RNA Sequencing Overview

Learning Objectives:

Explain what are the main benefits RNA-seq data can provide for researchers.

List commonly used software and its applications.

Describe different types of RNA-seq libraries.

Describe important considerations in experimental design.

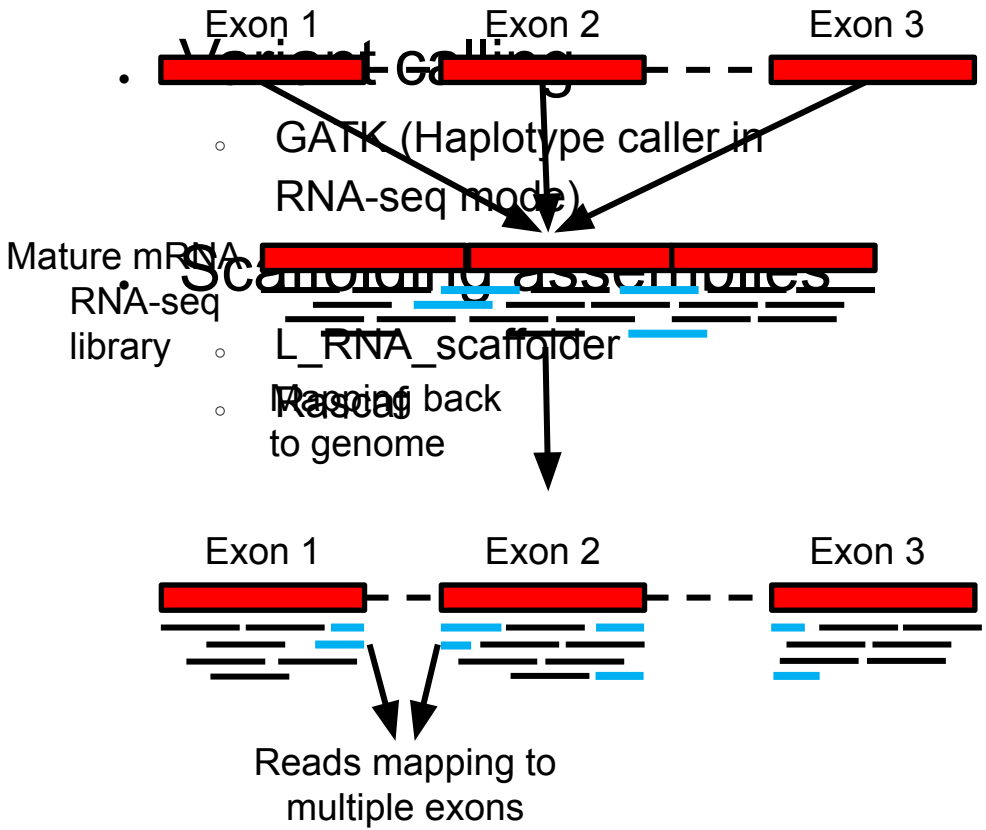
Project sponsored by the Texas Higher Education Coordinating Board

What Does RNA-seq Data Provide?

- Annotate genomes/transcripts or assembly transcriptomes
- Discover nucleotide variants
- Scaffold genome assemblies
- Measure gene expression and detect differences in expression between groups

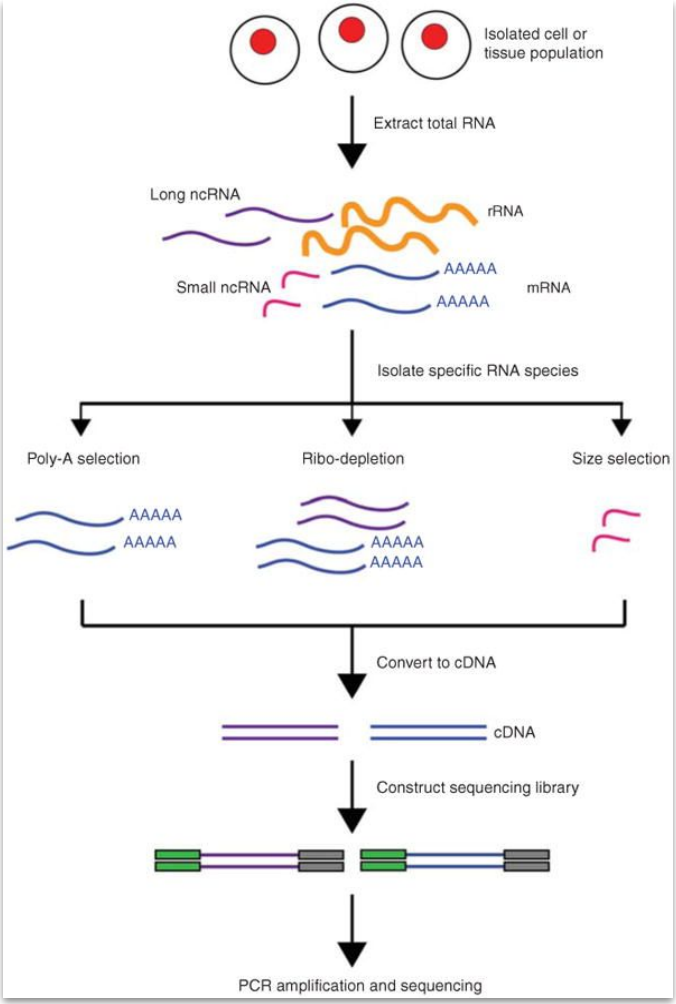
RNA-seq Applications

- Transcriptome assembly
  - de novo - Trinity, Oases, SOAPdenovo-Trans
  - Reference-based - Trinity, StringTie, Cufflinks
- Splice-aware alignment
  - HISAT2
  - STAR
  - Tophat
- File conversion/handling
  - SAMtools
  - Picard
- Variant calling
  - GATK (Haplotype caller in RNA-seq mode)
- Scaffolding assemblies
  - L_RNA_scaffolder
  - Rascaf



RNA Sequencing

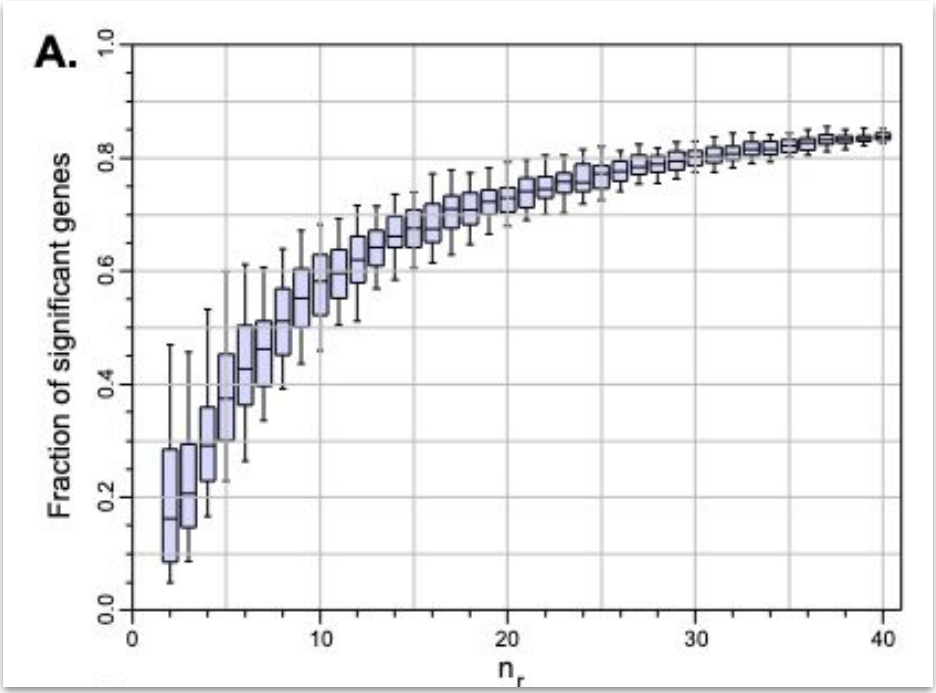
- Poly-A selection
  - Enriches for mRNA
- Ribosomal depletion
  - Removes rRNA
  - Leaves mRNA, lncRNA, and pre-RNA
- Size selection
  - Used for smRNA



Kukurba and Montgomery, 2016

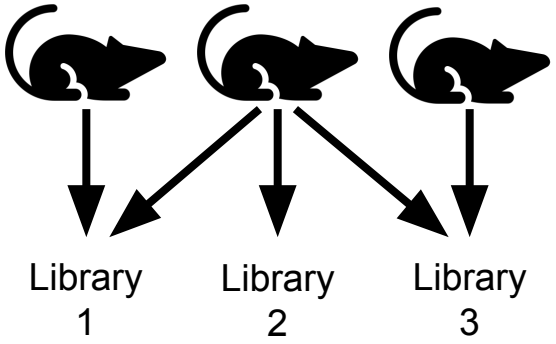
Experimental Design (Differential Expression)

- Sequencing Depth
  - Minimum 30 million aligned reads per replicate (ENCODE)
  - 30-60 million reads per sample (Illumina)
- Replicate Number
  - 3 replicates per condition minimum (will likely recover 20-40% of DEG)
  - Schurch et al. (2016) - 6 replicates per condition minimum, recommended 12 to capture all DEG



Experimental Design (Differential Expression)

- Biological replicates
  - Independent samples from different populations or individuals
- Technical replicates
  - Multiple preparations/libraries from the same individual
- 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



