How to use the python coulter counter fitting script

Updated 24/08/2022

Based on script from Rose McNelly

What does the script do?

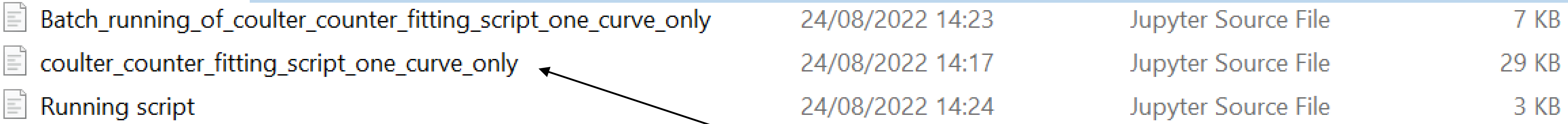
- The script fits 2 different models to unimodal coulter counter traces:

1) A normal curve
2) A lognormal curve

- It compares the different fittings using parameter uncertainties and a standard error of regression test. It produces estimates of granule diameter and for each model.

What are the scripts?

2) Batch_running_of_coulter_counter_script_one_curve_only – this script will take each coulter counter csv file in turn and feed them into the coulter_counter_fitting_script_one_curve_only. It also saves all the parameters from the different fittings and produces summary excel sheets (see later for info about outputs)



Batch_running_of_coulter_counter_fitting_script_one_curve_only	24/08/2022 14:23	Jupyter Source File	7 KB
coulter_counter_fitting_script_one_curve_only	24/08/2022 14:17	Jupyter Source File	29 KB
Running script	24/08/2022 14:24	Jupyter Source File	3 KB

1) Running script – this script is used to change our working directory and submit the other two scripts

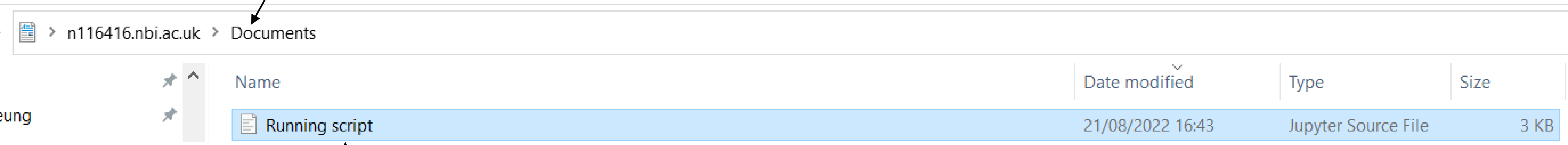
3) coulter_counter_fitting_script_one_curve_only – this script does the actual fittings. It takes one input file and fits the double normal, double lognormal and lognormal-normal curves to the data. It produces a PDF with graphs of the 3 different fittings and a comparison of the 3.

Why can't we just use the coulter_counter_fitting_script by itself? Technically we could but in order to get excel sheets with all the parameters in we need to use script 2 (batch_running_of_coulter_counter_script_one_curve_only). It also means that if we have multiple coulter counter files to analyse at once we only need to submit the script once and not for every input file. The running script also makes it easier so that we can tell the computer where are input files are (so that we don't have to copy them into our Jupyter lab directory – see later for info about Jupyter lab).

How should my data be organised?

I always place the Running script (only this has to be here) in my Documents folder as this is easy to navigate to using Jupyter notebook (see later)

Documents folder on your computer



n116416.nbi.ac.uk > Documents

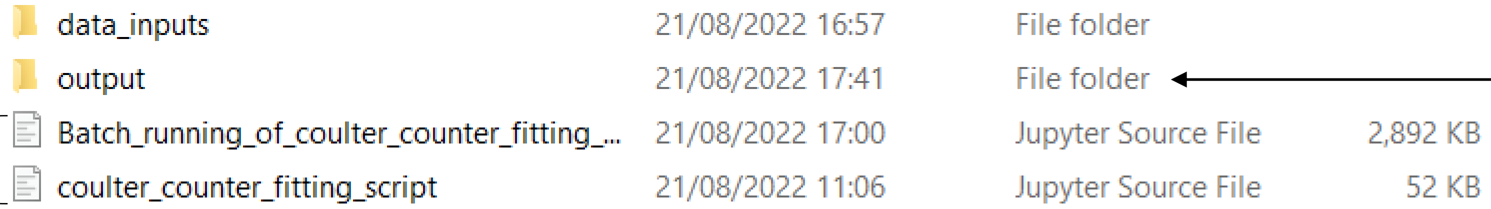
Name	Date modified	Type	Size
Running script	21/08/2022 16:43	Jupyter Source File	3 KB

Only the Running script needs to be here

How should my data be organised?

Everything else can be in a different folder, this can be anywhere and doesn’t have to be within your Documents folder, but if you want it in your Documents folder this is fine too. It should be organised as shown here:

data_inputs folder – contains all the csv files from the coulter counter (if it contains other file types it should run ok as the script should only read in csv files)



data_inputs	21/08/2022 16:57	File folder	
output	21/08/2022 17:41	File folder	
Batch_running_of_coulter_counter_fitting_...	21/08/2022 17:00	Jupyter Source File	2,892 KB
coulter_counter_fitting_script	21/08/2022 11:06	Jupyter Source File	52 KB

