Save Copy to Evernote

Last updated: Apr 23, 2020

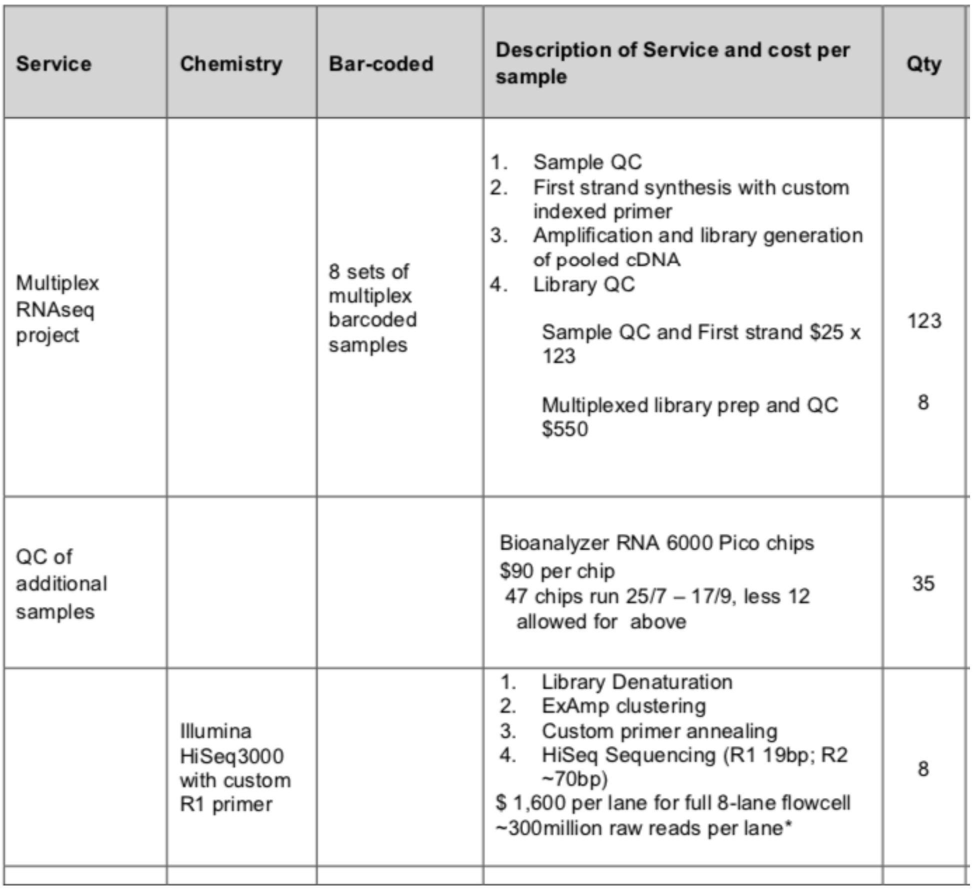
# Trev-seq data analysis pipeline (Original one from October-November):

- 000_Summary data type form Trevor Wilson at Hudson laboratory sin Melbourne
- 000_Project data sharing

- 000_Project data sharing
- 00_Demultiplexing (sabre)
- 01_Fastqc
- 02_cutadapt (we have contamination)
- 03_RNAsik
  - 03_STAR mapping mm10 genome and annotation
  - 04_Mapping analysis
- HTSEQ analysis on them….

## 000_Summary data type from Trevor Wilson at Hudson laboratories in Melbourne.

- HiSeq 3000
- Sample pooling 123 samples
- 8 Sets / Lanes of Multiplex RNA-seq project
- Up to 300M reads per lane
- “In-house developed multiplex method” by Trevor Wilson.
- R1:19 bp this is 20
- R2:76bp in reality, when checking this is 75



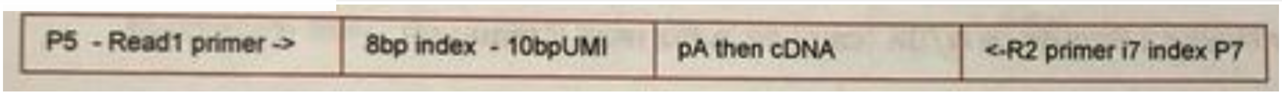
| Service | Chemistry | Bar-coded | Description of Service and cost per sample | Qty |
|---|---|---|---|---|
| Multiplex RNAseq project | | 8 sets of multiplex barcoded samples | 1. Sample QC<br>2. First strand synthesis with custom indexed primer<br>3. Amplification and library generation of pooled cDNA<br>4. Library QC<br>Sample QC and First strand $25 x 123<br>Multiplexed library prep and QC $550 | 123<br>8 |
| QC of additional samples | | | Bioanalyzer RNA 6000 Pico chips<br>$90 per chip<br>47 chips run 25/7 – 17/9, less 12 allowed for above | 35 |
| | Illumina HiSeq3000 with custom R1 primer | | 1. Library Denaturation<br>2. ExAmp clustering<br>3. Custom primer annealing<br>4. HiSeq Sequencing (R1 19bp; R2 ~70bp)<br>$ 1,600 per lane for full 8-lane flowcell<br>~300million raw reads per lane* | 8 |
| | | | | |

So already we know that if we put from 15-17 samples in a lane where we expel **~300 million raw reads per lane**

We expect between 15M-17M reads per sample! (For me it looks quite low, but talking with Christian he confirms that it is RNA poly A) and it is stem cells rather than tissue

So I guess it should be “neater”

Also we know that the data has the following format:



P5 - Read1 primer -> | 8bp index - 10bpUMI | pA then cDNA | <-R2 primer i7 index P7

So it contains an 8bp index + a 10 bp UMI!

