CCanalyser2

How to interpret the results,

And troubleshoot your run, if needed

Manual by Jelena 08/Sep/2015

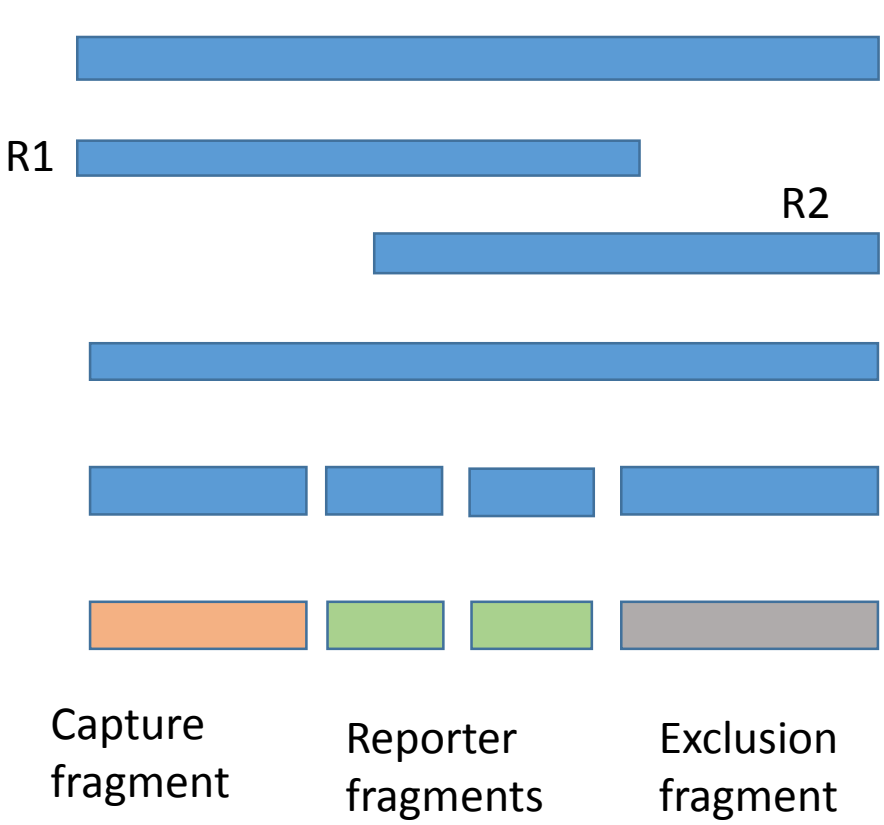
- 1) About “reads” and “fragments” – and possible wrongly given oligo coordinate text files

- 2) Output folder contents

- 3) Good and bad quality data – how does it look in UCSC browser ?

What is a “read” and what is a “fragment”

- In CaptureC analysis, we need to separate between various different fragment types.
- Here the nomenclature :



Sonication fragment. Captured and sent for sequencing.

Read1 and Read2 – paired end sequencing results from the fragment

Flash-merged “full read” for the sequence

In silico RE-digested read – showing different **captured fragments** within the read

In silico RE-digested read – showing different identities of the captured fragments

What is a “capture” and what is a “reporter” fragment

Analysis read (contains all these fragments)

In silico RE-digested read – showing different identities of the captured fragments

Capture fragment

Reporter fragments

Exclusion fragment

Capture fragment is the fragment, which contained the the biotinylated oligo.
It is marked as capture fragment, if it maps (in Bowtie) within the DpnII fragment where the capture oligo was located.

Reporter fragment is any other fragment than capture fragment or exclusion fragment. If there is a capture fragment within the same read, this reporter fragment is reported in the results.

Exclusion fragment is a captured fragment, which is not within the oligo-containing DpnII fragment, but is +/- 1000 bases from that fragment. These fragments are considered to be “too close to the capture site” to be reliable signal, and are excluded from the analysis.

Which reporter and capture fragments get reported ?

Capture fragment = orange

Reporter fragment = green

Exclusion Fragment = grey

In this case we report 2 reporter fragments

In this case we report 0 reporter fragments (no capture in the read)

In this case we report 0 reporter fragments (multi-capture, all fragments are discarded)

Remember also give NO EMPTY LINE in the end of oligo coordinate text file, otherwise the code will try to read the last line in as oligo having name (empty) and start site (empty) and end site (empty) and this make the perl script crumble.