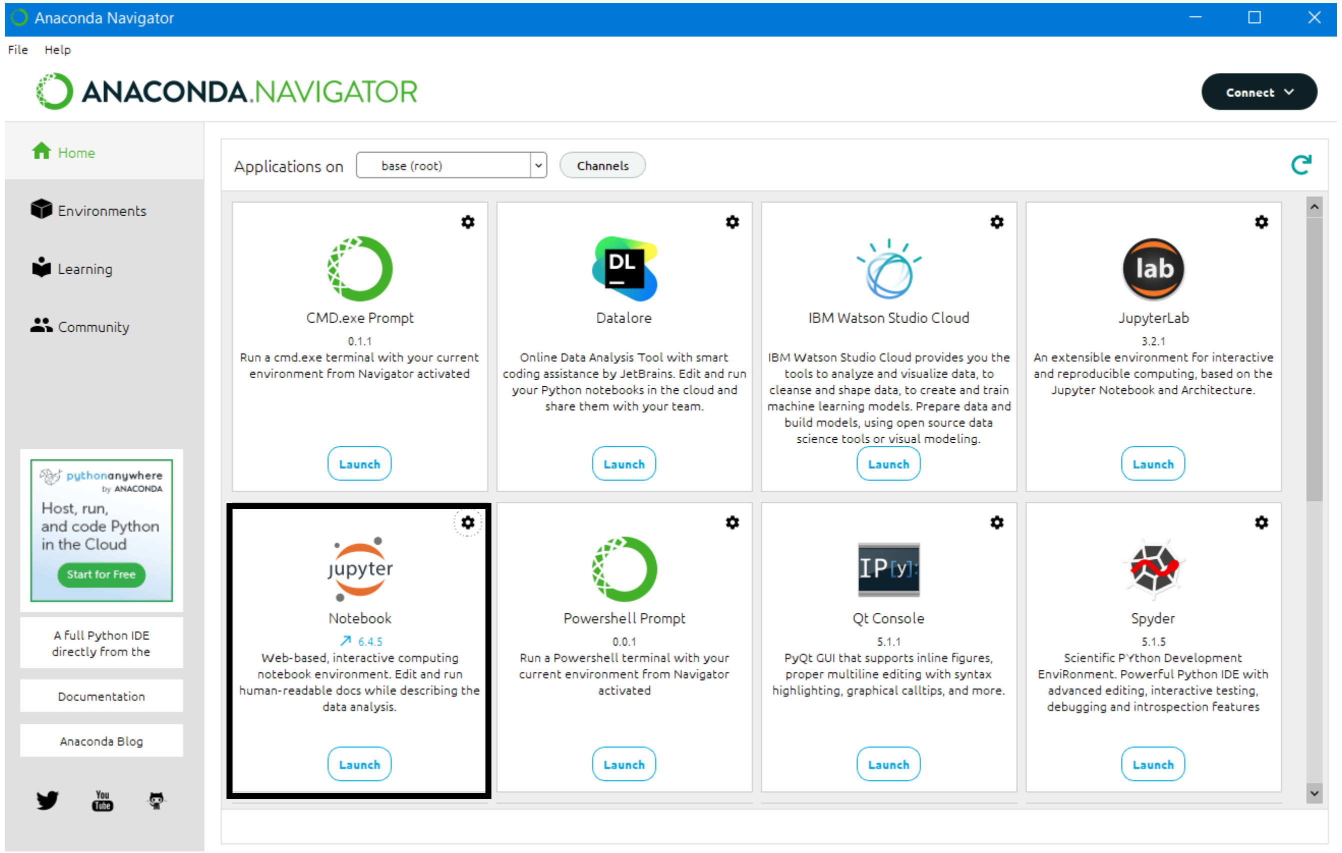
Empty folder called output – the output files (see later) will be added to this folder

The batch_running_of_coulter_counter_fitting_script and the coulter_counter_fitting_script

The folder which is organised like this is you **working directory,** take note of the name of this folder and where it is (the full path) as you’ll need it

How do I run the script?

1) Download Anaconda Navigator (https://docs.anaconda.com/anaconda/install/) and open Jupyter Notebook (this will open in an internet browser)



CMD.exe Prompt
0.1.1
Run a cmd.exe terminal with your current environment from Navigator activated

Datalore
Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud
IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

JupyterLab
3.2.1
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook
↗ 6.4.5
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

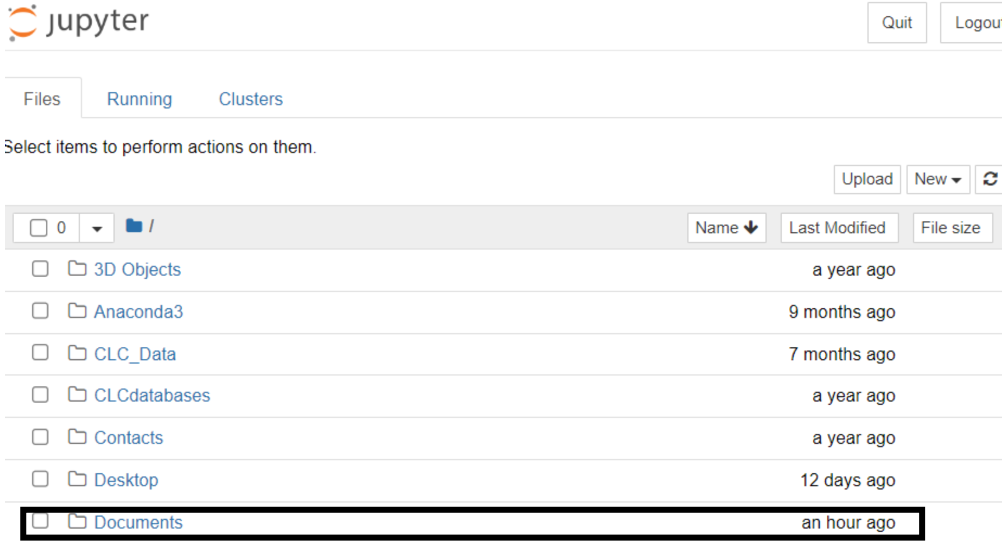
Powershell Prompt
0.0.1
Run a Powershell terminal with your current environment from Navigator activated

Qt Console
5.1.1
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Spyder
5.1.5
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

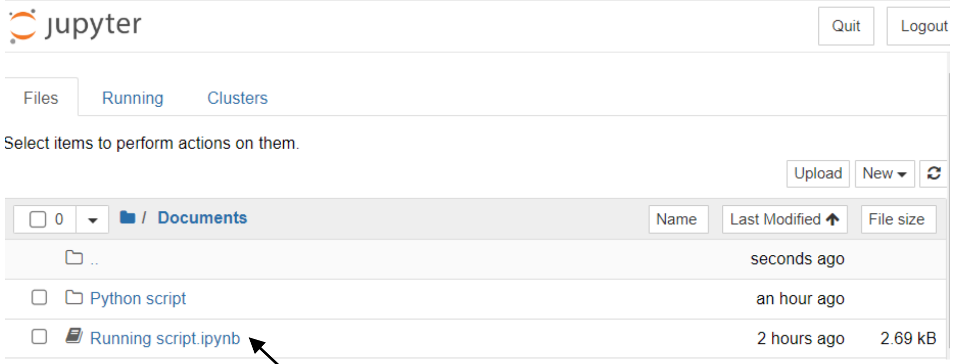
How do I run the script?

