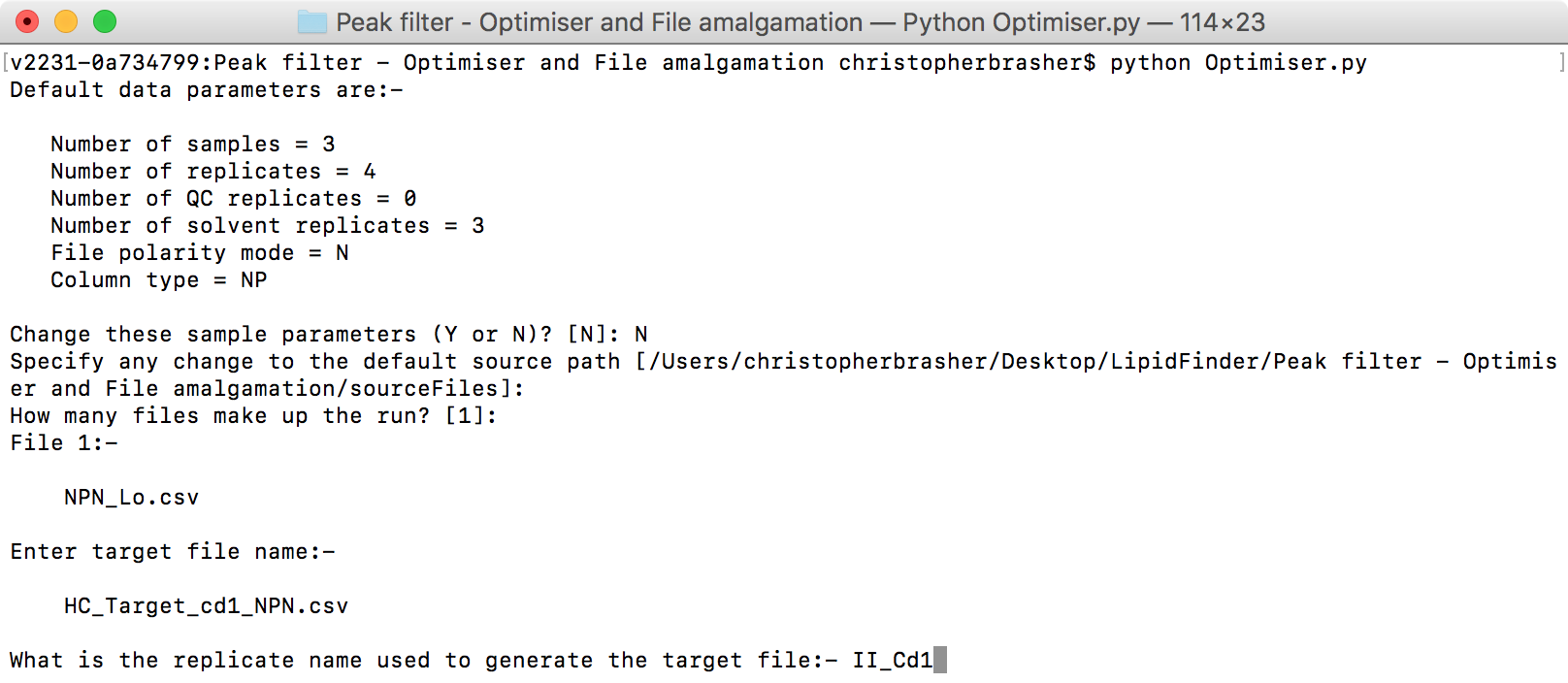
1. Optimiser will now prompt for each of the input files in turn, input files should be loaded in the order smallest retention time to largest retention time.



