Differential Gene Expression

Trinity Documentation: <https://github.com/trinityrnaseq/trinityrnaseq>

1. Create conda environment

conda create -n <envr name>

conda activate <envr name>

1. Check raw sequence quality with fastQC
2. Enter directory with raw data

cd <dir name>

1. Create submission script

nano <script name>

1. Paste following and change as necessary

#!/bin/bash

#$ -S /bin/sh

. /etc/profile

#$ -cwd

#$ -pe threads 50

PATH="/home-dir/home/mbrown/software/fastqc"

fastqc -o <output directory name> \*.fastq.gz

1. While in raw data folder, submit script to the cluster

qsub <script name>

1. Trim raw sequences with trimmomatic
2. Concatenate forward sequences together

cat \*R1\* > forward.gz

1. Concatenate reverse sequences together

cat \*R2\* > reverse.gz

1. Create submission script

#!/bin/bash

#$ -S /bin/sh

. /etc/profile

#$ -cwd

#$ -pe threads 70

PATH="/home-dir/home/mbrown/software/trinityrnaseq- v2.13.2:$PATH"

Trinity --seqType fq --max\_memory 100G --CPU 70 --trimmomatic --quality\_trimming\_params "ILLUMINACLIP://home-dir/home/mbrown/ software/trinityrnaseq-v2.13.2/Trimmomatic/adapters/NexteraPE- PE.fa:2:30:10 SLIDINGWINDOW:4:5 LEADING:5 TRAILING:5 MINLEN:25" --bflyHeapSpaceMax 100G --full\_cleanup --output salt\_trinity\_out --left /home-dir/home/lkirsch/salt\_experiment/foward.gz --right /home-dir/home/lkirsch/salt\_experiment/reverse.gz

1. Submit script

Transcript quantification with RSEM

1. Create samples\_ID.txt

* Tab delimited
* Columns => condition sample\_ID full path to forward full path to reverse
  + Headers not needed
  + Need to list out each individual sample with the full path

1. Create submission script

#!/bin/bash

#$ -S /bin/bash

. /etc/profile

#$ -cwd

#$ -pe threads 50

#$ -V

align\_and\_estimate\_abundance.pl --transcripts /home/lkirsch/salt\_experiment/raw\_data/Arcevulg-NoBact.outRef\_Transcriptome.fas --est\_method RSEM --aln\_method bowtie2 --prep\_reference --trinity\_mode --samples\_file salt\_IDs.txt --seqType fq

1. Build transcript and gene expression matrices

<path to Trinity utils>/abundance\_estimates\_to\_matrix.pl --est-method RSEM <list files> --gene\_trans\_map ‘none’

* A ‘quant file’ listing all target files can be used, but sometimes doesn’t work
* You can list each file, it’s annoying but will for sure work
* HAVE TO SPECIFY NO GENE TRANS MAP

1. Counting number of expressed genes
2. Run in command line

<Trinity path>/util/misc/count\_matrix\_features\_given\_MIN\_TPM\_threshold.pl genes\_matrix.TPM.not\_cross\_norm | tee genes\_matrix.TPM.not\_cross\_norm.counts\_by\_min\_TPM

and

<Trinity path>/util/misc/count\_matrix\_features\_given\_MIN\_TPM\_threshold.pl

trans\_matrix.TPM.not\_cross\_norm | tee trans\_matrix.TPM.not\_cross\_norm.counts\_by\_min\_TPM

1. Open R from command line

%R

1. Plot expressed gene counts

Data = read.table(“genes\_matrix.TPM.not\_cross\_norm.counts\_by\_min\_TPM”, header = T)

Plot(data, xlim = c(-100,0), ylim = c(0,100000), t=’b’)

1. Filter transcripts based on expression value

<Trinity path>/util/filter\_low\_expr\_transcripts.pl –matrix <string> --transcripts <string> --trinity\_mode

QC Samples and Biological Replicates

1. Create a new samples.txt

* Tab delimited
* Columns => condition sample\_ID
  + Header not needed

1. Compare reps for each sample

<Trinity path>/Analysis/DifferentialExpression/PtR --matrix counts.matrix --samples samples.txt --log2 --CPM --min\_rowSums 10 --compare\_replicates

1. Compare reps across samples
2. Create heat map

<Trinity path>/Analysis/DifferentialExpression/PtR --matrix counts.matrix --min\_rowSums 10 --s samples.txt --log2 --CPM --sample\_cor\_matrix

1. Plot PCA

<Trinity path>/Analysis/DifferentialExpression/PtR --matrix counts.matrix -s samples.txt --min\_rowSums 10 --log2 --CPM --center\_rows --prin\_comp 3

Differential Expression Analysis

1. DE Analysis
2. Open R in command line

%R

1. Install required libraries

BiocManager::install(c(“edgeR”, “limma”, “DESeq2”, “ctc”, “Biobase”, “gplots”, “ape”, “argparse”))

* Will need to reinstall if you close R

1. Run DESeq2 and edgeR

<Trinity path>/Analysis/DifferentialExpression/run\_DE\_analysis.pl --matrix counts.matrix --method <edgeR | DESeq2> --samples\_file samples.txt

* Use the 2nd samples.txt file you made with the conditions and IDs only
* For edgeR, need to add --dispersion <float> argument
* If you don’t use a samples.txt file, the program will compare every sample to every other sample

1. Extract and cluster DE transcripts

<Trinity path>/Analysis/DifferentialExpression/analyze\_diff\_expr.pl --matrix <TMM.EXPR.matrix> --samples samples.txt

* Use 2nd samples.txt

1. Partition genes into clusters

<Trinity path>/Analysis/DifferentialExpression/define\_clusters\_by\_cutting\_tree.pl -R <string> --Ptree 60

Annotation

Enrichment

1. For each of the DE output .subset file, create a .txt with only the transcript IDs
2. Open file in TextEdit
   1. Right click file name -> open with -> TextEdit
3. Command + shift + down arrow to highlight entire text
4. Command + c to copy
5. Open excel document and paste text
6. Delete all but the first column with the transcript IDs
   1. You can also remove the sampleA header
7. Save as tab delimited .txt
   1. You’ll need to add an additional line at the end
   2. Make sure to name appropriately so you will know to what each file corresponds
8. Format eggNOG annotation
9. Open eggNOG annotation with TextEdit
10. Remove all text above the line beginning with “#query” and save
11. Make sure you are in the directory with you annotation file and the transcript ID files
12. Run format.py

python format.py <filename> <filename.formatted>

1. Open R in command line

%R

1. Open cluster\_profile.r in a separate IDE window

* You will need to change this file to correspond with your data
  + Read each transcript ID file into R

<filename> <- read.lines(con = “filename.txt”)

* + Check enrichment

<variable> <- enricher(<filename>, pvalueCutoff = 0.05, pAdjustMethod = “fdr”, qvalueCutoff = 0.05, TERM2GENE = term2gene, TERM2NAME = NA)

1. In R, install the ‘clusterProfiler’ library

library(“clusterProfiler”)

* If the first time using, you will need to install from Bioconductor

if (!require(“BiocManager”, quietly = TRUE))

install.packages(“BiocManager”)

BiocManager::install(“clusterProfiler”)

1. Run cluster\_profile.r line by line in groups per transcript ID file

* Readlines
* Enricher
* If enriched, write.table to save output