User Manual for Capture C analysis

James Davies 14th May 2015 Contact: james.davies@trinity.ox.ac.uk
Jelena telenius 29th Oct 2015 jelena.telenius@gmail.com

A) Copy the run scripts to your analysis folder
B) Load the pre-requisites (modules) the code needs to run
C) Copy your MiSeq or HiSeq data to your analysis folder
D) Run the scripts one at a time - to prepare your data for captureC analysis

Overview of scripts

1. dpngenome.pl makes a file of all of the dpnII coordinates in the genome from a multiple FASTA file of the genome (the same format as used by bowtie). This script outputs the files to the same path as the input file.
2. dpnII2E.pl performs an in silico dpnII digestion of the fastq files. This script outputs the files to the same path as the input file.
3. CCanalyser2.pl analyses the sam file outputted by bowtie. It also needs the file of the dpnII coordinates in the genome (made by dpngenome.pl) and a file of the list of oligonucleotides. It outputs wig, windowed wig, gff and sam files of the data from each of the tracks (relating to each capture point) into a subdirectory of the script itself. It can also copy the files into your public folder and generate a track hub of the data so that you only need to paste a single url into UCSC to see all of the tracks.

A) Copy the abovementioned scripts to your folder :

Set of "for-sure-working" scripts – straight from Jelena :
(these are a little bit behind in development – but well tested, i.e. will work for you for sure) !

```
cp /t1-data/data/hugheslab/jelenatools/CC/CC2/RunScripts/dpnII2E.pl .
cp /t1-data/data/hugheslab/jelenatools/CC/CC2/RunScripts/dpngenome3_1.pl .
cp /t1-data/data/hugheslab/jelenatools/CC/CC2/RunScripts/CCanalyser2.pl .
```

Improved versions of the above codes is available below - it contains output report improvements and more reliable behaviour with input errors (wrong format oligo coordinate file etc). However, this set is BETA release, so report bugs to jelena.telenius@gmail.com :

```
cp /t1-data/data/hugheslab/jelenatools/CC/CC3/RunScripts/dpnII2E.pl .
cp /t1-data/data/hugheslab/jelenatools/CC/CC3/RunScripts/dpngenome3_1.pl .
cp /t1-data/data/hugheslab/jelenatools/CC/CC3/RunScripts/CCanalyser3.pl .
```

Change the permissions – to be able to run them :

```
chmod u=rwx *.pl
```

B) Load the pre-requisites (modules) the code needs to run

Load these modules before running this analysis pipeline :

```
module load trim_galore/0.3.1
module load bowtie/1.0.0
module load ucsctools/1.0
module load flash/1.2.8
```

The pipeline is tested for perl v5.18.1, and it requires perl modules Cwd Data::Dumper Getopt::Long

C) Copy your MiSeq or HiSeq data to your analysis folder

Load your data from miseq folder

1. Miseq data for Read1 and Read2 of your sample is located in `/hts/data6/miseq/username/date/samplename_R1_001.fastq.gz` and `/hts/data6/miseq/username/date/samplename_R2_001.fastq.gz` ,where username is your username and date is f.ex 2015-05-15 for 15th May 2015 (that is the starting date of the run – not the end date), and the samplename is the name you gave your sample whe

2. Check that the miseq run has finished by doing
   ```
   ls -lht /hts/data6/miseq/username/date
   ```
   If the output of the command shows you the `fastq.gz` files, and you see that the time point when they were last modified (the time it gives you in the listing) is not "now" but, say 1 hour in the past – the files are ready (they are not being modified any more), and you can start copying them.

3. Copy the files to your folder like this :
   ```
   nohup cp /hts/data6/miseq/username/date/samplename* .  &
   ```
   You can usually ignore the `Undetermined*.fastq.gz` files which contain the small amount of reads the sequencer had problems in recognizing adapters in.
4. Unpack your files in the folder like this :
   ```
   nohup gzip -d samplename_R1_001.fastq.gz
   ```
   , and after that has been finished, for Read2 :
   ```
   nohup gzip -d samplename_R2_001.fastq.gz
   ```

5. In the instructions below, we call these unpacked files (which now have the same file name as before, without the ending –gz) as `Paired_end_R1.fastq` and `Paired_end_R2.fastq`

D) Run the scripts one at a time

Note ! – if you need to log out and log back in when in the middle of the analysis – remember to re-load the needed modules (step B above) – the modules are "wiped out" when you log out !

