RNA-seq and Differential Expression

Texas A&M HPRC

March 22, 2024

High Performance
Research Computing
DIVISION OF RESEARCH

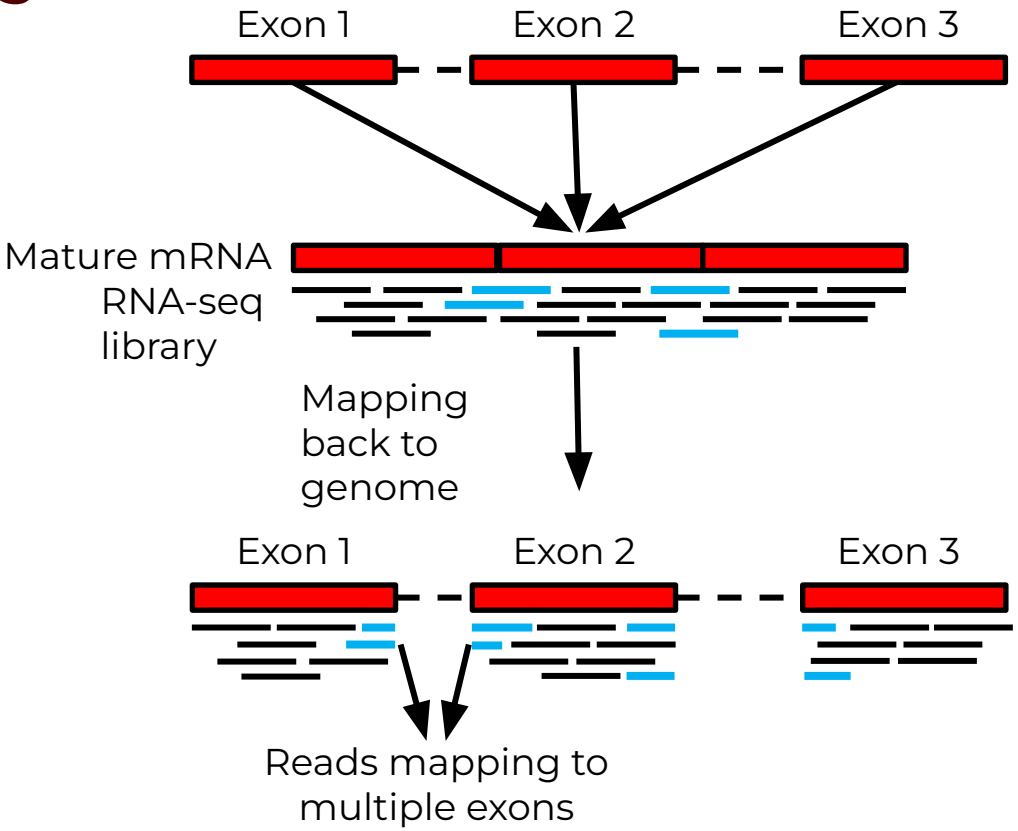
What does RNA-seq data provide?

- Measure gene expression
- Detect differences in expression between groups
- Annotate genomes
- Transcriptome assembly
- Nucleotide variant discovery
- Genome assembly scaffolding

RNA-seq Applications

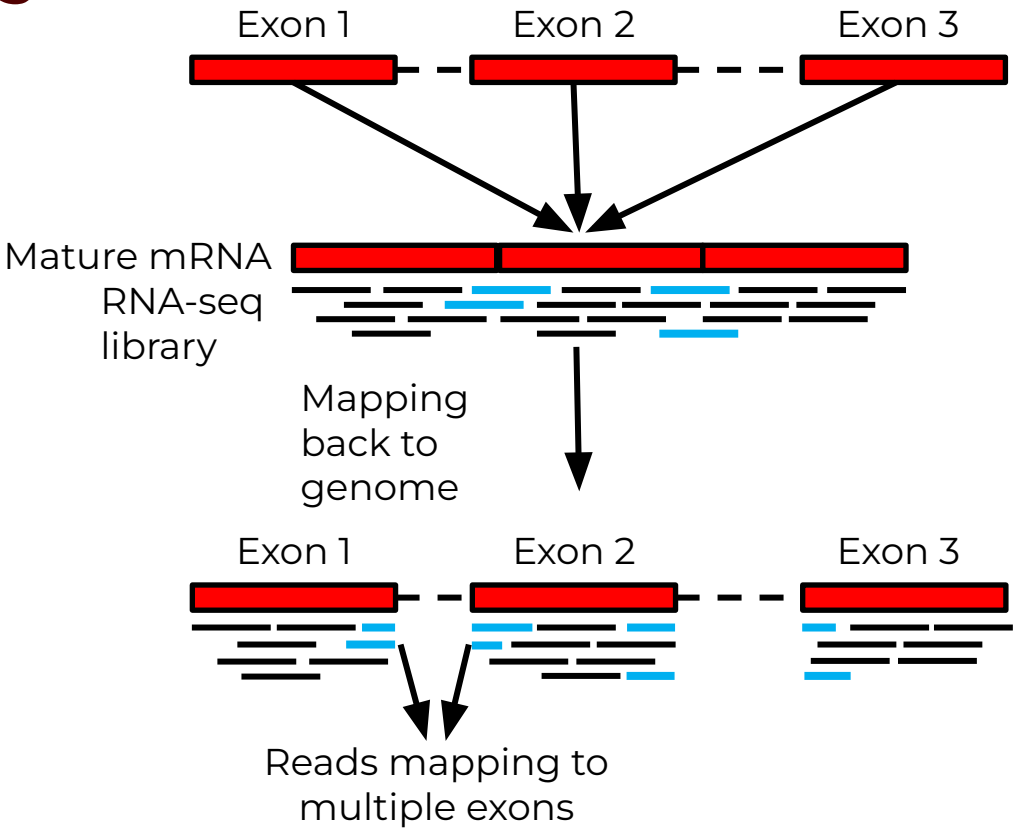
- Transcriptome Assembly
  - ***de novo***: Trinity, Oases, SOAPdenovo-Trans
  - **Reference-based**: Trinity, StringTie, Cufflinks

- Splice-aware alignment
  - HISAT2
  - STAR
  - Clara Parabricks (GPU-accelerated STAR)



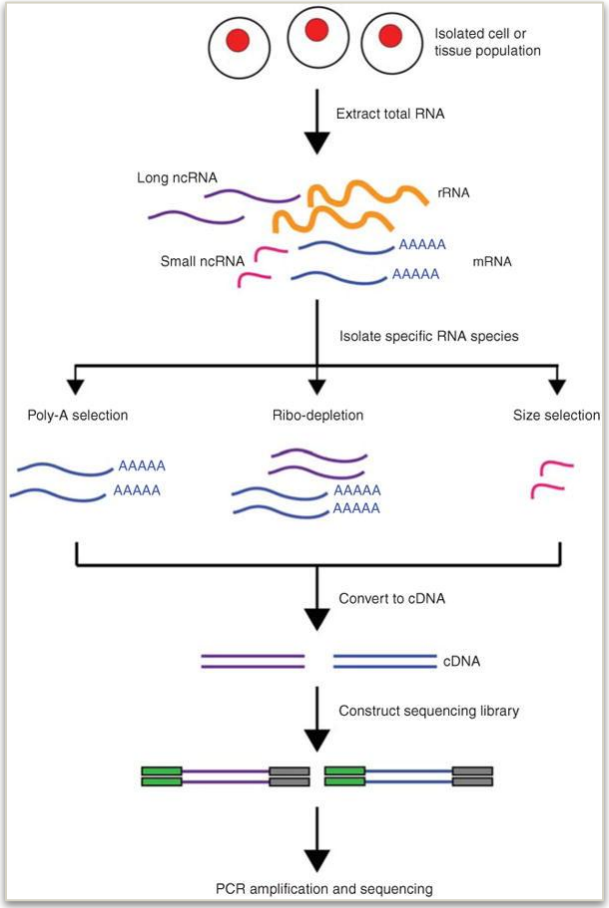
RNA-seq Applications

- File conversion and formatting
  - SAMtools
  - Picard tools
- Variant Calling
  - GATK (HaplotypeCaller in RNA-seq mode)
- Scaffolding Assemblies
  - L_RNA_scaffolder
  - Rascaf



Types of RNA-seq Libraries

- Poly-A selection - enriches for mRNA
- Ribosomal depletion - removes rRNA, leaving mRNA, lncRNA, and pre-mRNA
- Size selection (smRNA)



Kukurba and Montgomery, 2016

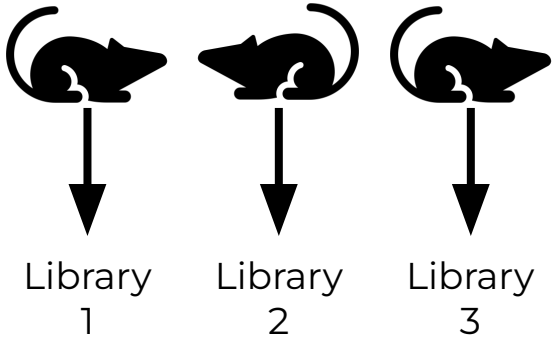
Experimental Design (for Differential Expression)

- Sequencing Depth
  - Minimum 30 million aligned reads per replicate (ENCODE)
  - 30-60 million reads per replicate (Illumina)

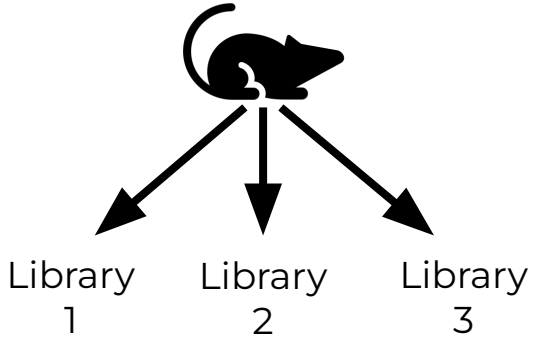
- Replicate Number
  - 3 replicates per condition minimum (will likely recover 20-40% of true DEGs)
  - Schurch et al. (2016) suggest 6 replicates per condition minimum, 12 replicates per condition optimal

Experimental Design (for Differential Expression)

- Biological Replicates
  - Independent samples from different populations or individuals