2) Navigate to your Documents folder



Select items to perform actions on them.

Name	Last Modified	File size
3D Objects	a year ago	
Anaconda3	9 months ago	
CLC_Data	7 months ago	
CLCdatabases	a year ago	
Contacts	a year ago	
Desktop	12 days ago	
Documents	an hour ago	



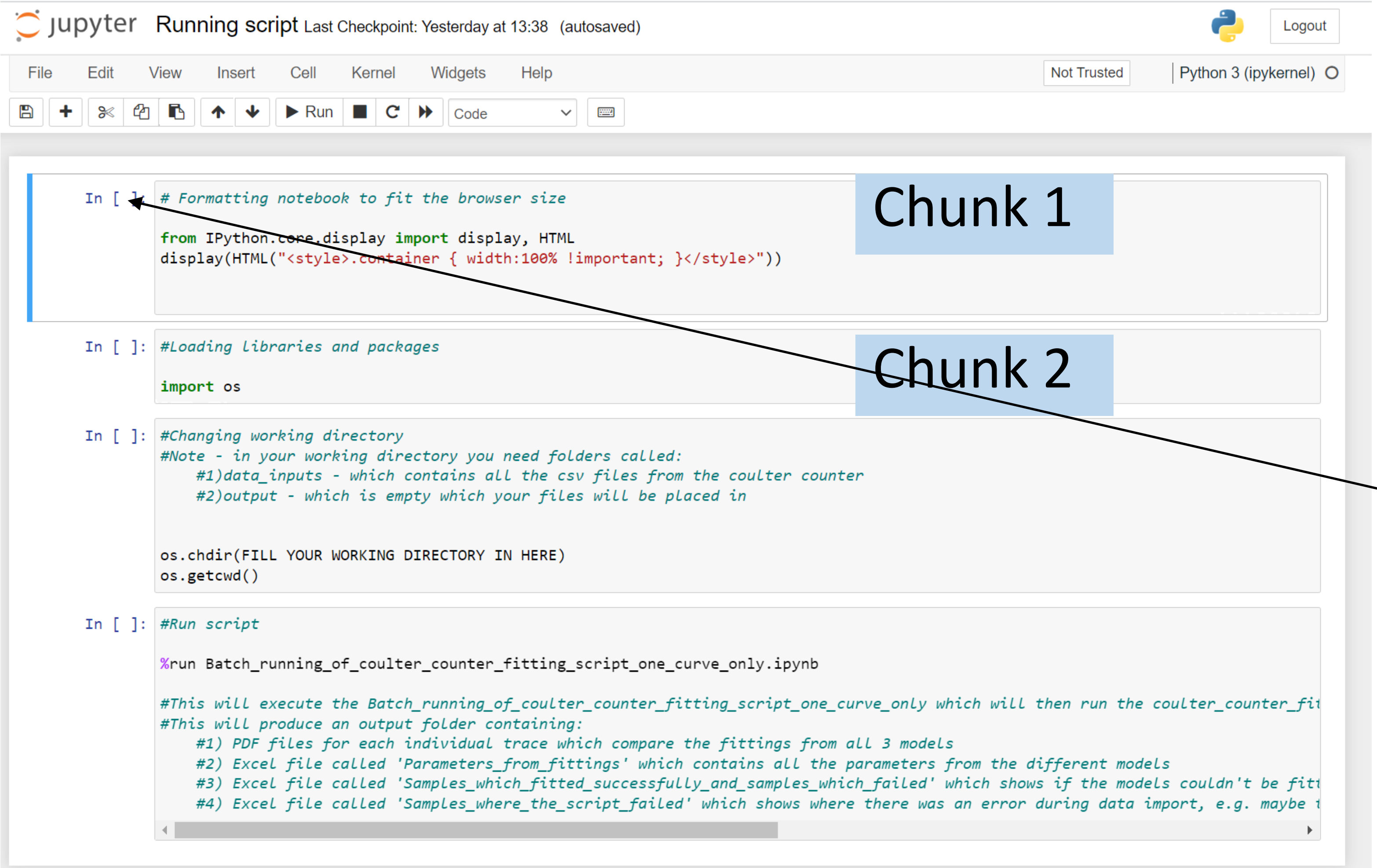
Select items to perform actions on them.

Name	Last Modified	File size
..	seconds ago	
Python script	an hour ago	
Running script.ipynb	2 hours ago	2.69 kB

You'll see the Running script here

3) Open the Running script

How do I run the script?



```
In [ ]: # Formatting notebook to fit the browser size

        from IPython.core.display import display, HTML
        display(HTML("<style>.container { width:100% !important; }</style>"))
```

```
In [ ]: #Loading libraries and packages

        import os
```