Be careful with these new concepts as I know barcoded for pool-seq and demultiplexing but never dealt with UMIs before

- Unique molecular identifiers (**UMIs**), or molecular barcodes (MBC) are short sequences or molecular "tags" added to DNA fragments in some next generation sequencing library preparation protocols to identify the input DNA molecule. They can be used to reduce errors and quantitative bias introduced by amplification.
- A **UMI is** then randomly assigned to each molecule and the pool of molecules amplified by duplicating each molecules for every in silico PCR cycle. The final pool of molecules **are** then “sequenced” to produce the reads.

**###Number of counts of the original data:**

| Set or Lane | # Reads in R1 | # Samples within set | # Of expected number of reads (+ 10 % of missing not included) | R1 bp | R2 bp |
|---|---|---|---|---|---|
| Set1 | 294,775,915 | 17 | 17,339,759.7 | 20 | 77 |
| Set2 | 303,318,042 | 14 | 21,665,574.4 | 20 | 76 |
| Set3 | 305,492,703 | 16 | 19,093,293.9 | 20 | 77 |
| Set4 | 299,233,225 | 14 | 21,373,801.8 | 20 | 77 |
| Set5 | 311,752,725 | 17 | 18,338,395.6 | 20 | 76 |
| Set6 | 283,569,294 | 16 | 17,723,080.9 | 20 | 77 |
| Set7 | 239,235,722 | 16 | 14,952,232.6 | 20 | 77 |
| Set8 | 250,823,201 | 13 | 19,294,092.4 | 20 | 77 |

## 000_Project data sharing:

The data of this project has been backed up in

- **a) External hard-rive**
- **b) UQMDR:** QRISdata/**Q1287**/00_fastq_gz
- On Froday 1st of November we have a meeting about UQRDM!

## 00_Demultiplexing

“The samples are barcoded, and therefore, demultiplexing is required”

You need to know the exact number of samples that you are demultiplexing. If you have 15 samples, you need to have 15 cellular barcodes similar to those shown above. In Trev-seq cellular barcodes are 8 bases long and UMIs are typically 10 bases long. There is an addition one base in R1 (can't explain this particular well) therefore R1 read has to be 19 bases long. Have a look inside FASTQ file to see that this is the case. Your R2 can be anywhere upto 150 bases (standard illumina length), but I think Trevor normally does 75 cycles - length.

The tool Kirill mentioned is sabre:
https://github.com/najoshi/sabre
Sabre is a tool that will demultiplex barcoded reads into separate files. It will work on both single-end and paired-end data in fastq format. It simply compares the provided barcodes with each read and separates the read into its appropriate barcode file, after stripping the barcode from the read (and also stripping the quality values of the barcode bases). If a read does not have a recognized barcode, then it is put into the unknown file.
Sabre also has an option (-m) to allow mismatches of the barcodes.

Finally, after demultiplexing, sabre outputs a summary of how many records went into each barcode file.

Be aware that if you do not format the barcode data file correctly, sabre will not work properly.

The one he points out is: 1. demultiplex with sabre tool https://github.com/serine/sabre to get your 15 individual fastq files, all single ended

It should be sent in PBS job scheduling system used on Awoonga!

```
uqmnaval@awoonga1:/QRISdata/Q1287/00_fasq_gz> /home/uqmnaval/software/sabre/src/sabre
-f Set1_S1_L001_R1_001.fastq.gz  -r Set1_S1_L001_R2_001.fastq.gz  -b
Set1_barcodes.txt  -c -u -m2 -l 10 -a 1 -s Set1_sabre.txt

  Running: sabre
  Command line args:
      --fq1 Set1_S1_L001_R1_001.fastq.gz
      --fq2 Set1_S1_L001_R2_001.fastq.gz
      --barcodes Set1_barcodes.txt
      --unassigned_R1 unassigned_R1.fq
      --unassigned_R2 unassigned_R2.fq
      --combine 1
      --umi 1
      --max-mismatch 2
      --min-umi-len 10  ###minimum UMI length to keep
      --max-5prime-crop 1
      --no-comment -1
      --stats Set1_sabre.txt
  BC TAAGGCGA
  BC CGTACTAG
  BC AGGCAGAA
  BC TCCTGAGC
  BC GGACTCCT
  BC TAGGCATG
  BC CTCTCTAC
  BC CGAGGCTG
  BC AAGAGGCA
  BC GTAGAGGA
  BC GCTCATGA
  BC ATCTCAGG
  BC ACTCGCTA
  BC GGAGCTAC
  BC GCGTAGTA
  BC CGGAGCCT
  BC TACGCTGC
```