- Technical Replicates
  - Multiple libraries from the same individual

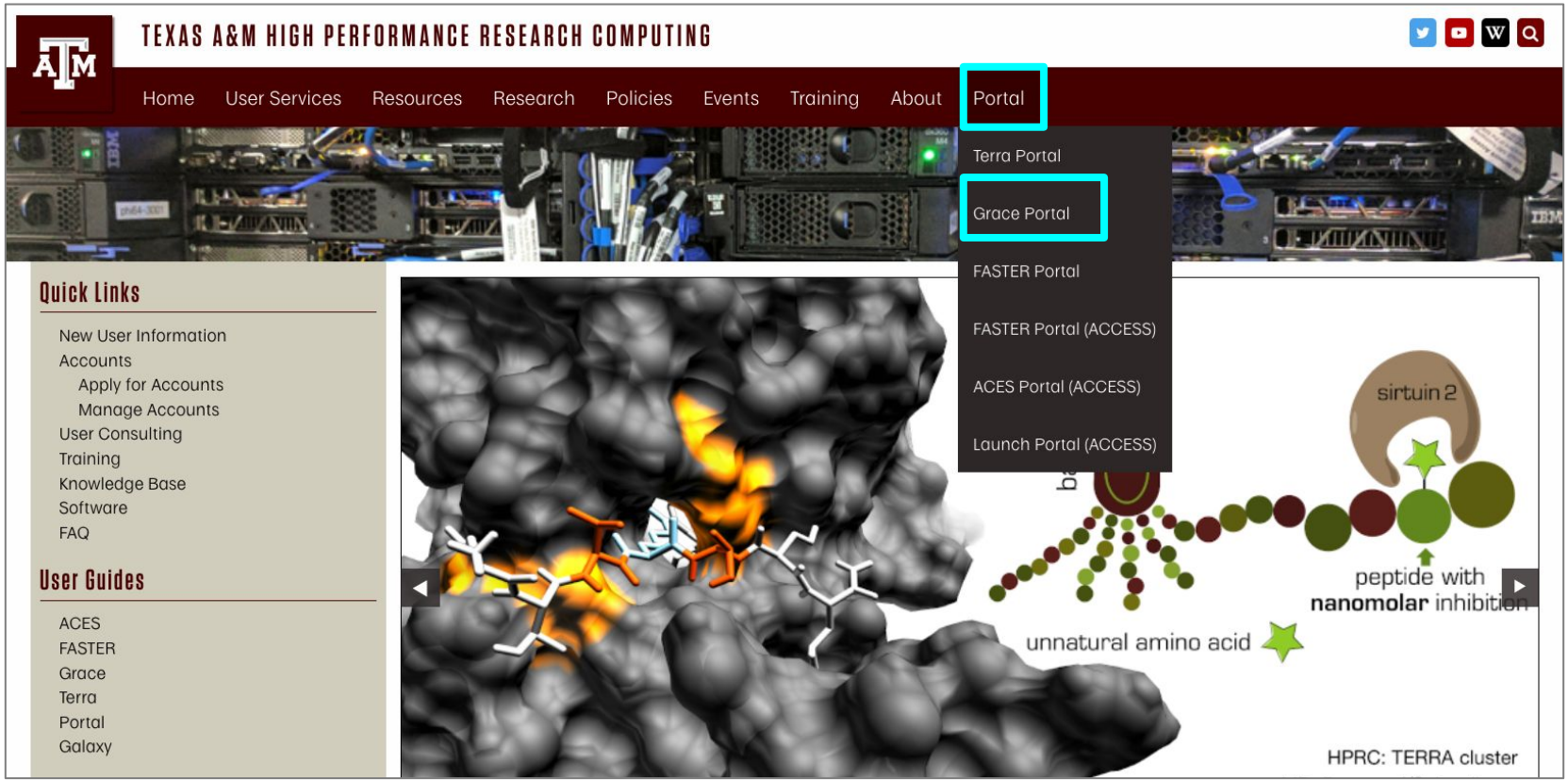


Experimental Design (for Differential Expression)

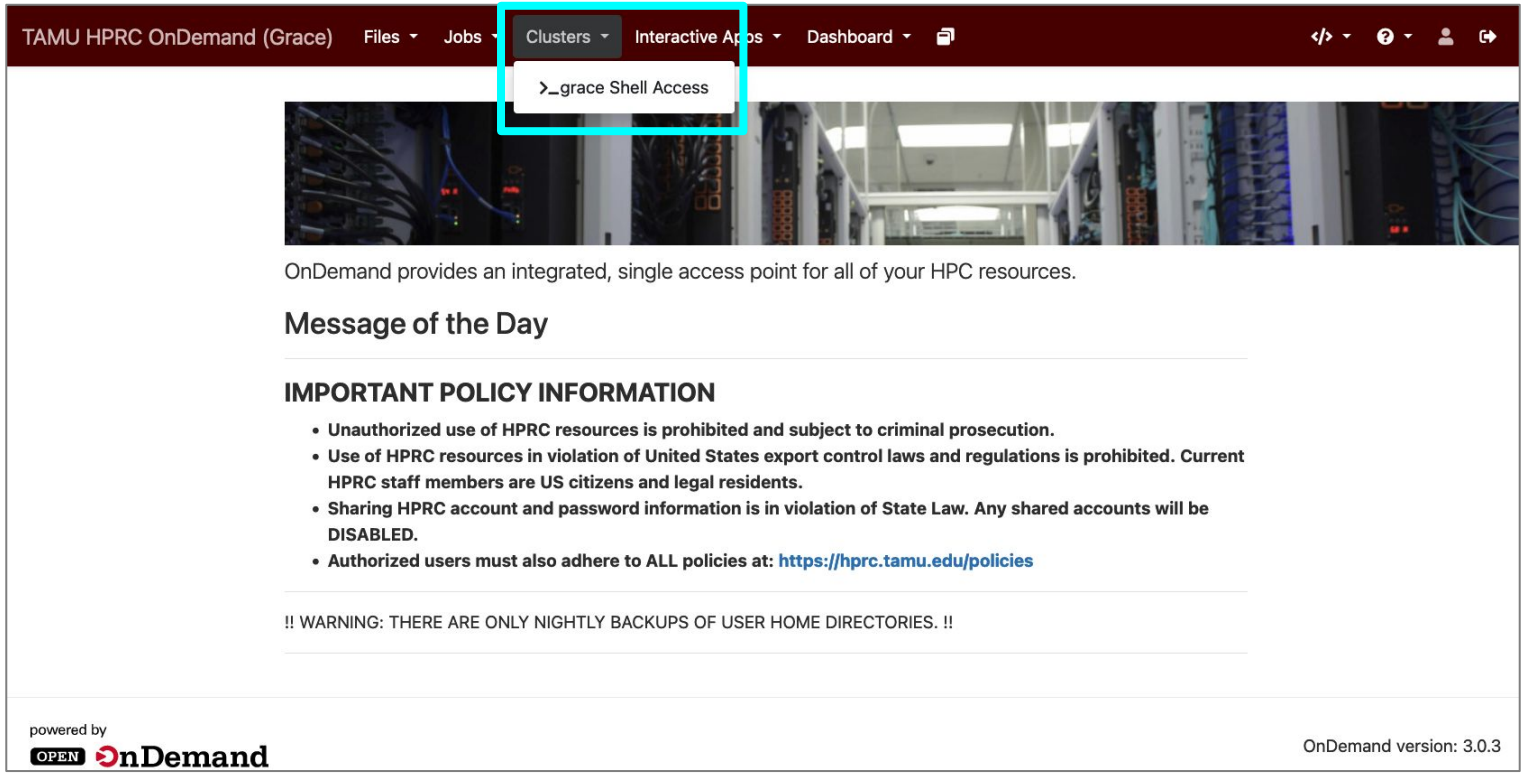
Replicates - Which to use?

- Biological replicates generally increase statistical power more than technical replicates
- Biological variability > Technical Variability
- Biological replicates contain both biological and technical variability

Accessing the HPRC Grace Portal



Accessing Grace Shell in OOD Portal



Accessing Grace Shell in OOD Portal

Example Data

- Create a new directory in your scratch space

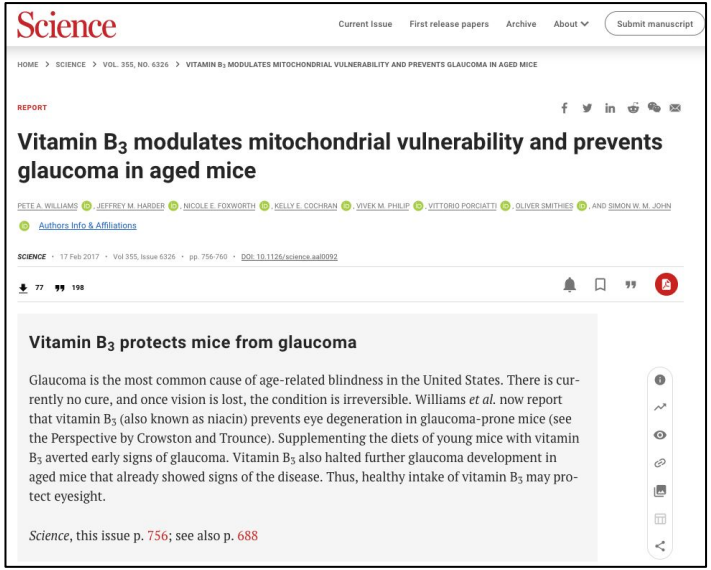
```
$ mkdir $SCRATCH/RNA_class
```

- Change your working directory to the one you just created

```
$ cd $SCRATCH/RNA_class
```

- Copy the example data to your directory

```
$ cp -r /scratch/training/bio/rna-seq/* .
```



Science

Current Issue First release papers Archive About Submit manuscript

HOME > SCIENCE > VOL. 355, NO. 6326 > VITAMIN B_3 MODULATES MITOCHONDRIAL VULNERABILITY AND PREVENTS GLAUCOMA IN AGED MICE

REPORT

Vitamin B_3 modulates mitochondrial vulnerability and prevents glaucoma in aged mice

PETE A. WILLIAMS, JEFFREY M. HARDER, NICOLE E. FOXWORTH, KELLY E. COCHRAN, VIVEK M. PHILIP, VITTORIO PORCIATTI, OLIVER SMITHIES, AND SIMON W. M. JOHN

Authors Info & Affiliations

SCIENCE • 17 Feb 2017 • Vol 355, Issue 6326 • pp. 756-760 • DOI: 10.1126/science.aal0092

