

AFRICAN CENTERS OF EXCELLENCE IN BIOINFORMATICS

KAMPALA, UGANDA

Processing and Analysis of RNAseq NGS Data

Today's Instructor

Brendan Jeffrey, Ph.D. Molecular and Cellular Biology

- Bioinformatics and Computational Biosciences Branch (BCBB), NIAID
- National Institutes of Health, Bethesda, MD USA
- NIH - Rocky Mountain Labs, Hamilton, MT USA
- Contact our team via email:
 - Listserv: bioinformatics@niaid.nih.gov
 - Instructor: brendan.jeffrey@nih.gov

Purpose of This Course



Introduction to RNAseq

An introduction to computational tools used for processing and analyzing RNAseq data – heavy on the R

Introduction – What is RNAseq



RNA sequencing uses next-generation sequencing to reveal presence and quantity of RNA in a biological sample at a given moment in time

Introduction – Why Sequence RNA

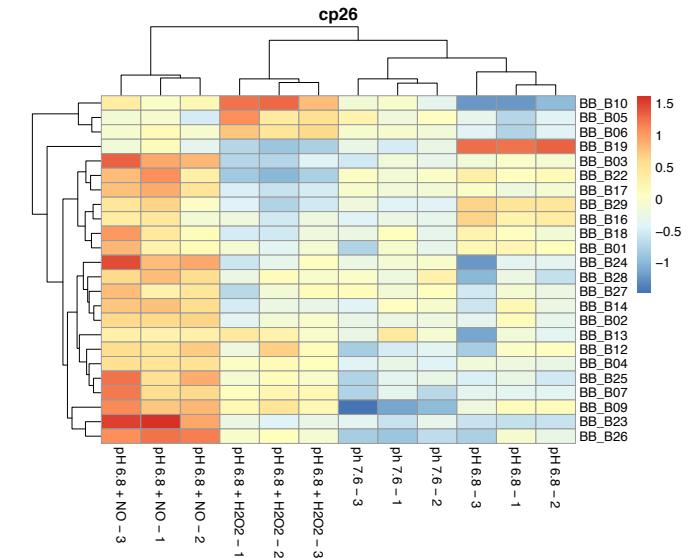
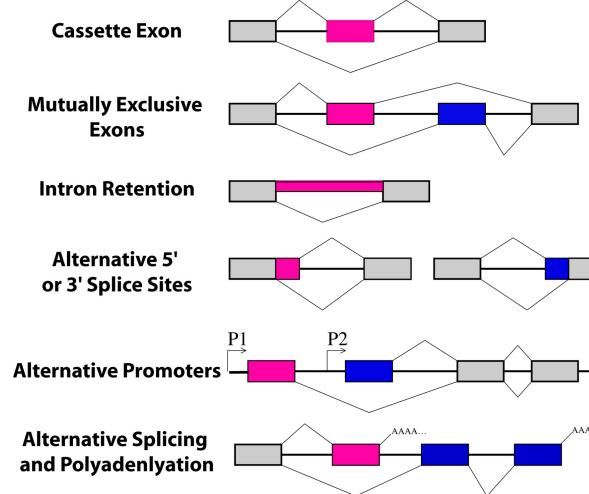


Qualitative - Annotation

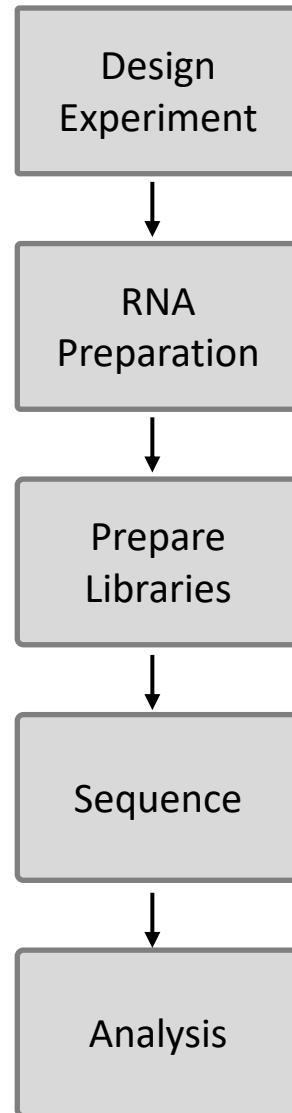
- Transcriptome assembly
- Novel gene finding
- Defining Exon/Intron boundaries
- SNP finding