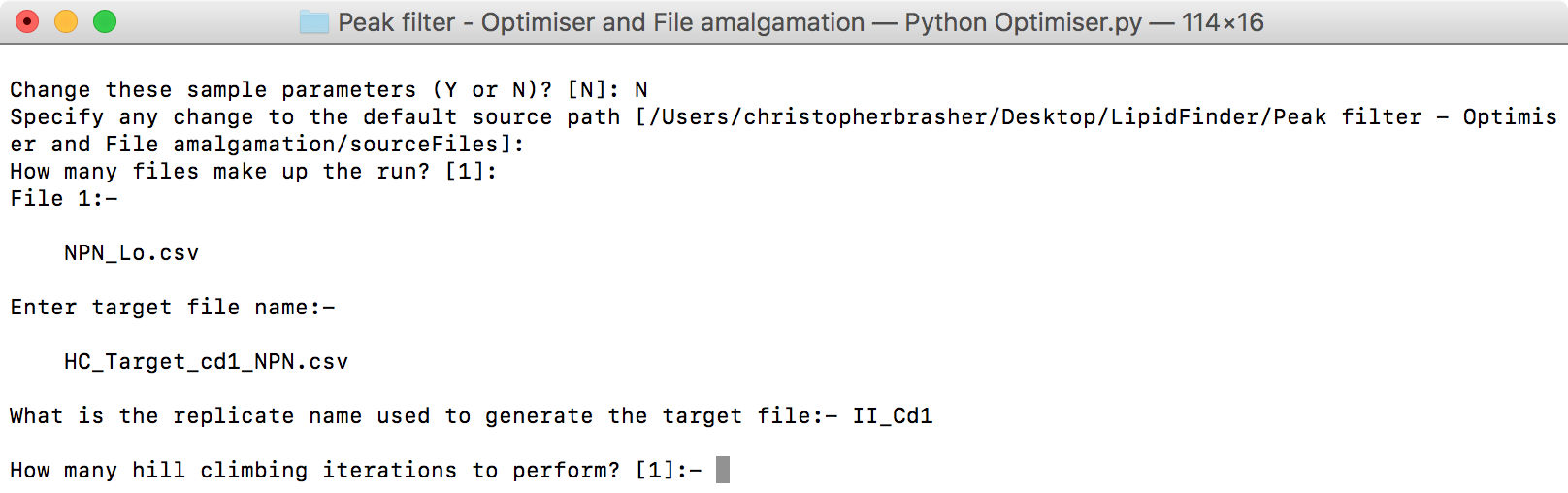
1. After the last input file is successfully loaded, Optimiser will prompt for the name of the target file.



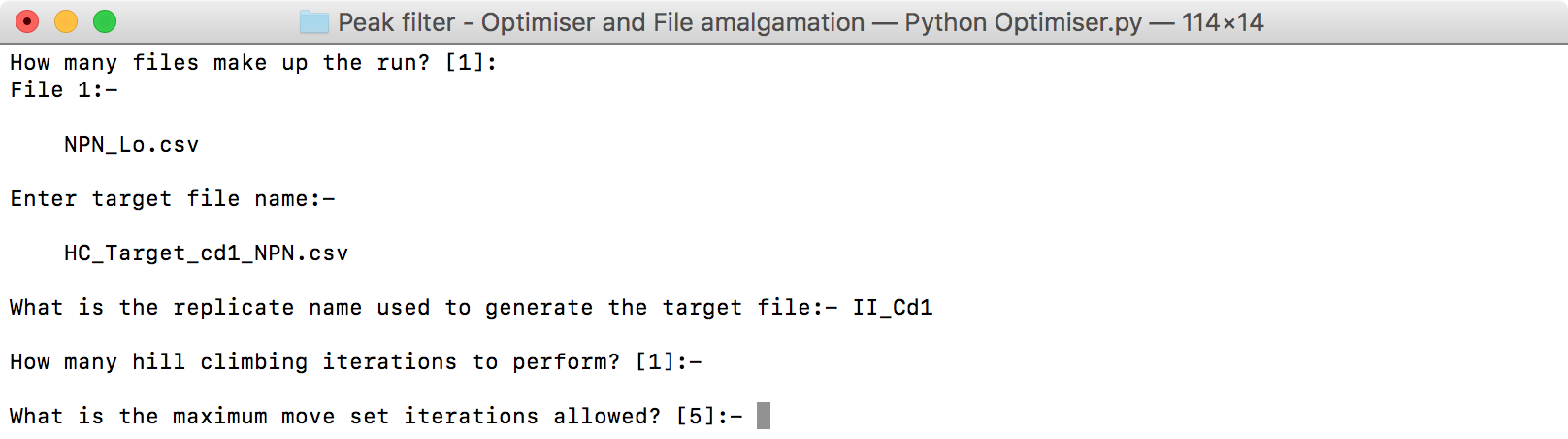
1. Optimiser will now prompt for the replicate name in the data set loaded in step 7 that the target file was generated from.