77 198

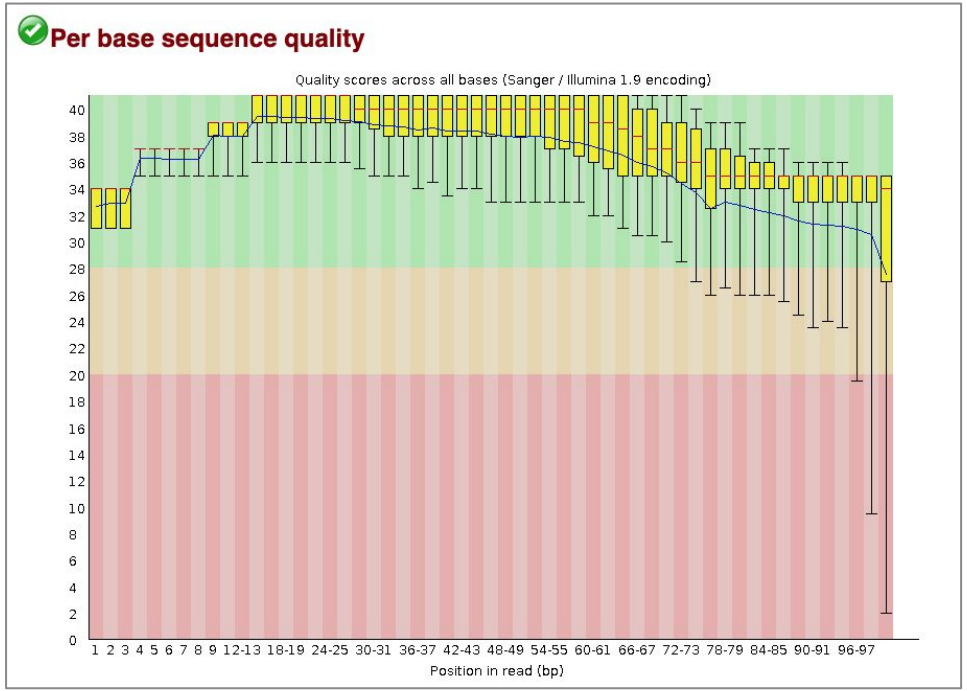
Vitamin B_3 protects mice from glaucoma

Glaucoma is the most common cause of age-related blindness in the United States. There is currently no cure, and once vision is lost, the condition is irreversible. Williams *et al.* now report that vitamin B_3 (also known as niacin) prevents eye degeneration in glaucoma-prone mice (see the Perspective by Crowston and Trounce). Supplementing the diets of young mice with vitamin B_3 averted early signs of glaucoma. Vitamin B_3 also halted further glaucoma development in aged mice that already showed signs of the disease. Thus, healthy intake of vitamin B_3 may protect eyesight.

Science, this issue p. 756; see also p. 688

Quality Control

- NGS libraries should be assessed for adapter content and low-quality reads before downstream analysis
- Low-quality bases and adapters can introduce errors and reduce map rates
- Avoid overly aggressive trimming practices



Quality Control

- Will use FastQC to examine the quality of our example data
- Look for the appropriate module on Grace:

```
$ module spider fastqc
```

```
$ module spider FastQC/0.11.9-Java-11
```

- Clear any previously loaded modules and load FastQC:

```
$ module purge
```

```
$ module load FastQC/0.11.9-Java-11
```

Running jobs on Grace

- Small jobs can be run on the login nodes (< 60 minutes, up to 8 cores)
- Larger jobs should be submitted to the compute nodes:
  - Slurm job scheduler
  - Can specify computing requirements:
    - Amount of memory required
    - Number of cores
    - Which modules to load
- Template job scripts are available:

  https://hprc.tamu.edu/kb/Software/useful-tools/GCATemplates/

Quality Control

- Run FastQC on our example fastqs:

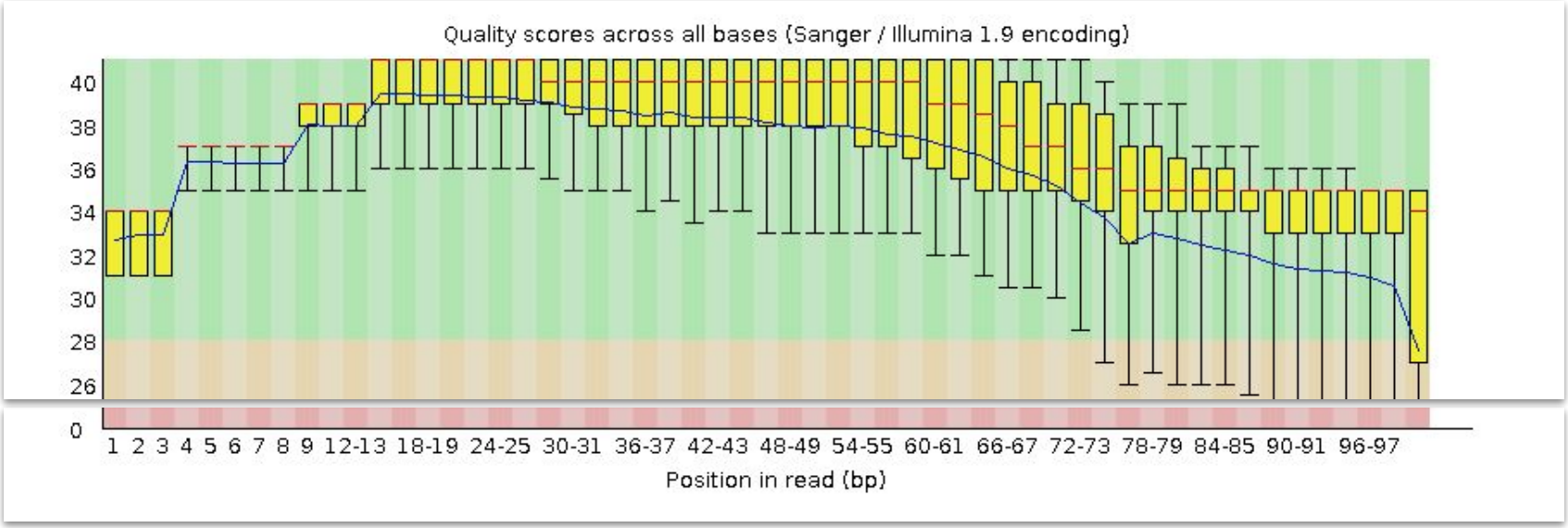
```
$ fastqc -t 2 -o . Control1_R1.fastq.gz Control1_R2.fastq.gz
```

- Go to “Files” tab in Grace portal and navigate to the RNA_class directory

- FastQC results saved as html files

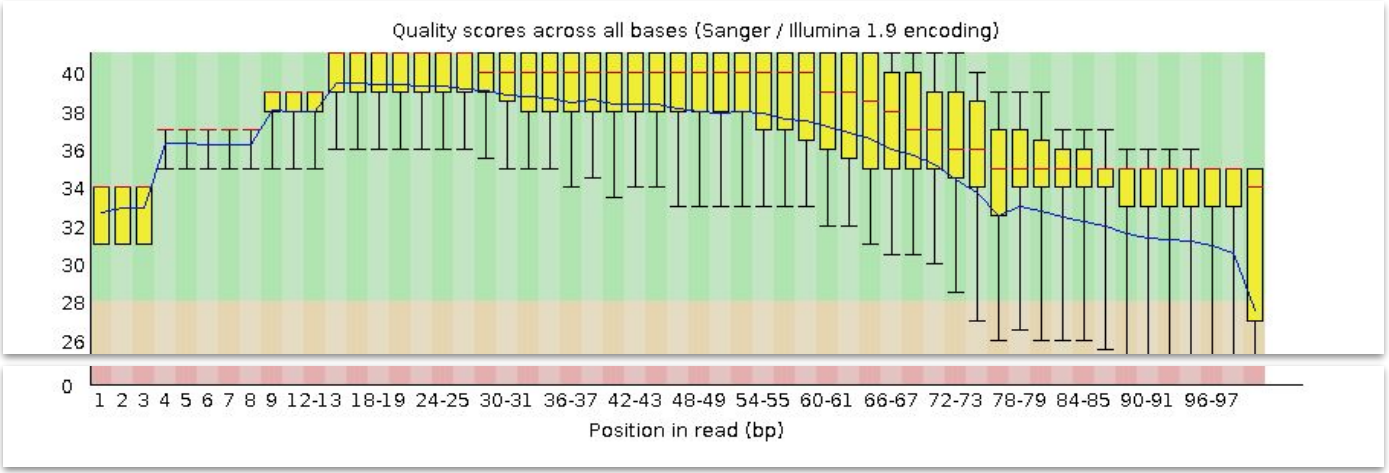
FASTQ Format

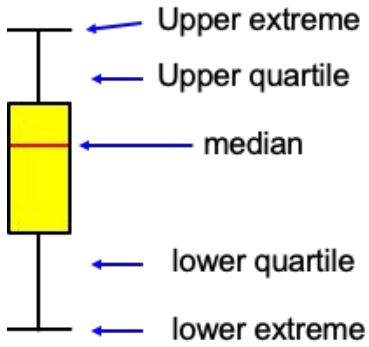
```
@ERR504787.2.1 M00368:15:000000000-A0HKH:1:5:21261:10968-1 length=100
GATCGGAAGAGCACACGTCTGAACTCCAGTCACGATCAGATCTCGTATGCCGTCTTCTGCTTGAAAAAAAAAAAAAAAAACAAAAAACATAATGCCGTAAA
+ERR504787.2.1 M00368:15:000000000-A0HKH:1:5:21261:10968-1 length=100
=:=4AD=B8A:+<A::1<:AE<C3*?F<B???<?:8:6?B*9BD;/638.=-'-.@7=).=A:6?DDDCBBBB9555&&)+((+2&&+(((((((&&&+
@ERR504787.3.1 M00368:15:000000000-A0HKH:1:3:12724:25677-1 length=100
GATGTTTTGTTACTGATTGGAACCATGATTGGTGCTTTACTTGGTTTCTTCCTATTTAACCACAAGCCTGCCAAAGTATTTATGGGAGATGTAGGTAGTT
+ERR504787.3.1 M00368:15:000000000-A0HKH:1:3:12724:25677-1 length=100
BCCFDEFFHHHHHJJJJIJJJJJJJJJIJJJFHIJJJIJJJJJIJJJJJJJIJJJIJIIJJJJJFHJJJJIJJHH=CHHFFFFFFEDDEDEEEDCCDCE
```



