A guide to using PeakR

Introduction

PeakR is an R shiny app which is designed to help speed up the processing of electropherogram data from the Applied Biosystems programs PeakScanner and GeneMapper. PeakR is not a finished program yet so certain data files or using it in a certain way may cause the program to crash without warning. Hopefully I can eventually improve PeakR so that it is more robust and usable.

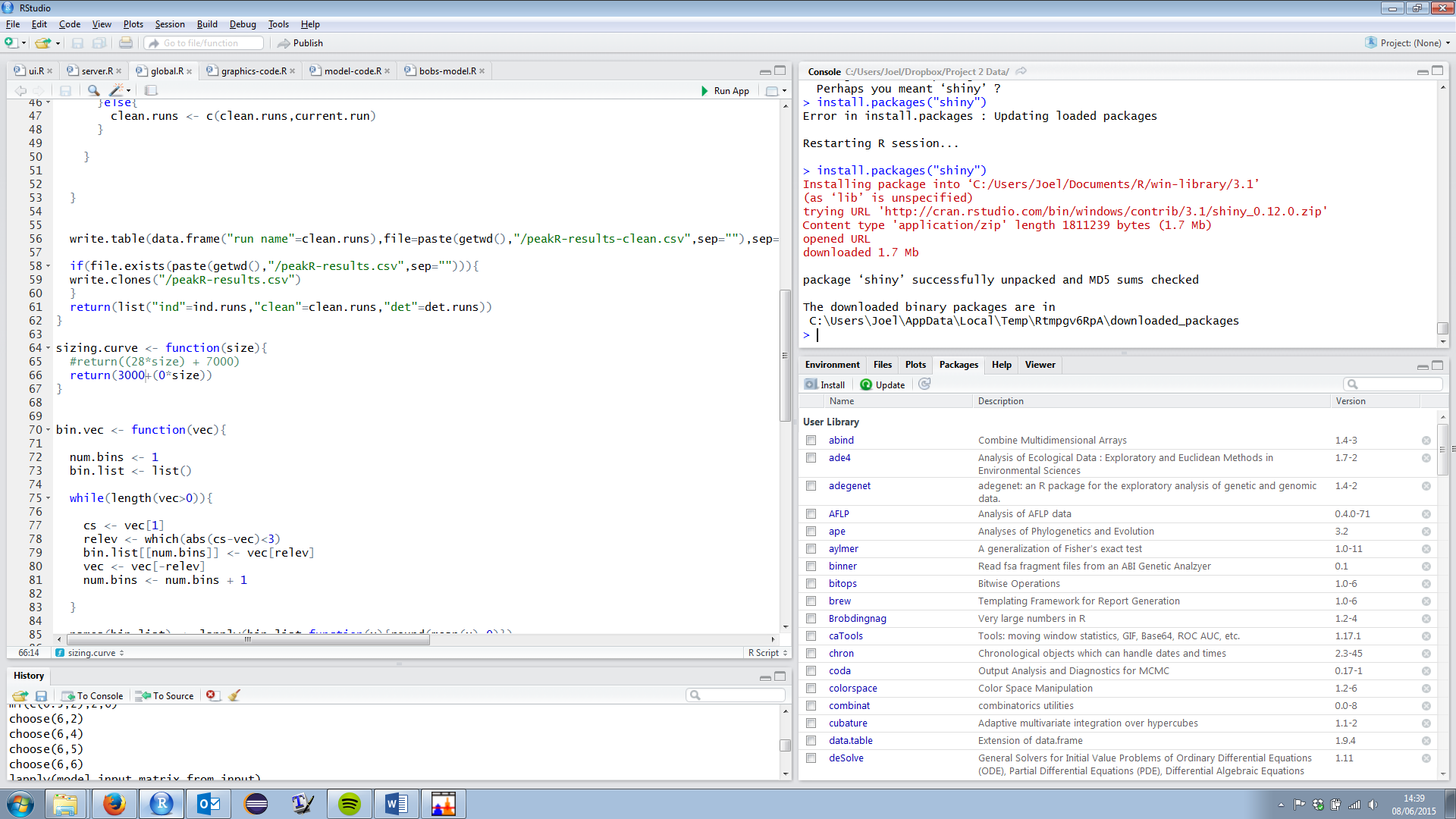
Downloading and installing PeakR

Firstly you will need to have the Applied Biosystems software Peak Scanner installed which is available from here (<http://www.appliedbiosystems.com/absite/us/en/home/support/software-community/free-ab-software.html>). This software will work the raw capillary electrophoresis data and turn it into a data format which PeakR can use.

Next you will need to install R Studio, an integrated development environment for the R statistical software (<http://www.rstudio.com/>). Once R Studio is installed, open it and type the following line into the box called *console*:

