**QuickStart guide to using genescan pipeline v2**

Cobbled together by Branduff McAllister

Contact: [mcallisterw@cardiff.ac.uk](mailto:mcallisterw@cardiff.ac.uk)

Fragman package1 used from <https://cran.r-project.org/web/packages/Fragman/index.html>

## 1. Set-up the pipeline the first time

First, download R & RStudio.

R: <https://cloud.r-project.org/>

RStudio: <https://rstudio.com/products/rstudio/download/> (the free version)

Place the script ‘mouse\_FSA\_pipeline.R’ someplace sensible (it doesn’t really matter where). Next, make a folder on your computer with the .FSA files you want to analyse. This can be anywhere, although the pipeline will take longer to run the more files you have in the folder. On my computer, I can run ~60 per minute, but this may be slower on other computers. Go ahead and open the script in RStudio.

Run the following code in the console window if you have just installed R for the first time:

Code to install necessary packages, 1

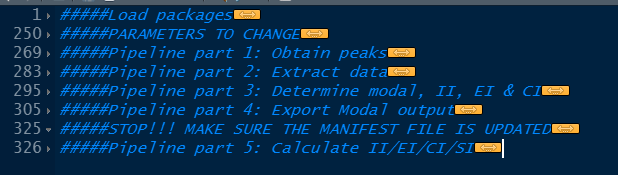
install.packages("tidyverse")

install.packages("Fragman")

This installs the pre-requisite packages to run the pipeline!

## 2. Select the parameters to run in the pipeline

Open the .R script file in RStudio if you haven’t done so already. To make it easier to read, you can hit the little arrow buttons to the left of the #####headings##### which will collapse each section (see red outline below).



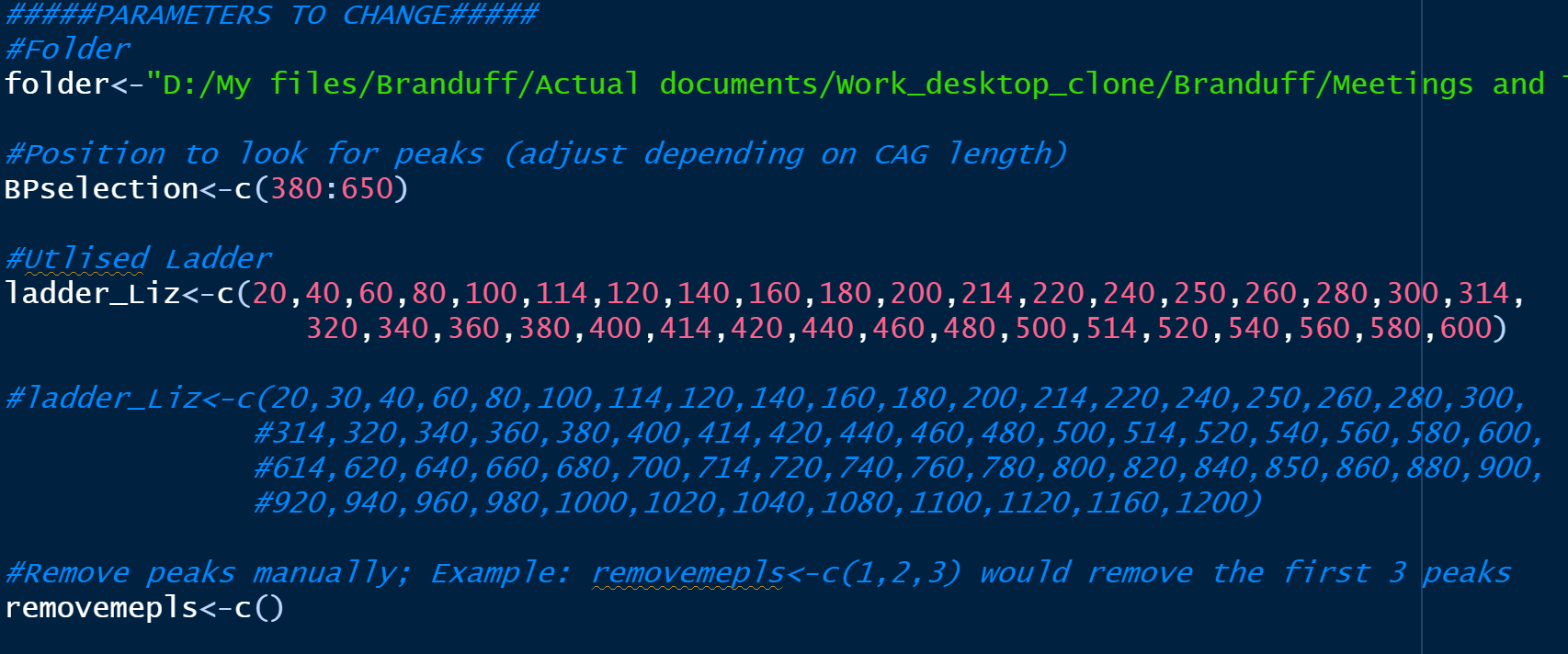
Scroll down to the ‘Parameters to change’ portion of the script. Here is an explanation for each variable:

**folder** = The folder where all your FSA files are. This is also where all your output files will go once the script runs. Make sure to end the path in a trailing slash or it can cause your output files to go in the directory above. Example: ‘D:/My Data/My FSA files/’

**BPselection** = The basepair sizes to look at. I find ~400-600 works pretty well for mouse data, but you can increase/decrease the size of the selection as necessary (check the output traces).