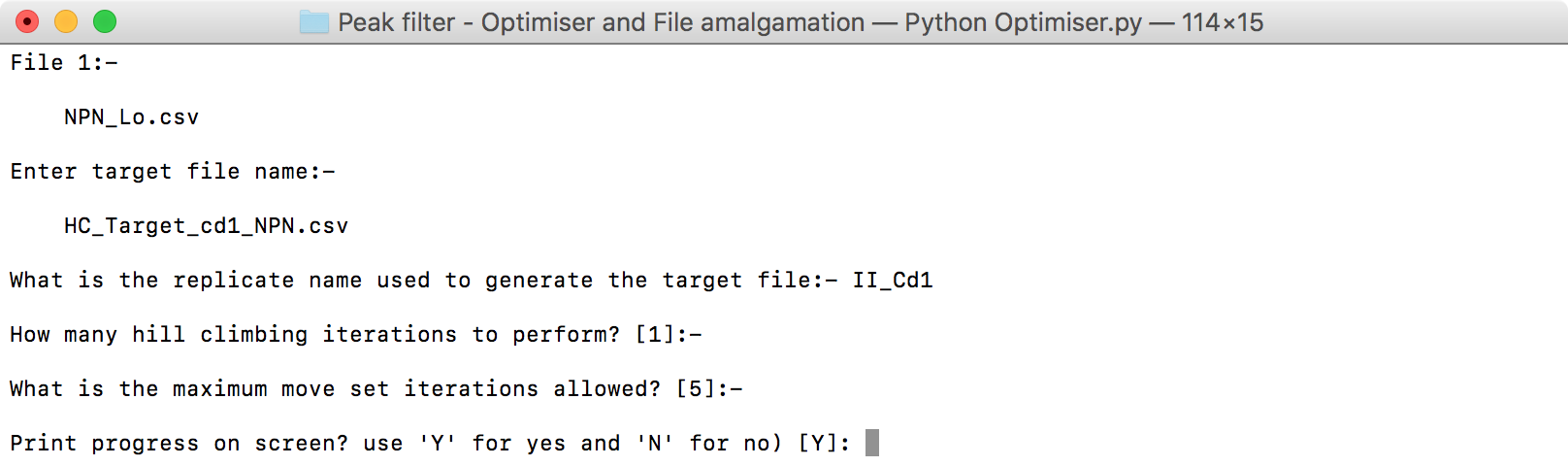
1. Each Optimiser run starts with a random parameter set, running optimiser over multiple iterations with multiple different starting parameters ensures a better chance of finding the global best parameter set for the dataset.