High Performance Research Computing
DIVISION OF RESEARCH

TEXAS A&M UNIVERSITY
Engineering Studio for Advanced Instruction & Learning

TEES EDGE

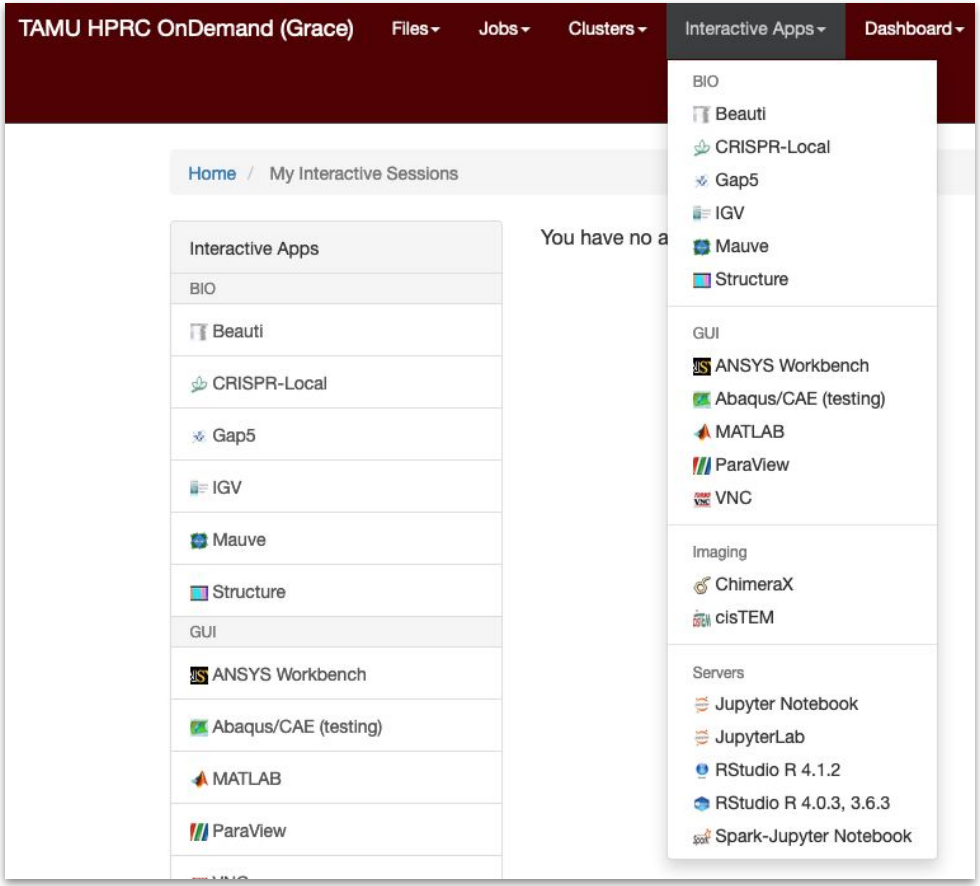
How to Access the Grace Cluster

Identify where to find NGS tools, find software for specific applications, and request the installation of additional NGS tools.

Project sponsored by the Texas Higher Education Coordinating Board

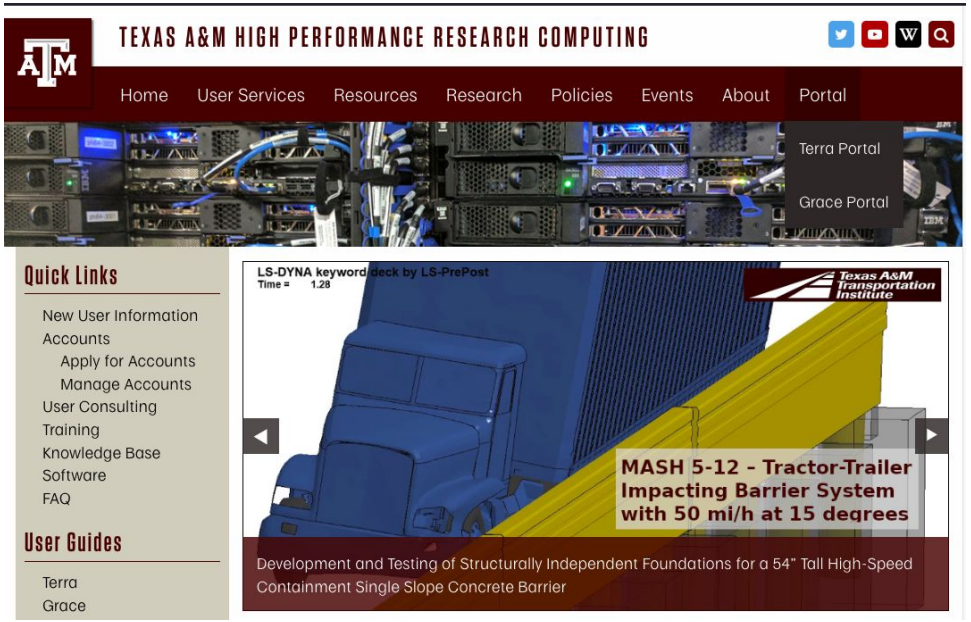
HPRC Portal

- Access is web browser based
- All HPRC software tools are available either as a GUI or via UNIX command line
- Best for GUI applications
  - RStudio
  - IGV



Demo

- Accessing the Grace Cluster through the HPRC Portal



https://www.youtube.com/watch?v=dqa2ZzsEmQs&list=PLHR4HLly3i4aJJDxKTZIpxyJG6uSqgAgd

How to Access the Grace Cluster

- Unix Command Line
  - Working knowledge of Unix and Slurm
  - Bioinformatics template scripts available
  - Not well-suited for GUI applications
  - Use Terminal Application on Max or Linux machines
  - Windows machines require and SSH client

```
 ==========================================================================
 |        Texas A&M University High Performance Research Computing        |
 |                                                                         |
 | Website:             https://hprc.tamu.edu                              |
 | Consulting:          help@hprc.tamu.edu (preferred) or (979) 845-0219   |
 | Grace Documentation: https://hprc.tamu.edu/wiki/Grace                   |
 | Terra Documentation: https://hprc.tamu.edu/wiki/Terra                   |
 | YouTube Channel:     https://www.youtube.com/texasamhprc                |
 ==========================================================================

 ***************************************************************************
*                  === IMPORTANT POLICY INFORMATION ===                    *
* - Unauthorized use of HPRC resources is prohibited and subject to        *
*   criminal prosecution.                                                  *
* - Use of HPRC resources in violation of United States export control     *
*   laws and regulations is prohibited. Current HPRC staff members are     *
*   US citizens and legal residents.                                       *
* - Sharing HPRC account and password information is in violation of       *
*   Texas State Law. Any shared accounts will be DISABLED.                 *
* - Authorized users must also adhere to ALL policies at:                  *
*                      https://hprc.tamu.edu/policies/                     *
 ***************************************************************************

   !! WARNING: THERE ARE ONLY NIGHTLY BACKUPS OF USER HOME DIRECTORIES. !!

      Please restrict usage to 8 CORES across ALL login nodes.
      Users found in violation of this policy will be SUSPENDED.

            To see these messages again, run the motd command.
```

Demo

- Accessing the Grace Cluster through the Unix command line
- Using the Terminal application on Mac/Linux machines
  - https://www.youtube.com/watch?v=KjHwfZI_ej4&list=PLHR4HLly3i4YrkNWcUE77t8i-AkwN5AN8&index=4
- Using MobaXTerm on Windows machines
  - https://www.youtube.com/watch?v=PXIGhqLJP3g&list=PLHR4HLly3i4YrkNWcUE77t8i-AkwN5AN8&index=3

Login to Grace and Download the Example Data

- Login through the portal or via command line
  - Make a new directory for the course

```
mkdir $SCRATCH/RNA_class
```

  - Change your working directory to the one you just created

```
cd $SCRATCH/RNA_class
```

  - Copy the example data for the course

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

Where can you find NGS tools?

Learning Objectives

Know where to find NGS tools, find software for particular applications, and request the installation of additional NGS tools

Project sponsored by the Texas Higher Education Coordinating Board

Where to Find the NGS Tools

- TAMU HPRC Documentation (https://hprc.tamu.edu/wiki/Bioinformatics)

- Use ‘module’ to search the cluster on the command line
  - module avail
  - module spider
  - module key

- If you would like a program installed on Grace, send an email with the URL link to help@hprc.tamu.edu

GCATemplates

Learning Objective:

Access and edit template job scripts to run jobs on the Grace cluster.

Project sponsored by the Texas Higher Education Coordinating Board

GCATemplates Use

- Jobs on Grace are limited to:
  - 60 minutes and
  - a max of 8 cores
- Longer jobs require more time or power require:
  - Submission of the job script to the scheduler
    - Includes variables like:
      - Amount of memory required
      - Number of CPU threads to use
      - Which modules need to be loaded
      - Command to run the software
- Template Job Scripts information:
  - https://hprc.tamu.edu/wiki/SW:GCATemplates

https://hprc.tamu.edu/wiki/Bioinformatics:Sequence_QC#FastQC

Evaluation

FastQC

GCATemplates available: grace

Click to see template script on github.tamu.edu

```
module spider FastQC
```

After running FastQC via the command line, you can ssh to an HPRC cluster enabling X11 forwarding by using the -X option and view the images using the eog tool.