Be aware of possible typos in oligo coordinate file !

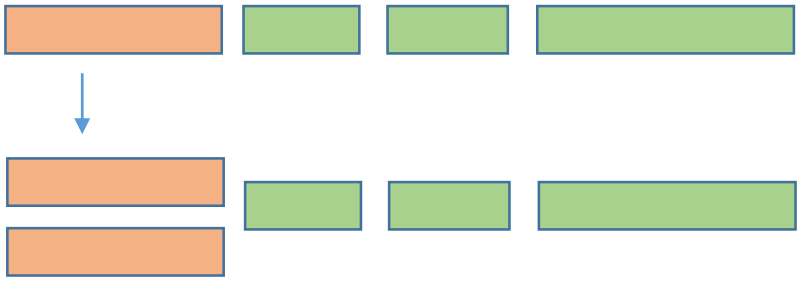
Capture fragment = orange

Reporter fragment = green

Exclusion Fragment = grey

In this case we report 3 reporter fragments

In this case we report 0 reporter fragments (no capture). This results with not carefully given oligo coordinates ! (the fragment has to be contained completely within the given DpnII oligo-coordinates to be considered capture !)



If you give same DpnII fragment many times under different names (many oligos to capture same fragment) – all these reads are discarded !

- The code sees them as “multi-captures” !

So, give each DpnII fragment only ONCE in the oligo coordinate file !

Run log file(s)

Error / output log

- If you ran automated pipeline run, you will have nohup.out or qsub.out and qsub.err files.

- These files contain all the output from the pipeline – how different tools processed the data, and if there were any errors in the run.

- CCanalyser2.pl script is not 100% stable yet, so it may claim “all went fine”, but you may get no output. In that case you need to have a look at the files the script generated and troubleshoot with that information (see “output folder contents” in next slides).

UCSC browser output

- The last lines of your nohup.out / qsub.out file (tail nohup.out or tail qsub.out) give you your data hub address.

- You can use the data hub address to load your data to UCSC browser. (If you don’t know how, see here : http://sara.molbiol.ox.ac.uk/public/telenius/DataHubs/ReadMe/HUBtutorial_AllGroups_160813.pdf)

Help me ! – when feeling puzzled..

- If you don’t know what went wrong – you can always send the data hub address (or the nohup.out or qsub.out and qsub.err files to Jelena (jelena.telenius@gmail.com) and she will have a look at your output log files to find out what happened !

Output folder contents

- If you ran automated pipeline run, you will have all these files and folders :

READ1_fastqc_ORIGINAL
READ1_fastqc_ORIGINAL.zip
READ2_fastqc_ORIGINAL
READ2_fastqc_ORIGINAL.zip

Quality control for the original input fastq files

read_trimming.log

Log file from read trimming

READ1_fastqc_TRIMMED
READ1_fastqc_TRIMMED.zip
READ2_fastqc_TRIMMED
READ2_fastqc_TRIMMED.zip

Quality control for the trimmed fastq files

Log files from "flashing" – i.e. combining overlapping ends of reads

flashing.log
out.hist
out.histogram

Output fastq file from "flashing"

Combined_reads.fastq
Combined_reads_fastqc
Combined_reads_fastqc.zip

Quality control for the "flashed" fastq file

In-silico restriction-enzyme digested fastq file

Combined_reads_REdig.fastq

Combined_reads_REdig.sam

Bowtie output - in-silico restriction-enzyme digested fastq file mapped to the genome

"The results" = output of CCanalyser2.pl script

Combined_reads_REdig_all_capture_reads_CC2.sam

dpnfragments.bed
dpnfragments.txt

test_3006_CC2

Output folder contents

"The results" = output of CCanalyser2.pl script

Combined_reads_REdig_all_capture_reads_CC2.sam

dpnfragments.bed
dpnfragments.txt

test_3006_CC2

Sam format output of all CAPTURE fragments

Bed file and original txt file of your given oligo coordinates

Within the output folder (here test_3006_CC2) you have a bunch of files for each capture site :

Sam files for all reported fragments. The other file has the reporter fragments, and the other the capture fragments.

Combined_reads_REdig_CC2_capture_A3.sam
Combined_reads_REdig_CC2_A3.sam

Combined reads REdig CC2 capture B3.sam
Combined_reads_REdig_CC2_B3.sam

Wig files for all reported fragments. These are also in bigwig format in the public folder – these are the UCSC visualisation tracks.