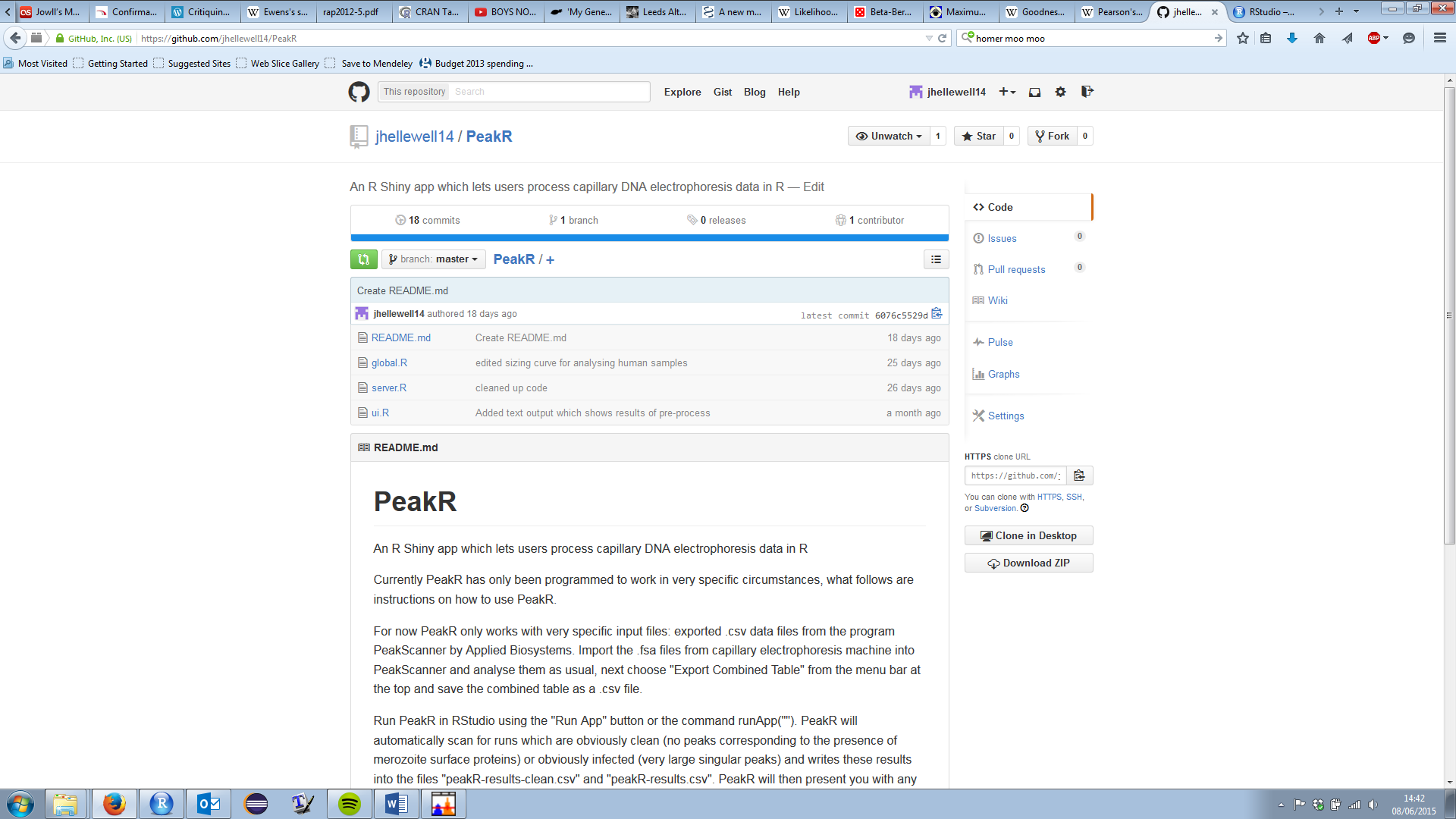
Install.packages(“shiny”)

Some output should appear which should look like this:



The next step before you can begin using PeakR is to download the application files from <https://github.com/jhellewell14/PeakR>

The easiest way to do this is to look for the Download ZIP button on the right hand side of the page. This is shown below in red. Unzip the zip file and put the contents wherever you want to keep them.



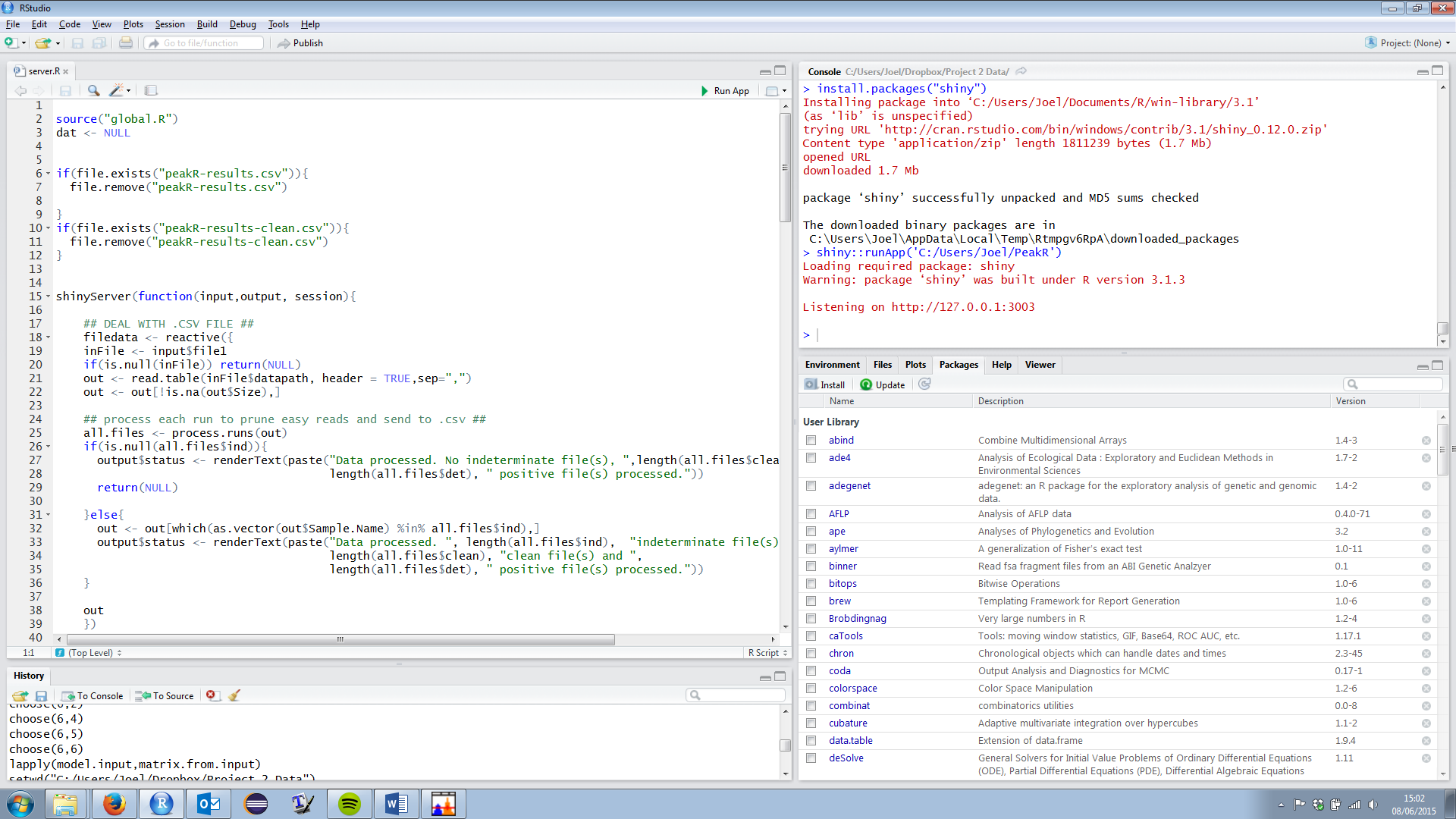
Hopefully now PeakR should be ready to use with your data.