6. Perform adaptor trimming of the raw fastq files if this hasn't been performed by the sequencer. I tend to use trim_galore to do this with the paired settings. For example:
   ```
   nohup trim_galore --paired Paired_end_R1.fastq Paired_end_R2.fastq&
   ```

7. If you are using 150 bp paired end reads from the Miseq then the overlapping reads need to be merged. I use FLASH to do this. For example:
   ```
   nohup flash --interleaved-output Paired_end_R1_val_1.fq Paired_end_R2_val_2.fq &
   ```
   Then concatenate the separate files of merged and unmerged reads:
   ```
   cat out.notCombined.fastq out.extendedFrags.fastq > Combined_reads.fastq
   ```

8. If you sequence the data with reads that are shorter than the fragments then you see a lot of reads in the `out.notCombined.fastq` file. It is ok – what you did with flash in that case, was not to "merge the reads" but rather to "sort them to interleaved order" which is needed for the rest of the pipeline.

9. In silico digest the generated `Combined_reads.fastq` file with above-mentioned `dpnII2E.pl` script., like this :
```
    nohup ./dpnII2E.pl Combined_reads.fastq &
```

10. It is crucial for the subsequent `CapCanalyser_5.pl` script that the files are in strict order with the reads of each name being adjacent.

11. To achieve this I align the fastq files with bowtie using one processor only. You can use more than one processor but then the files will need to be sorted. I also tend to use m2, best and strata – but this is partly because m2 is needed otherwise all of the reads for alpha globin will be thrown (because of the gene duplication) so you may well want to use your own setting. For example:

```
    nohup bowtie -p 1 -m 2 --best --strata --sam --chunkmb 256
      /databank/igenomes/Mus_musculus/UCSC/mm9/Sequence/BowtieIndex/genome
      Sample_REdig.fastq Sample_REdig.sam &
```

    where the `Sample_REdig.fastq` is your output from step 9, and `Sample_REdig.sam` is the output of this command – give any name of your liking, ending with .sam

12. You need a file containing whole-genome DpnII digest.

    For hg18,hg19 or mm9, you can use one of these :
```
/hts/data2/jdavies/07mm9_dpn/hg18_dpnII_coordinates.txt
/hts/data2/jdavies/07mm9_dpn/hg19_dpnII_coordinates.txt
/hts/data2/jdavies/07mm9_dpn/mm9_dpnII_coordinates.txt
```

    If you mapped to some other genome, you need to run `dpngenome3_1.pl` script to generate the file :

```
nohup ./dpngenome3.pl PathToGenomeFasta &
```

    here the path to genome fasta can be found in our system in the "igenomes" database – For example genome mm9 :
```
/databank/igenomes/Mus_musculus/UCSC/mm9/Sequence/WholeGenomeFasta/genome.fa
```

13. Generate oligonucleotide-coordinate file (or rather, a file of DpnII fragments which contain the coordinates of your biotinylated capture-oligos). Table below gives you instructions, and a step-by-step oligo-file guide is available in the same web site you downloaded this manual :
https://sites.google.com/site/gbgintranet/documents/capture-c-analysis-pipeline-vs-1
If you are a collaborator and cannot access that site – send email to jelena.telenius@imm.ox.ac.uk to receive a copy !

Oligonucleotide file generation – quick introduction :

The file is a tab-delimited file with these columns :

Name	Chr	Start	Stop	Chr	Start	Stop	SNP	SNP base

Where the first (leftmost) chr-start-stop denote to Dpn fragment coordinates, and second chr-start-stop denote exclusion coordinates (1000bases to both directions from the DpnII fragment).

For example (one capture fragment only) :

```
Hba-1  11    32182969     32183821     11    32181969     32184821     1     A
```

Note the exclusion bases added to columns 6 and 7.

Notice last columns "1" and "A" – this is kind of "default input" to those columns – they have to contain something (even if you wish no-snp specific run.)

To use above line for a SNP-specific run, you could, for example :

```
Hba-1  11    32182969     32183821     11    32181969     32184821     32182980   T
```

If you had a snip to T at location 32182980, that is.

Your capture oligo file will thus look something like this :

```
C1  11    32182969     32183821     11    32181969     32184821     1     A
C2  1      45182345     45182345     1      45182345     45182345     1     A
C3  15    32342342     32342342     15    32342342     32342342     1     A
```

Where the C1 C2 C3 are the names you gave to your capture sites, and columns 2-4 are the DpnII fragments within which the corresponding biotinylated oligos reside, and columns 5-7 are the exclusion coordinates (+/- 1000 bases both directions).