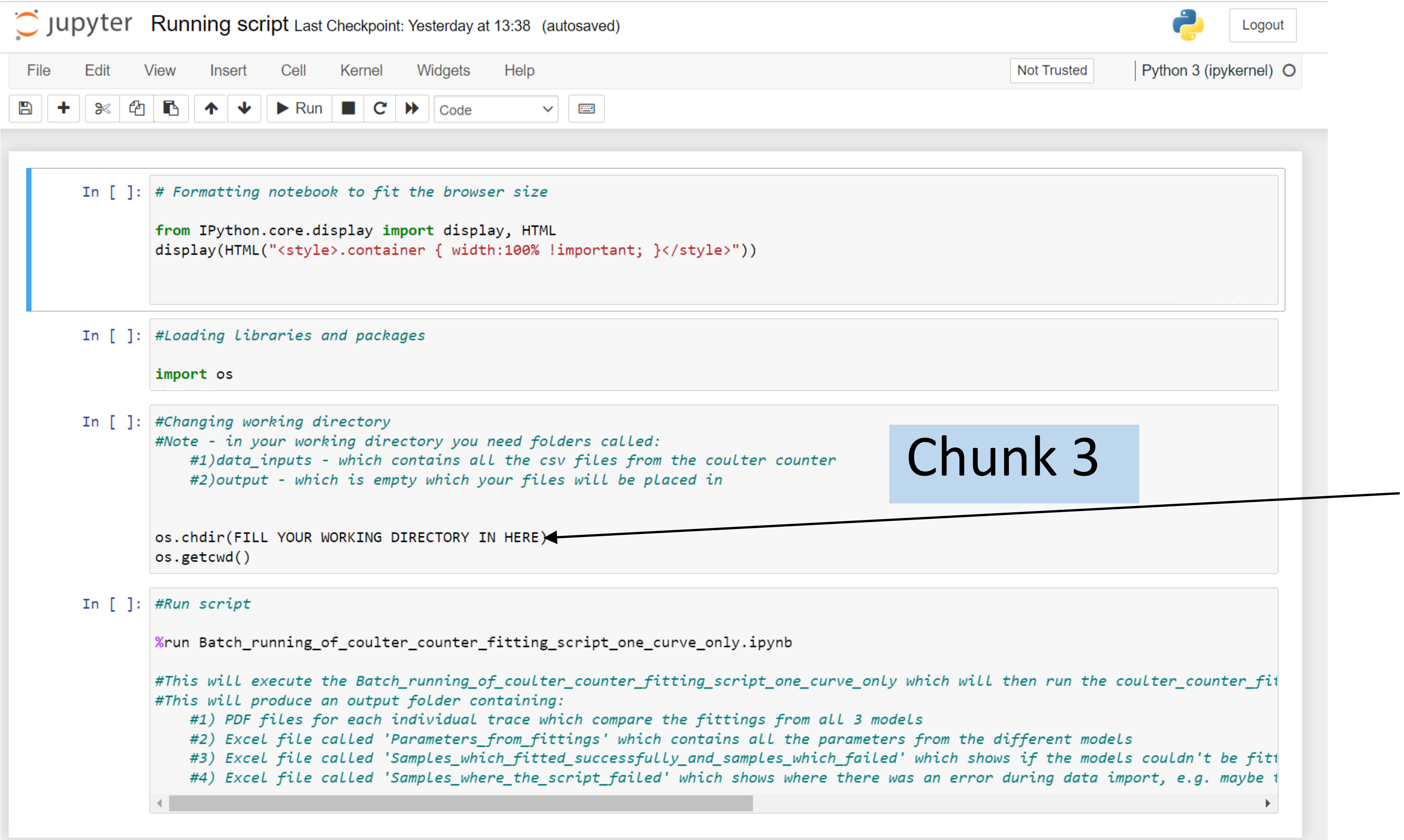
```
In [ ]: #Changing working directory
        #Note - in your working directory you need folders called:
            #1)data_inputs - which contains all the csv files from the coulter counter
            #2)output - which is empty which your files will be placed in


        os.chdir(FILL YOUR WORKING DIRECTORY IN HERE)
        os.getcwd()
```

```
In [ ]: #Run script

        %run Batch_running_of_coulter_counter_fitting_script_one_curve_only.ipynb

        #This will execute the Batch_running_of_coulter_counter_fitting_script_one_curve_only which will then run the coulter_counter_fit
        #This will produce an output folder containing:
            #1) PDF files for each individual trace which compare the fittings from all 3 models
            #2) Excel file called 'Parameters_from_fittings' which contains all the parameters from the different models
            #3) Excel file called 'Samples_which_fitted_successfully_and_samples_which_failed' which shows if the models couldn't be fitt
            #4) Excel file called 'Samples_where_the_script_failed' which shows where there was an error during data import, e.g. maybe t
```

4) Run the first two chunks (shown as grey boxes), by clicking in each chunk and pressing Ctrl and Enter.

If it has run successfully a small number should appear in the square brackets on the left of the chunk

How do I run the script?



```
In [ ]: # Formatting notebook to fit the browser size

        from IPython.core.display import display, HTML
        display(HTML("<style>.container { width:100% !important; }</style>"))
```

```
In [ ]: #Loading libraries and packages

        import os
```