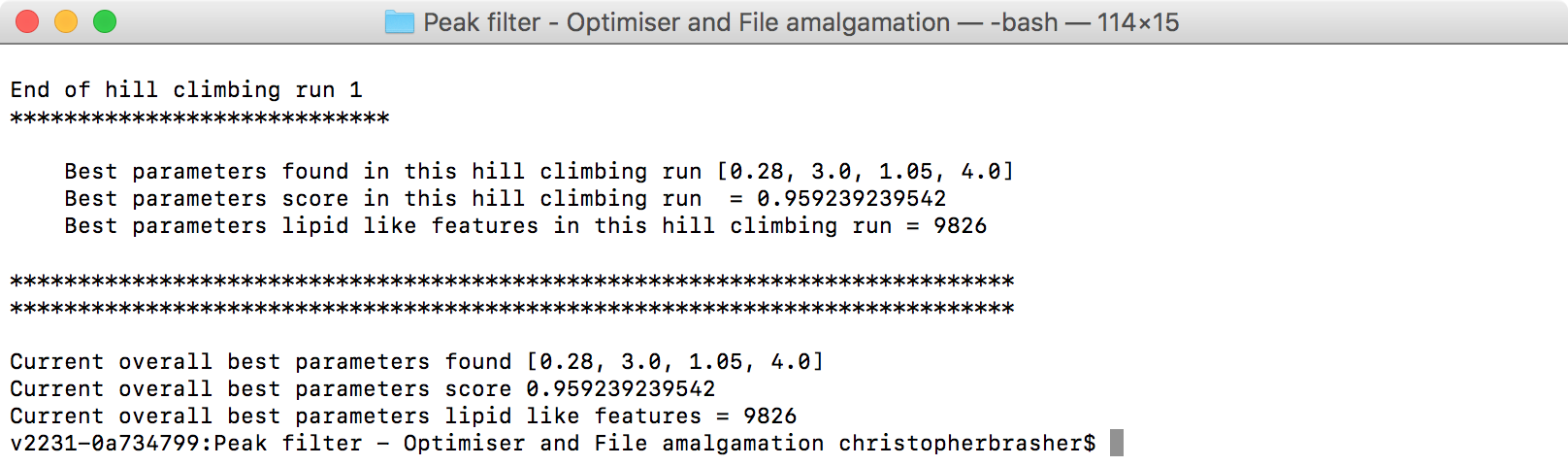
1. In order to speed up progression each the user is able to limit the number of times the parameter sets may be visited in a single Optimiser iteration.



1. Longer Optimiser runs may print out a lot of data to the screen, the user has the option of disabling the printing of progress to the screen.



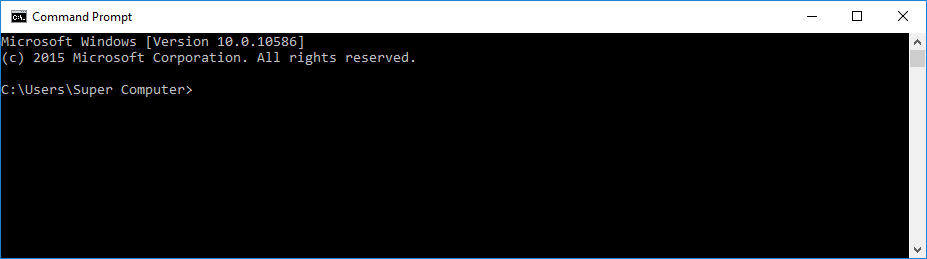
1. Optimiser will now run to completion, this may take a very long time (days) depending on the settings used. It is advisable to perform a trial run where just 1 hill climbing iteration is performed initially so the user can gauge how long multiple iterations will take.



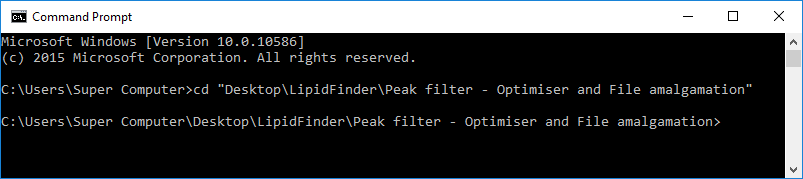
1. Optimiser has now finished; the best scoring parameters are displayed in the following order: *‘peakAdjacentFrameMaxRT*’, ‘*peakMinFoldCutOff*’, ‘*peakMaxRTWidth*’ and ‘*mzSizeErrorPPM’*. These parameters can now be set in the parameters.csv file and the data set run through the PeakFilter module. In addition, an output file named ‘hc\_exported\_file.csv’ detailing the progression of the Optimiser run is output to the current python working directory.

**Running Optimiser using Windows**

1. Open Command prompt.



1. Navigate to the folder where the Optimiser code is.



1. Type in the console – “python Optimser.py” (this is case sensitive) – This will start the Optimiser program and import all dependant libraries. Follow the instructions from section 4 under ‘Running Optimiser using Mac OSX/Linux above.