**### Recommended not working on QRIS directory!**

Directory: /30days/uqmnaval/RNAseq/00_fasq_gz

```
cat 00_demultiplex_set1.pbs
#!/bin/bash
#PBS -A UQ-IMB
#PBS -l walltime=02:00:00
#PBS -l select=1:ncpus=10:mem=20GB
#cd /QRISdata/Q1287/00_fasq_gz/
cd /30days/uqmnaval/RNAseq/00_fasq_gz/

/home/uqmnaval/software/sabre/src/sabre -f Set1_S1_L001_R1_001.fastq.gz  -r
Set1_S1_L001_R2_001.fastq.gz  -b Set1_barcodes.txt  -c -u -m2 -l 10 -a 1 -s
Set1_sabre.txt -t 8
mv unassigned_R2.fq Set1_unassigned_R2.fq
mv unassigned_R1.fq Set1_unassigned_R1.fq
/home/uqmnaval/software/sabre/src/sabre -f Set2_S2_L002_R1_001.fastq.gz  -r
Set2_S2_L002_R2_001.fastq.gz  -b Set2_barcodes.txt  -c -u -m2 -l 10 -a 1 -s
Set2_sabre.txt -t 8
mv unassigned_R2.fq Set2_unassigned_R2.fq
mv unassigned_R1.fq Set2_unassigned_R1.fq
/home/uqmnaval/software/sabre/src/sabre -f Set3_S3_L003_R1_001.fastq.gz  -r
Set3_S3_L003_R2_001.fastq.gz  -b Set3_barcodes.txt  -c -u -m2 -l 10 -a 1 -s
Set3_sabre.txt -t 8
mv unassigned_R2.fq Set3_unassigned_R2.fq
mv unassigned_R1.fq Set3_unassigned_R1.fq
/home/uqmnaval/software/sabre/src/sabre -f Set4_S4_L004_R1_001.fastq.gz  -r
Set4_S4_L004_R2_001.fastq.gz  -b Set4_barcodes.txt  -c -u -m2 -l 10 -a 1 -s
Set4_sabre.txt -t 8
mv unassigned_R2.fq Set4_unassigned_R2.fq
mv unassigned_R1.fq Set4_unassigned_R1.fq
/home/uqmnaval/software/sabre/src/sabre -f Set5_S5_L005_R1_001.fastq.gz  -r
Set5_S5_L005_R2_001.fastq.gz  -b Set5_barcodes.txt  -c -u -m2 -l 10 -a 1 -s
Set5_sabre.txt -t 8
mv unassigned_R2.fq Set5_unassigned_R2.fq
mv unassigned_R1.fq Set5_unassigned_R1.fq

cat 00_demultiplex_set6.pbs
cat 00_demultiplex_Set6.pbs
#!/bin/bash
#PBS -A UQ-IMB
#PBS -l walltime=02:00:00
#PBS -l select=1:ncpus=10:mem=20GB

cd /30days/uqmnaval/RNAseq/00_fasq_gz/

/home/uqmnaval/software/sabre/src/sabre -f Set6_S6_L006_R1_001.fastq.gz  -r
Set6_S6_L006_R2_001.fastq.gz  -b Set6_barcodes.txt  -c -u -m2 -l 10 -a 1 -s
Set6_sabre.txt -t 8
mv unassigned_R2.fq Set6_unassigned_R2.fq
mv unassigned_R1.fq Set6_unassigned_R1.fq
/home/uqmnaval/software/sabre/src/sabre -f Set7_S7_L007_R1_001.fastq.gz  -r
Set7_S7_L007_R2_001.fastq.gz  -b Set7_barcodes.txt  -c -u -m2 -l 10 -a 1 -s
Set7_sabre.txt -t 8
mv unassigned_R2.fq Set7_unassigned_R2.fq
mv unassigned_R1.fq Set7_unassigned_R1.fq
/home/uqmnaval/software/sabre/src/sabre -f Set8_S8_L008_R1_001.fastq.gz  -r
Set8_S8_L008_R2_001.fastq.gz  -b Set8_barcodes.txt  -c -u -m2 -l 10 -a 1 -s
Set8_sabre.txt -t 8
mv unassigned_R2.fq Set8_unassigned_R2.fq
mv unassigned_R1.fq Set8_unassigned_R1.fq

(rnasik-1.5.4) uqmnaval@awoonga1:.../uqmnaval/RNAseq/00_fasq_gz>
```

After demultiplexing this is 77bp!

All data then is outputted and we have 148 *fastq files… some have 0Gb that is because they are merged with other datatypes:

## We now check per sample and barcode

#Check on the distinct barcodes…

The different

### First of all understanding the output in Set1!

#Number of reads in Set1:

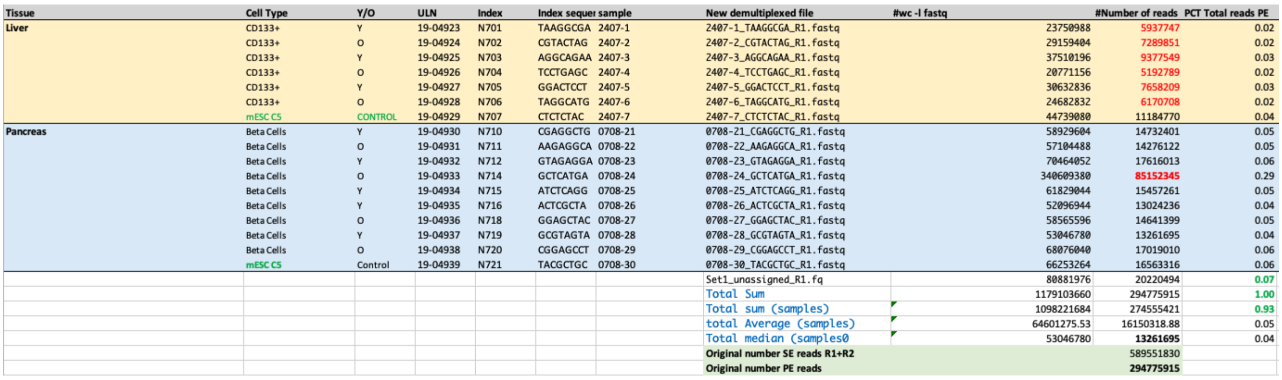
```
echo $(($(zcat Set1_S1_L001_R1_001.fastq.gz | wc -l)/4))
294775915 ### That matches with Hudson’s report “Lane 1 total reads passed filter
294.8 Million; Set 1 i7 index TAAGGCGA”
echo $(($(zcat Set1_S1_L001_R2_001.fastq.gz | wc -l)/4))
294775915

#so as SE: 294775915*2 = 589551830
```