Quantitative

- Functional studies
- Differential gene expression
- Alternative splicing



RNAseq – Experiment Workflow



Experiment
designed to address
your question

Isolate and
purify RNA

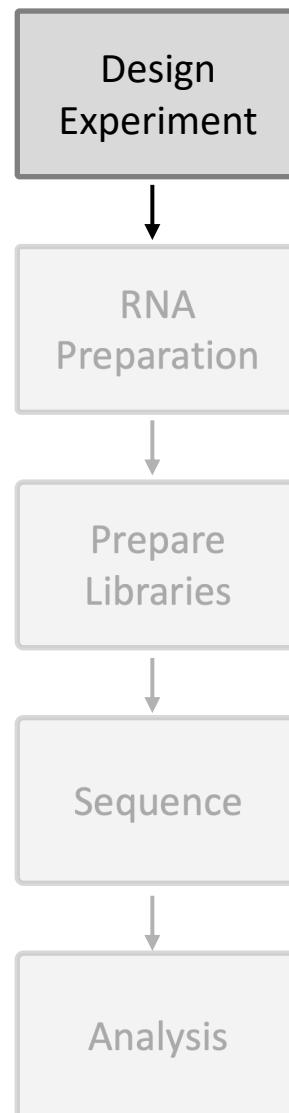
Convert RNA to cDNA
add adaptors,
multiplex barcodes

Sequence cDNA using
sequencing platform -
Illumina / Pac Bio

Analyzing
sequenced
short reads



RNAseq – Experiment Workflow



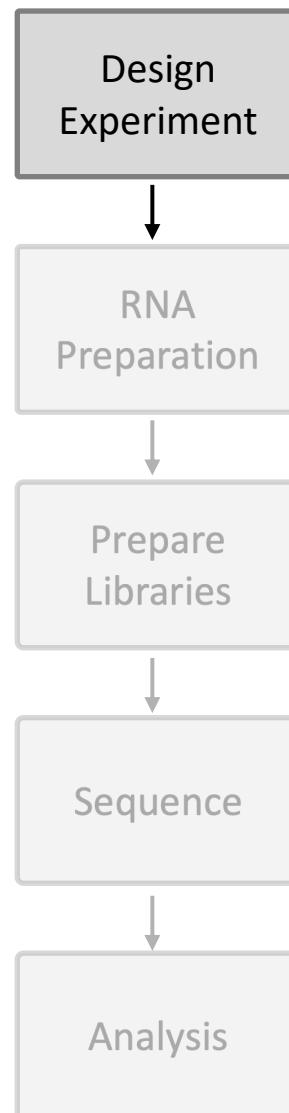
Qualitative/Annotation – analyze RNAseq reads to identify genes, architecture