FASTQ Format

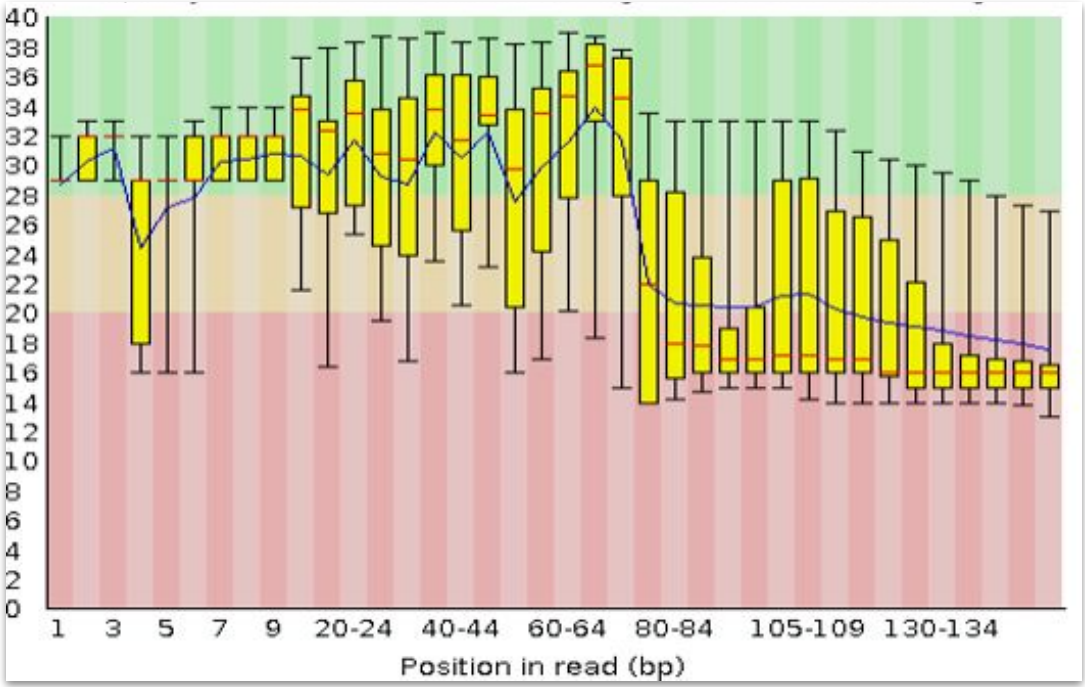
```
@ERR504787.2.1 M00368:15:000000000-A0HKH:1:5:21261:10968-1 length=100
GATCGGAAGAGCACACGTCTGAACTCCAGTCACGATCAGATCTCGTATGCCGTCTTCTGCTTGAAAAAAAAAAAAAAAAACAAAAAACATAATGCCGTAAA
+ERR504787.2.1 M00368:15:000000000-A0HKH:1:5:21261:10968-1 length=100
=:=4AD=B8A:+<A::1<:AE<C3*?F<B???<?:8:6?B*9BD;/638.=-'-.@7=).=A:6?DDDCBBBB9555&&)+((+2&&+((((((()&&&+
@ERR504787.3.1 M00368:15:000000000-A0HKH:1:3:12724:25677-1 length=100
GATGTTTTGTTACTGATTGGAACCATGATTGGTGCTTTACTTGGTTTCTTCCTATTTAACCACAAGCCTGCCAAAGTATTTATGGGAGATGTAGGTAGTT
+ERR504787.3.1 M00368:15:000000000-A0HKH:1:3:12724:25677-1 length=100
BCCFDEFFHHHHHJJJJIJJJJJJJJJIJJJFHIJJJIJJJJJIJJJJJJJIJJJIJIIJJJJJFHJJJJIJJHH=CHHFFFFFFEDDEDEEEDCCDCE
```





Failed QC Examples

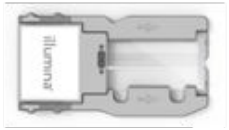
Example 1. Failed per base sequence quality - expired MiSeq kit

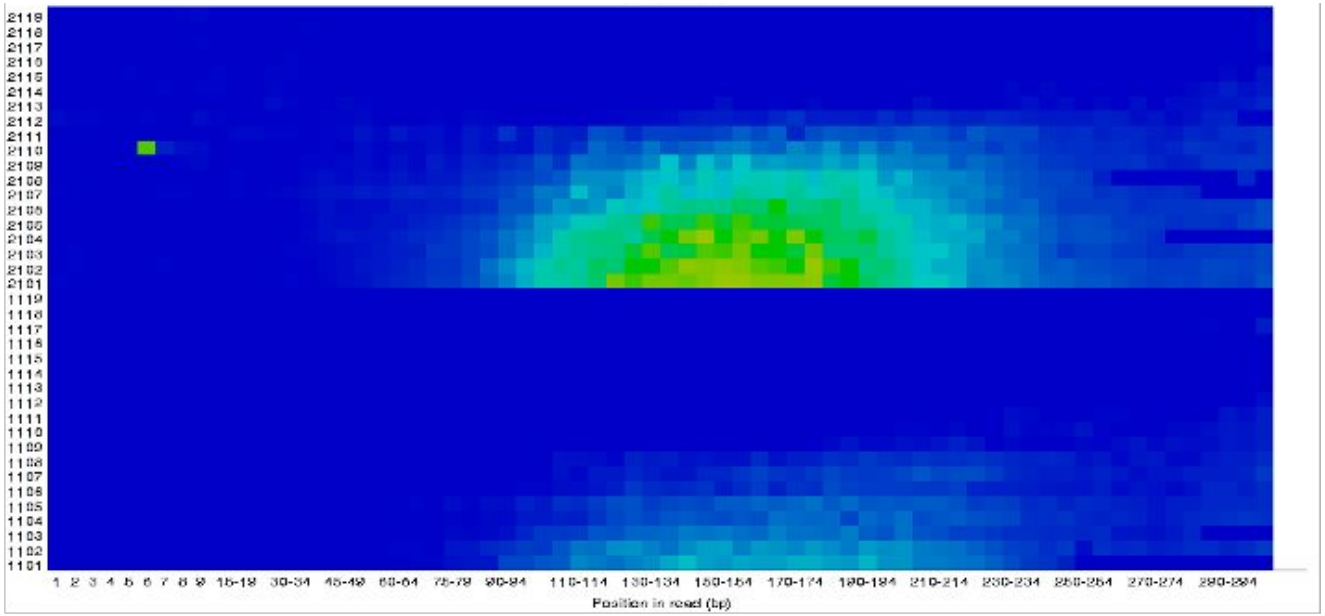


Failed QC Examples

Example 2. Faulty flowcell

MiSeq flowcell

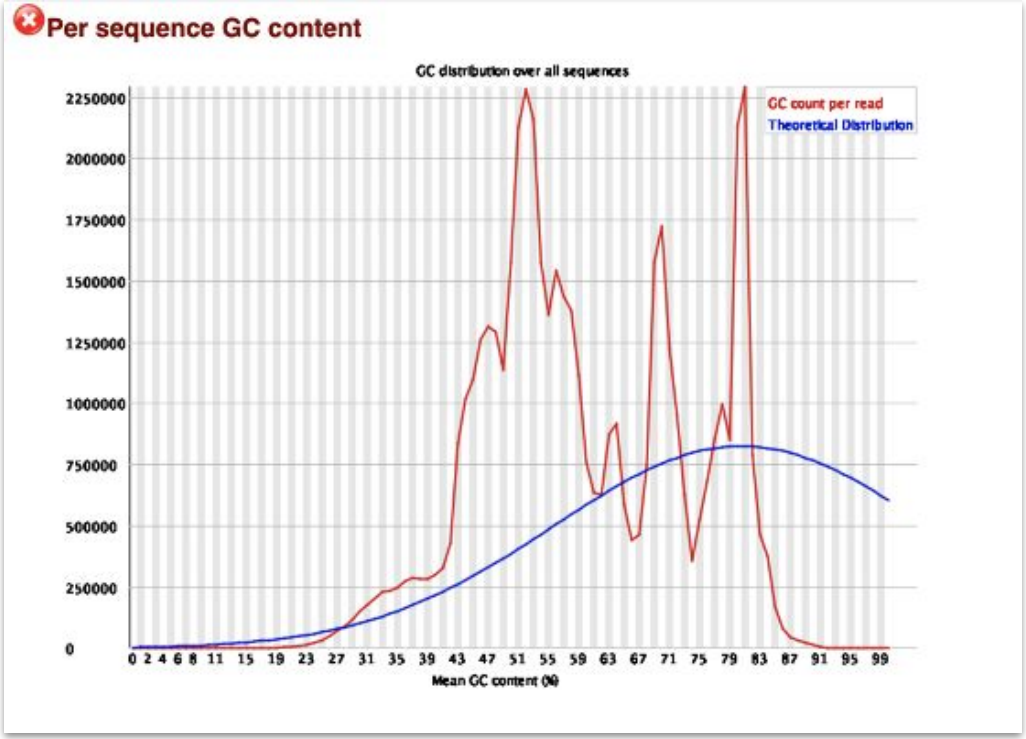




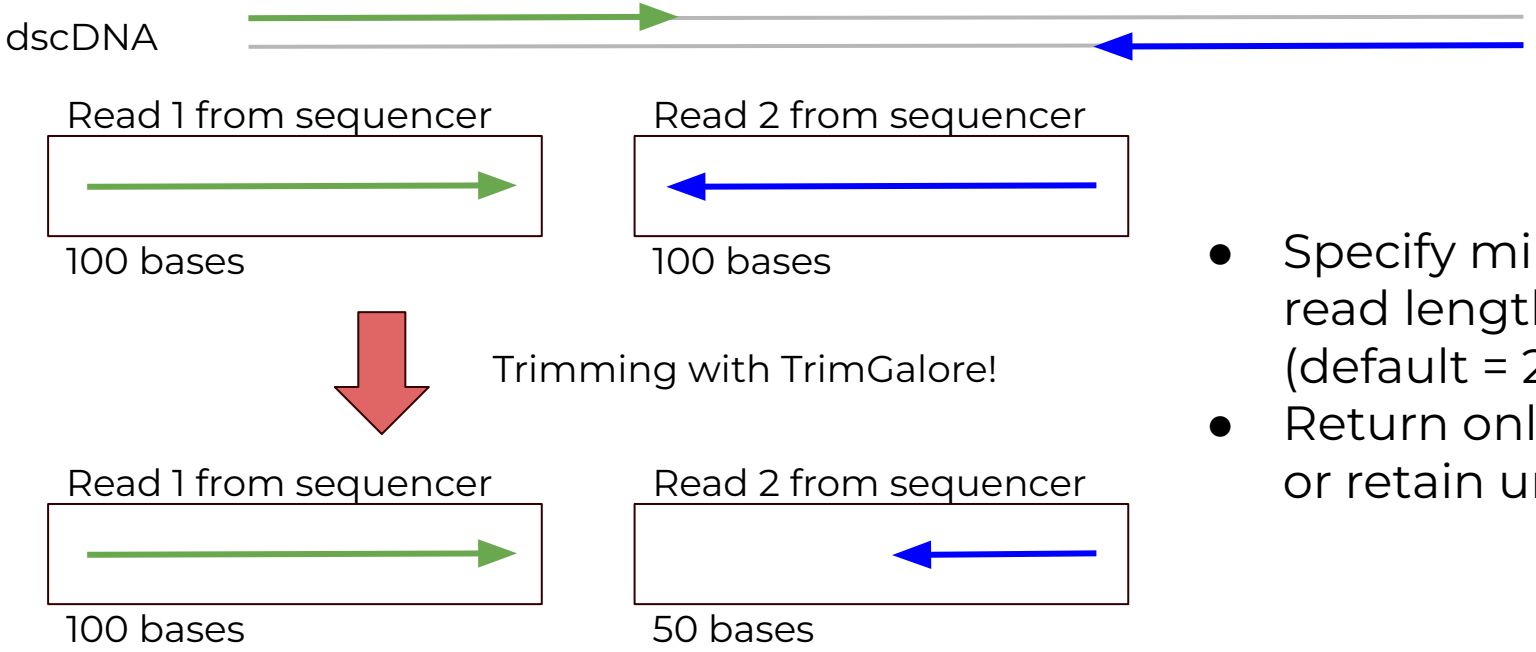
good quality — poor quality

Failed QC Examples

Example 3. Contamination



Library Trimming



- Specify minimum read length (default = 20)
- Return only paired or retain unpaired