From your desktop:

```
ssh -X username@grace.hprc.tamu.edu
```

From your FastQC working directory on Grace unzip the .zip results file then use eog to view the results in the Images directory:

```
eog sample_fastqc/Images/per_sequence_gc_content.png
```

You can also run FastQC interactively using the FastQC GUI by logging in using X11 forwarding and running the command:

```
fastqc
```

https://hprc.tamu.edu/wiki/Bioinformatics:Sequence_QC#FastQC

Executable File | 44 lines (34 sloc) | 1.9 KB

Raw | Blame

```
#!/bin/bash
#SBATCH --export=NONE                # do not export current env to the job
#SBATCH --job-name=fastqc            # job name
#SBATCH --time=01:00:00              # max job run time dd-hh:mm:ss
#SBATCH --ntasks-per-node=1          # tasks (commands) per compute node
#SBATCH --cpus-per-task=2            # CPUs (threads) per command
#SBATCH --mem=14G                    # total memory per node
#SBATCH --output=stdout.%x.%j        # save stdout to file
#SBATCH --error=stderr.%x.%j         # save stderr to file

module load FastQC/0.11.9-Java-11

<<README
    - FASTQC homepage: http://www.bioinformatics.babraham.ac.uk/projects/fastqc
    - FASTQC manual: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/Help
README

################################## VARIABLES ################################
# TODO Edit these variables as needed:

########## INPUTS ##########
pe1_1='/scratch/data/bio/GCATemplates/data/miseq/c_dubliniensis/DR34_R1.fastq.gz'
pe1_2='/scratch/data/bio/GCATemplates/data/miseq/c_dubliniensis/DR34_R2.fastq.gz'

```

Click “Raw” if you want to copy and paste from your web browser

```bash
#!/bin/bash
#SBATCH --export=NONE                # do not export current env to the job
#SBATCH --job-name=fastqc            # job name
#SBATCH --time=01:00:00              # max job run time dd-hh:mm:ss
#SBATCH --ntasks-per-node=1          # tasks (commands) per compute node
#SBATCH --cpus-per-task=2            # CPUs (threads) per command
#SBATCH --mem=4G                     # total memory per node
#SBATCH --output=stdout.%x.%j        # save stdout to file
#SBATCH --error=stderr.%x.%j         # save stderr to file

module load FastQC/0.11.9-Java-11

<<README
    - FASTQC homepage: http://www.bioinformatics.babraham.ac.uk/projects/fastqc
    - FASTQC manual: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/Help
README

################################ VARIABLES ################################
# TODO Edit these variables as needed:

########## INPUTS ##########
pe1_1='/scratch/data/bio/GCATemplates/data/miseq/c_dubliniensis/DR34_R1.fastq.gz'
pe1_2='/scratch/data/bio/GCATemplates/data/miseq/c_dubliniensis/DR34_R2.fastq.gz'

######## PARAMETERS ########
threads=$SLURM_CPUS_PER_TASK

########## OUTPUTS #########
output_dir='./'
```

These parameters are read by the job scheduler

Load the required module(s) first

Comment section

Variables to be edited

```
################################## COMMANDS ##################################
# use -o <directory> to save results to <directory> instead of directory where reads are located
#   <directory> must already exist before using -o <directory> option
# --nogroup will calculate average at each base instead of bins after the first 50 bp
# fastqc runs one thread per file; using 20 threads for 2 files does not speed up the processing

fastqc -t $threads -o $output_dir $pe1_1 $pe1_2

##############################################################################
<<CITATION
    - Acknowledge TAMU HPRC: https://hprc.tamu.edu/research/citations.html

    - FastQC: http://www.bioinformatics.babraham.ac.uk/projects/fastqc
CITATION
```

Command to run the application

Download and modify GCATemplate Script

- You can access GCATemplates through the command line or copy a script from the HPRC Wiki
- Modify it to run on the two fastq files in the example data
- Once the script is modified and in your working directory, submit the job to run on a compute node

```
sbatch <nameofjobscript>
```

RNA-seq Library QC

Learning Objective:

Run FastQC on a pair of fastq files, interpret and explain the output from the FastQC program, and identify common QC problems and their underlying causes.

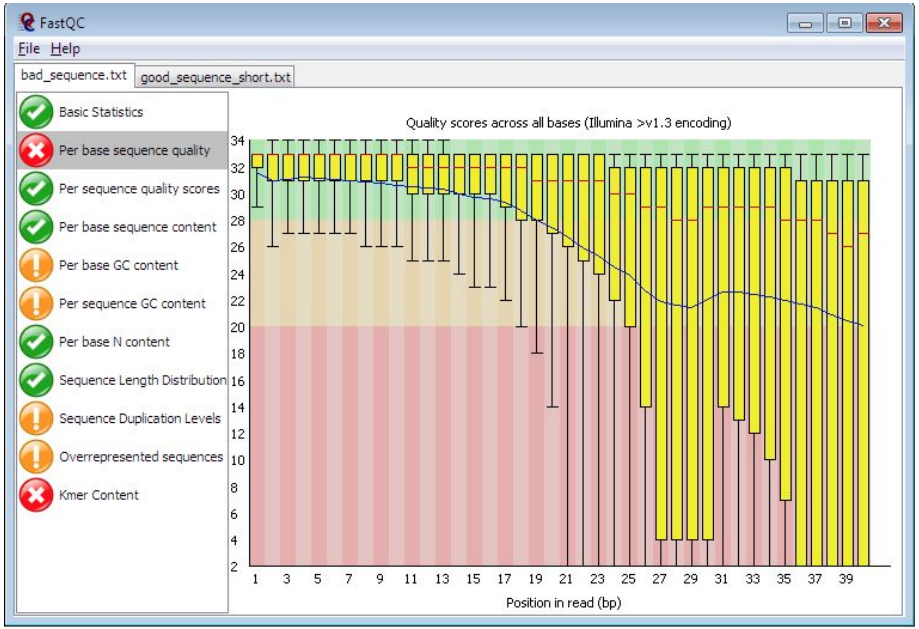
Project sponsored by the Texas Higher Education Coordinating Board

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

- Let’s see how to run FastQC using the script you modified and uploaded in the last module



Results Summary

- Read 1

Basic Statistics

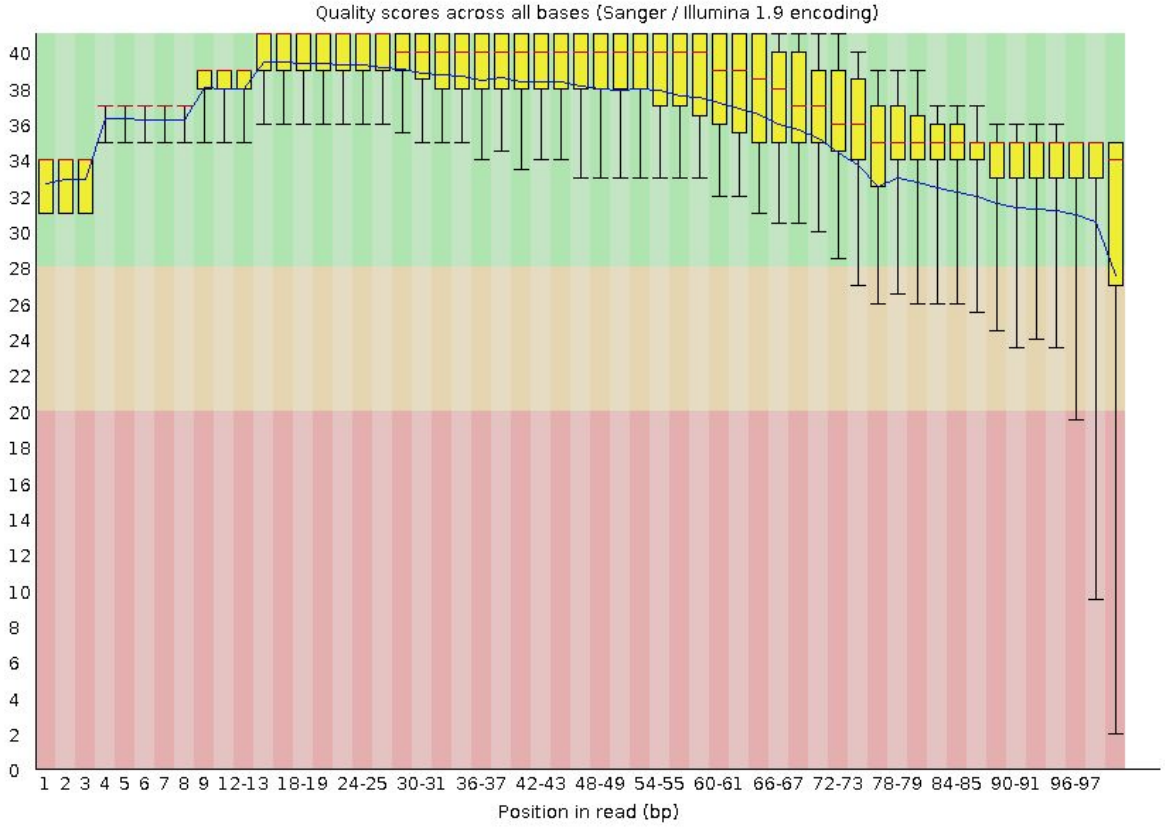
Measure	Value
Filename	Control1_R1.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	250000
Sequences flagged as poor quality	0
Sequence length	100
%GC	44

- Read 2

Basic Statistics

Measure	Value
Filename	Control1_R2.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	250000
Sequences flagged as poor quality	0
Sequence length	100
%GC	44