Combined_reads_REdig_CC2_A3.wig
Combined_reads_REdig_CC2_A3_win.wig

Combined_reads_REdig_CC2_B3.wig
Combined_reads_REdig_CC2_B3_win.wig

Reporter fragment counts per DpnII fragment, ready to be loaded to MIG (https://mig.molbiol.ox.ac.uk/mig/), or for differential analysis in DESeq.

Combined_reads_REdig_CC2_A3.gff

Combined_reads_REdig_CC2_B3.gff

Results log file – see next page !

Combined_reads_REdig_report_CC2.txt

Output folder contents

The results log file `Combined_reads_REdig_report_CC2.txt`

The results output file gives the statistics of each capture (also for the discarded ones : the multi-captures, and captures with no reporters).

To check how many of your reporter fragments actually got reported, check for the words :

Actual reported fragments (VS1.0 of scripts)

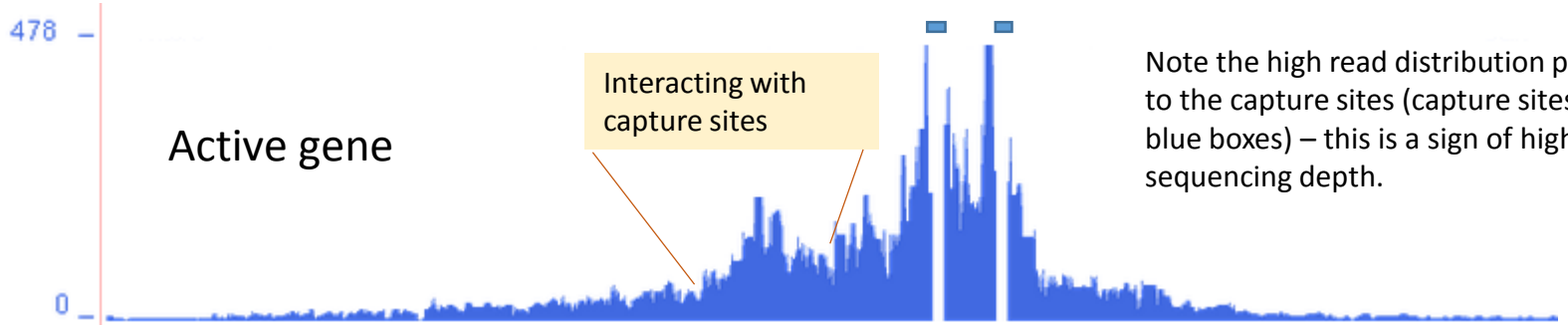
or

Final count Reads containing capture(s) in composition (VS 2.0 of scripts)

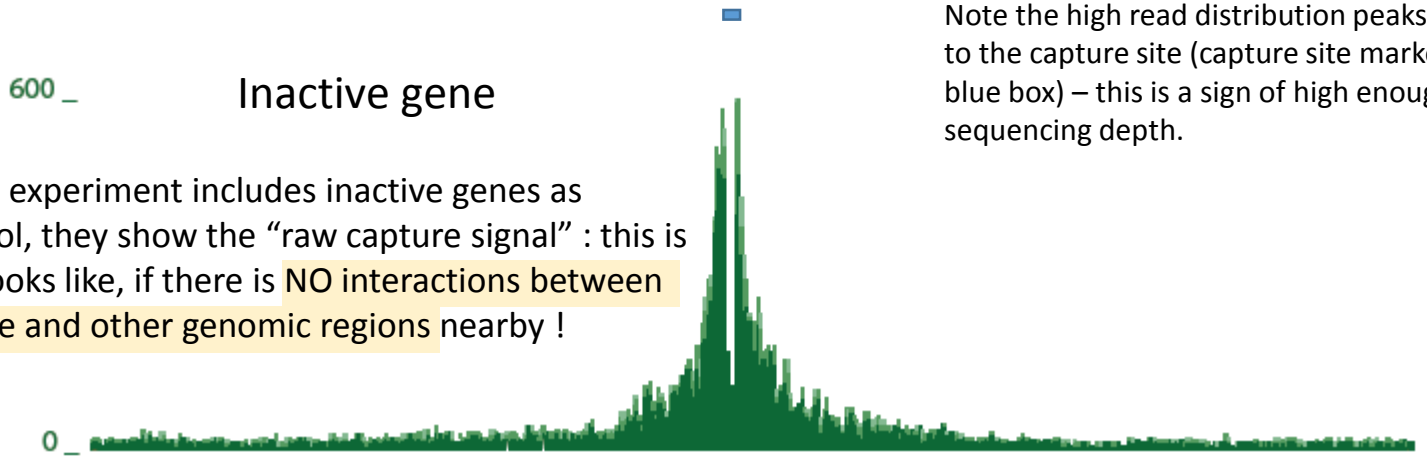
You should have ~10 000 reported fragments per capture site, to be able to do reliable analysis of the samples !