Library Trimming

- Remove loaded modules:

```
$ module purge
```

- Find and load the appropriate modules:

```
$ module spider trim_galore
```

```
$ module spider Trim_Galore/0.6.7
```

```
$ module load GCCcore/11.2.0 Trim_Galore/0.6.7
```

- Run Trim_Galore!

```
$ trim_galore --paired --fastqc \
    Control1_R1.fastq.gz Control1_R2.fastq.gz
```

Aligning Reads to a Reference Genome

- Popular splice-aware aligners
  - STAR (now available for GPUs!)
  - HISAT2
- Alignment software needs to have and indexed genome (software specific)
  - Only needs to be done once
  - HPRC maintains indexed genomes for popular aligners
  - Email help@hprc.tamu.edu if you would like us to add another indexed genome

Aligning Reads to a Reference Genome

- Clear any previously loaded modules:

```
$ module purge
```

- Search for and load the appropriate modules:

```
$ module spider hisat
```

```
$ module spider HISAT2/2.2.1
```

```
$ module load GCC/11.3.0 OpenMPI/4.1.4 HISAT2/2.2.1
```

- Get information on how to run the program:

```
$ hisat2 -h
```

Aligning Reads to a Reference Genome

- Align our trimmed reads to the mouse genome:
  - Path to previously indexed genome:

/scratch/data/bio/genome_indexes/ncbi/mm39/hisat2/GCF_000001635.27_GRCm39_genomic

  - Set the path to the indexed genome as a new variable:

```
$ idx_genome=/path/to/genome
```

  - Run the HISAT2 command

```
$ hisat2 -x $idx_genome -p 2 \
    -1 Control1_R1_val_1.fq.gz \
    -2 Control1_R2_val_2.fq.gz \
    -S Control1.sam
```

Aligning Reads to a Reference Genome

```
236499 reads; of these:
  236499 (100.00%) were paired; of these:
    30736 (13.00%) aligned concordantly 0 times
    197200 (83.38%) aligned concordantly exactly 1 time
    8563 (3.62%) aligned concordantly >1 times
    ----
    30736 pairs aligned concordantly 0 times; of these:
    3583 (11.66%) aligned discordantly 1 time
    ----
    27153 pairs aligned 0 times concordantly or discordantly; of these:
    54306 mates make up the pairs; of these:
    30660 (56.46%) aligned 0 times
    21188 (39.02%) aligned exactly 1 time
    2458 (4.53%) aligned >1 times
93.52% overall alignment rate
```

Processing Alignment Files

- Alignment files may need to be modified and/or converted before any downstream analyses:
  - Sorting (name or position/coord)
  - Adding read groups
  - Converting to binary format
- We will use SAMtools to process our alignment file:

```
$ module purge
```

```
$ module spider SAMtools
```

```
$ module spider SAMtools/1.17
```

```
$ module load GCC/12.2.0 SAMtools/1.17
```

Processing Alignment Files

- Run SAMtools sort to convert and sort the alignment file in one step:

```
$ samtools sort --threads 2 \
    -o Control1_sorted.bam  Control1.sam
```

- Index the new bam file:

```
$ samtools index Control1_sorted.bam
```

Generating Count Files

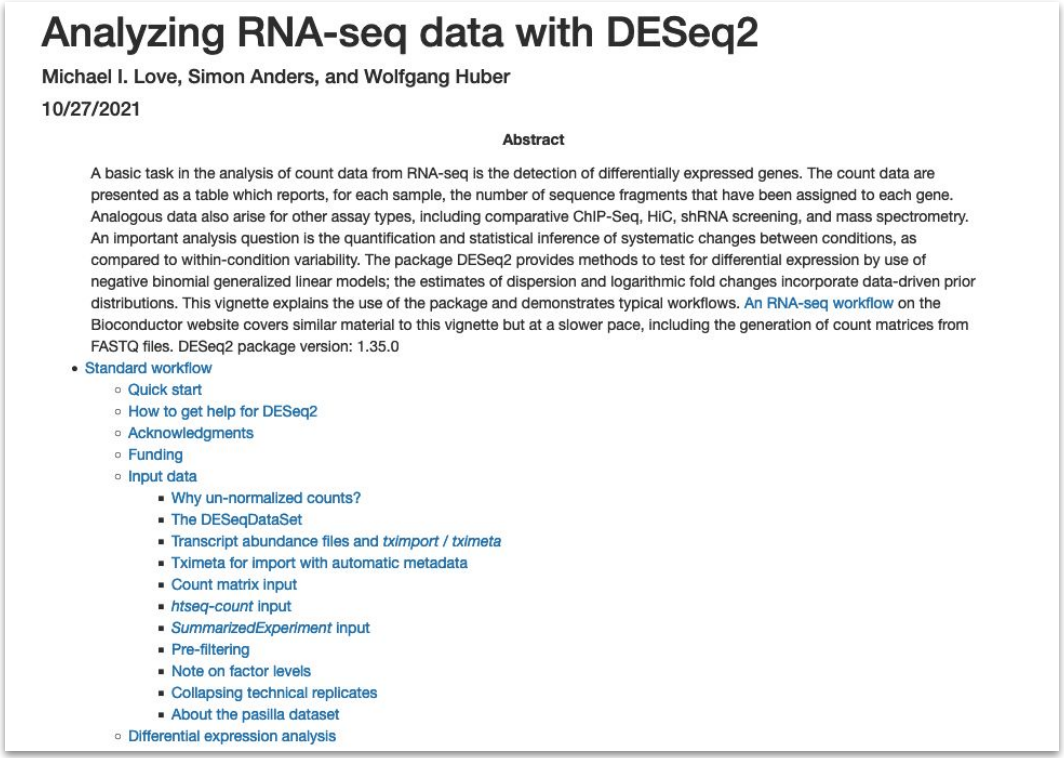
- There are many packages available to generate read counts:
  - featureCounts
  - GenomicRanges (R package)
  - HTSeq
- Load the required modules and produce the count table:

```
$ module purge
```

```
$ module load GCC/11.2.0 OpenMPI/4.1.1 HTSeq/2.0.1
```

```
$ htseq-count -r pos -i gene Control1_sorted.bam \
    GCF_000001635.27_GRCm39_genomic.gff > Control1_counts.txt
```

Differential Expression Analysis with DESeq2



Analyzing RNA-seq data with DESeq2

Michael I. Love, Simon Anders, and Wolfgang Huber

10/27/2021

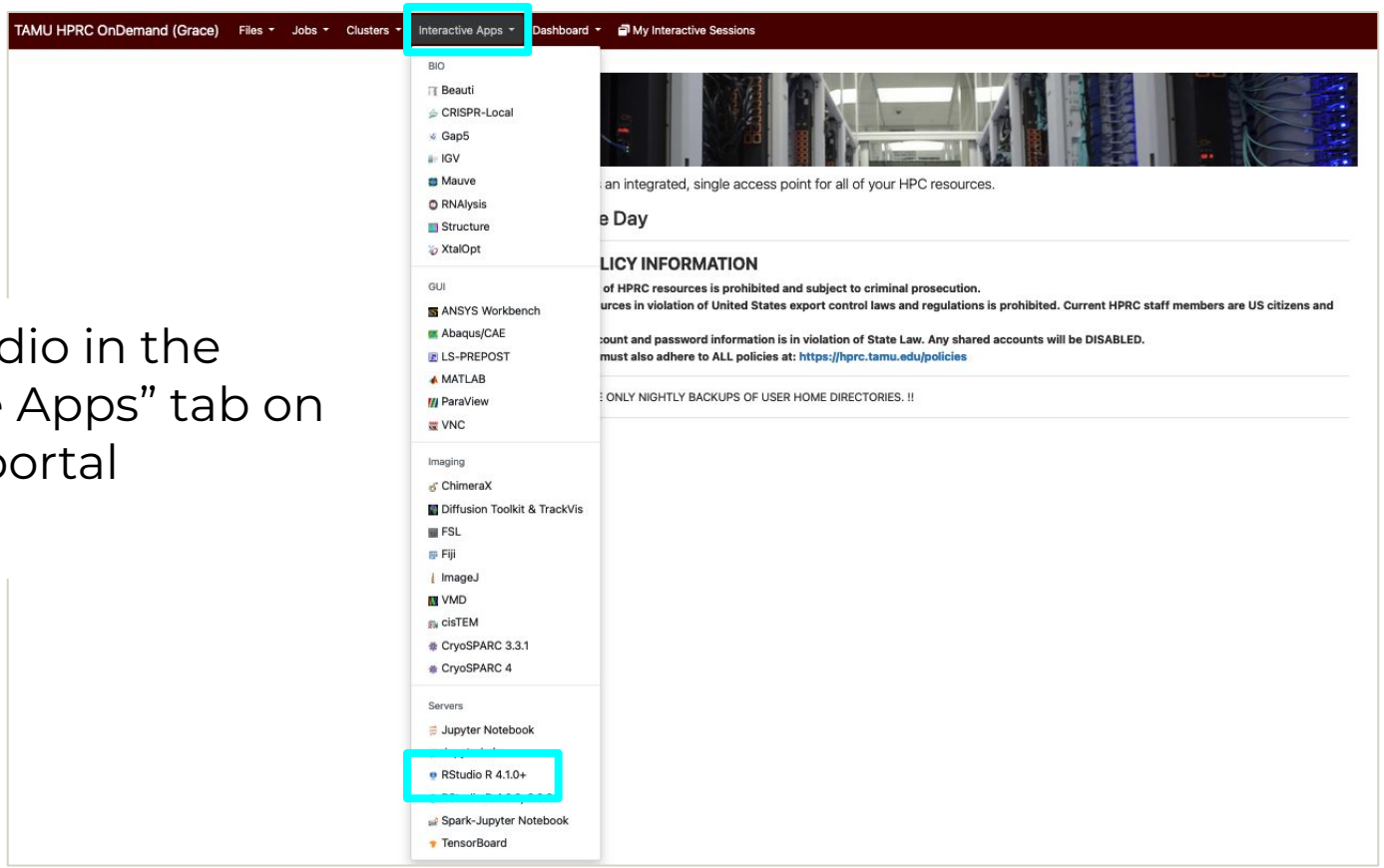
Abstract

A basic task in the analysis of count data from RNA-seq is the detection of differentially expressed genes. The count data are presented as a table which reports, for each sample, the number of sequence fragments that have been assigned to each gene. Analogous data also arise for other assay types, including comparative ChIP-Seq, HiC, shRNA screening, and mass spectrometry. An important analysis question is the quantification and statistical inference of systematic changes between conditions, as compared to within-condition variability. The package DESeq2 provides methods to test for differential expression by use of negative binomial generalized linear models; the estimates of dispersion and logarithmic fold changes incorporate data-driven prior distributions. This vignette explains the use of the package and demonstrates typical workflows. An RNA-seq workflow on the Bioconductor website covers similar material to this vignette but at a slower pace, including the generation of count matrices from FASTQ files. DESeq2 package version: 1.35.0