###Looking at Sabre output…. The Toal of reads is inferior 293.571.193 PE reads… that is 587,142,386 SE reads (total number of reads R1+R2)

**Set1_m0_sabre.txt**

```
Barcode     N_records    N_pairs    P_pairs
TACGCTGC    9040379     4520189    0.02
CGGAGCCT    16481778     8240889    0.03
GCGTAGTA    12632956     6316478    0.02
GGAGCTAC    13941528     6970764    0.02
ACTCGCTA    12662261     6331130    0.02
ATCTCAGG    15157226     7578613    0.03
GCTCATGA    77588366     38794183    0.13
GTAGAGGA    16244670     8122335    0.03
AAGAGGCA    13848293     6924146    0.02
CGAGGCTG    20684602     10342301    0.04
CTCTCTAC    10809287     5404643    0.02
TAGGCATG    5921828     2960914    0.01
GGACTCCT    7377530     3688765    0.01
TCCTGAGC    5002548     2501274    0.01
AGGCAGAA    8860252     4430126    0.02
CGTACTAG    6979547     3489773    0.01
TAAGGCGA    5792122     2896061    0.01
unassigned    71239682    35619841    0.12
total    587142386    293571193    1.00 ###These numbers are lower than the original
number of reads :  294775915-293571193=1,204,722 ##One million reads from the
original input to sabre are discarded.
```

## FILE output file Lane 1/ Set 1:



| Tissue | Cell Type | Y/O | ULN | Index | Index sequer | sample | New demultiplexed file | #wc -l fastq | #Number of reads | PCT Total reads PE |
|---|---|---|---|---|---|---|---|---|---|---|
| Liver | CD133+ | Y | 19-04923 | N701 | TAAGGCGA | 2407-1 | 2407-1_TAAGGCGA_R1.fastq | 23750988 | 5937747 | 0.02 |
|  | CD133+ | O | 19-04924 | N702 | CGTACTAG | 2407-2 | 2407-2_CGTACTAG_R1.fastq | 29159404 | 7289851 | 0.02 |
|  | CD133+ | Y | 19-04925 | N703 | AGGCAGAA | 2407-3 | 2407-3_AGGCAGAA_R1.fastq | 37510196 | 9377549 | 0.03 |
|  | CD133+ | O | 19-04926 | N704 | TCCTGAGC | 2407-4 | 2407-4_TCCTGAGC_R1.fastq | 20771156 | 5192789 | 0.02 |
|  | CD133+ | Y | 19-04927 | N705 | GGACTCCT | 2407-5 | 2407-5_GGACTCCT_R1.fastq | 30632836 | 7658209 | 0.03 |
|  | CD133+ | O | 19-04928 | N706 | TAGGCATG | 2407-6 | 2407-6_TAGGCATG_R1.fastq | 24682832 | 6170708 | 0.02 |
|  | mESC C5 | CONTROL | 19-04929 | N707 | CTCTCTAC | 2407-7 | 2407-7_CTCTCTAC_R1.fastq | 44739080 | 11184770 | 0.04 |
| Pancreas | Beta Cells | Y | 19-04930 | N710 | CGAGGCTG | 0708-21 | 0708-21_CGAGGCTG_R1.fastq | 58929604 | 14732401 | 0.05 |
|  | Beta Cells | O | 19-04931 | N711 | AAGAGGCA | 0708-22 | 0708-22_AAGAGGCA_R1.fastq | 57104488 | 14276122 | 0.05 |
|  | Beta Cells | Y | 19-04932 | N712 | GTAGAGGA | 0708-23 | 0708-23_GTAGAGGA_R1.fastq | 70464052 | 17616013 | 0.06 |
|  | Beta Cells | O | 19-04933 | N714 | GCTCATGA | 0708-24 | 0708-24_GCTCATGA_R1.fastq | 340609380 | **85152345** | 0.29 |
|  | Beta Cells | Y | 19-04934 | N715 | ATCTCAGG | 0708-25 | 0708-25_ATCTCAGG_R1.fastq | 61829044 | 15457261 | 0.05 |
|  | Beta Cells | Y | 19-04935 | N716 | ACTCGCTA | 0708-26 | 0708-26_ACTCGCTA_R1.fastq | 52096944 | 13024236 | 0.04 |
|  | Beta Cells | O | 19-04936 | N718 | GGAGCTAC | 0708-27 | 0708-27_GGAGCTAC_R1.fastq | 58565596 | 14641399 | 0.05 |
|  | Beta Cells | Y | 19-04937 | N719 | GCGTAGTA | 0708-28 | 0708-28_GCGTAGTA_R1.fastq | 53046780 | 13261695 | 0.04 |
|  | Beta Cells | O | 19-04938 | N720 | CGGAGCCT | 0708-29 | 0708-29_CGGAGCCT_R1.fastq | 68076040 | 17019010 | 0.06 |
|  | mESC C5 | Control | 19-04939 | N721 | TACGCTGC | 0708-30 | 0708-30_TACGCTGC_R1.fastq | 66253264 | 16563316 | 0.06 |
|  |  |  |  |  |  |  | Set1_unassigned_R1.fq | 80881976 | 20220494 | 0.07 |
|  |  |  |  |  |  |  | Total Sum | 1179103660 | 294775915 | 1.00 |
|  |  |  |  |  |  |  | Total sum (samples) | 1098221684 | 274555421 | 0.93 |
|  |  |  |  |  |  |  | total Average (samples) | 64601275.53 | 16150318.88 | 0.05 |
|  |  |  |  |  |  |  | Total median (samples0 | 53046780 | **13261695** | 0.04 |
|  |  |  |  |  |  |  | **Original number SE reads R1+R2** |  | 589551830 |  |
|  |  |  |  |  |  |  | **Original number PE reads** |  | **294775915** |  |