Depth of coverage is important versus replicates

Library prep methods important to get even coverage across entire transcript

SMARTer system - clontech

RNAseq – Experiment Workflow



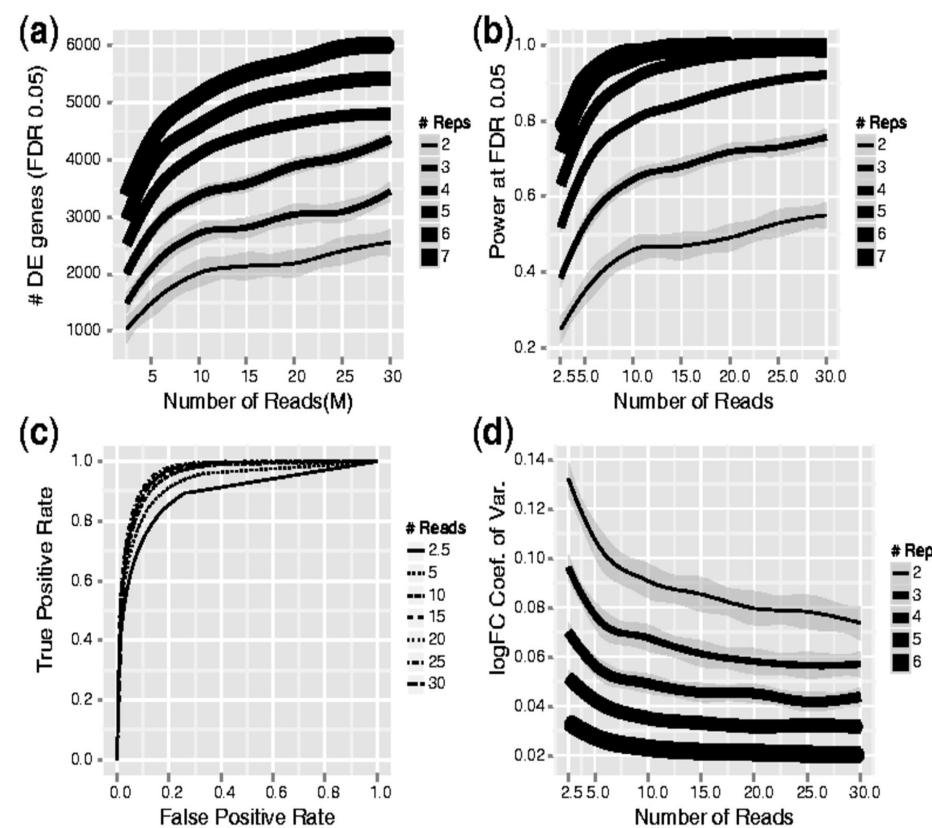
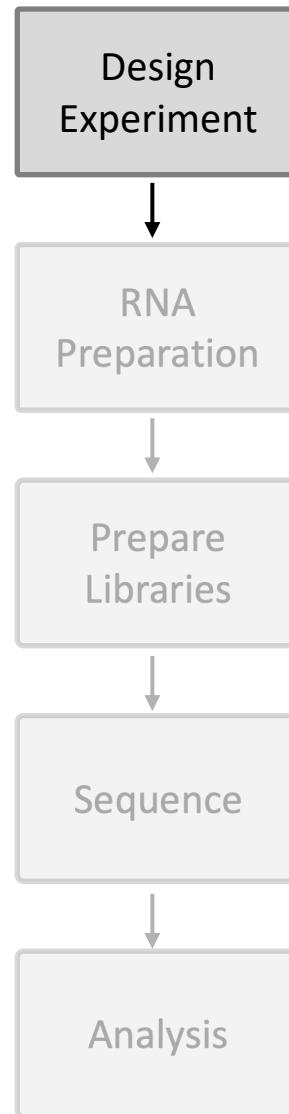
Differential Gene Expression

DGE experiments designed to accurately measure both the counts of each gene (coverage) and the variances that are associated with those counts (replicates)

Biological replicates >> Technical replicates

[Biometry: The Principles and Practices of Statistics in Biological Research](#) – Sokal and Rohlf

RNAseq – Experiment Workflow



Drug 1 with 2 treatments (+ or -)

+ N = 6	- N = 6
---------	---------

Total # ind. = 12

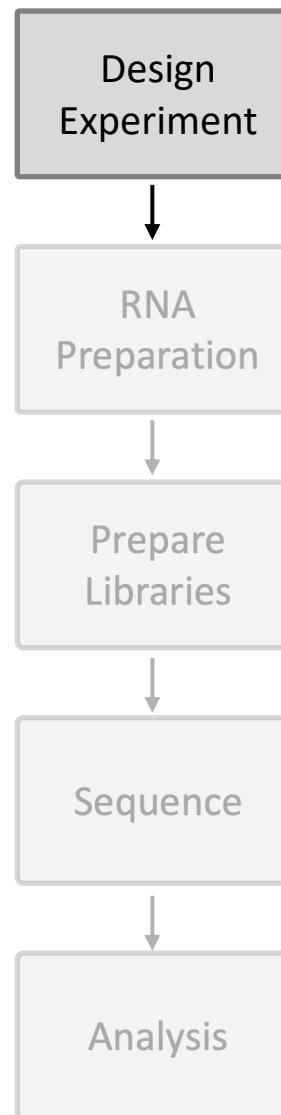
Drug 1 with 2 treatments (+ or -) by Drug 2 with 2 treatments (+ or -) by Drug 3 with 2 treatments (+ or -)