Per base sequence quality



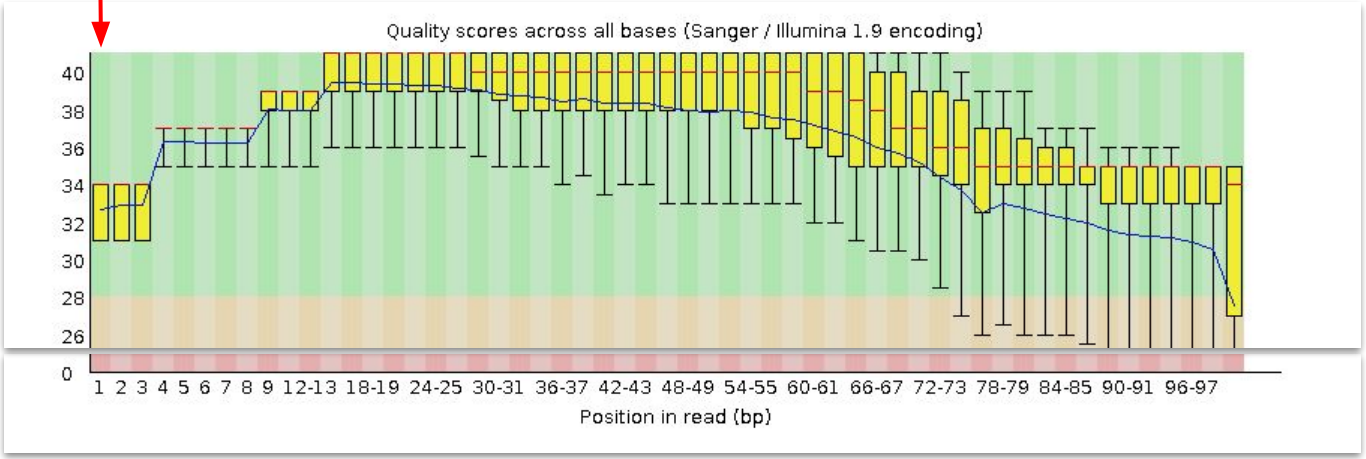
FASTQ Format

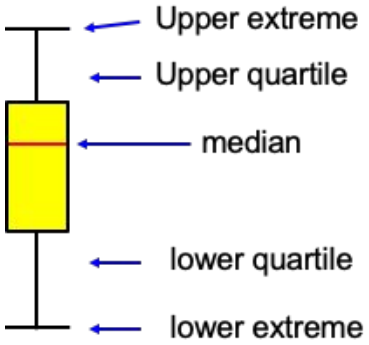
```
@ERR504787.2.1 M00368:15:000000000-A0HKH:1:5:21261:10968-1 length=100
GATCGGAAGAGCACACGTCTGAACTCCAGTCACGATCAGATCTCGTATGCCGTCTTCTGCTTGAAAAAAAAAAAAAAAACAAAAAACATAATGCCGTAAA
+ERR504787.2.1 M00368:15:000000000-A0HKH:1:5:21261:10968-1 length=100
=:=4AD=B8A:+<A::1<:AE<C3*?F<B???<?:8:6?B*9BD;/638.=-'-.@7=).=A:6?DDDCBBBB9555&&)+((+2&&+(((((((&&&+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

Average quality score distribution at position one

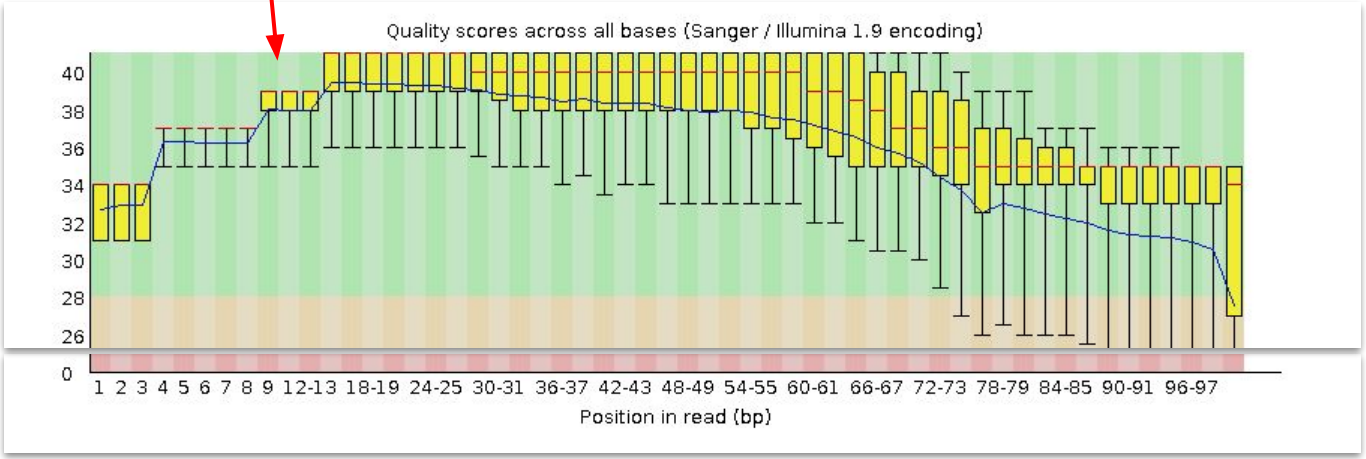


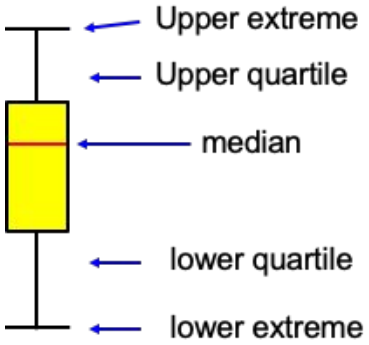


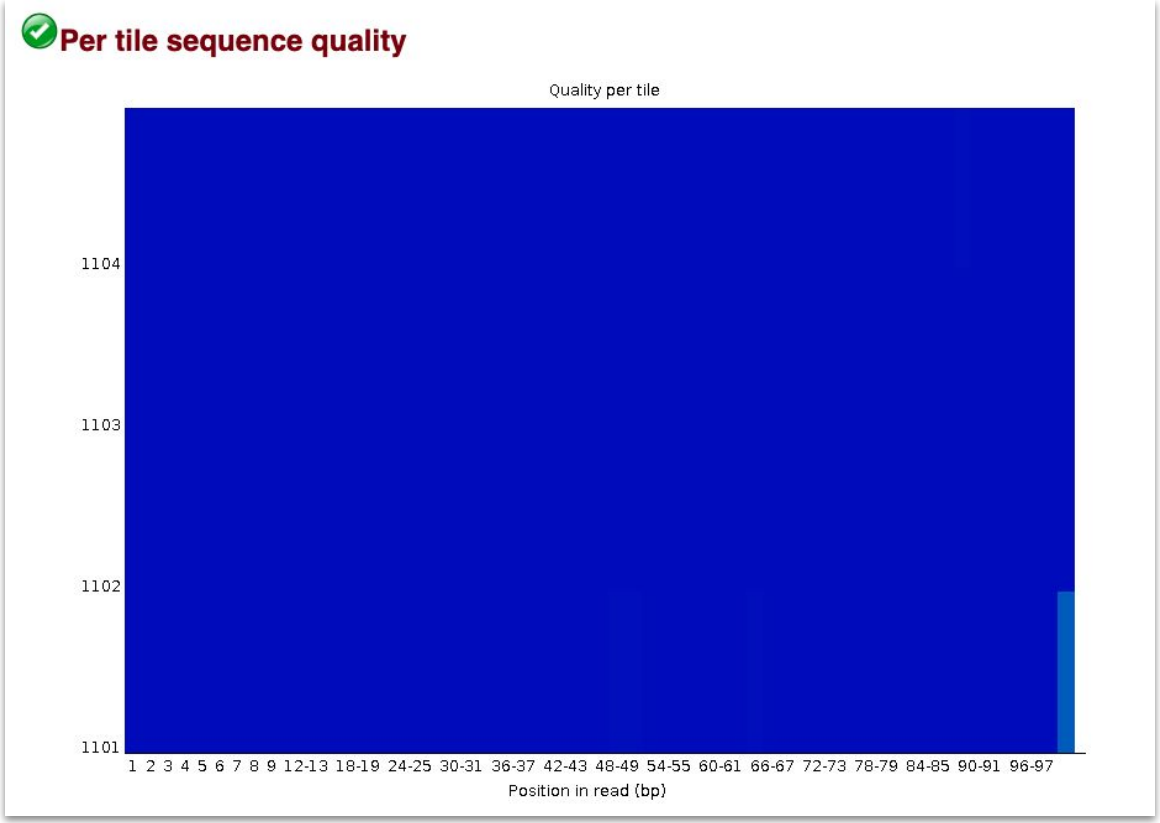
FASTQ Format