How does the data look in UCSC browser ?

Very good quality data – these track are high enough resolution for differential peak analysis



Note the high read distribution peaks right next to the capture sites (capture sites marked with blue boxes) – this is a sign of high enough sequencing depth.

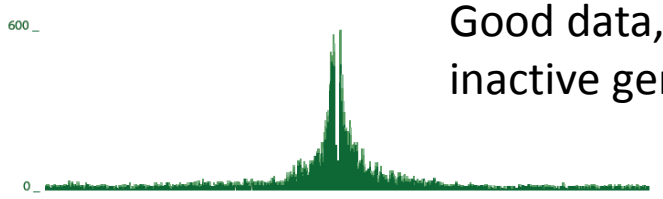


Note the high read distribution peaks right next to the capture site (capture site marked with a blue box) – this is a sign of high enough sequencing depth.

If your capture experiment includes inactive genes as negative control, they show the "raw capture signal" : this is how capture looks like, if there is NO interactions between the capture site and other genomic regions nearby !

Including this kind of negative control to your design, makes it easy to see if your sequencing depth was high enough – shape like this in a negative control is a sure sign of high quality data.

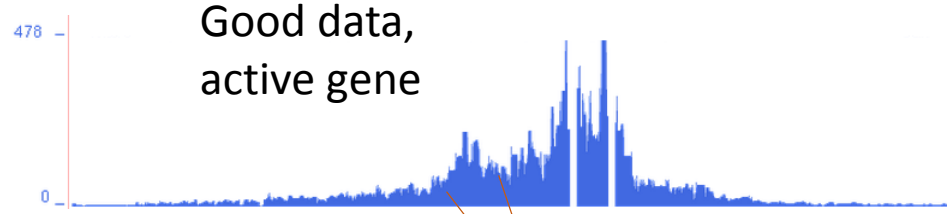
How does the data look in UCSC browser ?



Good data, inactive gene

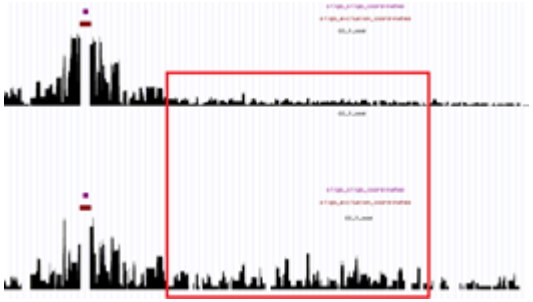
Below some data of mediocre and low quality samples

– replicates and/or deeper sequencing, and even possibly repeating the experiment is needed !