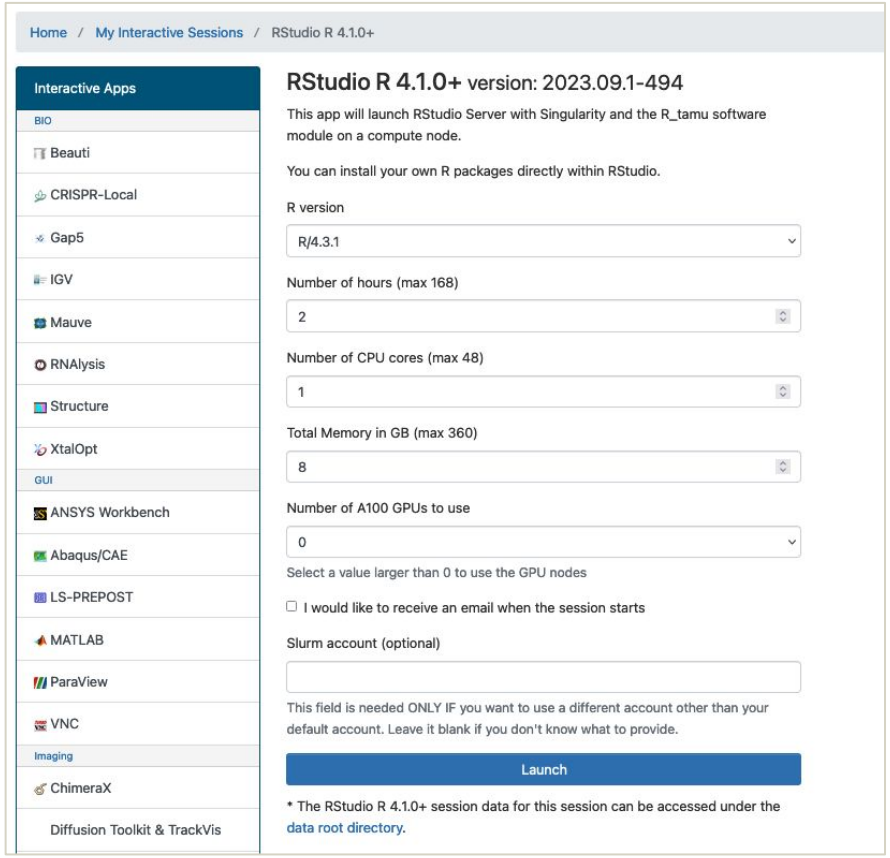
http://bioconductor.org/packages/devel/bioc/vignettes/DESeq2/inst/doc/DESeq2.html

RStudio on Grace

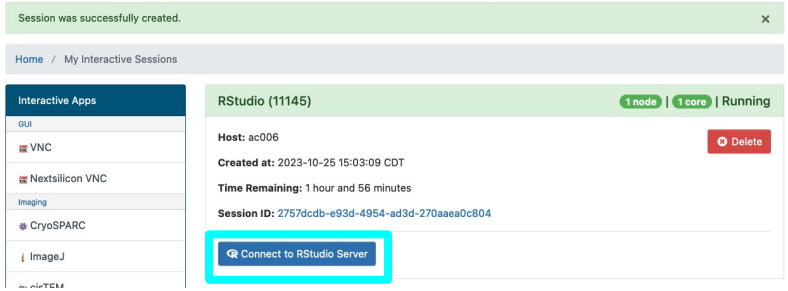
- Open RStudio in the “Interactive Apps” tab on the Grace portal



RStudio on Grace



- Set the number of hours to 2
- Set the number of cores to 1
- Set the Total GB memory to 8
- Click Launch Button
- Wait for the session to start
- Click “Connect to RStudio Server



Differential Expression Analysis

Open a new R script and set your working directory

```
setwd("/scratch/user/username/RNA_class/counts")
```

- Let's look at the contents of the directory and the sample table (in the console):

```
> list.files()
```

```
> system("cat sampleTable.csv")
```

Differential Expression Analysis

- Load the required packages:

```
library(ggplot2)
library(pheatmap)
library(DESeq2)
library(EnhancedVolcano)
```

- Highlight this section of code in the script and click “Run”

Differential Expression Analysis

- Read in the sample table and reformat it:

```
sampleTable = read.csv("sampleTable.csv", header=TRUE)
sampleTable = as.data.frame(sampleTable)
sampleTable$condition = factor(sampleTable$condition)
sampleTable
```

- Output:

```
> sampleTable
   sampleName            fileName      condition
1    Control1 Control1_counts.txt        Control
2    Control2 Control2_counts.txt        Control
3    Control3 Control3_counts.txt        Control
4    Control4 Control4_counts.txt        Control
5    Control5 Control5_counts.txt        Control
6        NAD1     NAD1_counts.txt NAD_supplement
7        NAD2     NAD2_counts.txt NAD_supplement
8        NAD3     NAD3_counts.txt NAD_supplement
9        NAD4     NAD4_counts.txt NAD_supplement
10       NAD5     NAD5_counts.txt NAD_supplement
>
```

Differential Expression Analysis

- Create the dds object

```
dds = DESeqDataSetFromHTSeqCount(sampleTable = sampleTable,
                                 directory = ".",
                                 design = ~ condition)
dds
```

- Output:

```
> dds
class: DESeqDataSet
dim: 46316 10
metadata(1): version
assays(1): counts
rownames(46316): 0610005C13Rik 0610006L08Rik ... n-TYgta9 n-Tcgca44
rowData names(0):
colnames(10): Control1 Control2 ... NAD4 NAD5
colData names(1): condition
>
```

Differential Expression Analysis

- Filter out genes with low read counts:

```
keep <- rowSums(counts(dds)) >= 10
dds <- dds[keep,]
```

- Run the differential expression analysis:

```
dds <- DESeq(dds)
res <- results(dds)
res
```

Differential Expression Analysis

DESeq Results Explained:

```
> res
log2 fold change (MLE): condition NAD supplement vs Control
Wald test p-value: condition NAD supplement vs Control
DataFrame with 46316 rows and 6 columns
                     baseMean     log2FoldChange             lfcSE                stat              pvalue               padj
                    <numeric>          <numeric>         <numeric>           <numeric>           <numeric>          <numeric>
0610005C13Rik  5.99463012842517  0.847110388480526   1.0536757176372  0.803957398183298   0.4214215791485 0.626767086797856
0610006L08Rik 0.595406936513421  -1.33338402962542  2.80181545117752 -0.475900020133387  0.634145607845708                NA
0610009B22Rik  229.572854136365  -0.46059738889209 0.272726760296267   -1.688860265827 0.0912462113650002 0.227131423458176
0610009E02Rik  52.7148015454124  -1.18516447577791 0.483501805720158  -2.45121003015211 0.0142376849790884 0.058533268492583
0610009L18Rik  5.27096640148362  0.500548878654153   1.0060551707554  0.497536211933899  0.618810973397835 0.779869206330055
```

Differential Expression Analysis

DESeq Results Explained:

```
> res
log2 fold change (MLE): condition NAD supplement vs Control
Wald test p-value: condition NAD supplement vs Control
DataFrame with 46316 rows and 6 columns
                  baseMean     log2FoldChange             lfcSE                stat              pvalue               padj
                 <numeric>          <numeric>         <numeric>           <numeric>           <numeric>          <numeric>
0610005C13Rik  5.99463012842517  0.847110388480526   1.0536757176372  0.803957398183298   0.4214215791485  0.626767086797856
0610006L08Rik 0.595406936513421  -1.33338402962542  2.80181545117752 -0.475900020133387  0.634145607845708                 NA
0610009B22Rik  229.572854136365  -0.46059738889209 0.272726760296267   -1.688860265827 0.0912462113650002 0.227131423458176
0610009E02Rik  52.7148015454124  -1.18516447577791 0.483501805720158   -2.45121003015211 0.0142376849790884 0.058533268492583
0610009L18Rik  5.27096640148362  0.500548878654153   1.0060551707554  0.497536211933899  0.618810973397835 0.779869206330055
```

Mean of normalized counts for all samples

Differential Expression Analysis

DESeq Results Explained:

```
> res
log2 fold change (MLE): condition NAD supplement vs Control
Wald test p-value: condition NAD supplement vs Control
DataFrame with 46316 rows and 6 columns
                     baseMean     log2FoldChange             lfcSE                stat              pvalue               padj
                    <numeric>          <numeric>         <numeric>           <numeric>           <numeric>          <numeric>
0610005C13Rik 5.99463012842517  0.847110388480526  1.0536757176372  0.803957398183298   0.4214215791485 0.626767086797856
0610006L08Rik 0.595406936513421 -1.33338402962542 2.80181545117752 -0.475900020133387 0.634145607845708                NA
0610009B22Rik  229.572854136365 -0.46059738889209 0.272726760296267   -1.688860265827 0.0912462113650002 0.227131423458176
0610009E02Rik  52.7148015454124 -1.18516447577791 0.483501805720158  -2.45121003015211 0.0142376849790884 0.058533268492583
0610009L18Rik  5.27096640148362  0.500548878654153  1.0060551707554  0.497536211933899  0.618810973397835 0.779869206330055
```

Log2 fold change: NAD supplement vs Control

Differential Expression Analysis

DESeq Results Explained:

```
> res
log2 fold change (MLE): condition NAD supplement vs Control
Wald test p-value: condition NAD supplement vs Control
DataFrame with 46316 rows and 6 columns
                     baseMean     log2FoldChange             lfcSE                stat              pvalue               padj
                    <numeric>          <numeric>         <numeric>           <numeric>           <numeric>          <numeric>
0610005C13Rik  5.99463012842517  0.847110388480526  1.0536757176372  0.803957398183298   0.4214215791485 0.626767086797856
0610006L08Rik 0.595406936513421  -1.33338402962542  2.80181545117752 -0.475900020133387  0.634145607845708                NA
0610009B22Rik  229.572854136365  -0.46059738889209 0.272726760296267   -1.688860265827 0.0912462113650002 0.227131423458176
0610009E02Rik  52.7148015454124  -1.18516447577791 0.483501805720158  -2.45121003015211 0.0142376849790884 0.058533268492583
0610009L18Rik  5.27096640148362  0.500548878654153  1.0060551707554  0.497536211933899  0.618810973397835 0.779869206330055
```