```
@ERR504787.2.1 M00368:15:000000000-A0HKH:1:5:21261:10968-1 length=100
GATCGGAAGAGCACACGTCTGAACTCCAGTCACGATCAGATCTCGTATGCCGTCTTCTGCTTGAAAAAAAAAAAAAAAACAAAAAACATAATGCCGTAAA
+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
```

Positions are binned after first few bases

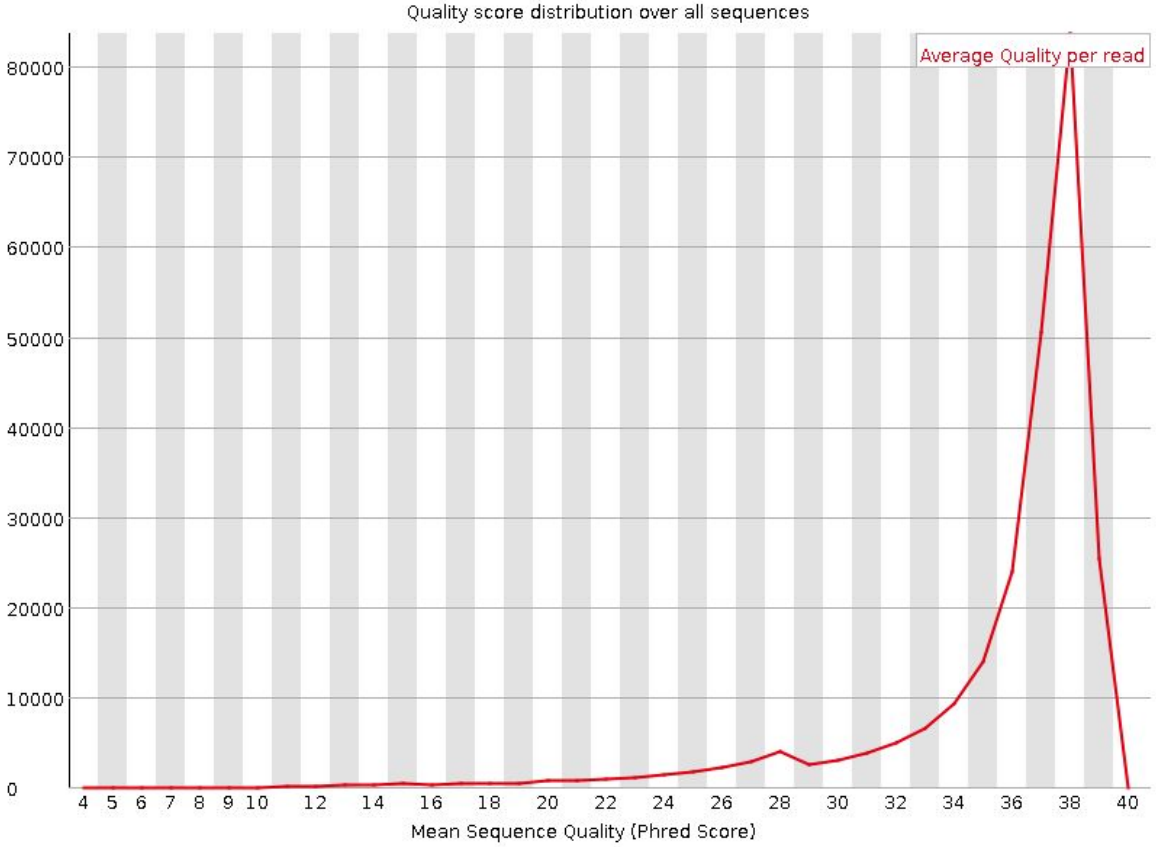




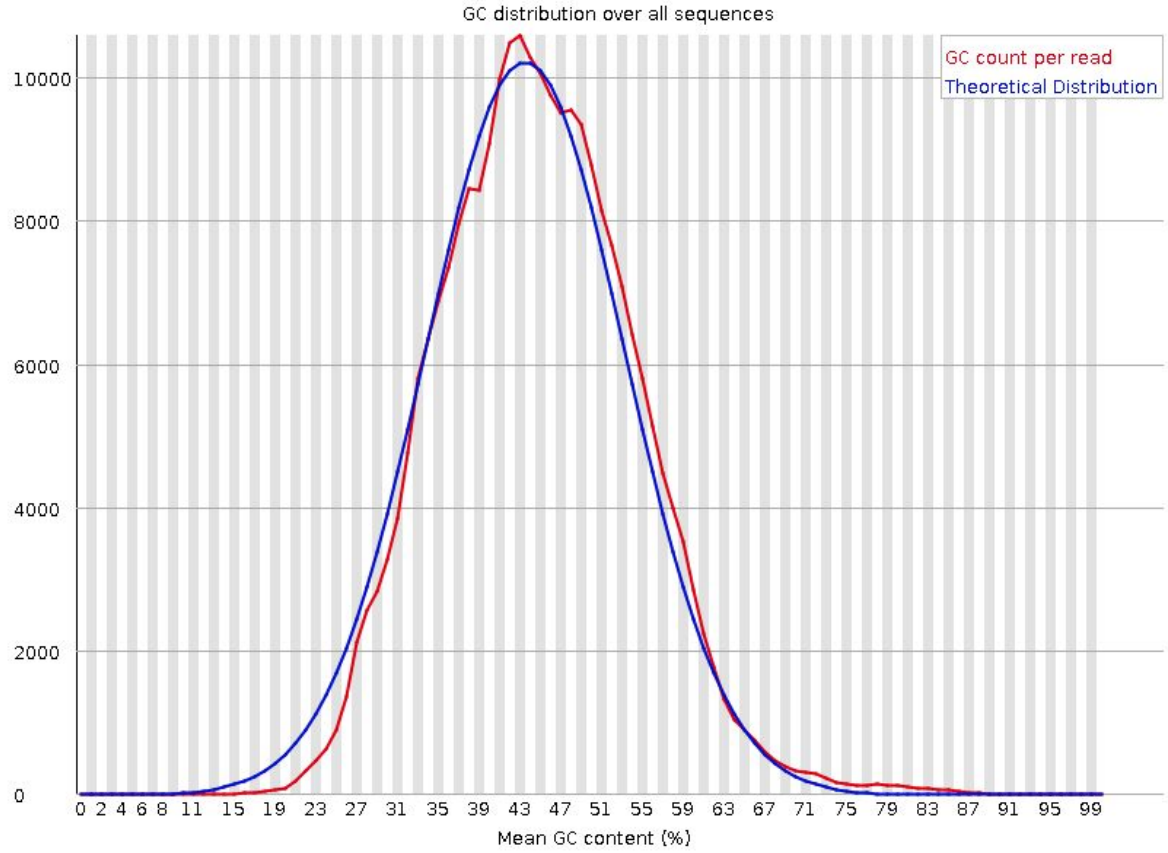


High quality  Low quality

Per sequence quality scores



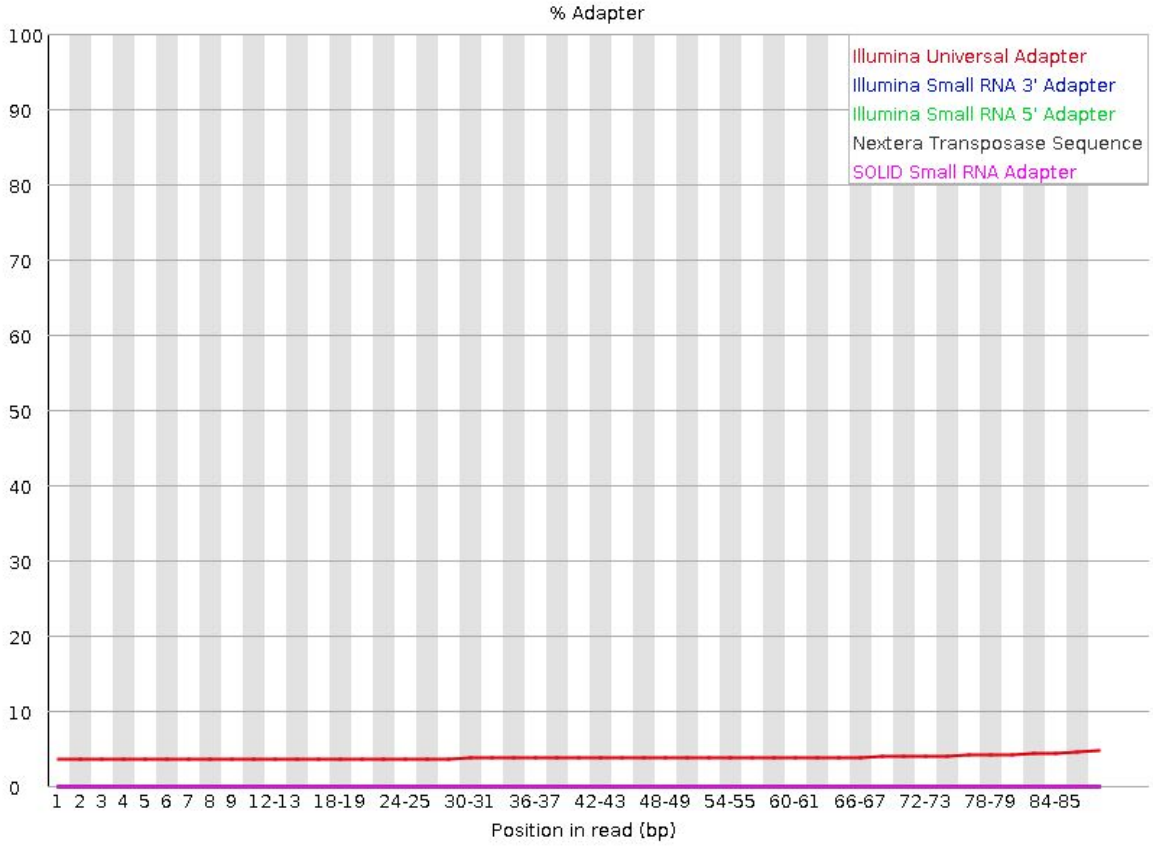
Per sequence GC content



Overrepresented sequences

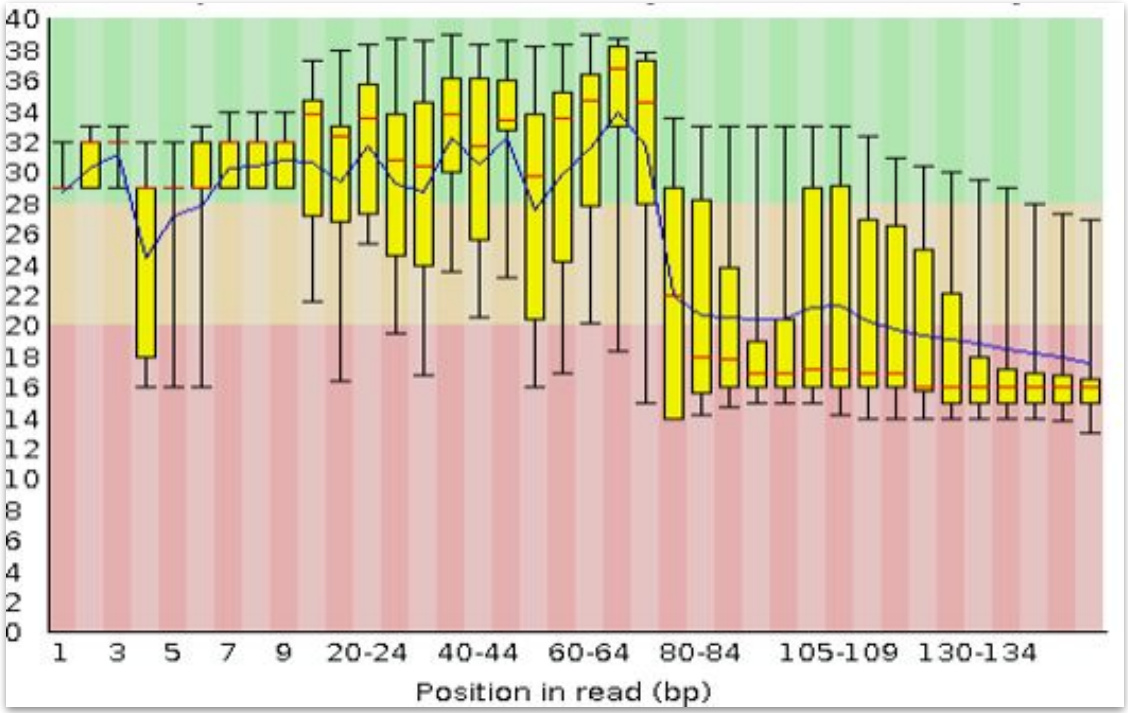
Sequence	Count	Percentage	Possible Source
AGATCGGAAGAGCACACGTCTGAACTCCAGTCACCGTACGTAATCTCGTA	8350	3.34	TruSeq Adapter, Index 22 (97% over 40bp)
GATCGGAAGAGCACACGTCTGAACTCCAGTCACCGTACGTAATCTCGTAT	1312	0.5248	TruSeq Adapter, Index 22 (97% over 40bp)