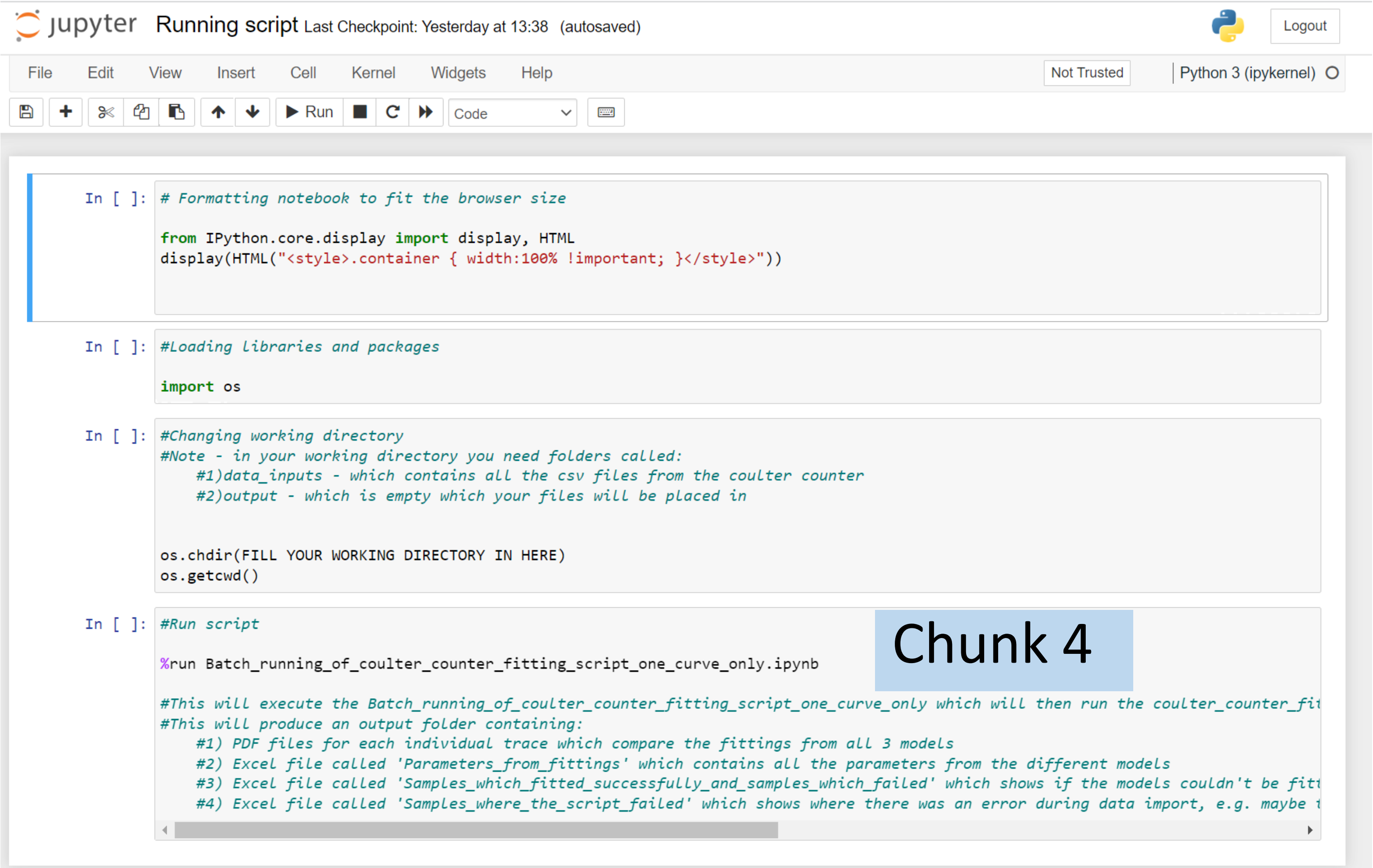
```
In [ ]: #Changing working directory
        #Note - in your working directory you need folders called:
            #1)data_inputs - which contains all the csv files from the coulter counter
            #2)output - which is empty which your files will be placed in


        os.chdir(FILL YOUR WORKING DIRECTORY IN HERE)
        os.getcwd()
```

```
In [ ]: #Run script

        %run Batch_running_of_coulter_counter_fitting_script_one_curve_only.ipynb

        #This will execute the Batch_running_of_coulter_counter_fitting_script_one_curve_only which will then run the coulter_counter_fit
        #This will produce an output folder containing:
            #1) PDF files for each individual trace which compare the fittings from all 3 models
            #2) Excel file called 'Parameters_from_fittings' which contains all the parameters from the different models
            #3) Excel file called 'Samples_which_fitted_successfully_and_samples_which_failed' which shows if the models couldn't be fitt
            #4) Excel file called 'Samples_where_the_script_failed' which shows where there was an error during data import, e.g. maybe t
```

5) In chunk 3, change your working directory to the location which contains the other scripts, your data_inputs and output folder. You need to put in the full path, not just the folder name, and put it within quotation marks, e.g. ‘U:\Working_Directory_Folder’

Run the chunk, if it has worked then it should print the working directory folder path directly underneath the chunk.

How do I run the script?



```
In [ ]: # Formatting notebook to fit the browser size

        from IPython.core.display import display, HTML
        display(HTML("<style>.container { width:100% !important; }</style>"))
```

```
In [ ]: #Loading libraries and packages

        import os
```

```
In [ ]: #Changing working directory
        #Note - in your working directory you need folders called:
            #1)data_inputs - which contains all the csv files from the coulter counter
            #2)output - which is empty which your files will be placed in


        os.chdir(FILL YOUR WORKING DIRECTORY IN HERE)
        os.getcwd()
```

```
In [ ]: #Run script

        %run Batch_running_of_coulter_counter_fitting_script_one_curve_only.ipynb

        #This will execute the Batch_running_of_coulter_counter_fitting_script_one_curve_only which will then run the coulter_counter_fit
        #This will produce an output folder containing:
            #1) PDF files for each individual trace which compare the fittings from all 3 models
            #2) Excel file called 'Parameters_from_fittings' which contains all the parameters from the different models
            #3) Excel file called 'Samples_which_fitted_successfully_and_samples_which_failed' which shows if the models couldn't be fitt
            #4) Excel file called 'Samples_where_the_script_failed' which shows where there was an error during data import, e.g. maybe t
```

Chunk 4

6) Run chunk 4 – this submits the Batch_running_of_coulter_counter_script_one_curve_only which itself will submit the coulter_counter_fitting_script_one_curve_only (you won't have to do anything)

You'll start to see the outputs appear in the output folder

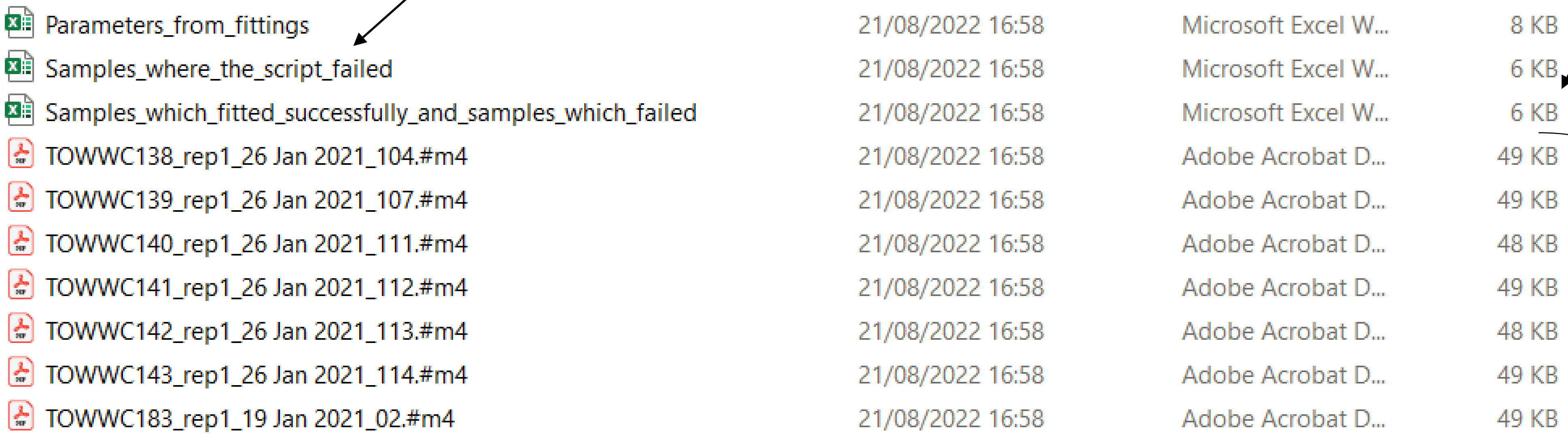
If you want to know how the script works see later

What are the outputs?

