PLATE-seq data analysis steps:

Data:

Graphical user interface, text

Description automatically generated

Graphical user interface, application, table, Excel

Description automatically generated

Text

Description automatically generated

Checking the reads in the files

Gunzip -c Schwabe\_S7\_L003\_R1\_001.fastq.gz | wc -l

5398357364

5,398,357,364 lines, 4 lines per read. So 1,349,589,341 reads for 384 wells

Analysis:

Table

Description automatically generated

Steps:

1. Remove the 5’ 1-26bp common adapter seq from all the well barcodes.
2. Use the remaining 27-60bp, 33nt long, well barcodes to demultiplex the fastq files (with 2 mismatches)

‘Sabre’ could not run on the files and was quitting due to ‘Segmentation fault 11’ error: not enough memory issues. Then I used ‘cutadapt’ for removing the barcodes and demultiplexing the well reads.

cutadapt --error-rate 0.061 --no-indels --cores 10 --action=trim -g file:barcodes\_trimmed.fasta -o trimmed-{name}.R1.fastq.gz -p trimmed-{name}.R2.fastq.gz Schwabe\_S7\_L003\_R1\_001.fastq.gz Schwabe\_S7\_L003\_R2\_001.fastq.gz

explanation:

--error-rate 0.061 because 0.061\*33=2.01; so two mismatches allowed for a 33nt long trimmed adapter

-g file:barcodes\_trimmed.fasta is the file containing the trimmed adapter in fastq format

1. Run Kallisto on the demultx fastq files to get transcript level abundance (3’ over hang option --single-overhang)

# index is prebuild for mus so no need to build myself

https://github.com/pachterlab/kallisto-transcriptome-indices/releases

Ensembl Transcriptomes v96 Latest

These index files were produced using kallisto version 0.45.1 on Ensembl v96 transcriptomes. The transcripts\_to\_genes.txt files were made with t2g.py. DOWNLOADED ON 2022-07-27

THEN QUANTIFY USING KALLISTO

I tried kallisto on paired-end and single-end R2 fastq files. Kallisto on single-end fastq files were giving ~35-60% more fragments/counts per gene than paired-end. This could be due to barcode/polyA capture sequences being present in the R1 fastq files. The PLATE-seq paper also suggests using just R2 reads.

kallisto quant -i transcriptome\_mus.idx -o counts\_R2/output\_R2\_ A1 -b 100 --single -l 180 -s 20 --threads=10 --single-overhang fastq/trimmed- A1.R2.fastq.gz

--single -l 180 -s 20 for single-end fastq files with fragment length 180 and sd 20. It is difficult to get these values for our dataset since it was paired-end, so used the values suggested in kallisto manual.

--single-overhang for 3’ capture.

-b 100 for 100 bootstraps

--threads=10 for 10 parallel threads of processing

Diagram

Description automatically generatedDiagram

Description automatically generated

Chart, box and whisker chart

Description automatically generated

1. Run tximport to get the gene level counts

Used the transcripts to gene mapping "transcripts\_to\_genes.txt" present in the mouse dataset from kallisto

transcripts\_to\_genes <- read.delim("transcripts\_to\_genes.txt", header=FALSE)

tx2gene <- data.frame(transcripts=transcripts\_to\_genes[,1],genes=transcripts\_to\_genes[,3])

txi.kallisto.tsv <- tximport(files, type = "kallisto", tx2gene = tx2gene)

1. Use Deseq2 to get the DEGs