Log fold change standard error

Differential Expression Analysis

DESeq Results Explained:

```
> res
log2 fold change (MLE): condition NAD supplement vs Control
Wald test p-value: condition NAD supplement vs Control
DataFrame with 46316 rows and 6 columns
                     baseMean     log2FoldChange             lfcSE                stat              pvalue               padj
                    <numeric>          <numeric>         <numeric>           <numeric>           <numeric>          <numeric>
0610005C13Rik  5.99463012842517  0.847110388480526   1.0536757176372  0.803957398183298   0.4214215791485 0.626767086797856
0610006L08Rik 0.595406936513421  -1.33338402962542  2.80181545117752 -0.475900020133387  0.634145607845708                NA
0610009B22Rik  229.572854136365  -0.46059738889209 0.272726760296267   -1.688860265827 0.0912462113650002 0.227131423458176
0610009E02Rik  52.7148015454124  -1.18516447577791 0.483501805720158  -2.45121003015211 0.0142376849790884 0.058533268492583
0610009L18Rik  5.27096640148362  0.500548878654153   1.0060551707554  0.497536211933899  0.618810973397835 0.779869206330055
```

Wald statistic: NAD supplement vs Control

Differential Expression Analysis

DESeq Results Explained:

```
> res
log2 fold change (MLE): condition NAD supplement vs Control
Wald test p-value: condition NAD supplement vs Control
DataFrame with 46316 rows and 6 columns
                     baseMean     log2FoldChange             lfcSE                stat              pvalue              padj
                    <numeric>          <numeric>         <numeric>           <numeric>           <numeric>         <numeric>
0610005C13Rik  5.99463012842517  0.847110388480526   1.0536757176372  0.803957398183298   0.4214215791485 0.626767086797856
0610006L08Rik 0.595406936513421  -1.33338402962542  2.80181545117752 -0.475900020133387  0.634145607845708                NA
0610009B22Rik  229.572854136365  -0.46059738889209 0.272726760296267   -1.688860265827 0.0912462113650002 0.227131423458176
0610009E02Rik  52.7148015454124  -1.18516447577791 0.483501805720158  -2.45121003015211 0.0142376849790884 0.058533268492583
0610009L18Rik  5.27096640148362  0.500548878654153   1.0060551707554  0.497536211933899  0.618810973397835 0.779869206330055
```

Wald test p value
(unadjusted)

Differential Expression Analysis

DESeq Results Explained:

```
> res
log2 fold change (MLE): condition NAD supplement vs Control
Wald test p-value: condition NAD supplement vs Control
DataFrame with 46316 rows and 6 columns
                     baseMean     log2FoldChange             lfcSE                stat              pvalue               padj
                    <numeric>          <numeric>         <numeric>           <numeric>           <numeric>          <numeric>
0610005C13Rik  5.99463012842517  0.847110388480526   1.0536757176372  0.803957398183298   0.4214215791485 0.626767086797856
0610006L08Rik 0.595406936513421  -1.33338402962542  2.80181545117752 -0.475900020133387  0.634145607845708                NA
0610009B22Rik  229.572854136365  -0.46059738889209 0.272726760296267   -1.688860265827 0.0912462113650002 0.227131423458176
0610009E02Rik  52.7148015454124  -1.18516447577791 0.483501805720158  -2.45121003015211 0.0142376849790884 0.058533268492583
0610009L18Rik  5.27096640148362  0.500548878654153   1.0060551707554  0.497536211933899  0.618810973397835 0.779869206330055
```

BH corrected p-values (corrected for multiple testing)

Differential Expression Analysis

- How many genes are differentially expressed?

```
sum(res$padj <= 0.05, na.rm = TRUE)
```

- Collect all the DEGs and write them to file:

```
sigGenes <- res[ which(res$padj < 0.05), ]
sigGenes
write.csv(sigGenes,
          "Differentrially_Expressed.csv",
          row.names = TRUE)
```

PCA (Principal Component Analysis)

- Log transform the results and calculate the row variance

```
logTran <- rlog(dds)
rv <- rowVars(assay(logTran))
```

- Create a list of genes with the greatest variance:

```
select <- order(rv, decreasing = TRUE)[1:100]
```

PCA plot

- Run the principal component analysis (PCA)

```
PCA <- prcomp(t(assay(logTran)[select, ]), scale = FALSE)
summary(PCA)
```

- Output:

```
> summary(PCA)
Importance of components:
                          PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9      PC10
Standard deviation     13.1129 2.50384 1.94479 1.45805 1.42247 1.24092 1.08253 0.52065 0.38289 3.066e-15
Proportion of Variance  0.9084 0.03312 0.01998 0.01123 0.01069 0.00814 0.00619 0.00143 0.00077 0.000e+00
Cumulative Proportion   0.9084 0.94156 0.96154 0.97278 0.98347 0.99160 0.99779 0.99923 1.00000 1.000e+00
>
```

PCA plot

- Set up the PCA for ggplot2

```
percentVar <- round(100*PCA$sdev^2/sum(PCA$sdev^2),1)
ggPCA_out <- as.data.frame(PCA$x)
ggPCA_out <- cbind(ggPCA_out, sampleTable)
head(ggPCA_out)
```

- Output:

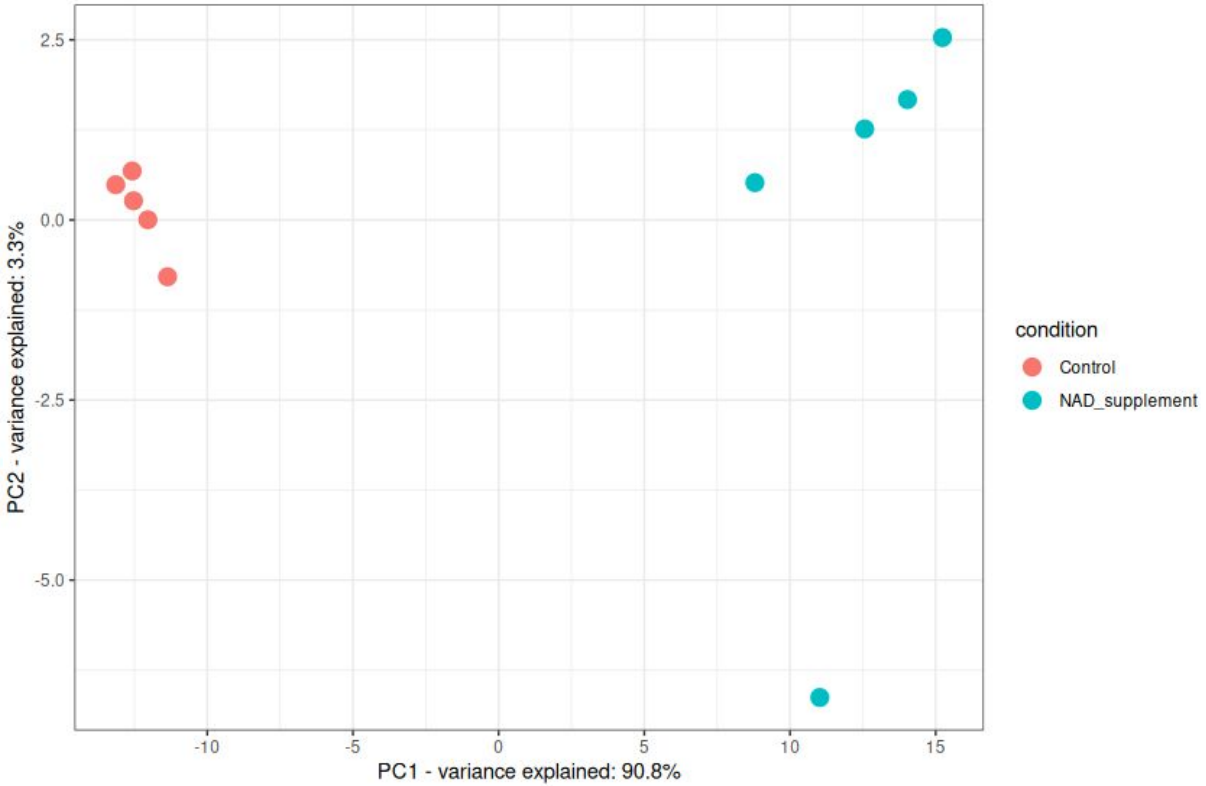
```
> head(ggPCA_out)
                PC1         PC2        PC3        PC4        PC5         PC6        PC7        PC8         PC9
Control1 -12.576882  0.679757091  1.4677571  1.4408177 -0.9772907 -2.58170153 -0.8901816  0.12856743  0.057730319
Control2 -11.362119 -0.789437801 -4.1149258  0.5846590 -1.6247299  0.15350772  1.1461662 -0.21539578  0.001565017
Control3 -12.038043  0.002152241  0.5811305 -3.0992919  1.4657618 -0.97863917  1.0515499 -0.04523746 -0.046467189
Control4 -13.139919  0.487982477  0.6723550  2.2531953  2.6066210  1.29314839  0.3152618 -0.07647994  0.001933003
Control5 -12.530993  0.265744874  1.7077315 -1.1646465 -1.8470308  2.10751112 -1.1715371  0.07993858 -0.013573324
NAD1       8.795471  0.517771986 -2.7475597 -0.6142266  1.1887028 -0.04330035 -1.5880439  0.71154077  0.323683075
                PC10 sampleName            fileName      condition
Control1 3.175046e-15   Control1 Control1_counts.txt        Control
Control2 2.950899e-15   Control2 Control2_counts.txt        Control
Control3 2.730071e-15   Control3 Control3_counts.txt        Control
Control4 3.300727e-15   Control4 Control4_counts.txt        Control
Control5 2.949020e-15   Control5 Control5_counts.txt        Control
NAD1     2.826141e-15       NAD1     NAD1_counts.txt NAD_supplement
>
```