Parameters_from_fittings – contains all the parameters from the 3 different fittings and also contains the uncertainties and the standard error of regression

Samples_where_the_script_failed – contains a list of filenames where the script failed (most likely during data import), if a filename appears in this spreadsheet it could be because the coulter counter CSV file is corrupted or not in the correct format

Samples_which_fitted_successful_and_samples_which_failed – contains a list of filenames for each of the different fitting methods for samples which worked and samples where the fitting failed. The fitting might have failed if the data is not bimodal in shape.

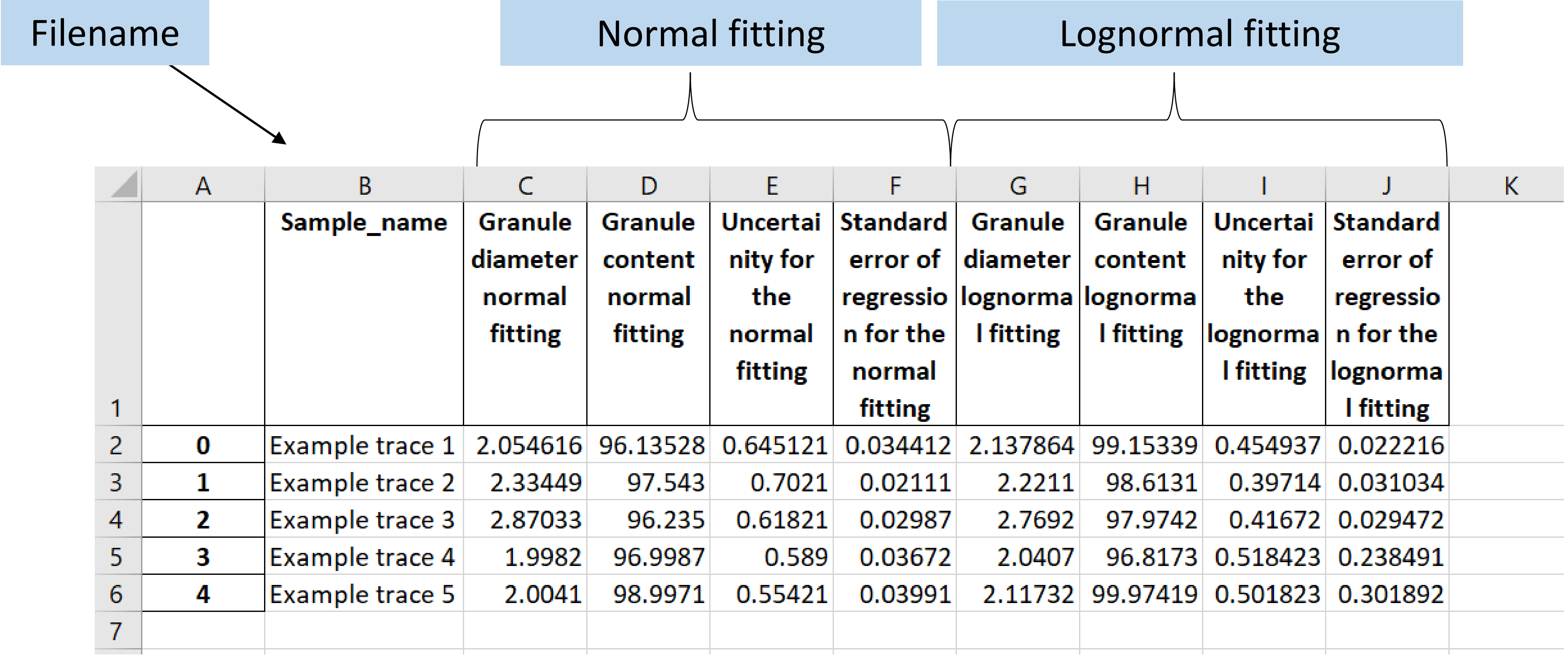


Parameters_from_fittings	21/08/2022 16:58	Microsoft Excel W...	8 KB
Samples_where_the_script_failed	21/08/2022 16:58	Microsoft Excel W...	6 KB
Samples_which_fitted_successfully_and_samples_which_failed	21/08/2022 16:58	Microsoft Excel W...	6 KB
TOWWC138_rep1_26 Jan 2021_104.#m4	21/08/2022 16:58	Adobe Acrobat D...	49 KB
TOWWC139_rep1_26 Jan 2021_107.#m4	21/08/2022 16:58	Adobe Acrobat D...	49 KB
TOWWC140_rep1_26 Jan 2021_111.#m4	21/08/2022 16:58	Adobe Acrobat D...	48 KB
TOWWC141_rep1_26 Jan 2021_112.#m4	21/08/2022 16:58	Adobe Acrobat D...	49 KB
TOWWC142_rep1_26 Jan 2021_113.#m4	21/08/2022 16:58	Adobe Acrobat D...	48 KB
TOWWC143_rep1_26 Jan 2021_114.#m4	21/08/2022 16:58	Adobe Acrobat D...	49 KB
TOWWC183_rep1_19 Jan 2021_02.#m4	21/08/2022 16:58	Adobe Acrobat D...	49 KB