### check EXCEL file:

<u>Recommendations this far: </u>re-run liver CD133 and Eye  RP cohort 2 pooled! # reads less than 10M

- The number of unassigned reads is as Kirill mentioned < than 10% of reads..
- The number of reads we got “single end this time is between 250-300 Millon rather than 400-500Millon
- The average number of reads per sample is 15 Million then with some fewer than 10!
- Kirill mentioned the aim was to have between 20-25 Million. This is not eh case in here! (The rule of thumb for RNA-seq gen expression is ~ 20-25K reads per sample. And I'm pretty sure this is what Trevor would aim for when doing the libraries.)

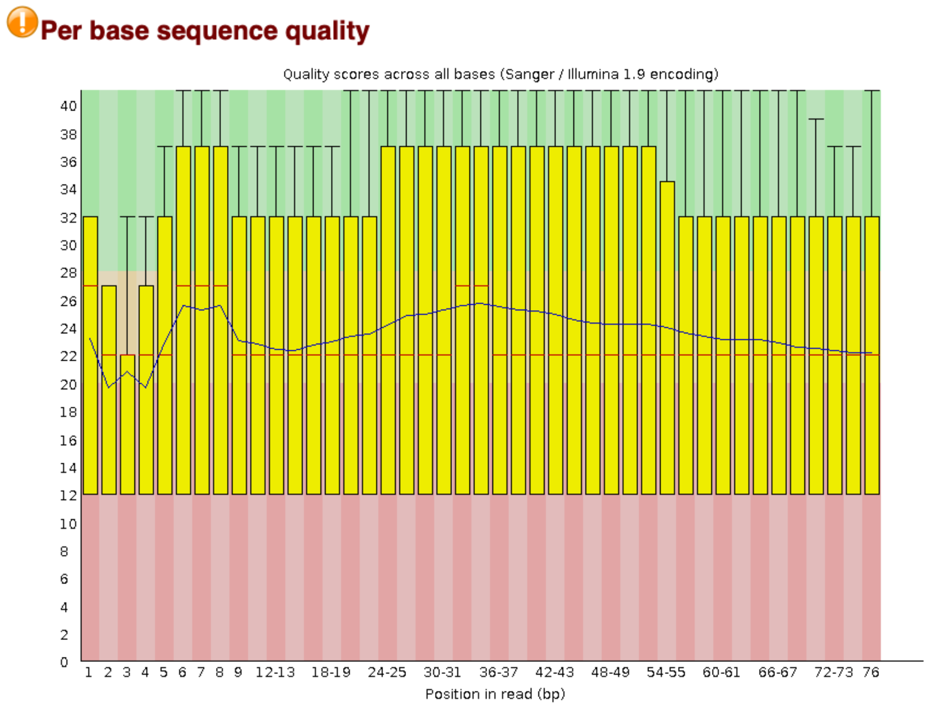
# 01_Data quality with fastqc

Well we had a general data fastqc report from Trevor and I am not sure if I am convinced on data. Usually at the

RNA-seq level you would like to have a Phred score above 30 and almost close to 40!

In here we are dealing with quality scores on the “red” area ==> danger !

We know that Phred Score is the same as probability of having an error.. I



When performing FastQC on the data we observer that we have “contamination of primers…” Therefore, before mapping they should be removed.
In particular, there is contamination of the following primer 19% of reds + 10% of reads!They have Cutadapt installed and working

Module load cutadapt **(version cutadapt/1.18)**

```
uqmnaval@tinaroo2:.../uqmnaval/RNAseq/01_demultiplexed> cutadapt -b
GCAGCGTAGTACTCTGCGTTGATACCACTGCTTCCGCGGACAGGCGTGTA -o 2607-9_clean_test.fastq 2607-
9_TACGCTGC_R1.fastq
This is cutadapt 1.18 with Python 3.6.8
Command line parameters: -b GCAGCGTAGTACTCTGCGTTGATACCACTGCTTCCGCGGACAGGCGTGTA -o
2607-9_clean_test.fastq 2607-9_TACGCTGC_R1.fastq
Processing reads on 1 core in single-end mode ...
```

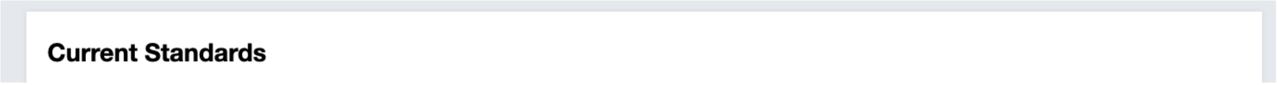
Cutadapt all options… so let’s clean the adapters first on them!

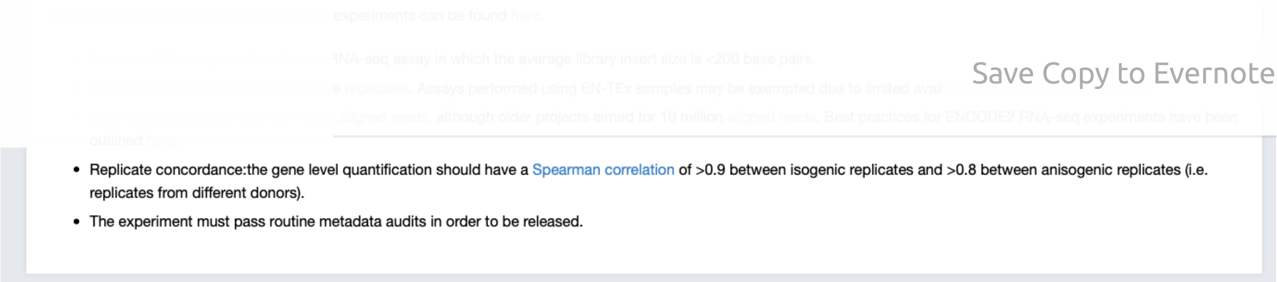
- `I should, I do have that information from Christian in Slack where he inputted all the info on how to perform the library preparations…`

And let’s compare our dataset with ENCODE current standards for small RNA-seq pipeline.