**ladder\_Liz** = The ladder you used. By default it uses the Liz600 ladder, but change this if you are using a different ladder. The blue commented text is the Liz1200 ladder (this is not used unless you uncomment it).

**removemepls** = A way to manually remove peaks if necessary. If not using, just ignore.

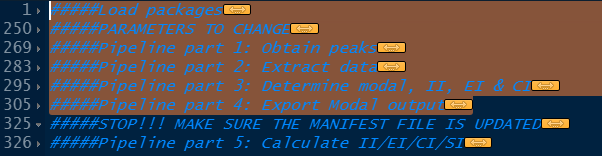


Parameters to change in the script

## 3. Running the pipeline (Part 1-4)

After selecting the parameters in 2., select the ‘load packages’, ‘parameters to change’ and the first four parts of the pipeline to run (see below). You can just select the headings if this is easier, and hit cntrl + enter to run. This will generate five files in the folder you selected earlier with your FSA files:

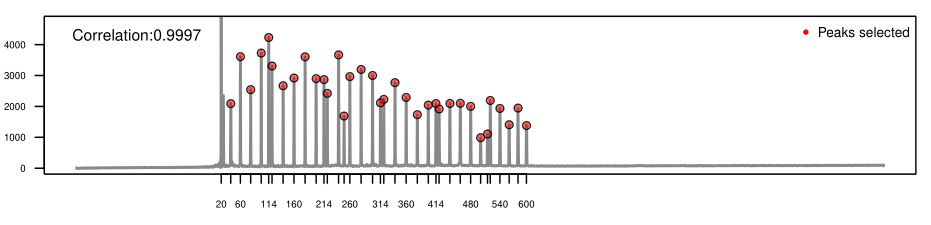
* Ladder.pdf
* Mainplot.pdf
* Summary\_modalonly.csv
* filelist.csv
* All\_peak\_report.csv



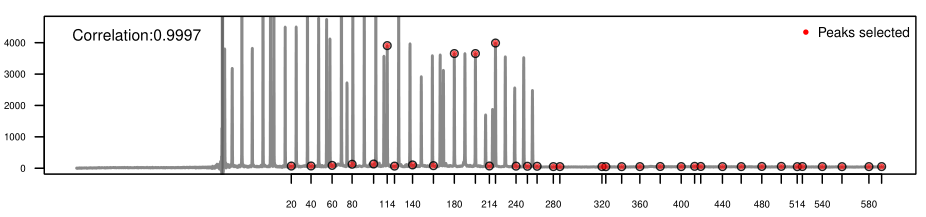
## 4. Checking the outputs of the pipeline

### a) Ladder output

First, open the Ladder.csv file. This is the peaks that the pipeline has selected. You should check each graph (quickly) to check the peaks assigned make sense. See below for a good and a bad example. This is an absolutely critical step, so make sure there’s no weird peaks, especially in the BP size that you’re investigating. See ‘ladder selection’ under troubleshooting for help if this is a problem.



Good example – ladder peaks selected successfully. The first peak notch is 20 and the last 600, and it has 36 individual peaks.

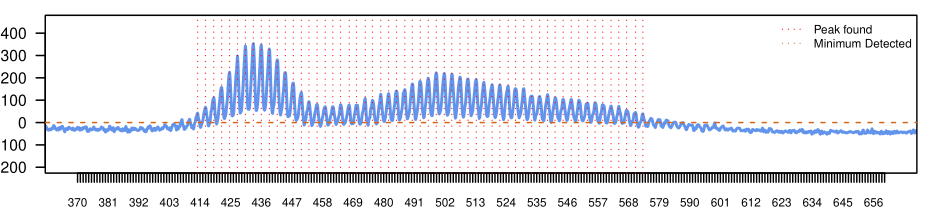


Very bad example – it’s unlikely they’d ever be this messed up, but obviously here true ladder peaks weren’t selected and instead noise was selected.

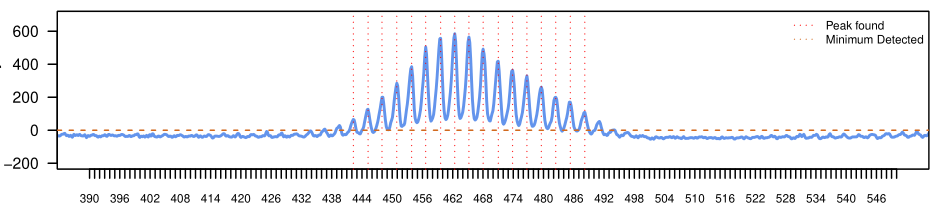
### b) Peak selection

Open mainplots.pdf and have a look at the traces/peak selections. The dashed red lines indicate peaks that have been selected for analysis, and the blue line is the trace from the genescan. By default, the peaks selected should be at least 3bp apart (this can be modified, see below). Note that peaks that occur outside the BPselection will *not* be counted, so you may want to change this if your peak selection is cutting off part of your traces.

