Normalising the CaptureC data without a pipeline !
Instructions by Jelena - 291015

 # Make a directory

```
mkdir normalisation
```

 # Go to the directory

```
cd normalisation
```

 # Copy all your to-be-normalised files to that folder
Let's assume you have 2 tissue types "Pri" (primary cells) and "E" (erythroid cells).
You have 1 replicate (Pri1) for Pri, and 2 replicates (E1,E2) for E cells.
You are interested in two of your oligos - HbaCombined and Hbax

These are the files you want :

Pri1_CC3/Combined_reads_REdig_CC2_HbaCombined.gff
Pri1_CC3/Combined_reads_REdig_CC2_Hbax.gff
E1_CC3/Combined_reads_REdig_CC2_HbaCombined.gff
E1_CC3/Combined_reads_REdig_CC2_Hbax.gff
E2_CC3/Combined_reads_REdig_CC2_HbaCombined.gff
E2_CC3/Combined_reads_REdig_CC2_Hbax.gff

 # These are the copy commands you want to do - note how the ES1 and ES2 file names are changed to avoid OVERWRITING files (we are copying files with same names!)

```
cp /hts/data1/user/Pri1_CC3/Combined_reads_REdig_CC2_HbaCombined.gff Pri1_HbaCombined.gff
cp /hts/data1/user/Pri1_CC3/Combined_reads_REdig_CC2_Hbax.gff Pri1_Hbax.gff
cp /hts/data1/user/E1_CC3/Combined_reads_REdig_CC2_HbaCombined.gff E1_HbaCombined.gff
cp /hts/data1/user/E1_CC3/Combined_reads_REdig_CC2_Hbax.gff E1_Hbax.gff
cp /hts/data1/user/E2_CC3/Combined_reads_REdig_CC2_HbaCombined.gff E2_HbaCombined.gff
cp /hts/data1/user/E2_CC3/Combined_reads_REdig_CC2_Hbax.gff E2_Hbax.gff
```

 # Now we can list what we have : ls

We see we have the 6 gff files :

```
Pri1_HbaCombined.gff
Pri1_Hbax.gff
E1_HbaCombined.gff
E1_Hbax.gff
E2_HbaCombined.gff
E2_Hbax.gff
```

 # Now we will run the first parts of the normalisation with a script I wrote :

```
/t1-data/data/hugheslab/jelenatools/CC/norm/VS100/run.sh &> normalisation.log
```

The above command normalises the data for 100 000 reads, using James' codes.

Now you should have file

```
ls | grep gfc
```

normalised.gfc.out

Check that you got no error messages in

```
cat normalisation.log
```

The subsequent steps depend on the ordering of columns in the file - so we prepare our data for those steps manually !

 # Generating the mean for replicates !

We have 7 columns in the file normalised.gfc.out

chr:start-stop NAME1 NAME2 NAME3 NAME4 NAME5 NAME6

We want to take the "correct columns" out - the ones which are the replicates of each others !
We use cut command to take out the columns we are interested in.

If our file heading looks like this :

chr:start-stop Prim_Hbx Prim_Hba ES1_Hbx ES1_Hba ES2_Hbx ES2_Hba

We can write the example commands below :

Example command for taking out ES1 and ES2 Hba columns :
cut -f 1,2,4 normalised.gfc.out > ES1_ES2_Hba_forMean.in
The above takes out the chr:start-stop and ES1_Hba and ES2_Hba columns.

Example command for taking out ES1 and ES2 Hbx columns :
cut -f 1,3,5 normalised.gfc.out > ES1_ES2_Hbx_forMean.in

Check your column numbers - they are not necessarily the same as above :

Check with head commands that you got the right columns :
head *forMean*

The file names have to end _forMean.in - that is how the subsequent script knows which files to average !

Now we will run the averaging for these files :

```
/t1-data/data/hugheslab/jelenatools/CC/norm/VS100/mean.sh &> mean.log
```

Now you should have files which end _mean.gfc , for example :

```
ls | grep gfc
```

ES12_Hbx_mean.gfc
ES12_Hba_mean.gfc

Check that you got no error messages in `cat mean.log`

 # Now we can make the substraction of the tracks !

We already have correct file format for the ES

Now we need to take the columns out for the Prim - just like we did for the ES above.
Now we don't have replicates, so taking out 2 columns is enough :

If our file heading looks like this :

chr:start-stop Prim_Hbx Prim_Hba ES1_Hbx ES1_Hba ES2_Hbx ES2_Hba

We can write the example commands below :

Example command for taking out Prim Hba :
```
cut -f 1,6 normalised.gfc.out > Prim_Hba.gfc
```
The above takes out the chr:start-stop and Prim_Hba columns.

Example command for taking out Prim Hbx :
```
cut -f 1,7 normalised.gfc.out > Prim_Hbx.gfc
```

Check your column numbers - they are not necessarily the same as above !

Check with head commands that you got the right columns :
```
head Prim*gfc
```

Now you should have files :

```
ls | grep gfc
```

ES1_ES2_Hbx_mean.gfc
ES1_ES2_Hba_mean.gfc
Prim_Hbx.gfc
Prim_Hba.gfc

Now we need to put the "pairs" of SAMPLE-INPUT to a same file like this :

```
paste Prim_Hbx.gfc ES1_ES2_Hbx_mean.gfc > Hbx_Prim_ES12_forSubtract.in
paste Prim_Hba.gfc ES1_ES2_Hba_mean.gfc > Hba_Prim_ES12_forSubtract.in
```
The file name has to end _forSubtract.in - that is how the script below recognizes the files it should do subtraction to.

 # We will now do the substraction like this :

```
/t1-data/data/hugheslab/jelenatools/CC/norm/VS100/subtract.sh &> subtract.log
```

Check that you got no error messages in

```
cat subtract.log
```

Now we should have files :

```
ls | grep gfc
```

Hbx_subtracted.gfc
Hba_subtracted.gfc

Prim_Hba.gfc
Prim_Hbx.gfc

ES1_ES2_Hbx_mean.gfc
ES1_ES2_Hba_mean.gfc

 # The last step is to make the visualisation files for all our data !

We should have files :

```
ls | grep gfc
```

Hbx_subtracted.gfc
Hba_subtracted.gfc

Pri_Hba.gfc
Pri_Hbx.gfc

ES12_Hbx_mean.gfc
ES12_Hba_mean.gfc

We need to generate data hub for these tracks - to see them in USCS !

```
/t1-data/data/hugheslab/jelenatools/CC/norm/VS100/hub.sh --genome mm9 --hubname testhub050815
--pf /public/telenius/testhub050815 &> hubbing.log
```

Check that you didn't get any error messages :

```
cat hubbing.log
```

The hub address is given in the end of the log file :

```
tail hubbing.log
```