**### Check on the ENCODE quality standards:**

https://www.encodeproject.org/data-standards/rn


Current Standards



experiments can be found here.

- RNA-seq assay in which the average library insert size is <200 base pairs.
- e replicates. Assays performed using EN-TEx samples may be exempted due to limited avail
- aligned reads, although older projects aimed for 10 million aligned reads. Best practices for ENCODE2 RNA-seq experiments have been outlined here.
- Replicate concordance:the gene level quantification should have a Spearman correlation of >0.9 between isogenic replicates and >0.8 between anisogenic replicates (i.e. replicates from different donors).
- The experiment must pass routine metadata audits in order to be released.

a-seq/small-rnas/

# 01_ Mapping to mm10 with STAR aligner / RNAsik in Awoonga / Tinaroo HPC

Next step is mapping against genome….

We have lots of reads being thrown out….. why? That is weird.. I have never seen sth like that that…. Too shot… why?

# not sure….

#Some issues with too short alignments…. % of reds unmapped: too short | 24.87%

So I think in our case

We need to lower the trimming value or increase the trimming value…

Because if we have contamination of an adapter to 26bp

I changed the parameters to 0.3

And by default the

--outFilterMatchNmin
default: 0

int: alignment will be output only if the number of matched bases is higher

than or equal to this value.

--outFilterMatchNminOverLread
default: 0.66

real: sam as outFilterMatchNmin, but normalized to the read length

That is minimum 50 bp, (75 *.66=49.5bp) but if we have already 75- 26 bp from the Clontech SMART CDS Primer II A (100% over 26bp) that is already 49!!!! So it is already inferior that what expected

I think is just that…. But on the other hand is pointing at quite a bit of contamination in the reads… alsmot 26bp-30bp of read are with adapter sequences

```
Hi All
```

```
Sorry for the delayed reply. Do not hesitate to bump up your posts if I do not answer
within a week.
Setting --outFilterScoreMinOverLread 0 --outFilterMatchNminOverLread 0 removes any
limits on the mapped length, so even if short portions of the reads align, the reads
will be considered mapped. This explains why you are seeing high mapping rate -
albeit such short alignments are often multimappers. I do not think this is generally
a good idea, as such short alignments will have a high rate of wrong alignments. It
is better to figure out why the mappability is poor, e.g.
1. File formatting issues
2. Poor sequencing quality
3. Contamination
```

I also did what Kirill was mentioning that is to look at the database of unmapped reads and check out what's in there….

```
uqmnaval@tinaroo2:.../uqmnaval/RNAseq/01_demultiplexed> head -21 t
21384 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
326 GACGTCGAGTACTCTGCGTTGATACCACTGCTTCCGCGGACAGGCGTGTAGATCTCGGTGGTCGCCGTATCATTAA
314 ATCTTTAACTCTAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
290 CGTCGAGTACTCTGCGTTGATACCACTGCTTCCGCGGACAGGCGTGTAGATCTCGGTGGTCGCCGTATCATTAAAA
239 CTTATATATTGAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
224 TTATATACCTATGAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
215 NAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
203 CATATAATAAATAAAAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
197 ACGTCGAGTACTCTGCGTTGATACCACTGCTTCCGCGGACAGGCGTGTAGATCTCGGTGGTCGCCGTATCATTAAA
159 ACAGTGTACATACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
140 TATTATAACAGTAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
126 CAGCTACAGTGTACAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
125 GTATATACACCTCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
125 GTATATACCCCTCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
113 CTGTTAAATAAAGCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
111 CAGCTACAGTGTACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
107 CATATTGATCAAGCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
105 TTTGTGGACTGTGAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
103 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
99 TAATAAATAAATAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
97 CAGCTACAGTGTACTAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
uqmnaval@tinaroo2:.../uqmnaval/RNAseq/01_demultiplexed> tail t
1 TTTTTTTTTTTTTTTTTAACCGCACCGGGCAGCCCGACGCGCCGCGGCCCCCCCCCGGCCCGCGTGGCTGGCTGGG
1 TTTTTTTTTTTTTTTTTAGGGGTCAGCCAGCCCACAAGACAACGGGCGGCCCCGCCTGCCCTCCGGGCGTTTT
1 TTTTTTTTTTTTTTTTTTATTTTAAGAAAAAAGAAAACAAAAACAGTACTTTTTAATGGAAACAACTTGACCAAAA
1 TTTTTTTTTTTTTTTTTTTTCGAAGACCGAAGCACACCCAAAGCAACAACCCCACAAACCAAGCGACGGGGTCGG
1 TTTTTTTTTTTTTTTTTTTTTAGTGTGCCAGGACGACGACCGAACCCCAAACACTTCTGCGGGGGTGCCGCTGGGT
1 TTTTTTTTTTTTTTTTTTTTTCAACAACCACCCCCCCCCCGCCCCCACCCCCCCACGCGCCCCCACGCGTGTGCA
1 TTTTTTTTTTTTTTTTTTTTTCCTGAGACAGCGCCGAGCGGGCTCCGGCCCCGCCCCGCGCCCTCCGTCTTGTTCG
1 TTTTTTTTTTTTTTTTTTTTTGACAAACCCAAGAAGCCCAGTGGCGCACACACCTGGGTGACACCTTGTGGATGCA
1 TTTTTTTTTTTTTTTTTTTTTTTTAATAGCGCGCCCTCCCGGCCCGGCCTGCCTTTTTCTCTGGTGCGTTGTCT
1 TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTGATACCACTTCTTCCGCTGTCTGCCGTGTAGGTCTCGTTGGTTGGT
```