Getting data for PeakR to use

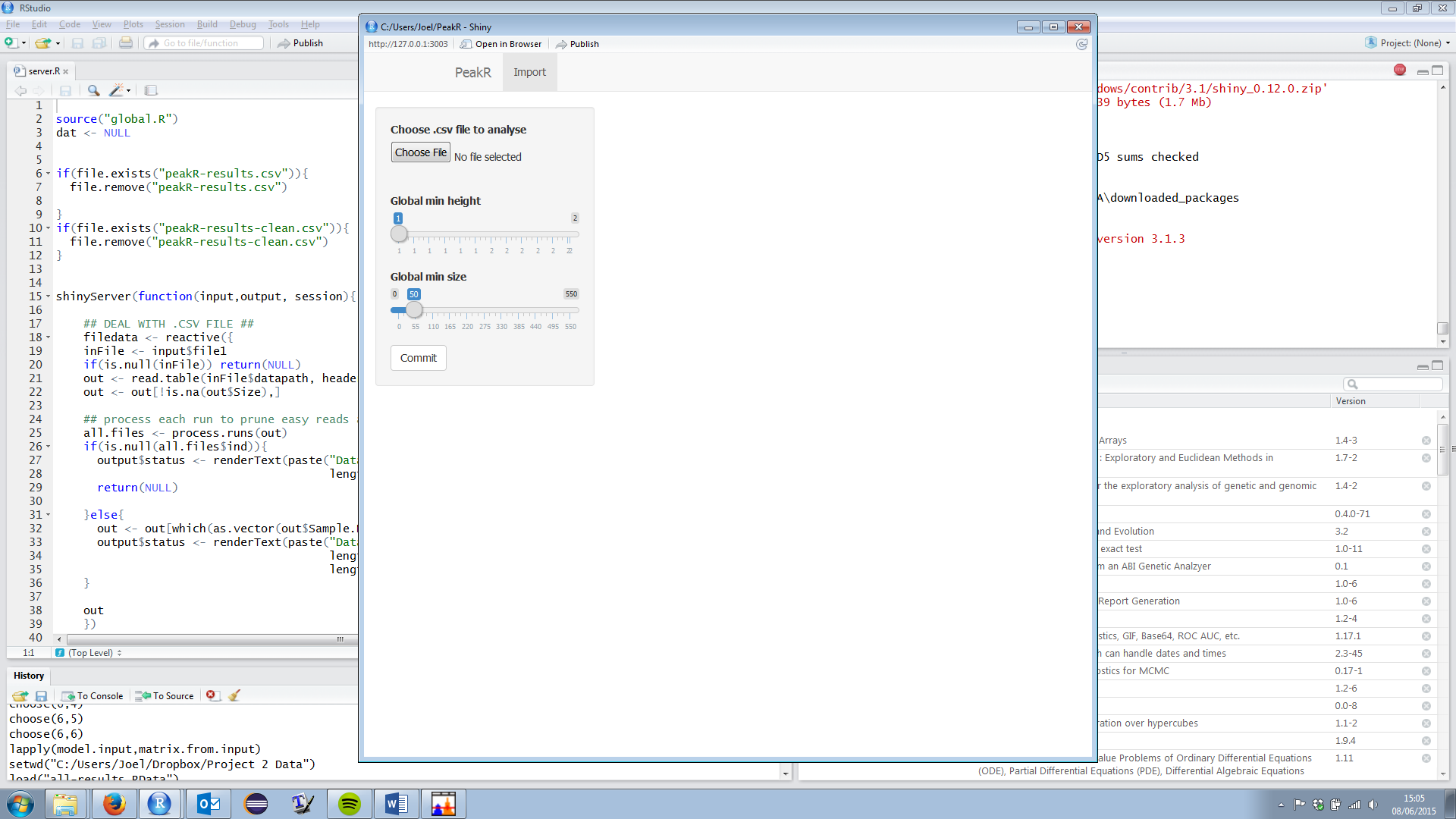
In PeakScanner 2 start a new project and add all of the data files that you want to analyse to the project. Set the size standard and analysis method and run the analysis. Once the analysis is complete press the small down arrow next to the Export button at the top and select “Export Combined Table”. Make sure to change the save as type from “Tab Delimited” to “Comma Separated Variables” and save the file wherever you like.

Using PeakR

To run PeakR open R Studio and go to “File” and then “Open File…”, then open either “ui.R” or “server.R” which can be found wherever you saved the files from <https://github.com/jhellewell14/PeakR> earlier. To run PeakR press the “Run App” button which is located here:



If everything has gone well so far a program should open which looks like this:



Click the “Choose File” button and open your .csv that you saved from Peak Scanner. PeakR will now search through the file and look at each sample. For each sample PeakR will choose to do one of three things: If no peaks in the electropherogram are above a certain amount then the run will be designated clean; If all peaks in the electropherogram are above a certain threshold then the run is decisively infected; if there are any peaks which fall just below a certain threshold then the run is indecisively infected. Currently this threshold is set at 3000 real fluorescent units and any peaks between 1000 and 3000 rfu make the run indecisive.

You can change the value of this threshold if you need to by changing the following code in the “global.R” file:

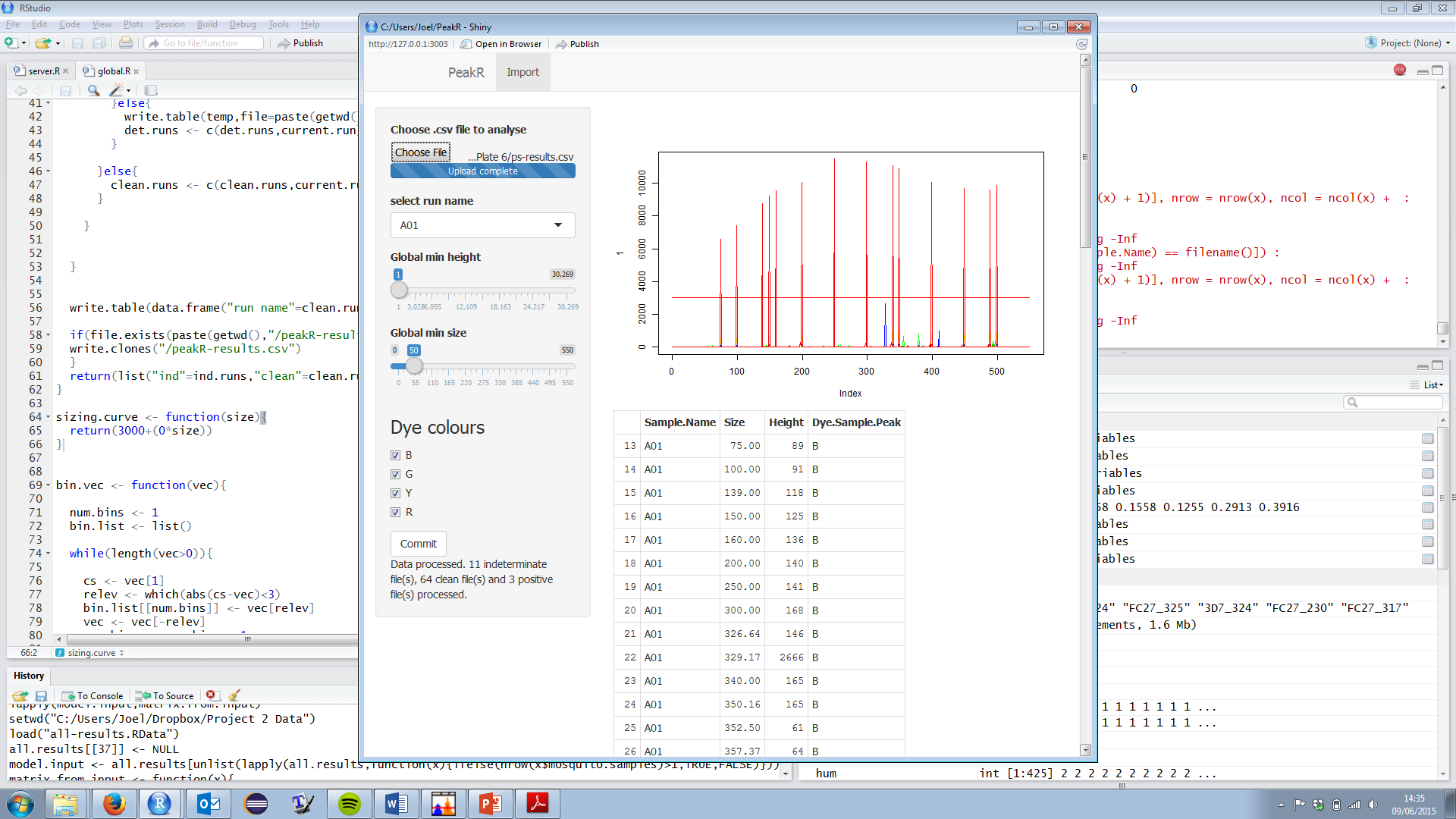
sizing.curve <- function(size){

return(3000+(0\*size))

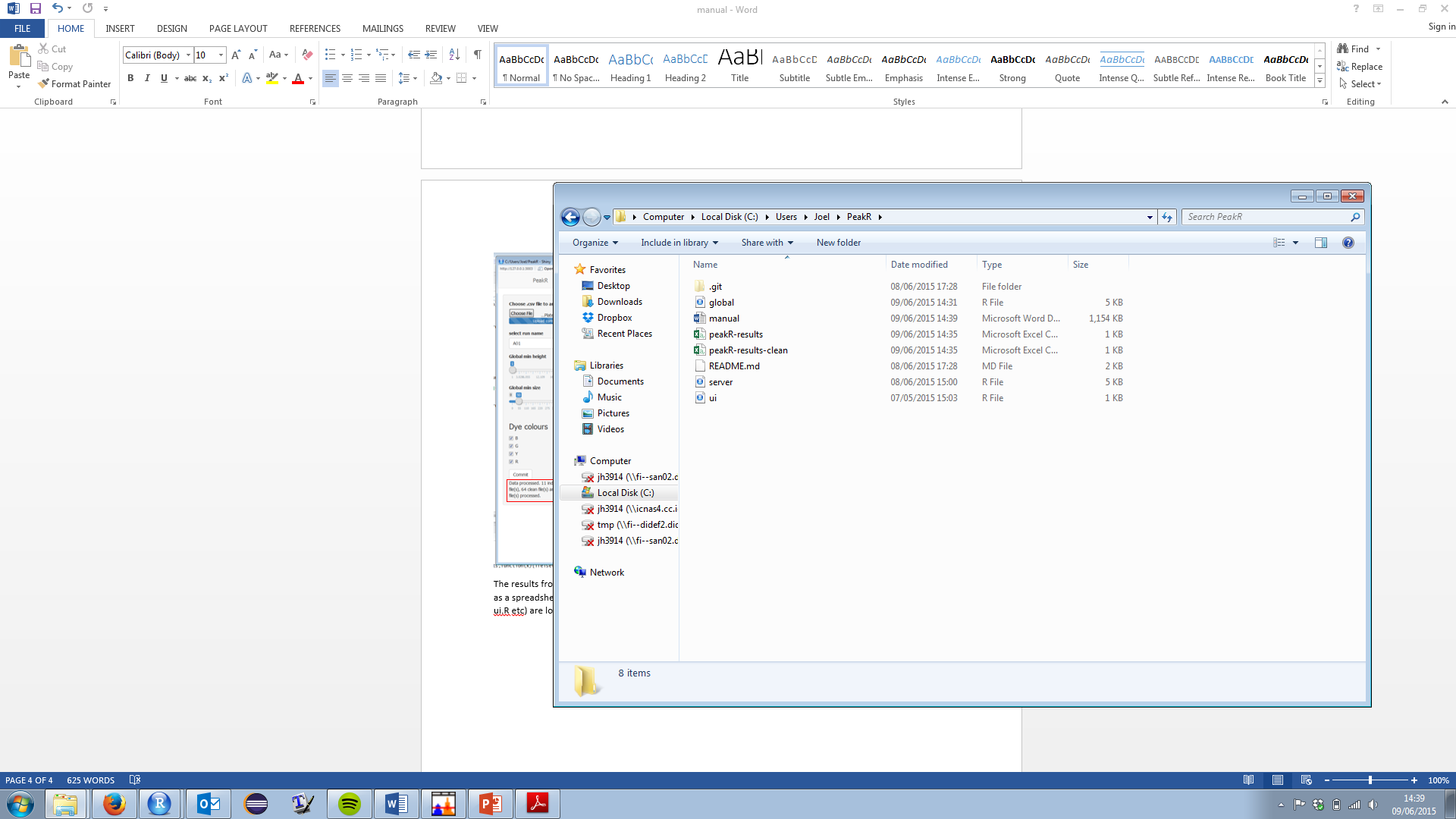
}

Change the value of “3000” to whatever value you desire, save the file, then re-open PeakR. If you are familiar with R you can specify any formula in the function to be your threshold.