Please avoid using whitespace in the "name" column (i.e. no names containing spaces) ! Also all other "weird characters" are forbidden (%$&* etc). Only _ (underscore) is allowed.

14. Run the capture-C analyser script `CCanalyser2.pl`

Running CCanalyser2.pl

The help an manual of the CCanalyser script are available with the commands :

```
perl CCanalyser2.pl --man
perl CCanalyser2.pl --help
perl CCanalyser2.pl -h
```

A) Input files for the script
B) Parameters you need to give to the script
C) Run command examples
D) Inspecting the output – did your run finish properly ?
E) Finetuning your run – optional parameters

A) Input files for the script :

1. File of capture oligonucleotide coordinates (step 13 above). Please note that the coordinates need to include the restriction enzyme sequence at both ends of the fragments otherwise a large proportion of your useful reads will be excluded. This is best checked by looking at the sam file output containing the reads that are being classified as capture fragments.
   Please avoid using whitespace in the name of the captures.

2. File of the restriction enzyme fragments in the genome (step 12 above)
3. Bowtie-mapped reads from the captureC experiment (step 11 above)

B) Parameters you have to give to the script :

1. Sample name :
   -s CapTest or **–s HgA** or –**s MySample** ,whatever pleases you. Don't use weird characters like @*~{ or whitespace in the name – only underscore _ and azAZ09 are allowed, and don't start with a number.

2. File of capture oligonucleotide coordinates (step 13 above).
   -o oligo_coordinates_file.txt
   You can give this as a full path /hts/dataX/user/path/to/oligo_coordinates_file.txt
   , if you don't have the file in your running folder.

3. File of the restriction enzyme fragments in the genome (step 12 above)
   -r genome_dpnII_coordinates.txt
   You can give this as a full path
   /hts/dataX/user/path/to/genome_dpnII_coordinates.txt
   , if you don't have the file in your running folder.

4. Bowtie-mapped reads from the captureC experiment (step 11 above)
   -f Sample_REdig.sam
   You can give this as a full path /hts/dataX/user/path/to/Sample_REdig.sam
   , if you don't have the file in your running folder.
5. Folder to put your data for visualisation : generate this file before running the script :

   ```
   mkdir /public/username/MyCaptureExperiment
   ```

   For example : `mkdir /public/telenius/CAPTURE1_test_250515`

   You will give the location of this folder in two different parameters :
   --pf /public/username/MyCaptureExperiment
   --pu userweb.molbiol.ox.ac.uk/public/username/MyCaptureExperiment

   Here userweb.molbiol.ox.ac.uk is the server address of our WIMM public server.
   You can also use sara.molbiol.ox.ac.uk (they are two different names for the same server)

C) Your run command will look something like this :

```
nohup perl CCanalyser2.pl -f Sample_REdig.sam -r genome_dpnII_coordinates.txt
--genome mm9 -o oligo_coordinates_file.txt -s MyTestCaptureRun
--pf /public/username/MyCaptureExperiment
--pu userweb.molbiol.ox.ac.uk/public/username/MyCaptureExperiment &
```

When running : remember the double hyphens -- for long options
(more than 1 letter long) ! – writing **--pu** or **-pu** is completely different
(to see why this is so important : http://perldoc.perl.org/Getopt/Long.html)
In short : -pu means the same as -p -u and not at all "--pu" – you risk OVERWRITING --pf and --pu with each others' values if you try to set them with -pu and -pf instead of --pf and --pu !!

If the above format run command does not work – just give full paths to all input files (to be on the safe side) :

```
nohup perl CCanalyser2.pl -f /hts/dataX/user/path/to/Sample_REdig.sam -r
/hts/dataX/user/path/to/genome_dpnII_coordinates.txt -genome mm9 -o
/hts/dataX/user/path/to/oligo_coordinates_file.txt -s MyCaptureRun
--pf /public/username/MyCaptureExperiment
--pu userweb.molbiol.ox.ac.uk/public/username/MyCaptureExperiment &
```

D) Did your run finish without complications ?

Check your output files :

The script will generate to you a folder named with the name you gave in parameter –s (sample name). Go to that folder `cd sampleName` – You should see a lot of wig and gff and sam files (as many as you had capture fragments in your oligo input file). Check their sizes : `ls -lh`
If you see files with non-zero size for most of the capture sites – you apparently have data for the run – the run succeeded !