PCA plot

- Plot the PCA

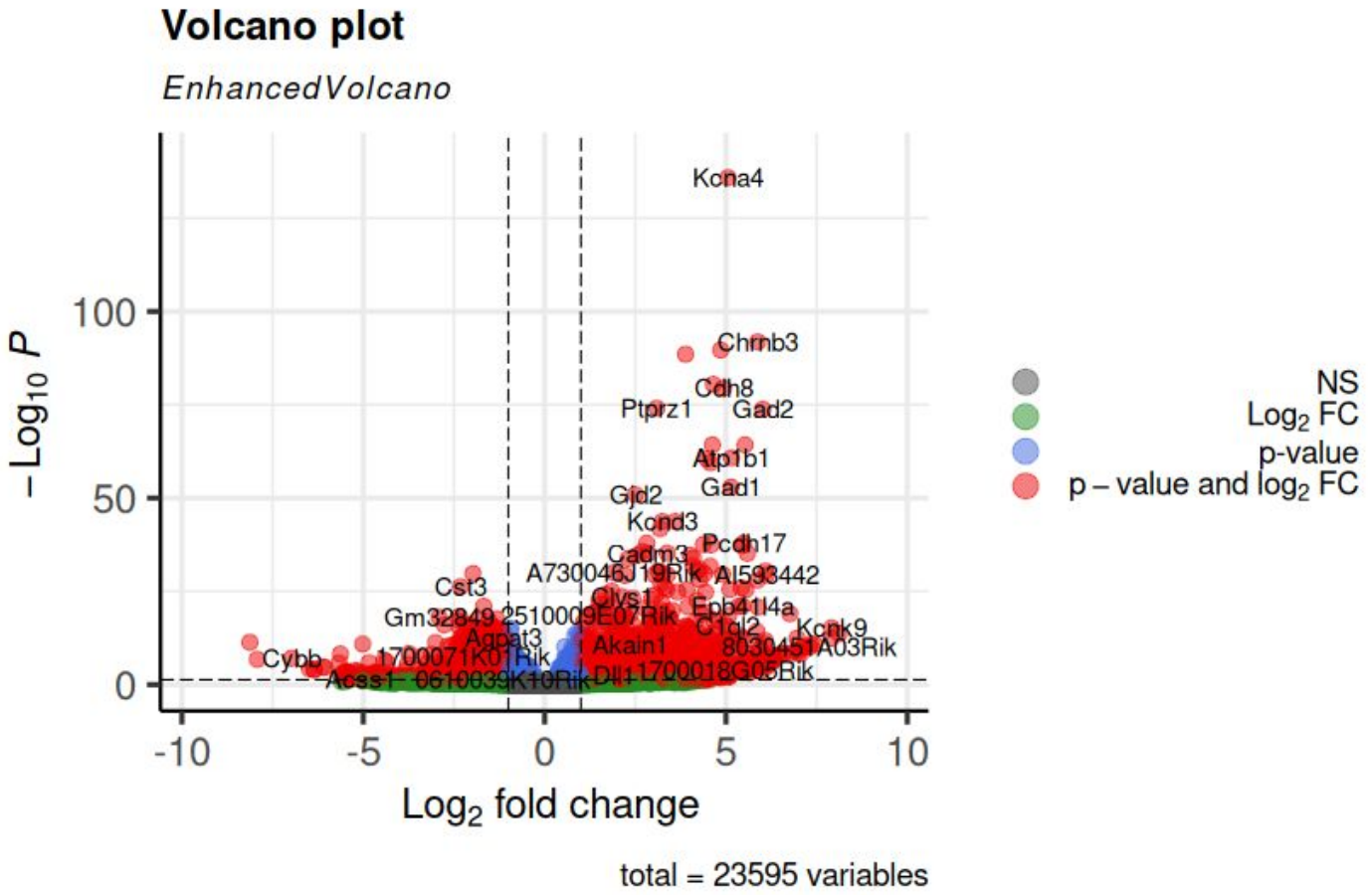
```
ggplot(ggPCA_out, aes(x=PC1,y=PC2,color=condition)) +
    geom_point(size=4) +
    labs(x = paste0("PC1 - variance explained: ", percentVar[1], "%"),
        y = paste0("PC2 - variance explained: ", percentVar[2], "%")) +
    theme_bw()
```

PCA plot


2.5
0.0
-2.5
-5.0
PC2 - variance explained: 3.3%
-10
-5
0
5
10
15
PC1 - variance explained: 90.8%
condition
Control
NAD_supplement

Volcano Plot

```
EnhancedVolcano(res,
                lab = rownames(res),
                x = 'log2FoldChange',
                y = 'padj',
                pCutoff = 0.05,
                FCcutoff = 1.0,
                pointSize = 3.0,
                labSize = 4.0,
                colAlpha = 1/2,
                drawConnectors = FALSE,
                legendPosition = "right")
```


Volcano plot
EnhancedVolcano
Kcna4
Chrnb3
Cdh8
Ptprz1
Gad2
Atp1b1
Gad1
Gjd2
Kcnd3
Pcdh17
Cadm3
A730046J19Rik
AI593442
Cst3
Clys1
Epb41l4a
Gm32849
2510009E07Rik
C1ql2
Kcnk9
Agpat3
Akain1
8030451A03Rik
Cybb
1700071K01Rik
Acss1
0610039K10Rik
Dll1
1700018G05Rik
100
50
0
-10
-5
0
5
10
$-\text{Log}_{10}\ P$
Log_2 fold change
NS
Log_2 FC
p-value
p – value and log_2 FC
total = 23595 variables

Heatmap

- Reorder the results based on adjusted p-values
- Assign genes with adjusted p-values below 0.05 and absolute log2 fold changes >= 6.5 to the variable ‘sig’

```
resorted_deresults <- res[order(res$padj),]
sig <- resorted_deresults[!is.na(resorted_deresults$padj) &
                            resorted_deresults$padj < 0.05 &
                            abs(resorted_deresults$log2FoldChange) >=
6.5,]
```

Heatmap

- Assign the gene names from ‘sig’ to a new variable named ‘selected’
- We will use the list of gene names for the heatmap

```
selected <- rownames(sig)
selected
```

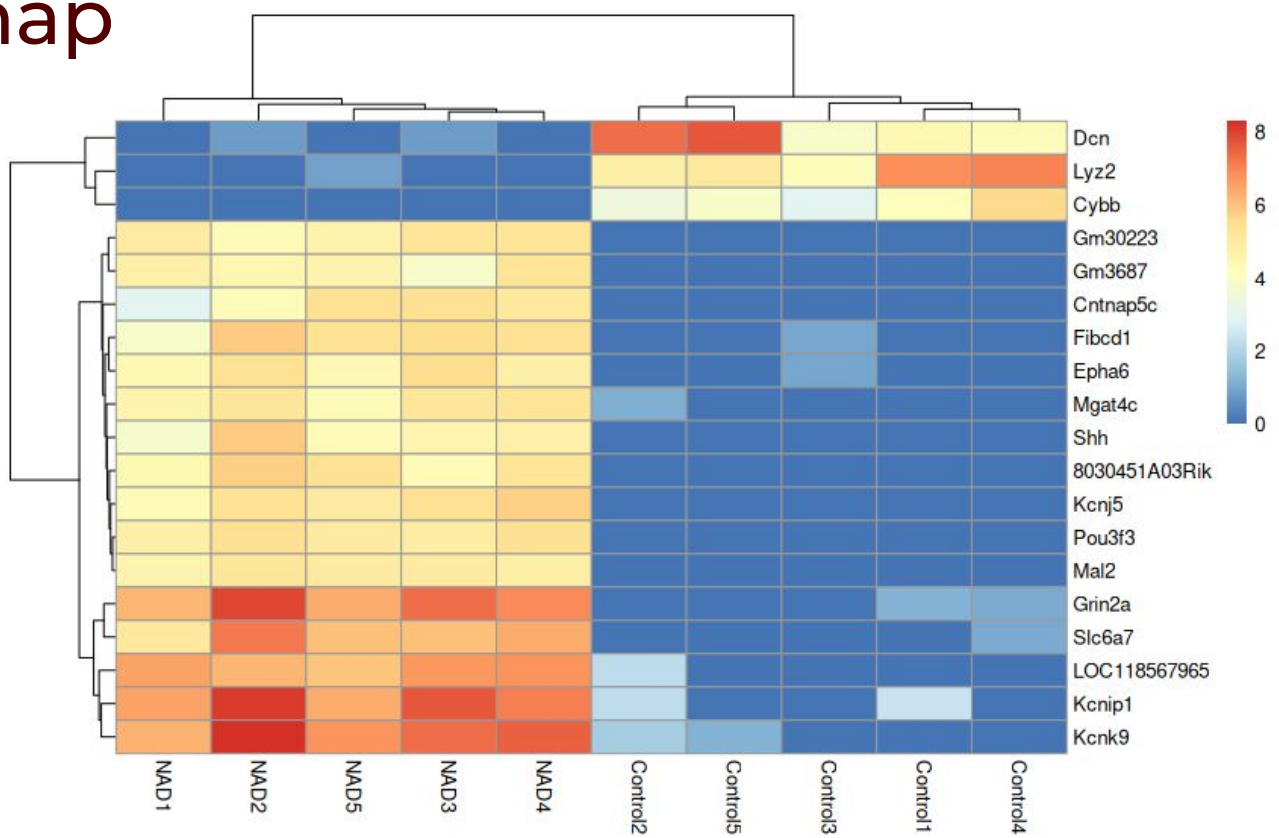
```
> selected
 [1] "Kcnip1"       "Kcnk9"       "Grin2a"       "Slc6a7"         "LOC118567965"  "Lyz2"
 [7] "Pou3f3"       "Kcnj5"       "Mal2"         "8030451A03Rik"  "Gm30223"       "Fibcd1"
[13] "Gm3687"       "Shh"         "Mgat4c"       "Cntnap5c"       "Epha6"         "Cybb"
[19] "Dcn"
>
```

Heatmap

- We need to normalize the data
- Then we can create a heatmap using the pheatmap package

```
transformed_readcounts <- normTransform(dds)
pheatmap(assay(transformed_readcounts)[selected,],
             cluster_rows = TRUE, show_rownames = TRUE,
             cluster_cols = TRUE,
             labels_col = colData(dds)$sampleName)
```

Heatmap


Dcn
Lyz2
Cybb
Gm30223
Gm3687
Cntnap5c
Fibcd1
Epha6
Mgat4c
Shh
8030451A03Rik
Kcnj5
Pou3f3
Mal2
Grin2a
Slc6a7
LOC118567965
Kcnip1
Kcnk9
8
6
4
2
0
NAD1
NAD2
NAD5
NAD3
NAD4
Control2
Control5
Control3
Control1
Control4

HIGH PERFORMANCE RESEARCH COMPUTING
TEXAS A&M UNIVERSITY

https://hprc.tamu.edu

HPRC Helpdesk:

help@hprc.tamu.edu
Phone: 979-845-0219

Help us help you. Please include details in your request for support, such as, Cluster (Faster, Grace, Terra, ViDaL), NetID (UserID), Job information (Job ID(s), Location of your jobfile, input/output files, Application, Module(s) loaded, Error messages, etc), and Steps you have taken, so we can reproduce the problem.