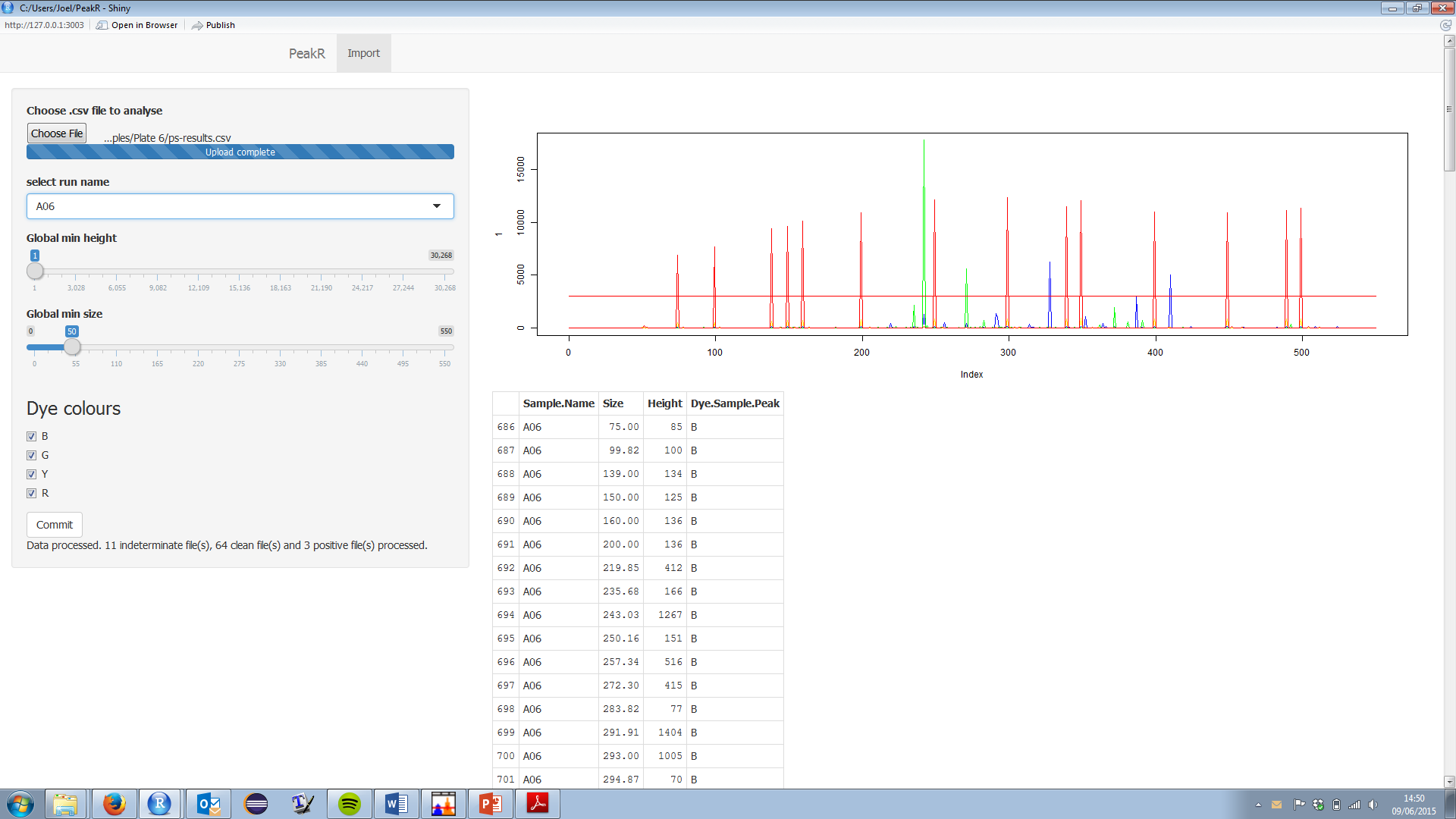
Once PeakR has performed this initial run it will tell you how many files were placed in each of the three categories in a text box at the bottom left of the program.



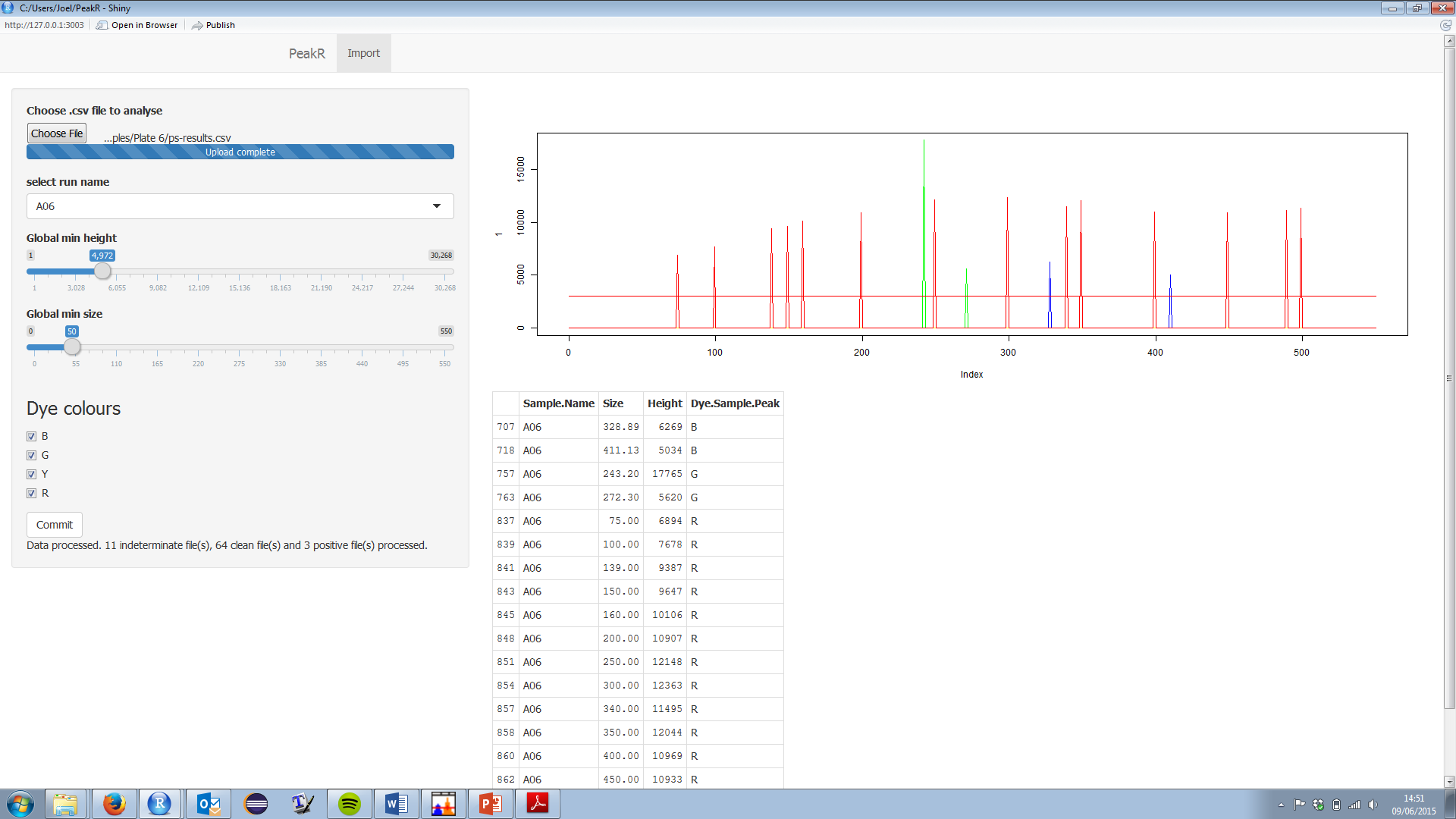
The results from all of the decisively infected and clean files are written into two .csv files which can be opened as a spreadsheet with Excel. You can find these spreadsheets in the folder where your PeakR files (server.R, ui.R etc) are located. “peakR-results” is the file for the infected samples and “peakR-results-clean” is the file for the clean samples. If there are no indeterminate files then your analysis is finished, all of the results are in the two spreadsheets mentioned above. However I would recommend having a look through these spreadsheets to see if they make sense, it might be the case that the threshold for infected peaks is too high and therefore too many files are being categorised as clean.