Adapter Content



Failed QC Examples

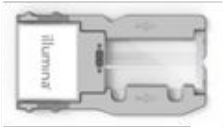
Example 1. Failed per base sequence quality - expired MiSeq kit

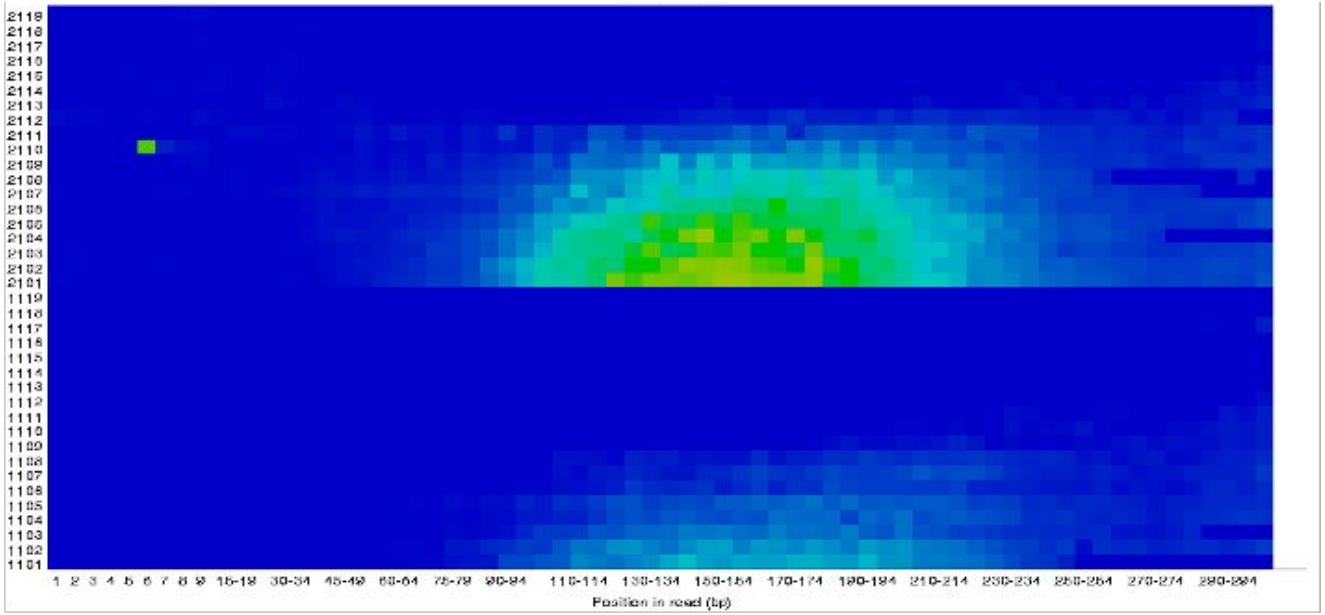


Failed QC Examples

Example 2. Faulty flowcell

MiSeq flowcell

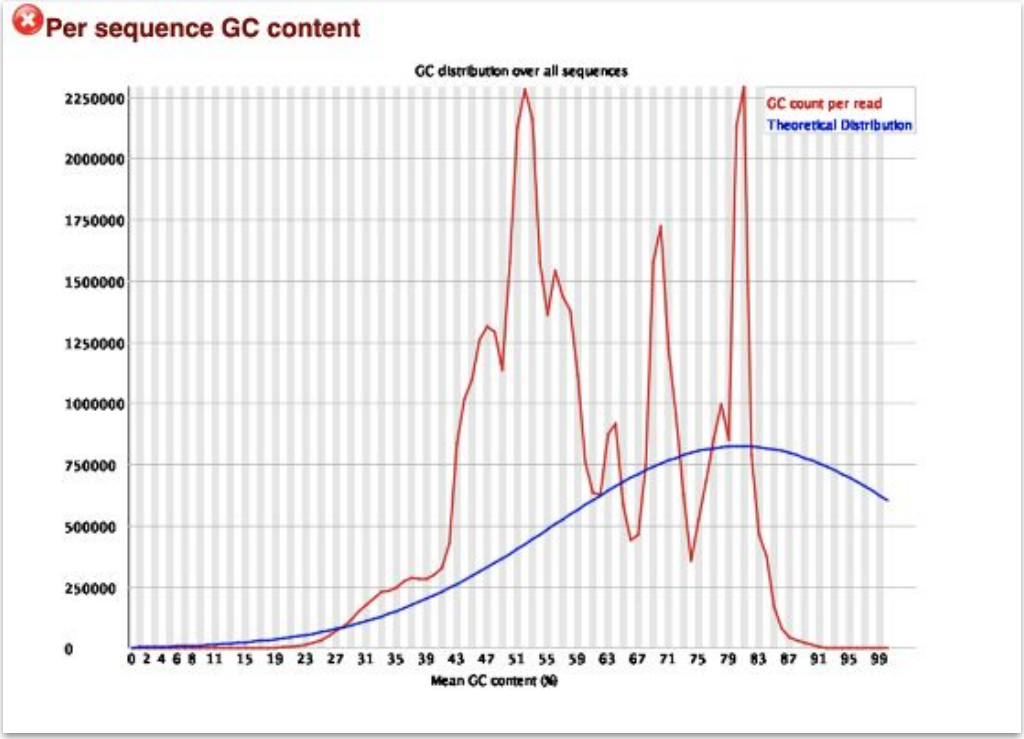




good quality  poor quality

Failed QC Examples

Example 3. Contamination



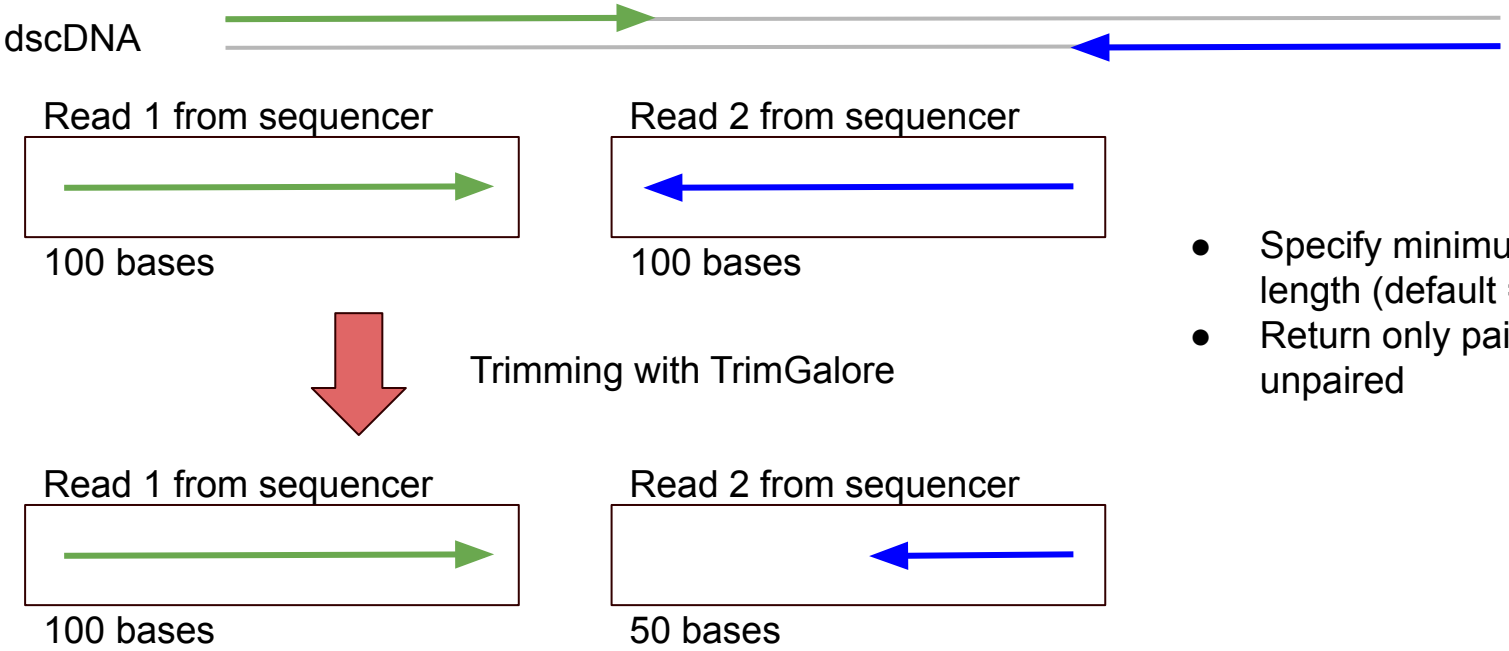
Library Trimming

Learning Objective:

Describe how library trimming works and use TrimGalore! to trim Illumiina libraries.

Project sponsored by the Texas Higher Education Coordinating Board

Library Trimming



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

Library Trimming

- Make sure you are in the RNA_class directory and complete the following commands in Grace to trim our RNA-seq libraries:

```
module purge
```

```
module spider trim_galore
```

Search for available versions of the program

```
module spider Trim_Galore/0.6.6-Python-3.8.2
```

See what modules need to be loaded with a specific version

Library Trimming

Continue in the same terminal with the following commands:

```
module load GCCcore/9.3.0 Trim_Galore/0.6.6-Python-3.8.2
```

```
trim_galore --paired Control1_R1.fastq.gz Control1_R2.fastq.gz
```

Library Trimming

Once the trimming is complete, check the results with FastQC:

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

```
fastqc Control1_R1_val_1.fq.gz
```

```
unzip Control1_R1_val_1_fastqc.zip
```

```
eog Control1_R1_val_1_fastqc/Images/per_base_quality.png
```

Aligning Reads to a Reference Genome

Learning Objective:

List commonly used RNA-seq aligners, use HISAT2 to align an RNA-seq library, and generate count files from alignments.

Project sponsored by the Texas Higher Education Coordinating Board

Aligning Reads to a Reference Genome

Read Mapping

- Popular splice-aware aligners
  - STAR
  - HISAT2
- Both programs need to index genome before aligning reads
  - Only needs to be done once
  - HISAT2 faster and more memory efficient
  - Some genomes already indexed on Grace

```
/scratch/data/bio/genome_indexes/
```

Send an email to help@hprc.tamu.edu if you need a genome indexed that is not found in the genome_indexes directory

Aligning Reads to a Reference Genome

- Read Mapping

- We'll use HISAT2 to align our library to the mouse reference genome - use the following command and then follow the instructions below to create and edit the job script

```
gcatemplates
```

- Type 11 to select "Sequence alignments"
- Type 3 to select "align mRNA reads to a reference"
- Type 1 to select "hisat2_2.2.1"
- Type 1 to select "pe library"
- Type y to copy the script to your current directory

```
gedit run_hisat2_2.2.1_pe_grace.sh
```

Aligning Reads to a Reference Genome

Read Mapping

- Modify the script

```
#!/bin/bash
#SBATCH --export=NONE                # do not export current env to the job
#SBATCH --job-name=hisat2            # job name
#SBATCH --time=1-00:00:00            # max job run time dd-hh:mm:ss
#SBATCH --ntasks-per-node=1          # tasks (commands) per compute node
#SBATCH --cpus-per-task=48           # CPUs (threads) per command
#SBATCH --mem=360G                   # total memory per node
#SBATCH --output=stdout.%x.%j        # save stdout to file
#SBATCH --error=stderr.%x.%j         # save stderr to file

<<README
    - HISAT2 manual: http://ccb.jhu.edu/software/hisat2/manual.shtml
README

module load GCC/9.3.0 OpenMPI/4.0.3 HISAT2/2.2.1
module load Python/3.8.2 SAMtools/1.10

######### SYNOPSIS #########
# This template script aligns paired end reads and sorts the output into a bam file

################################## VARIABLES ##################################
# TODO Edit these variables as needed:

########## INPUTS ##########
pe_1='/scratch/data/bio/GCATemplates/miseq/a_fumigatus/DRR022927_1.fastq.gz'
pe_2='/scratch/data/bio/GCATemplates/miseq/a_fumigatus/DRR022927_2.fastq.gz'
```

Change path to our trimmed reads

Aligning Reads to a Reference Genome

Read Mapping

- Modify the script

```
########## INPUTS ##########
pe_1='/scratch/data/bio/GCATemplates/miseq/a_fumigatus/DRR022927_1.fastq.gz'
pe_2='/scratch/data/bio/GCATemplates/miseq/a_fumigatus/DRR022927_2.fastq.gz'

# you can use an already prefixed genome found at: /scratch/data/bio/genome_indexes/
genome_index_prefix='/scratch/data/bio/genome_indexes/gmod_genomes/Aspergillus_fumigatus_Af293/hisat2/A_fumigatus_Af293'
```

Change path to the indexed mouse reference sequence

To this

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

Aligning Reads to a Reference Genome

Read Mapping

- Modify the script

```
######## PARAMETERS ########
threads=$SLURM_CPUS_PER_TASK
# read group information>
id='af_amp'
library='sra'
platform='ILLUMINA'
sample='DRR022927'

########## OUTPUTS #########
output_bam="${sample}_pe_aln.bam"
```

Change the read group information:

- id = ‘SRR5061328’
- library = ‘sra’
- platform = ‘Illumina’
- sample = ‘Control1’

Aligning Reads to a Reference Genome

Read Mapping

- Now submit the job and examine the output

```
sbatch run_hisat2_2.2.1_pe_grace.sh
```

```
squeue -u username
```

Check status of your job with this command

```
more stderr.hisat2 jobid
```

```
236499 reads; of these:
  236499 (100.00%) were paired; of these:
    ----
    32979 pairs aligned concordantly 0 times; of these:
      3583 (10.86%) aligned discordantly 1 time
    ----
    29396 pairs aligned 0 times concordantly or discordantly; of these:
      58792 mates make up the pairs; of these:
        33529 (57.03%) aligned 0 times
        22657 (38.54%) aligned exactly 1 time
        2606 (4.43%) aligned >1 times
92.91% overall alignment rate
[bam_sort_core] merging from 0 files and 48 in-memory blocks...
```

Generating Count Files

Learning Objective:

Generate count files from alignments.

Project sponsored by the Texas Higher Education Coordinating Board

Generating Count Files

- What to do after alignment?
- May need to convert your alignment
  - SAM to BAM
  - Coordinate-based or Read-based sorting

- Generate count files for DE
  - Number of reads aligning to each of a given GTF attribute

Several Options Are Available

- Summarize Overlaps
  - In R package “GenomicAlignments”
  - Generates DESeq object directly from sorted BAM files

- HTSeq-Count
  - Generates read count files for each sample
  - Commonly used – can be fed directly into DESeq2

Practice

- Type in the following commands to generate a count file for our mapped library:

```
module purge
```

```
module load GCC/10.2.0 OpenMPI/4.0.5 HTSeq/0.11.3 SAMtools/1.11
```

```
samtools index Control1_pe_aln.bam
```

```
htseq-count -f bam -r pos -i gene Control1_pe_aln.bam \
GCF_000001635.27_GRCm39_genomic.gff > Control1_counts.txt
```

High Performance Research Computing
DIVISION OF RESEARCH

TEXAS A&M UNIVERSITY
Engineering Studio for Advanced Instruction & Learning

TEES EDGE

Differential Expression Analysis with DESeq2

Learning Objectives:

Import read count data from HTSeq-Count into R, learn to load the R packages necessary for differential expression, create a DESeq data object, and run DESeq2 to complete a differential expression analysis.

Project sponsored by the Texas Higher Education Coordinating Board

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

Differential Expression using DESeq2

- Open RStudio through the Grace portal (Interactive apps > RStudio R 4.0.3, 3.6.3) or download the counts folder and open RStudio on your computer
- Open a new script where you will type all of the commands
- All R commands will be shown in green text on dark grey background:

```
command
```

- Set your working directory to the ‘counts’ folder

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

- Run commands by placing the cursor at the end of the command or by highlighting the command(s) you want to run and pressing ‘Run’

Differential Expression Analysis with DESeq2

You might need to install some packages:

```
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

BiocManager::install("EnhancedVolcano")
```

Differential Expression Analysis with DESeq2

Load the necessary libraries/packages:

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

Differential Expression Analysis with DESeq2

Load the count and sample information:

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

Output

	sampleName	fileName	condition
1	Control1_counts.txt	Control1_counts.txt	Control
2	Control2_counts.txt	Control2_counts.txt	Control
3	Control3_counts.txt	Control3_counts.txt	Control
4	Control4_counts.txt	Control4_counts.txt	Control
5	Control5_counts.txt	Control5_counts.txt	Control
6	NAD1_counts.txt	NAD1_counts.txt	NAD_supplement
7	NAD2_counts.txt	NAD2_counts.txt	NAD_supplement
8	NAD3_counts.txt	NAD3_counts.txt	NAD_supplement
9	NAD4_counts.txt	NAD4_counts.txt	NAD_supplement
10	NAD5_counts.txt	NAD5_counts.txt	NAD_supplement

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

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

Differential Expression Analysis with DESeq2

Load the count and sample information:

```
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 with DESeq2

Filter out genes with less than 10 total reads:

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

Run the differential expression analysis:

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

Output

```
> res
log2 fold change (MLE): condition NAD supplement vs Control
Wald test p-value: condition NAD supplement vs Control
DataFrame with 23595 rows and 6 columns
                baseMean log2FoldChange     lfcSE       stat    pvalue      padj
               <numeric>      <numeric> <numeric>  <numeric> <numeric> <numeric>
0610005C13Rik   5.99463       0.847191  1.066907   0.794063 0.4271590 0.6270460
0610009B22Rik 229.57285      -0.460666  0.276228  -1.667702 0.0953749 0.2297077
0610009E02Rik  52.71480      -1.185370  0.492924  -2.404771 0.0161826 0.0626895
0610009L18Rik   5.27097       0.500679  1.014609   0.493470 0.6216804 0.7782450
0610010F05Rik 217.32132       0.454101  0.177215   2.562429 0.0103943 0.0444339
...                 ...            ...       ...        ...       ...       ...
n-TRtct2        2.10872      1.6381445  1.355088  1.2088845 0.2267072  0.416331
n-TScga3        5.50077     -0.4540586  0.737908 -0.6153322 0.5383353  0.717329
n-TTagt5        1.22482     -3.3625847  1.792925 -1.8754738 0.0607276        NA
n-TWcca1        3.96509     -0.0540574  1.117400 -0.0483778 0.9614151  0.981356
n-TYgta1        1.05416      1.7174587  1.757956  0.9769635 0.3285872        NA
```

Differential Expression Analysis with DESeq2

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 with DESeq2

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 with DESeq2

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 with DESeq2

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 with DESeq2

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 with DESeq2

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 with DESeq2

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 with DESeq2

How many genes are differentially expressed at a significant level?

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

```
> sum(res$padj < 0.05, na.rm = TRUE)
[1] 5453
```

Collect all DEGs and write the results to file

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

Data Visualization

Learning Objectives:

Transform and prepare data for plotting in R, and generate publication-quality figures from your differential expression results.