- / - / - N = 6	- / - / + N = 6	- / + / - N = 6	- / + / + N = 6
+ / - / - N = 6	+ / - / + N = 6	+ / + / - N = 6	+ / + / + N = 6

Total # ind. = 48

Sample treatment/processing randomization

RNAseq – Experiment Workflow



We need more power! - Scotty

Inputs

Pilot Data: Upload your own pilot data or used a stored dataset as a model for your experiment. [\(?\)](#)

CAUTION

Power analysis results will not be predictive of the actual results unless the power analysis is performed on data that closely matches the experiment. Please read about [generating pilot data](#) and [selecting preloaded datasets](#) before continuing.

Upload Data

Upload a file containing the number of reads per gene for pilot data as a tab delimited text file. [See format info.](#)

No file chosen

Number of Replicates in Control:

Number of Replicates in Test (enter 0 if none):

Use a stored dataset [\(?\)](#)

Choose a model dataset (*Less Accurate*): [Dataset Descriptions](#)

Cost Data [\(?\)](#)

Cost per replicate, excluding reads:

Control:

Test:

Cost per million reads sequenced: [\(?\)](#)

Alignment Rate (to genes or transcripts): % [\(How to calculate?\)](#)

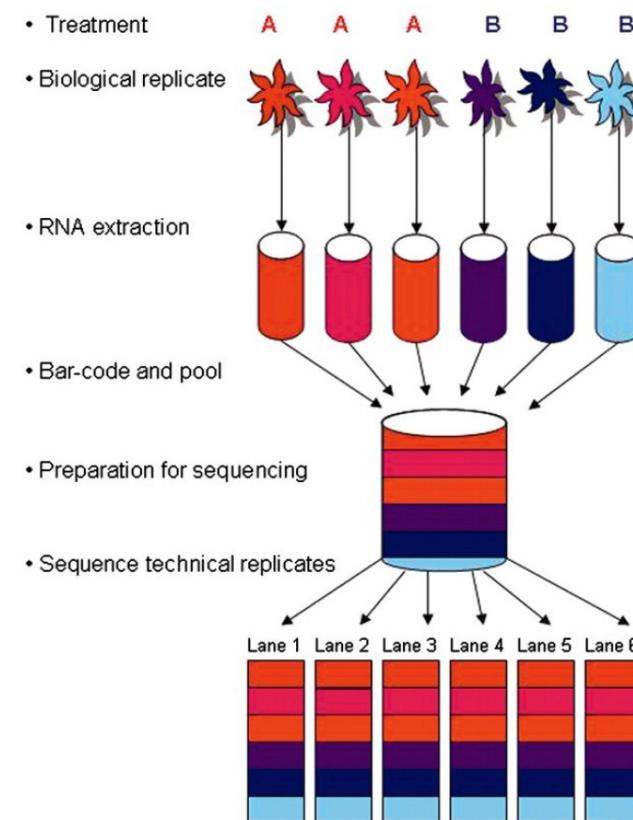
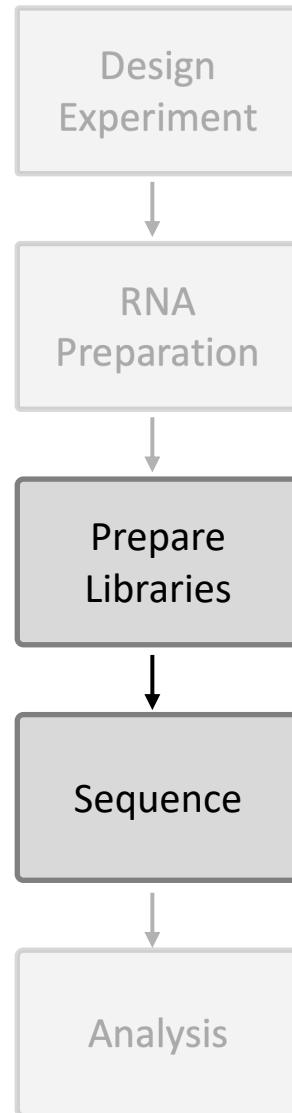
Constraints for Power Optimization [\(?\)](#)