The run statistics will be written into a file in this folder – you will find file called (something)_captures_report_CC2.txt

Have a look at the file, to find out which kind of capture fragments your data contained !

Now – you want to have a look at the data in the UCSC web browser !
A data hub has already been generated for you – it will be located in the folder you provided in you parameters – in folder `/public/username/MyCaptureExperiment`
You can check that you actually have files in there – if you see no bigwig files (.bw) when you `ls -lh /public/username/MyCaptureExperiment` , then you probably forgot to `module load ucsctools/1.0` before you ran the script – the bigwig files weren't generated as the command was not found.

Go out of the "sampleName" directory – to the folder you ran CCanalyser2.pl in .
You can now see the custom track load lines in the end of your nohup.out file :

```
tail -n 30 nohup.out
```

You search for lines like these :

```
track type=bigWig name="testfile_chr7captures_CC2_A3" description="CaptureC_gene_A3"
bigDataUrl=http:///public/telenius/CAPTUREC_DATA/TESTFILEtest1/testfile_chr7captures_
CC2_A3.bw
```

To load these to UCSC – you log in in our UCSC mirror : https://genome.molbiol.ox.ac.uk/
go to the genome you have your data in (say, mm9) , and select "My Data" → "Custom Tracks" , and paste the track description lines (above) to the field, and press "submit".

If you want to generate a data hub of these tracks instead (to not to need to load all of them into UCSC separately) – have a look at this data hub generation tutorial :
http://sara.molbiol.ox.ac.uk/public/telenius/DataHubs/ReadMe/HowToCreateA_DataHUB_070613.pdf

E) Optional parameters to finetune your run !

The help an manual of the CCanalyser script are available with the commands :

```
perl CCanalyser2.pl --man
perl CCanalyser2.pl --help
perl CCanalyser2.pl -h
```

Complete list of options for the script :

When running : remember the double hyphens -- for long options
(to see why this is so important : http://perldoc.perl.org/Getopt/Long.html)

-f	Input filename (this is the sam file you get as output in bowtie)
-r	Restriction coordinates filename (output of dpngenome3.pl script)
-o	Oligonucleotide position filename (generated along the table above)
--pf	Your public folder (something like /public/username/folder or so)
--pu	Your public url (something like userweb.molbiol.ox.ac.uk/public/username/folder – compare to –pf parameter)
-s	Sample name (and the name of the folder it goes into – no whitespace or weird characters like * } = or so , underscore _ is allowed)
-w	Window size (default = 2kb)
-i	Window increment (default = 200bp)
--dump	Print file of unaligned reads (sam format)
--snp	Force all capture points to contain a particular SNP
--limit	Limit the analysis to the first n reads of the file
--genome	Specify the genome (for example mm9 or hg18)
--globin	Combines the two captures from the gene duplicates (HbA1 and HbA2)

Using the --globin parameter :