Project sponsored by the Texas Higher Education Coordinating Board

Data Visualization

- We'll be using our results from DESeq2 to generate several plots that are useful for analyzing our data further.
- PCA plots
- Volcano plot (with the EnhancedVolcano library)
- Heatmaps (with the pheatmap library)

PCA Plot

- Log transform the results and calculate the variance

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

- Create a list of genes with the greatest variance (top 100)

```
select <- order(rv, decreasing = TRUE)[seq_len(min(100, length(rv)))]
```

PCA Plot

- Run the PCA and look at the results

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

```
> 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 3.353e-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 PCA results 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)
```

```
> head(ggPCA_out)
                PC1          PC2        PC3        PC4        PC5         PC6        PC7         PC8          PC9         PC10 sampleName            fileName      condition
Control1 -12.576882  0.679757091  1.4677571  1.4408177 -0.9772907 -2.58170153 -0.8901816  0.12856743  0.057730319 3.080869e-15   Control1 Control1_counts.txt        Control
Control2 -11.362119 -0.789437801 -4.1149258  0.5846590 -1.6247299  0.15350772  1.1461662 -0.21539578  0.001565017 2.699230e-15   Control2 Control2_counts.txt        Control
Control3 -12.038043  0.002152241  0.5811305 -3.0992919  1.4657618 -0.97863917  1.0515499 -0.04523746 -0.046467189 2.220446e-15   Control3 Control3_counts.txt        Control
Control4 -13.139919  0.487982477  0.6723550  2.2531953  2.6066210  1.29314839  0.3152618 -0.07647994  0.001933003 2.414735e-15   Control4 Control4_counts.txt        Control
Control5 -12.530993  0.265744874  1.7077315 -1.1646465 -1.8470308  2.10751112 -1.1715371  0.07993858 -0.013573324 3.108624e-15   Control5 Control5_counts.txt        Control
NAD1       8.795471  0.517771986 -2.7475597 -0.6142266  1.1887028 -0.04330035 -1.5880439  0.71154077  0.323683075 3.469447e-15       NAD1     NAD1_counts.txt NAD_supplement
```

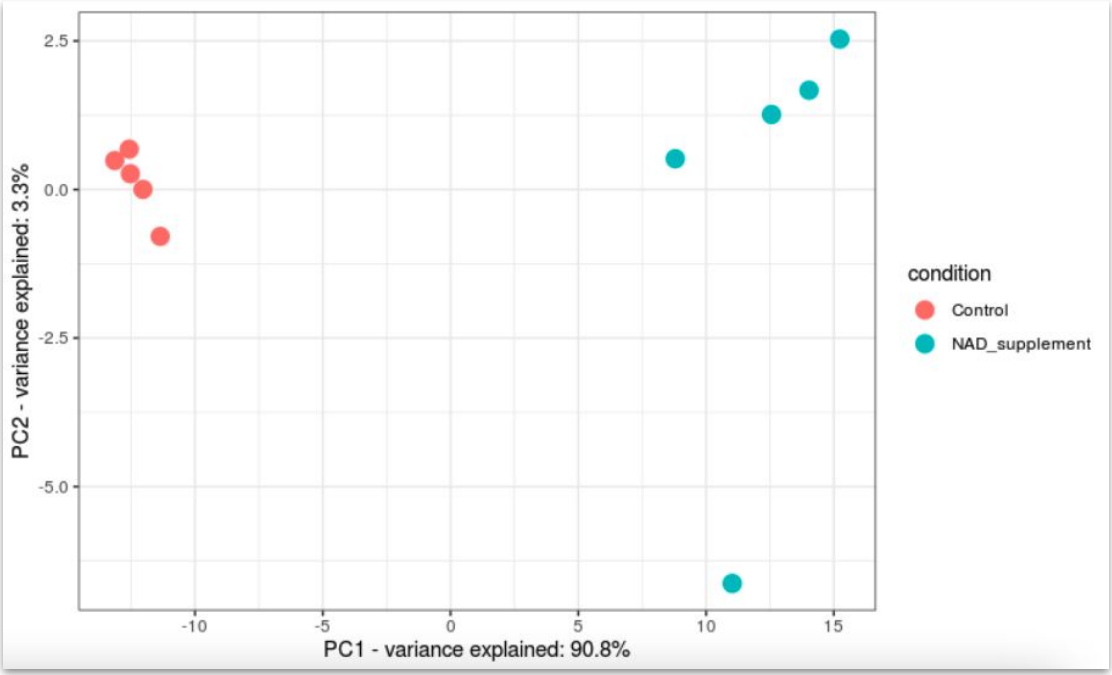
PCA Plot

- Plot the results

```
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()
```

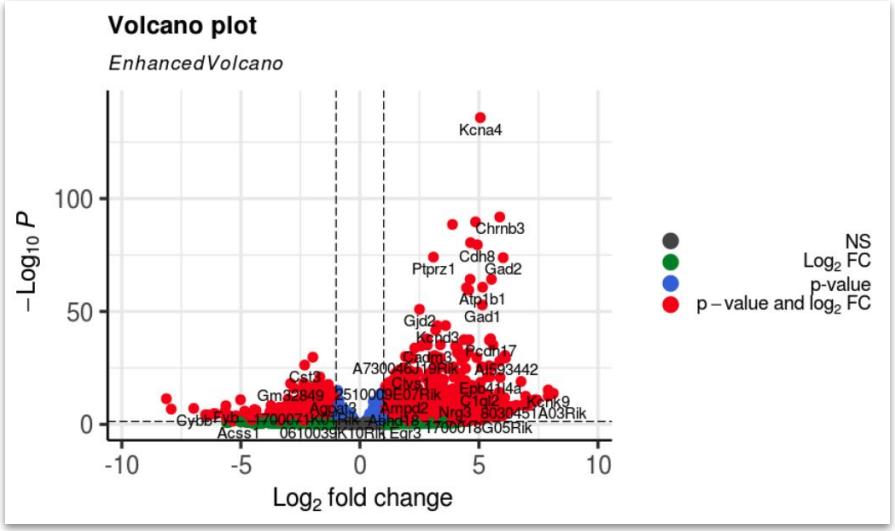
PCA Plot

- Plot the results



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")
```



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 ‘selected’
- [illegible]

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

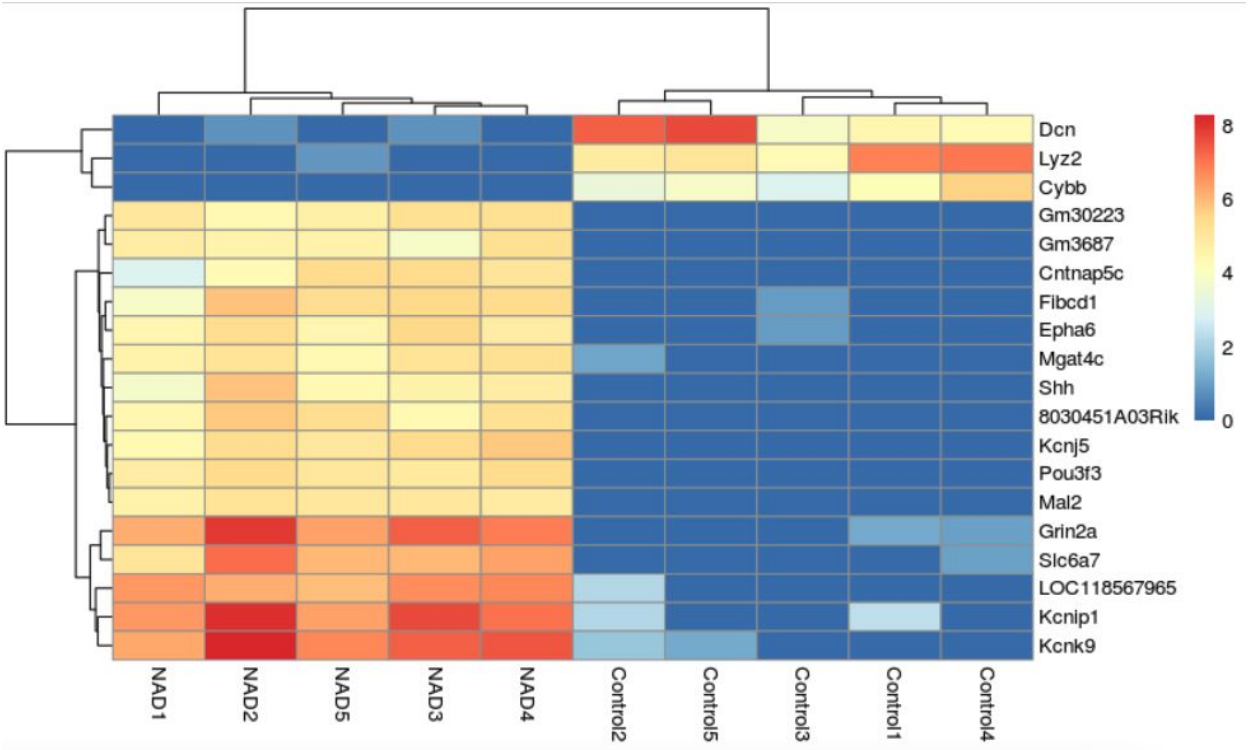
```
> selected
 [1] "Kcnip1"        "Kcnk9"         "Grin2a"        "Slc6a7"        "LOC118567965"  "Lyz2"
 [7] "Pou3f3"        "Kcnj5"         "Mal2"          "8030451A03Rik" "Gm30223"       "Fibcd1"
[13] "Gm3687"        "Shh"           "Mgat4c"        "Cntnap5c"      "Epha6"         "Cybb"
[19] "Dcn"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

Heatmap

- We'll transform the data
- Finally, we'll create the 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
NAD1
NAD2
NAD5
NAD3
NAD4
Control2
Control5
Control3
Control1
Control4
8
6
4
2
0