PDF files – contain graphs of all the different fittings

See subsequent slides for more info about all the different outputs

Parameters_from_fittings



Filename

Normal fitting

Lognormal fitting

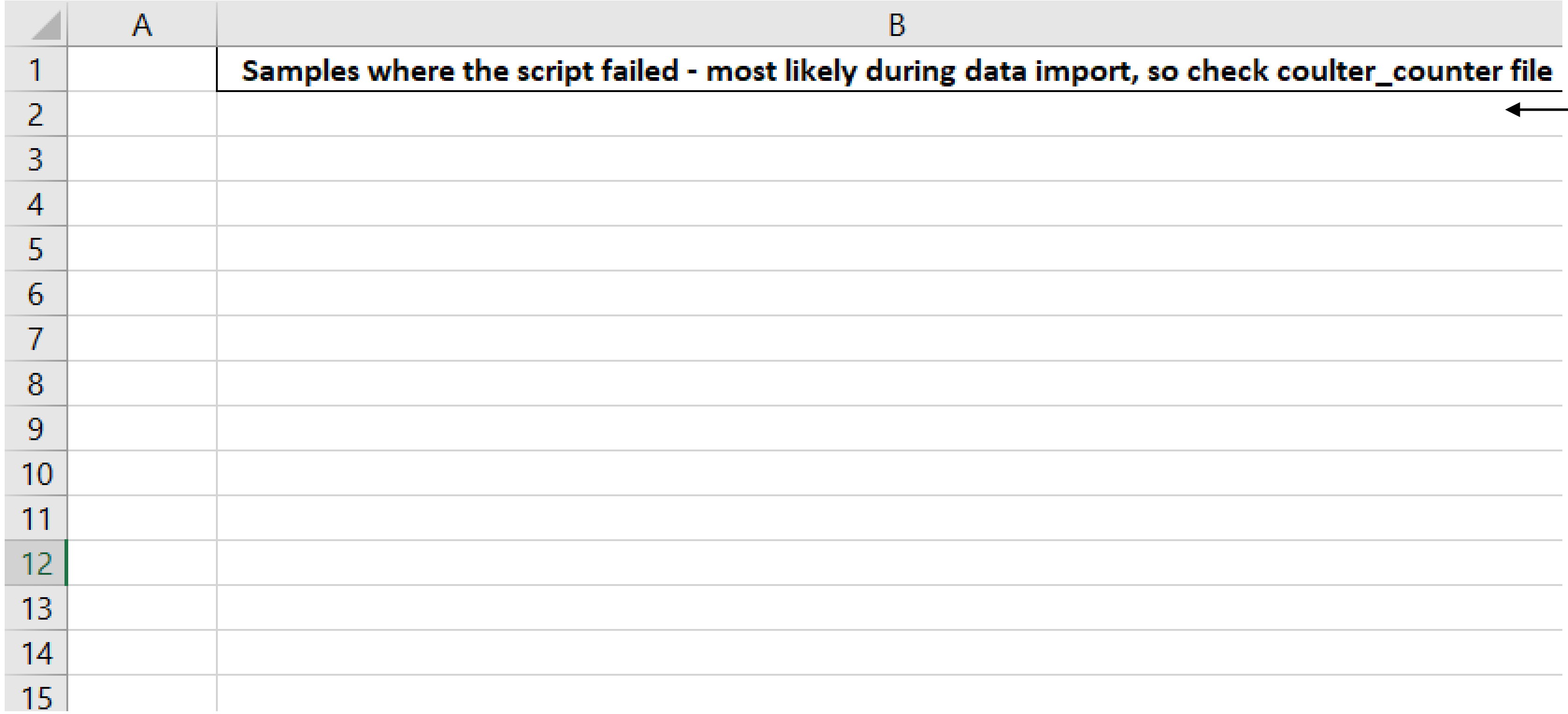
	A	B	C	D	E	F	G	H	I	J	K
1		Sample_name	Granule diameter normal fitting	Granule content normal fitting	Uncertainity for the normal fitting	Standard error of regression for the normal fitting	Granule diameter lognormal fitting	Granule content lognormal fitting	Uncertainity for the lognormal fitting	Standard error of regression for the lognormal fitting	
2	**0**	Example trace 1	2.054616	96.13528	0.645121	0.034412	2.137864	99.15339	0.454937	0.022216	
3	**1**	Example trace 2	2.33449	97.543	0.7021	0.02111	2.2211	98.6131	0.39714	0.031034	
4	**2**	Example trace 3	2.87033	96.235	0.61821	0.02987	2.7692	97.9742	0.41672	0.029472	
5	**3**	Example trace 4	1.9982	96.9987	0.589	0.03672	2.0407	96.8173	0.518423	0.238491	
6	**4**	Example trace 5	2.0041	98.9971	0.55421	0.03991	2.11732	99.97419	0.501823	0.301892	
7											

Granule content is the area under the curve, it is the percentage of the granules which are modelled by the fitting in question

Each row contains the parameters from a different input file

Uncertainty is a measure of how sure we can be about the parameters. Higher uncertainties mean that the parameters in the models could take lots of different values and still result in the same fitting. Therefore **we want the uncertainty to be as low as possible** as this means we are more certain that the model parameters are 'correct'. Standard error of regression is a form of R^2 for non-linear models. It measures how well the predicted model fits the data. Higher standard errors of regression means that the data deviates from the model more. So **we want the standard error of regression to be as low as possible**.

Samples_where_the_script_failed



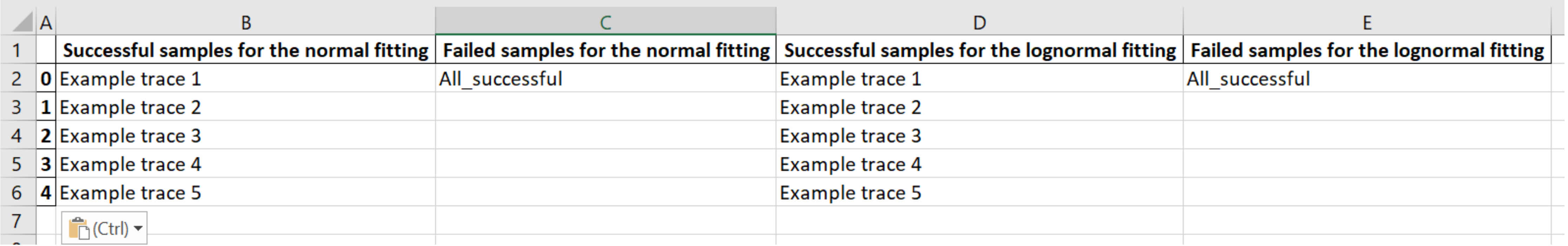
	A	B
1		**Samples where the script failed - most likely during data import, so check coulter_counter file**
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		

If any of the input files failed to load into the script properly they will be listed in this column.