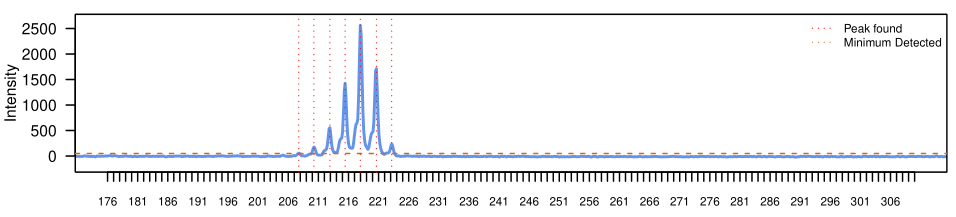
I have tried to optimise the peak selection for *HTT* expansion data, but check ‘peak selection’ under troubleshooting if there’s issues.



Example of a good output from peak selection (mice)



Example of a good output from peak selection (IPSCs)



Example of a good output from peak selection (patient sample)

### c) The filelist.csv and setting up the manifest file

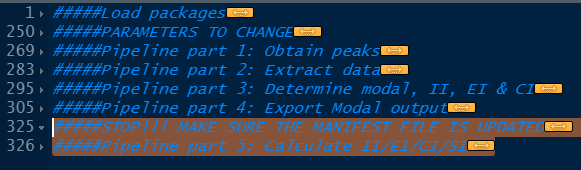
The final important file created is called filelist.csv. This has four main columns: files, startcag, weighted\_startcag and foundcag. The files column is a list of all the files in the folder selected; startcag and weighted\_startcag are by default NA and foundcag lists the modal peaks found in the pipeline. For the final step of the pipeline you MUST do the following:

* Select a value for startcag and weighted\_startcag for each file (or keep those you don’t want to run as NA)
* Rename the ‘filelist.csv’ to ‘manifest.csv’

As long as you don’t change any of the column names, the data from the manifest.csv will be loaded into R in the final part of the pipeline (see next section). For mouse data, copy and paste ‘foundcag’ into the ‘startcag’ cag for both striatal and tail data. For cells, input what the starting CAG should be. For patients, just copy and paste from the foundcag.

## 5. Running the pipeline (Part 5)

After you’ve included a manifest.csv file in the selected folder, as described in 4c), the rest of the pipeline should run without issue. Just select the final part of the pipeline, and use cntrl-enter to run. This will generate a final file called ‘Summary.csv’. This contains a list of summary variables for each sample.



The current operations performed:

**weighted\_CAGchange** = The weighted average repeat for the sample subtracted by starting point control

**CAG\_modal** = Modal CAG (the *first* largest peak, if there are >=2 peaks with the same size that are the largest)

**InstabIndex** = Instability index , where n=all peaks >10% of the modal peak

**ExpIndex** = Expansion index, as above but only counting peaks to the right of the modal

**ConIndex** = Contraction index, as above but only counting peaks to the left of the modal

**NeueExpIndex** = Alternative expansion index, based on the relative position to the modal peak as opposed to the starting CAG

**SomMos** = Somatic mosaicism, , where n = peaks >10% of the modal, and n0 = the modal peak.

## Troubleshooting

### Ladder selection

*Pro-tip*: If the ladders are messed up, try modifying the ‘ladder.info.attach’ function in pipeline part 1:

**ladd.init.thresh** (by default 400; try increasing if it’s selecting noise, or decreasing if the ladder is very weak on a particular run)

**attempt** (by default 10; you can increase which will take longer, but may lead to more accurate selection of ladder peaks).

**method** (by default ‘iter2’; you may have more luck with ‘iter’)

You can also use the **ladder.corrector** function if it’s really really messed up even after trying the above, but I would only recommend this if nothing else works as it’s quite tedious and not easy to use.

### Peak selection

As well as the following variables, you may want to change the BPselection to reduce the amount of noise (if you are aberrantly selecting noise).

**storing.inds**

**fourier** = T or F; you can change this to be false (F) if you don’t want Fourier transformed peaks. By default true (T)

**channels** = How many channels to read. LIZ ladders are usually 5, which is the default here.

**score.markers2**

**channel** = If your data is in another castle, er channel

**init.thresh** = If you want a hard intensity cut-off to select peaks

**ploidy** = The number of maximum peaks, default = 60

**shift** = Number of bp to discard neighboring peaks. By default 1.5, but you can change if necessary.

**left.cond / right.cond** = You can change the first number to select peaks of different sizes compared to the modal. By default, only finds peaks >10% of the intensity of the modal peak.

## References

1. Covarrubias-Pazaran, G., Diaz-Garcia, L., Schlautman, B., Salazar, W. & Zalapa, J. Fragman: an R package for fragment analysis. *BMC Genet.* **17**, 62 (2016).