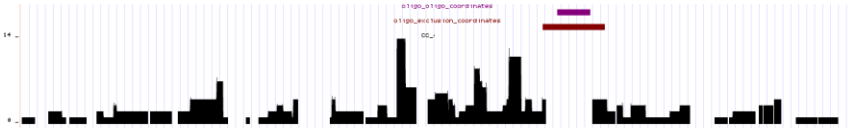
Good data, active gene

Interacting with capture sites



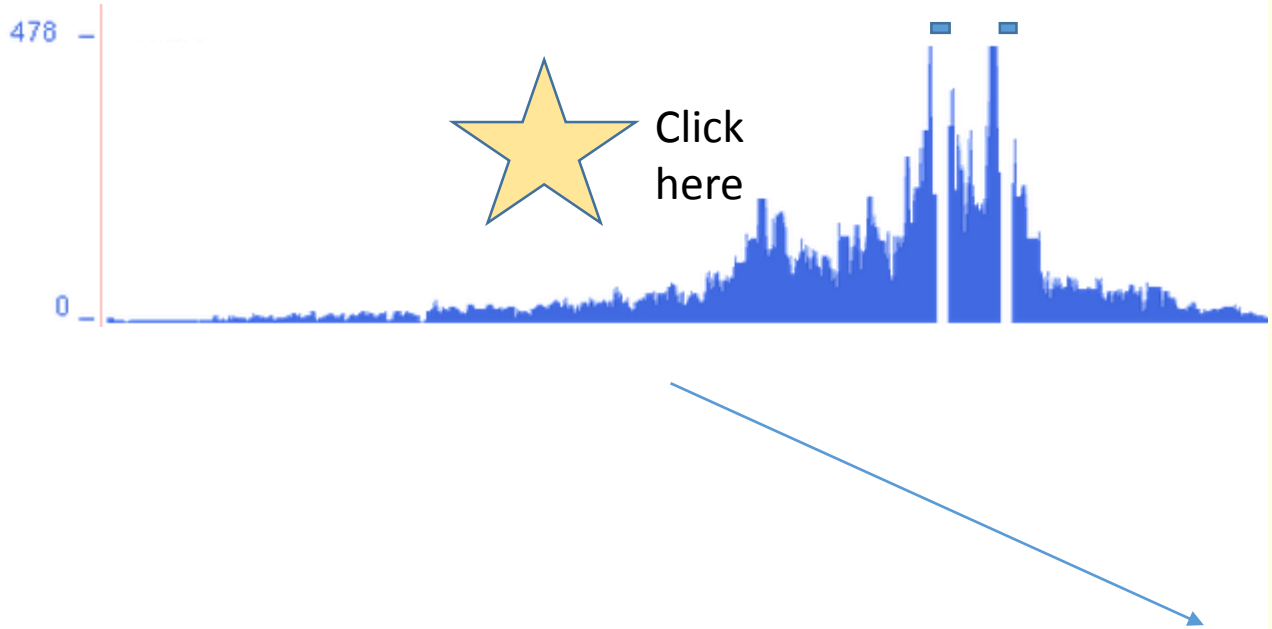
This one (on the left) is a somewhat low density sample – one can see differences between sample and control, but the “signature peaks” next to capture fragments are not so strong in the sample below – indicating a possibly little too low sequencing depth. This sample has potential, but cannot be analysed 100% reliably on this sequencing depth. If there are replicates available, they can also help in determining, which part of the signal is “real”.

This one (on the right) is very low sequencing depth sample –the “signature peaks” next to capture fragments are almost entirely absent. This may also be a “not-so-well” succeeded experiment, in which case the higher sequencing depth would not fix the issue. Usually signal higher than 10 reads in capture MEANS true signal, however, so the sample may just be recover-able, if sequenced on higher depth.



You can access all run log files, quality control reports etc, from your data hub !

Click on any track in the browser :



CC_test_2807d_H9 (hub_1242_test_2807d_H9)

Position: chr16:1-864,000
Total Bases in view: 864,000

No data overlapping current position.

Go to test_2807d_H9 track controls

Data last updated: 2015-08-03 16:51:55

Data produced Mon Aug 3 16:51:58 BST 2015 with Ca

Oligo coordinates given to the run :

Hba-1	chr7	73264944	73265558
Hba-2	chr7	73265770	73265993
Hbb-b1	chr7	73265990	73266610
Hbb-no	chr7	73269860	73270411
E3	chr7	73269860	73270411
G2	chr7	73223036	73224739
H2	chr7	73223036	73224739
Hbb-b2	chr10	81352436	81353064
H9	chr10	81354328	81355017
D10	chr10	81355445	81355701

Data located in : /hts/data6/telenius/runsAndAnalysis/c

Sample name : test_2807d, containing fastq files : /hts/
/hts/data6/telenius/runsAndAnalysis/captureDownload_

- Run log files available in : qsub.out , and qsub.err
- Capture script log file available in : Combined_reads_
- Capture script coordinate string : Combined_reads_F

FASTQC results here :

- FastQC results (untrimmed) : READ1_fastqc_ORIGI
- FastQC results (trimmed) : READ1_fastqc_TRIMME
- FastQC results (flashed, combined) : Combined_rea

Trimming/flashing log files here :

- Harsh trim_galore trim : read_trimming.log
- Flashing : flashing.log
- Histogram of flashed reads : flash.hist