Experimental Configurations to Test:

Maximum number of biological replicates per condition:

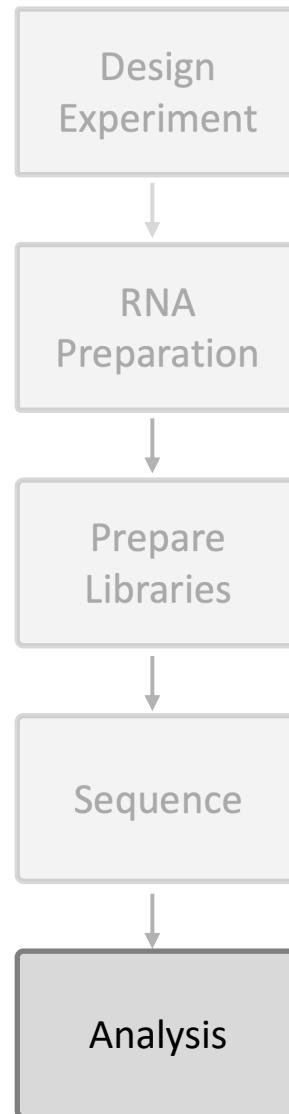
Assess the power of sequencing depths between and reads aligned to genes per replicate

RNAseq – Experiment Workflow



Multiplex and pool
when sequencing

RNAseq – Sequence Quality Control



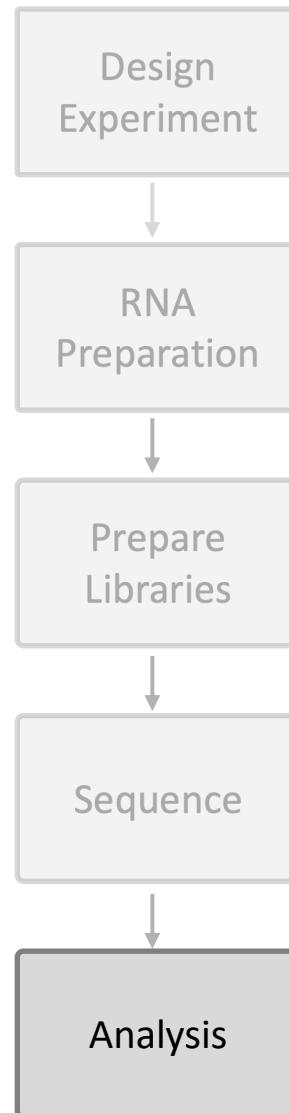
Several million reads generated - Fastq format

```
@M03264:1:00000000-AB9MU:1:1101:9844:1771 1:N:0
TATTCTGTGACTAGATTTTGAGAATAATCAGAATACAATTCTTGATATACAAGTATGCCAT
+
CGGGGGGGGGGGGGGGGGGGGGGGGGGGGGFC<FCFFDGGGGGGGGCGFGGGFFFGGGGGGGGGG
```

Table 1 ASCII Characters Encoding Q-scores 0–40

Symbol	ASCII Code	Q-Score	Symbol	ASCII Code	Q-Score	Symbol	ASCII Code	Q-Score
!	33	0	/	47	14	=	61	28
"	34	1	0	48	15	>	62	29
#	35	2	1	49	16	?	63	30
\$	36	3	2	50	17	@	64	31
%	37	4	3	51	18	A	65	32
&	38	5	4	52	19	B	66	33
'	39	6	5	53	20	C	67	34
(40	7	6	54	21	D	68	35
)	41	8	7	55	22	E	69	36
*	42	9	8	56	23	F	70	37
+	43	10	9	57	24	G	71	38
,	44	11	:	58	25	H	72	39
-	45	12	;	59	26	I	73	40
.	46	13	<	60	27			