Here this column is empty so all the files were inputted correctly.

NOTE – this does NOT mean that the fitting won’t work as it has failed before the fitting has been attempted.

Samples_which_fitted_successful_and_samples_which_failed

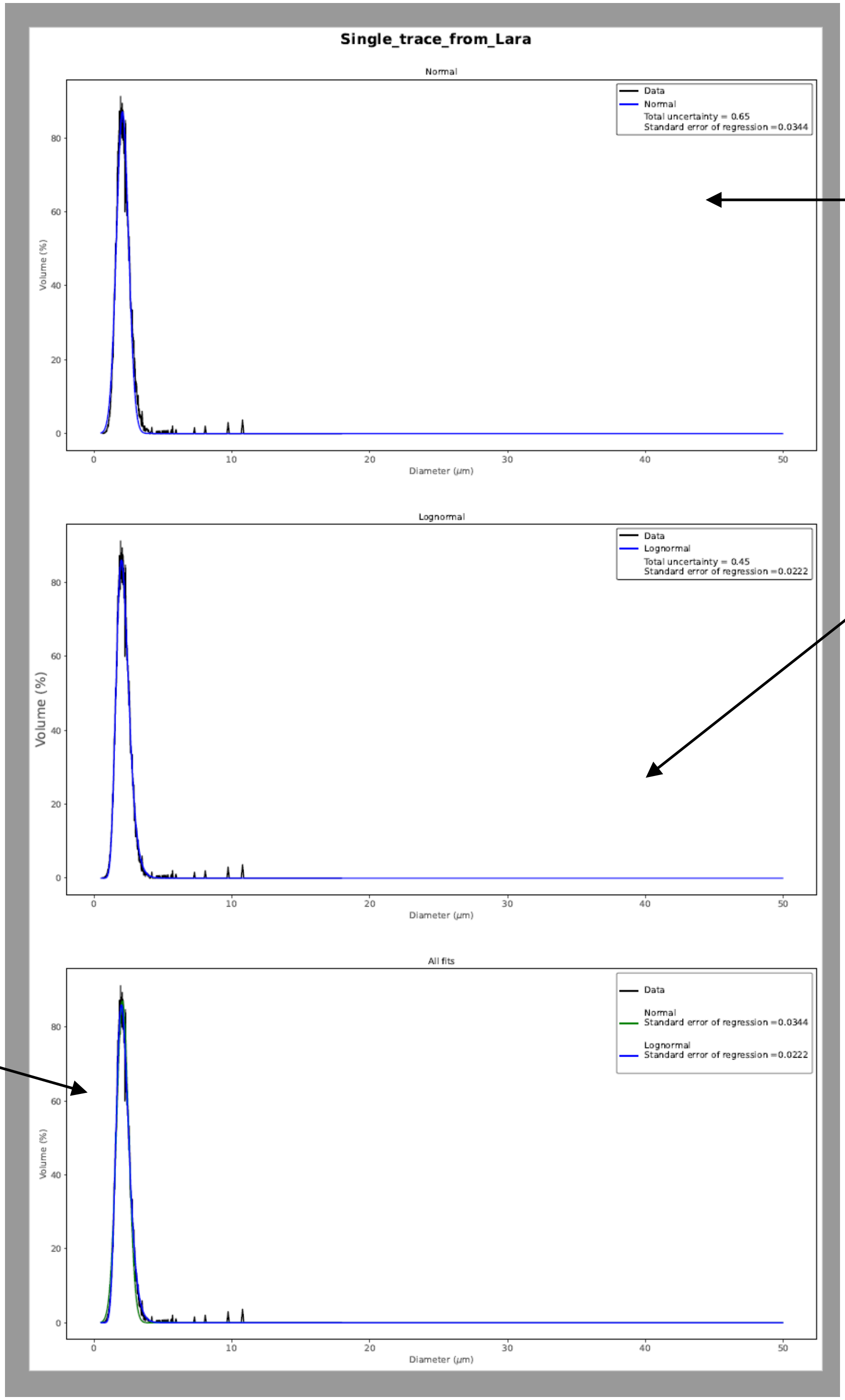


	A	B	C	D	E
1		Successful samples for the normal fitting	Failed samples for the normal fitting	Successful samples for the lognormal fitting	Failed samples for the lognormal fitting
2	0	Example trace 1	All_successful	Example trace 1	All_successful
3	1	Example trace 2		Example trace 2	
4	2	Example trace 3		Example trace 3	
5	3	Example trace 4		Example trace 4	
6	4	Example trace 5		Example trace 5	
7					

Tells us if the fitting failed for the normal (column C) and lognormal (column E) fittings, if any of the fittings failed then the filename will be listed in these columns. This means that the model couldn't be fitted to the data most likely as it significantly deviates from being unimodally distributed. It could be the case that one model might not fit but the other two might.

What should I do if the fitting is failing quite often? You could go into the coulter_counter_fitting_script_one_curve_only and adjust the initial parameter values to fit your data more closely (HINT – use the visualisation of the initial parameter values to help). If not this could suggest that this is not an appropriate model for your data and you might need to use something else, e.g. is it more bimodal?

PDF files



Normal fitting

Lognormal fitting

Black = coulter counter data, blue = overall fitting. Also includes the uncertainty and standard error of regression measurements.

Comparison of normal (green), lognormal (blue) fittings with the raw data (black) and the standard error of regression for each model

How do the scripts work?

Please see the detailed overview in the ‘How to use the python script to fit bimodal traces to coulter counter traces’ guide

The scripts here work in more or less the same way (e.g. curves are defined using the same equations, means are calculated the same way…) the only exception is that instead of fitting two curves per trace we only fit one

Overview

1) You submit it using the Running script

You only need to do this first step, everything else is automated

2) The Running script starts the Batch_running_of_coulter_counter_script_one_curve_only

3) The Batch_running_of_coulter_counter_script_one_curve_only takes every csv file in the data_inputs folder and submits it to the coulter_counter_fitting_script_one_curve_only (one at a time)

4) The coulter_counter_fitting_script_one_curve_only performs the fitting