Shell commands that need to be run are in black monospaced font
Shell commands that are just examples or outputs are in green monospaced font
Directory names are in italics

1) Set up the project directory :
It needs motion corrected micrographs in their own folder, I used *micrographs/* here. I called my test images img_01.mrc and img_02.mrc but the actual ones will have probably have EPU style file names like FoilHole_28152728_Data_28129874_28129876_20200302_090000-253445.mrc

2) Prepare the ctf information
Because the scripts are written for Relion 2, it will need to be used for the initial steps: Start it with

```
/fbs/emsoftware2/LINUX/fbsmi/relion2-stable/bin/relion
```

Import the micrographs as "2D micrographs/Tomograms *.mrc" using Import, and then Run a CtfFind job on them. This has to be done in Relion2.

2) Get the scripts

```
git clone https://github.com/attamatti/fibril_segment_analysis.git
```

All script files are now in *fibril_segment_analysis/*

3) Initial picking
I used gpu01 only – it didn't work on workstations

```
module load eman2/2.12
```

(use the older EMAN it works better)

```
e2boxer.py micrographs/* boxsize=10
```

Pick splines along the fibril, using as many points as you want, if there are multiple fibrils pick a point off the image to start a new fibril, but don't do this after the last fibril.

Put the boxfiles from EMAN in a new dir called *FibCoords/*

4) Made the bfil param files - have to run it once for each file
The make bfil_parfile script needs to be run on each file individually with this command:

```
fibril_segment_analysis/make-bfil-parfile.py FibCoords/img_01.box micrographs/img_01.mrc
```

The easiest way to do this is with the following unix shell loop

```
for f in FibCoords/*.box; do f2=${f//.box/.mrc};
f3=${f2//FibCoords/micrographs}; fibril_segment_analysis/make-bfil-
parfile.py $f $f3; done
```

this will dump a ton of .star files into the working dir. Put them in a new directory *parfiles/*

5) Used Bfil to extract straightened fibrils

```
module load bsoft

mkdir Straight_fibrils
```

the bfil command looks like this:

```
bfil -extract 400 -split -base img_02_fil -extension mrc -path
Straight_fibrils/ parfiles/img_02.star
```

The only value that needs to be set is the extract size (-extract 400 here) it needs to be big enough that it is larger than the biggest crossover length you are expecting.

Again, this command needs to be run individually on all the files so do this with a shell loop:

```
for f in parfiles/*; do f2=${f##*/}; f3=${f2//.star/_fil}; bfil -extract
400 -split -base $f3 -extension mrc -path Straight_fibrils/ $f; done
```

Straightened fibrils will be written in *Straight_fibrils/*

6) Pick the crossovers

```
e2boxer.py Straight_fibrils/* --boxsize=10
```

For each fibril put a box on the end (make sure it's on the image though), one at the centre of each crossover in order, and then one at the other end.
Save the box files and put them in a directory called a dir called *Xover_coords/*

7) Run the classification script

```
fibril_segment_analysis/segment_classification.py
```

It will ask for some inputs. The answers are in red

Output will look like this:

```
*** 1-D Segment Classification v1.2.1 ***

files search string: Xover_coords/*.box
write script for extraction for RELION? (y/n) y

** fibrils stats ***************************************
count    min  max  mean std
```

```
3    235.02    829.10    466.73    259.53

** segments stats ************************************
count     min  max  mean std
7    168.03    235.02    200.03    23.44

** Finding the right number of classes
*************************************
optimum number of breaks by goodness of varience fit 1st past 0.8
# classes goodness of varience fit
3         0.310601579086
4         0.74495872112
5         0.816339524037
using 5 classes
class breaks: [168.02678357928536, 168.04761230080004, 198.00252523642217,
206.0024271701671, 235.01914815605983]

** Fibrils evaluation *************************************
fibril        # segments    meanclassdist curvaturescore
Xover_coords/img_02_fil002-0001  2         0.000         0.07
Xover_coords/img_01_fil001-0001  1         XX        XX
Xover_coords/img_02_fil001-0001  4         1.667         0.09

** Segment classification *************************************
5 classes
class     count     mean std  range     boxsize
0    2    168.04    0.01 0.02 220
1    1    198.00    0.00 0.00 260
2    1    202.01    0.00 0.00 260
3    1    206.00    0.00 0.00 270
4    2    229.05    5.97 11.94     310

wrote seg-analysis-relion.sh

6 boxfiles written for RELION/EMAN
```

9) Fix the location of the box files

The script wrote the boxfiles in a directory called boxfiles, but the directory structure inside that needs to be the same as the straightened fibrils images so:

```
mkdir boxfiles/Straight_fibrils

mv boxfiles/*.box boxfiles/Straight_fibrils/
```

8) Attach ctf info to straightened fibrils

make a text file containing all the straightened fibils

```
ls Straight_fibrils/* > straight_fibs.txt
```

then use the ctf matching script

```
fibril_segment_analysis/rln_match_ctf_to_straigntened.py
CtfFind/job002/micrographs_ctf.star straight_fibs.txt
```

9) Extract the segments
This can be done in Relion3.1 which then allows all subsequent steps to be done in Relion3.1

```
Module load relion

sh seg-analysis-relion.sh fibrils_ctf.star
```

10) Do the classifications
The extraction script created a dir called *Particles/*. In it is a sub dir for each class, in each of those is a star file (IE class001.star) and a *Straight_fibrils/* dir with the actual particle images mrcs file.

Now run Relion3 and do a separate 2D classification job for each Particles/classxxx/clasxxx.star file.