RNAseq – Sequence Quality Control

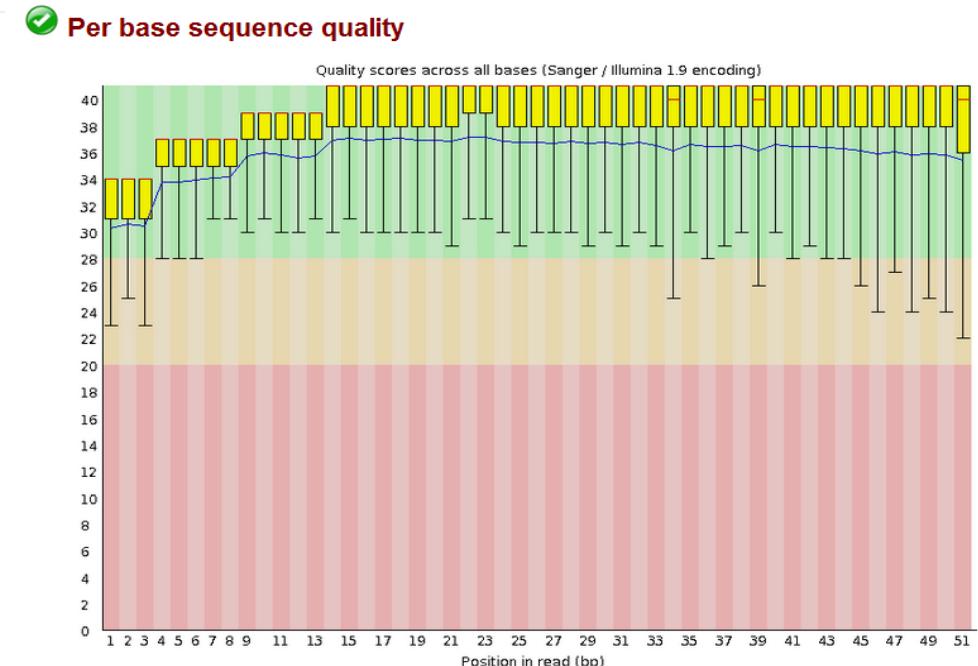
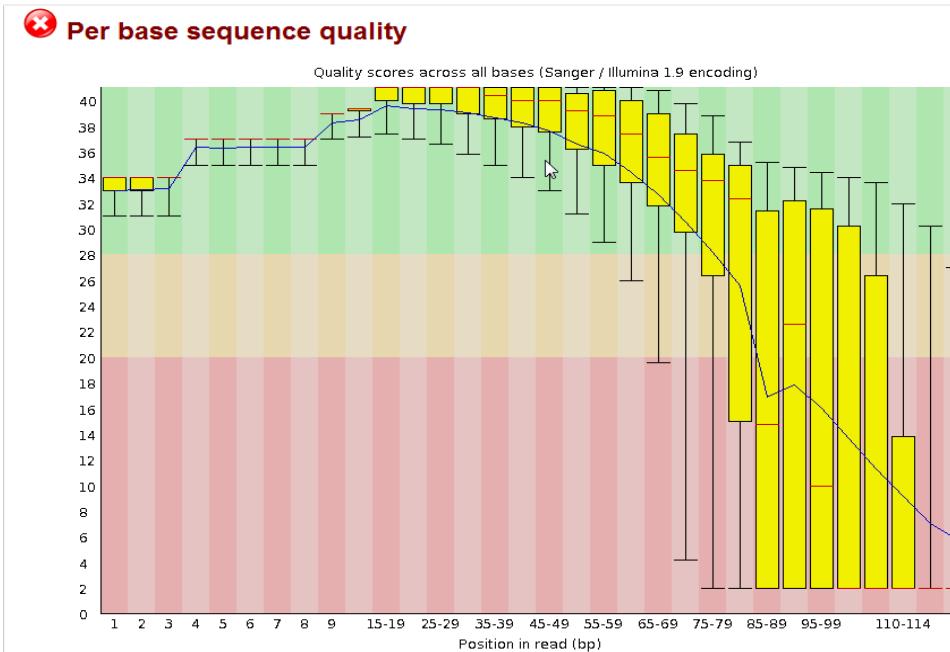
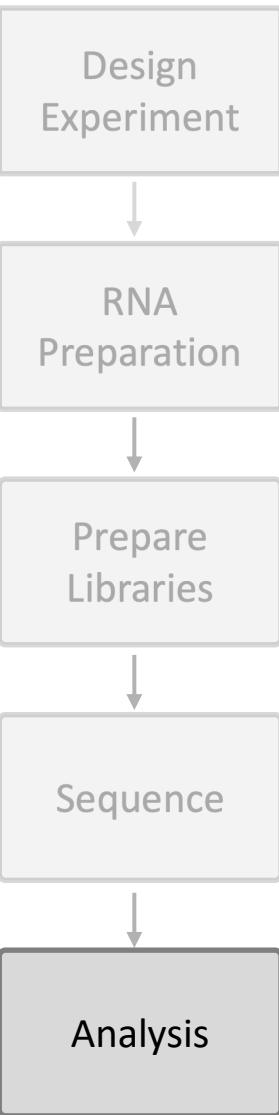