#### Mapping option: 01_demultiplexed/STAR_aling_03.pbs

```
#!/bin/bash
#PBS -A UQ-IMB
#PBS -N STAR
#PBS -l walltime=02:00:00
```

```
=10:mem=40GB
Aseq/01_demultiplexed/

mnaval/miniconda/etc/profile.d/conda.sh
conda activate rnasik-1.5.4
STAR  --runThreadN 12 --runMode alignReads --genomeDir
/90days/uqmnaval/genomes/Mus_musculus.GRCm38.dna.primary_assembly.starIdx  --
readFilesIn ${FILE} --readFilesCommand zcat  --outFileNamePrefix ${FILE}.03.  --
outSAMtype BAM SortedByCoordinate --outSAMunmapped Within ls -
```

#### Alternatively… using STAR we can use the transcriptome to check if the mapping is better as well as quantMode GeneCount

```
uqmnaval@tinaroo2:.../uqmnaval/RNAseq/01_demultiplexed> cat STAR_align_03.pbs
#!/bin/bash
#PBS -A UQ-IMB
#PBS -N STAR
#PBS -l walltime=02:00:00
#PBS -l select=1:ncpus=10:mem=40GB
cd /30days/uqmnaval/RNAseq/01_demultiplexed/
module load star
source /gpfs1/homes/uqmnaval/miniconda/etc/profile.d/conda.sh
conda activate rnasik-1.5.4
STAR  --runThreadN 12 --runMode alignReads --genomeDir
/90days/uqmnaval/genomes/Mus_musculus.GRCm38.dna.primary_assembly.starIdx  --
readFilesIn ${FILE} --readFilesCommand zcat  --outFileNamePrefix ${FILE}.03.annot.  -
-outSAMtype BAM SortedByCoordinate --outSAMunmapped Within --
outFilterMatchNminOverLread 0.3 --outFilterScoreMinOverLread 0.3 --twopassMode Basic
--sjdbGTFfile
/90days/uqmnaval/genomes/gencode.vM23.chr_patch_hapl_scaff.annotation.gtf --
sjdbOverhang 75 --quantMode GeneCounts
```

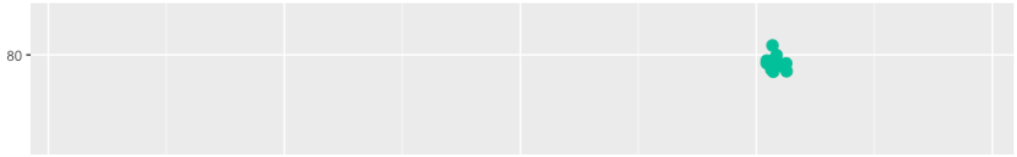
OK!

Now we go with featureCounts:
On them…..

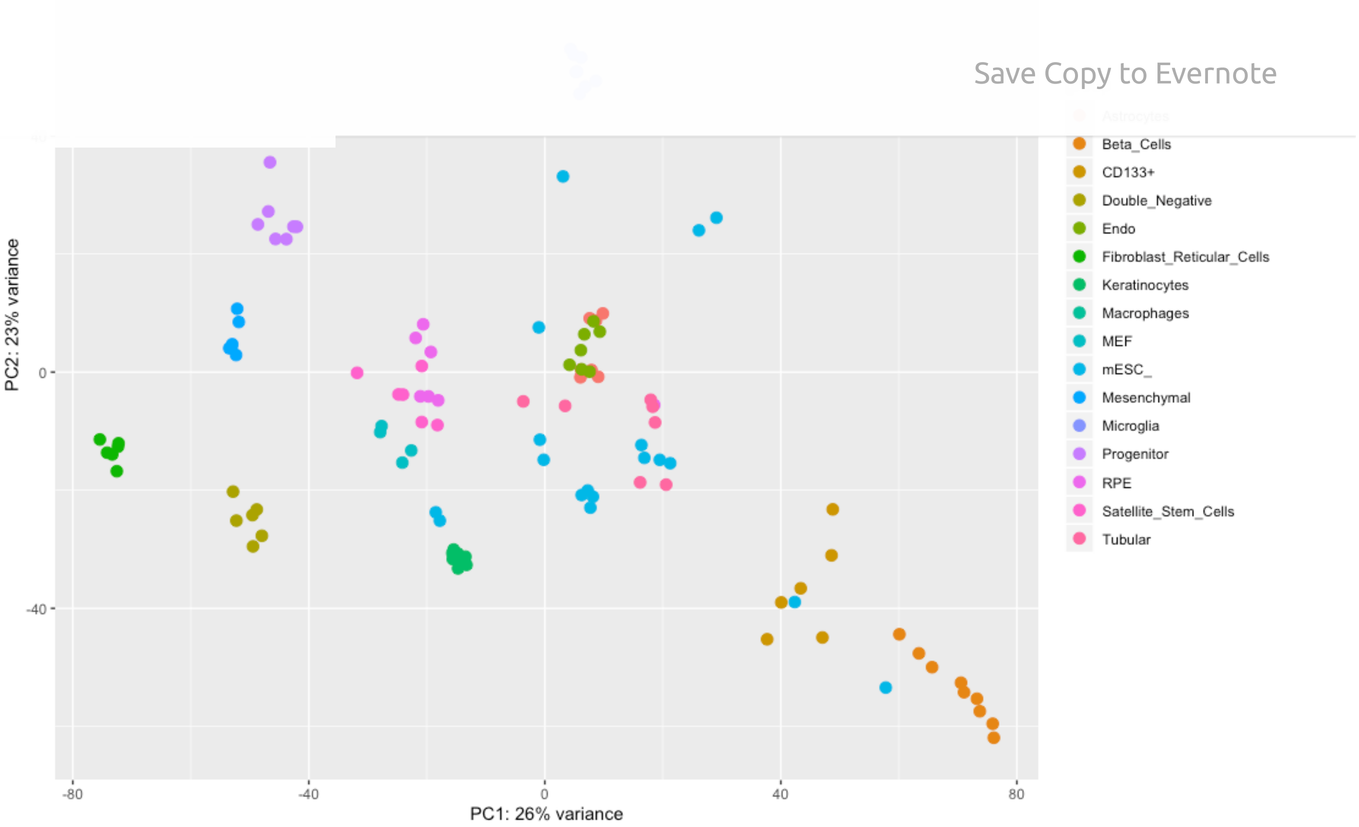
However, shall we remove duplicates on them? Though PICARD? … in RNA-seq data??