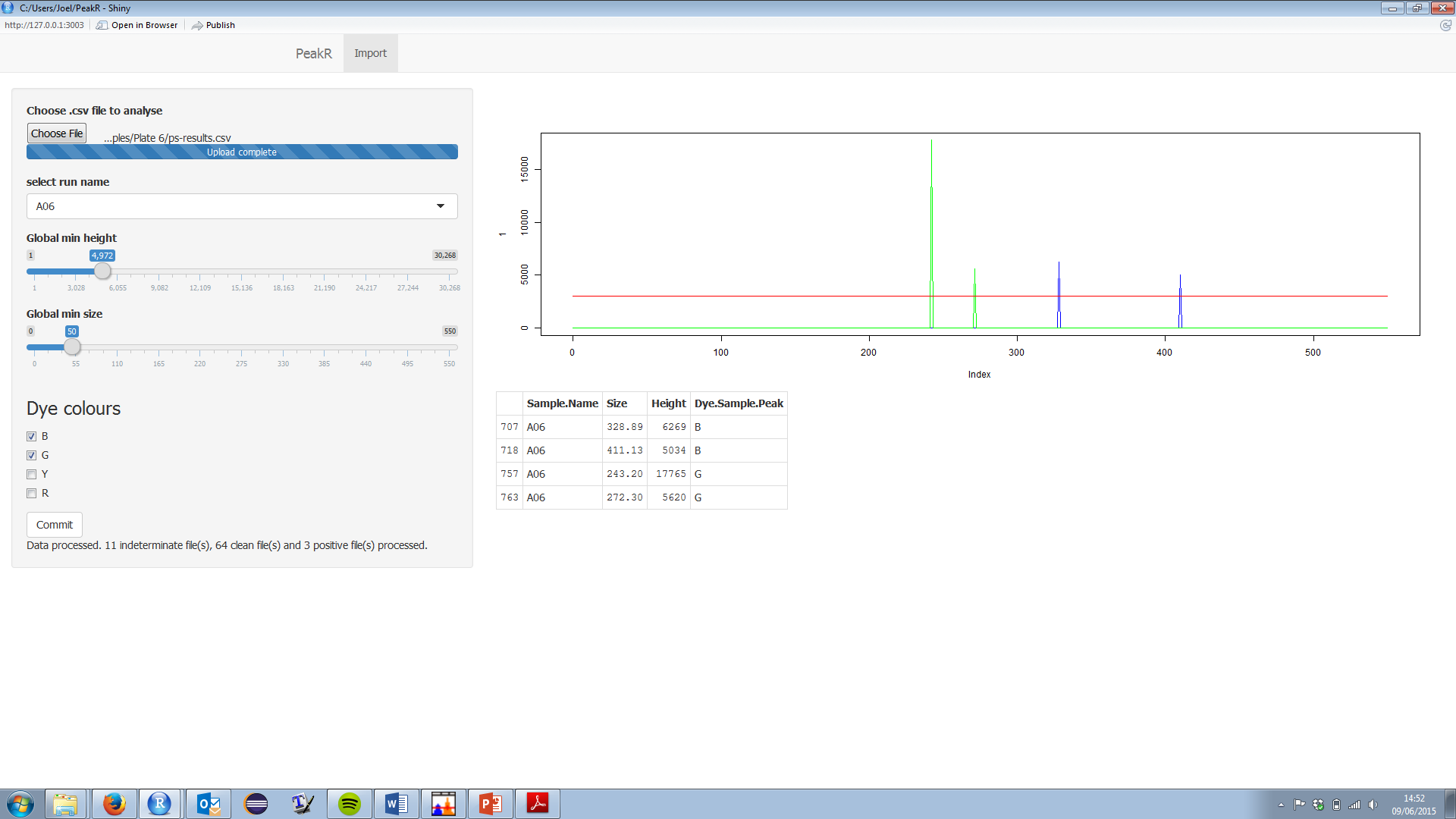
If there are indeterminate files the next step is to tell PeakR which peaks you think are real. You can do this using the sliders and checkboxes in the panel to the left. If I have an electropherogram which looks like this:



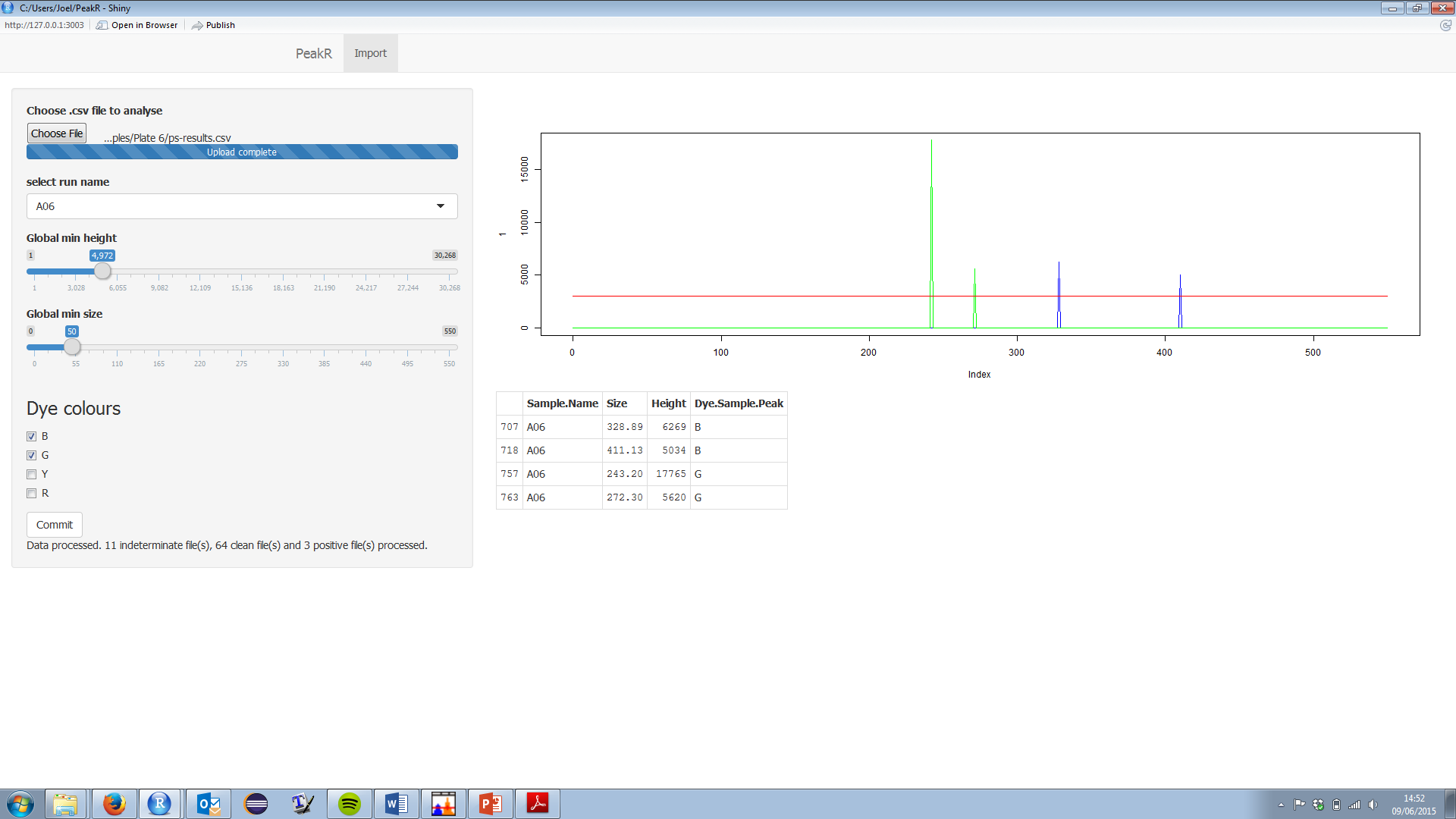
I can raise the global minimum size slider to get rid of some of the smaller peaks:



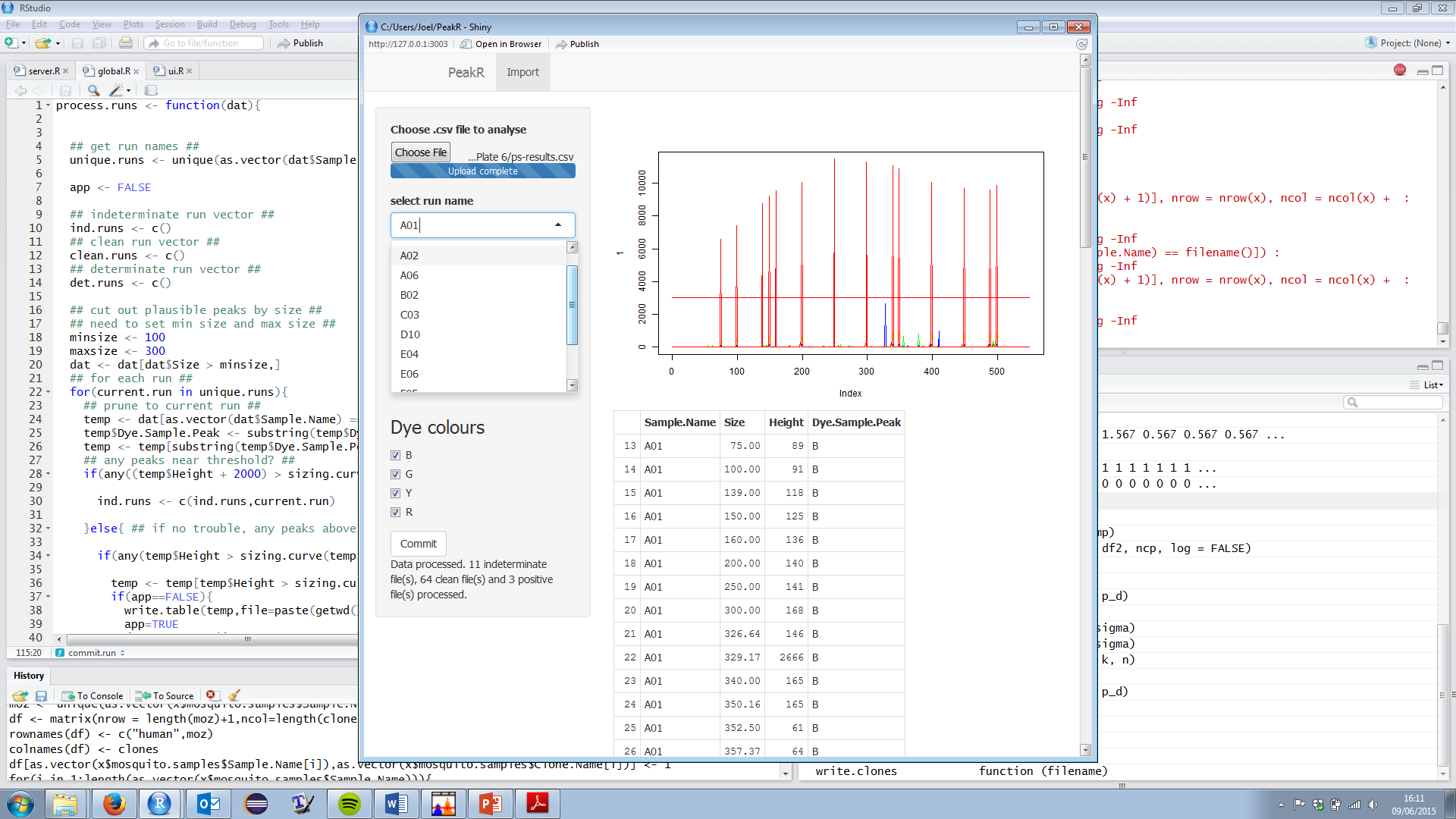
I can also untick the “R” checkbox to remove the red peaks:



I can also look in the table below the electropherogram, this tells me the details of the peaks which are left in the plot:



If I think that these are all peaks which show the presence of malaria then pressing the “Commit” button at the bottom left will write this data into the “peakR-results.csv” spreadsheet. If you press “Commit” and then change your mind you can change the electropherogram and press “Commit” again and this will overwrite the old results. (If you come across an indeterminate run which is actually clean just don’t commit anything and add it manually to peakR-results-clean.csv, I am working on a way around this.) Clicking on the drop down menu on the left will show you all of the indeterminate runs which PeakR would like you to look at, once you have committed results for all of the runs the analysis is finished.



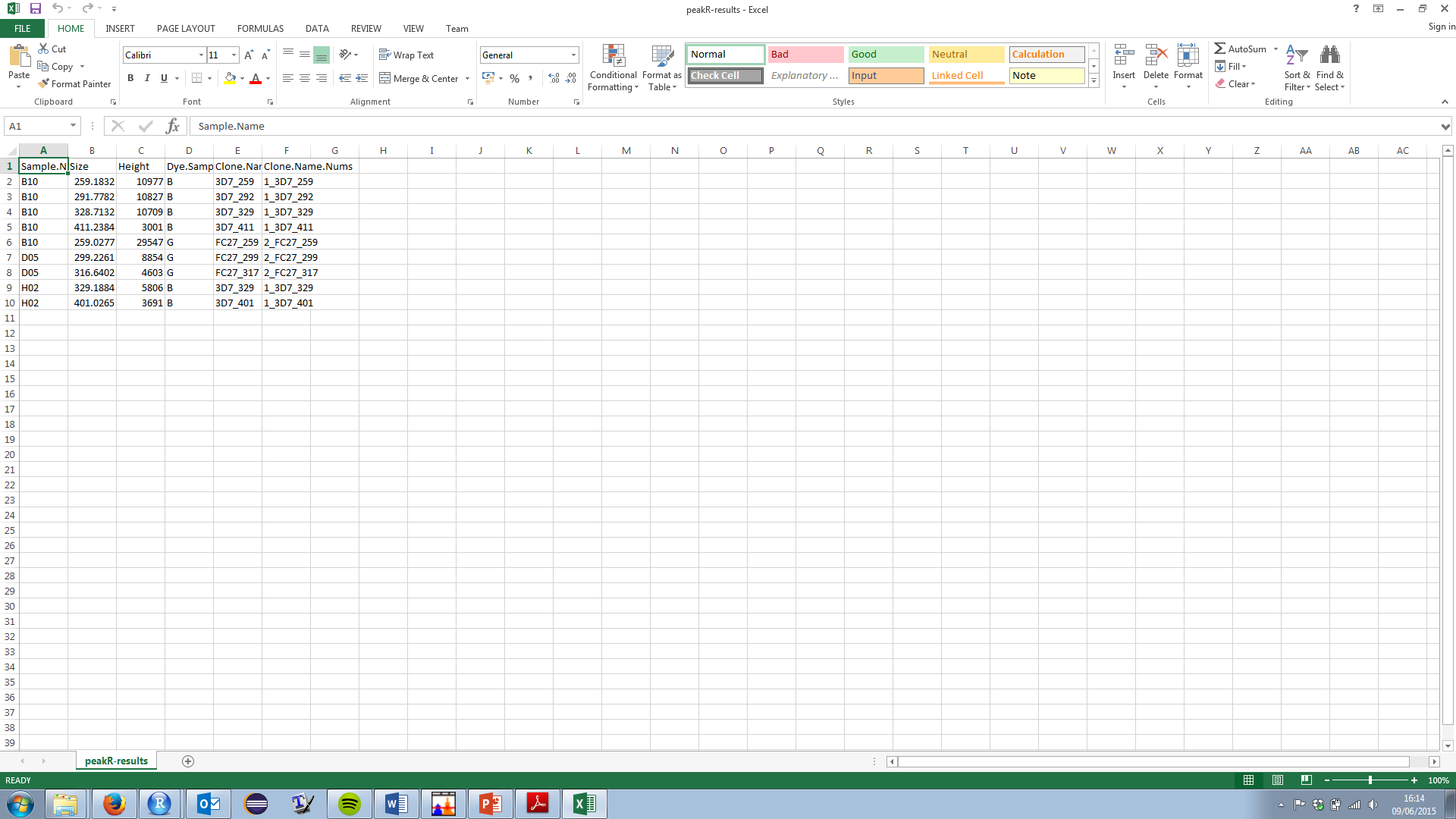
Interpreting PeakR output

**Clean runs**

The peakR-clean-results.csv is simple a list of the sample names for all runs which are clean, because these sample names are the same on each plate it is least confusing to analyse one plate of samples at a time with PeakR.

**Infected runs**

The peakR-results.csv spreadsheet is a list of all of the peaks found in each sample:



It lists the sample name, the height of the peak, the size (in base pairs) where the peak occurred, the dye colour of the peak and the program’s own attempt to name the clones. The program has a simple naming algorithm where it takes the first clone in the list, finds all clones from the same allelic family within 3 base pairs of the clone in size, groups these clones together and gives them a clone name based on the mean size of these clones (rounded up or down). This algorithm is a quick and easy way to name the clones but if you need to be more accurate when naming the clones it is perhaps best to do this yourself (This is because the algorithm might select clone A which is within 3 base pairs of clone B so these two clones are grouped, however clone C might be within 3 base pairs of clone B but not clone A so it is not included in the group. If the algorithm had selected clone B first then clones A and C would both be in the group. Discrepancies such as this are hard to account for without a much more complex naming algorithm – which I don’t have time to do for now).