Several million reads generated - Fastq format

```
@M03264:1:00000000-AB9MU:1:1101:9844:1771 1:N:0
TATTCTGTGACTAGATTTTGAGAATAATCAGAATACAATTCTTGATATACAAGTATGCCAT
+
CGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGFC<FCFFDGGGGGGGGCGFGGGFFFGGGGGGGGGG
```

Quality Score	Probability of Incorrect Base Call	Inferred Base Call Accuracy
10 (Q10)	1 in 10	90%
20 (Q20)	1 in 100	99%
30 (Q30)	1 in 1000	99.9%

RNAseq – Sequence Quality Control

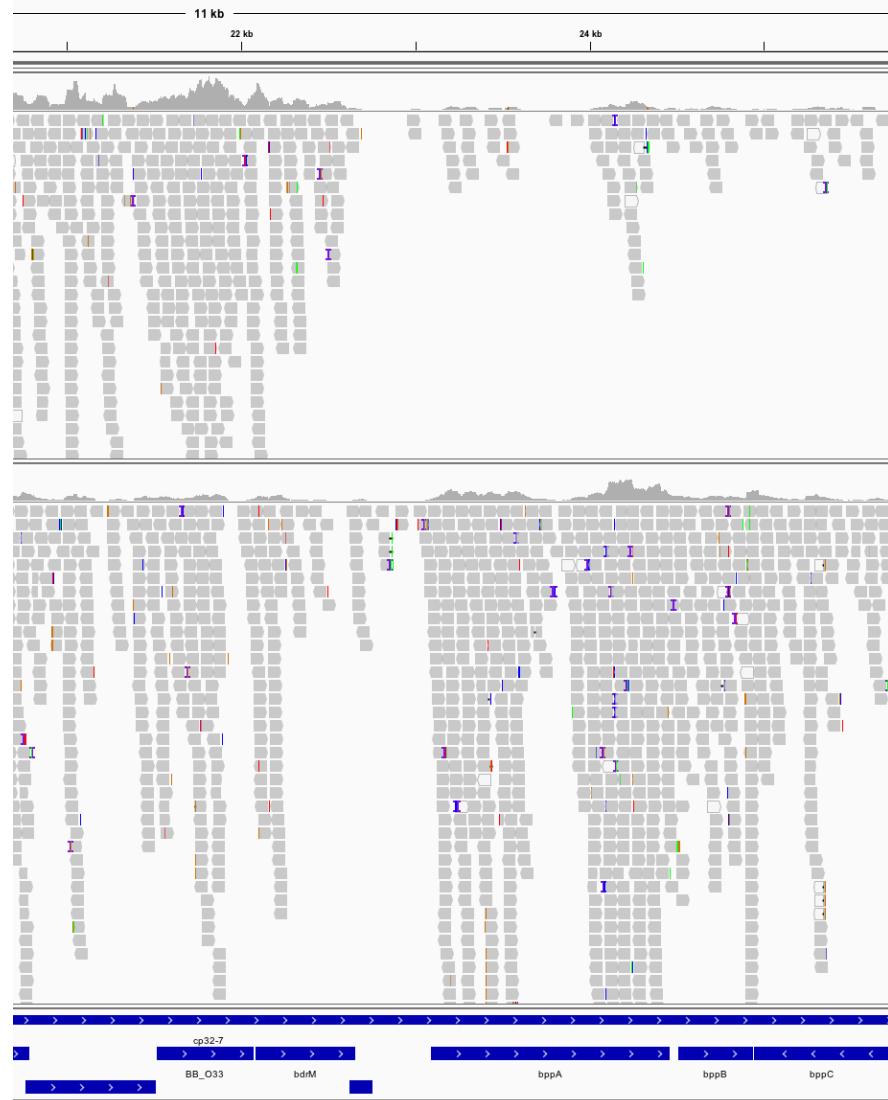
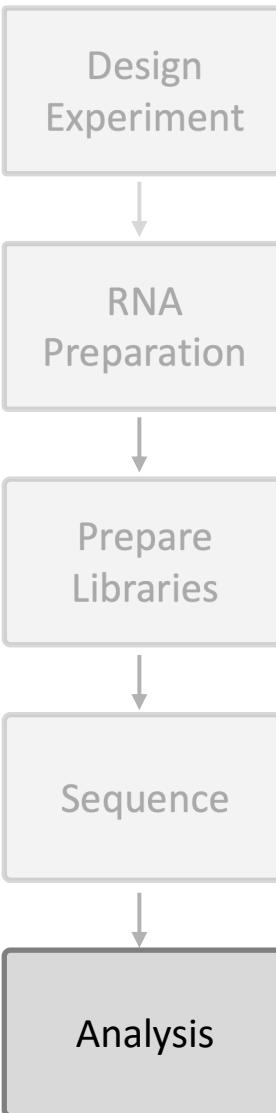