**Also the counts could be performed with STAR!**

- All_counts.txt
- On the data…. Of them...
-

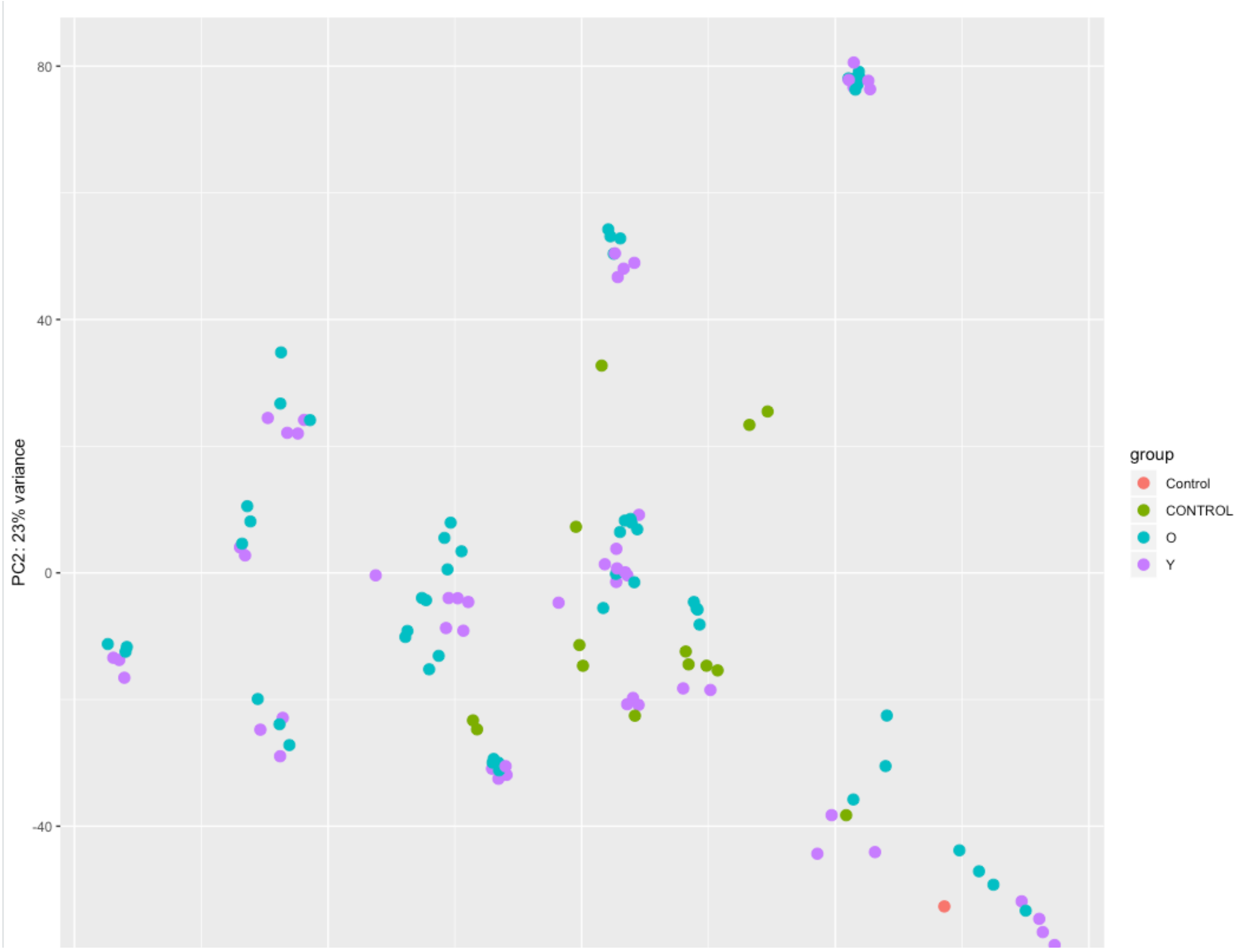


Save Copy to Evernote



##Check here the "batch effect on them"

plotPCA(vsd,intgroup='V3')



#### Now we perform on them