The logic is a bit un-conventional

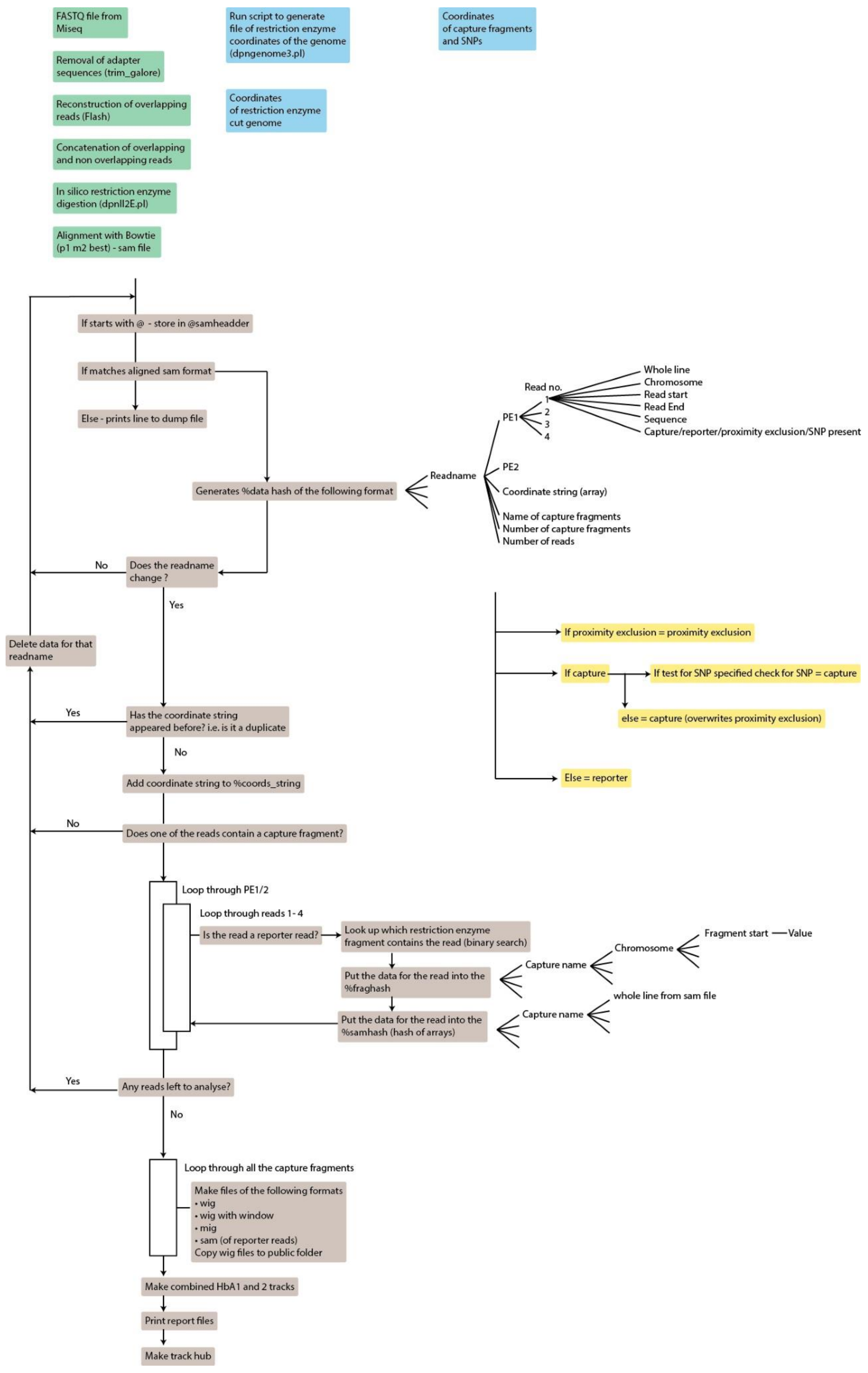
So it goes like this :

--globin 1
combine for alpha globin

--globin 2
combine for both alpha and beta.

You have to call the oligos in the oligo file exactly like this to make the --globin to work :

Hba-1
Hba-2
Hbb-b1
Hbb-b2


FASTQ file from Miseq
Removal of adapter sequences (trim_galore)
Reconstruction of overlapping reads (Flash)
Concatenation of overlapping and non overlapping reads
In silico restriction enzyme digestion (dpnII2E.pl)
Alignment with Bowtie (p1 m2 best) - sam file
Run script to generate file of restriction enzyme coordinates of the genome (dpngenome3.pl)
Coordinates of restriction enzyme cut genome
Coordinates of capture fragments and SNPs
If starts with @ - store in @samheadder
If matches aligned sam format
Else - prints line to dump file
Generates %data hash of the following format
Readname
PE1
Read no.
1
2
3
4
Whole line
Chromosome
Read start
Read End
Sequence
Capture/reporter/proximity exclusion/SNP present
PE2
Coordinate string (array)
Name of capture fragments
Number of capture fragments
Number of reads
Does the readname change ?
No
Yes
Delete data for that readname
Has the coordinate string appeared before? i.e. is it a duplicate
Yes
No
Add coordinate string to %coords_string
Does one of the reads contain a capture fragment?
No
If proximity exclusion = proximity exclusion
If capture
If test for SNP specified check for SNP = capture
else = capture (overwrites proximity exclusion)
Else = reporter
Loop through PE1/2
Loop through reads 1 - 4
Is the read a reporter read?
Look up which restriction enzyme fragment contains the read (binary search)
Put the data for the read into the %fraghash
Capture name
Chromosome
Fragment start
Value
Put the data for the read into the %samhash (hash of arrays)
Capture name
whole line from sam file
Any reads left to analyse?
Yes
No
Loop through all the capture fragments
Make files of the following formats
• wig
• wig with window
• mig
• sam (of reporter reads)
Copy wig files to public folder
Make combined HbA1 and 2 tracks
Print report files
Make track hub