Read QC overview with [FastQC](#)

Illumina adaptor removal, quality trimming of reads

[Trimmomatic](#), Sickle, FASTX-Toolkit, BBduk

RNAseq – Mapping Reads



[**Bowtie2**](#) – great general use

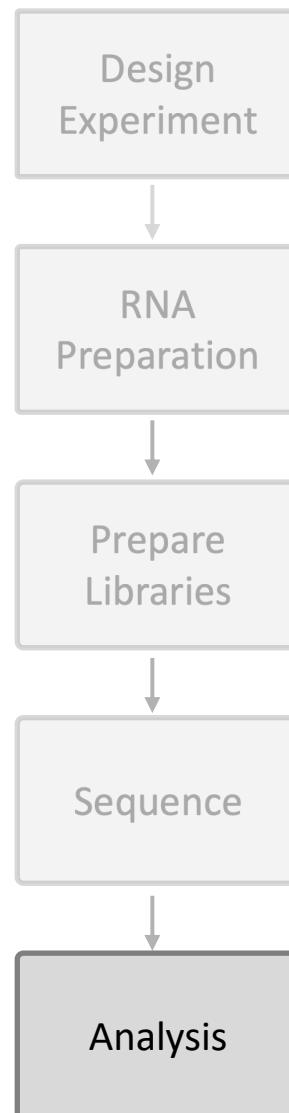
Splice aware?

- [**HISAT2**](#)
- [**BWA-MEM**](#)
- [**STAR**](#) – gene counts as output

Pseudoalignment – very fast

- [**Kallisto**](#)
- [**Salmon**](#)

RNAseq – Alignment files – SAM/BAM

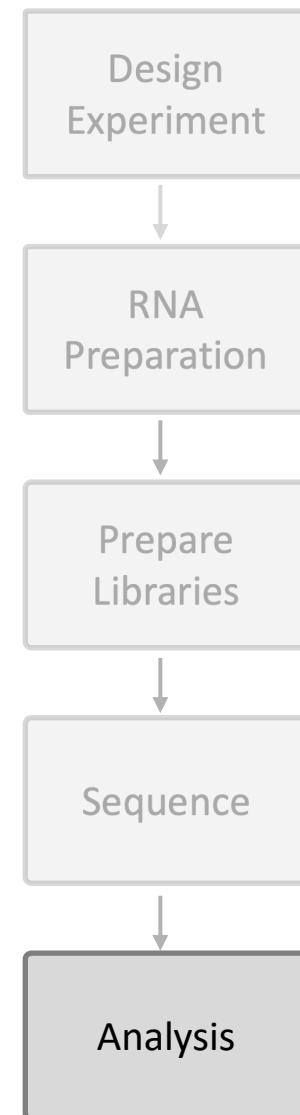


SAM/BAM format

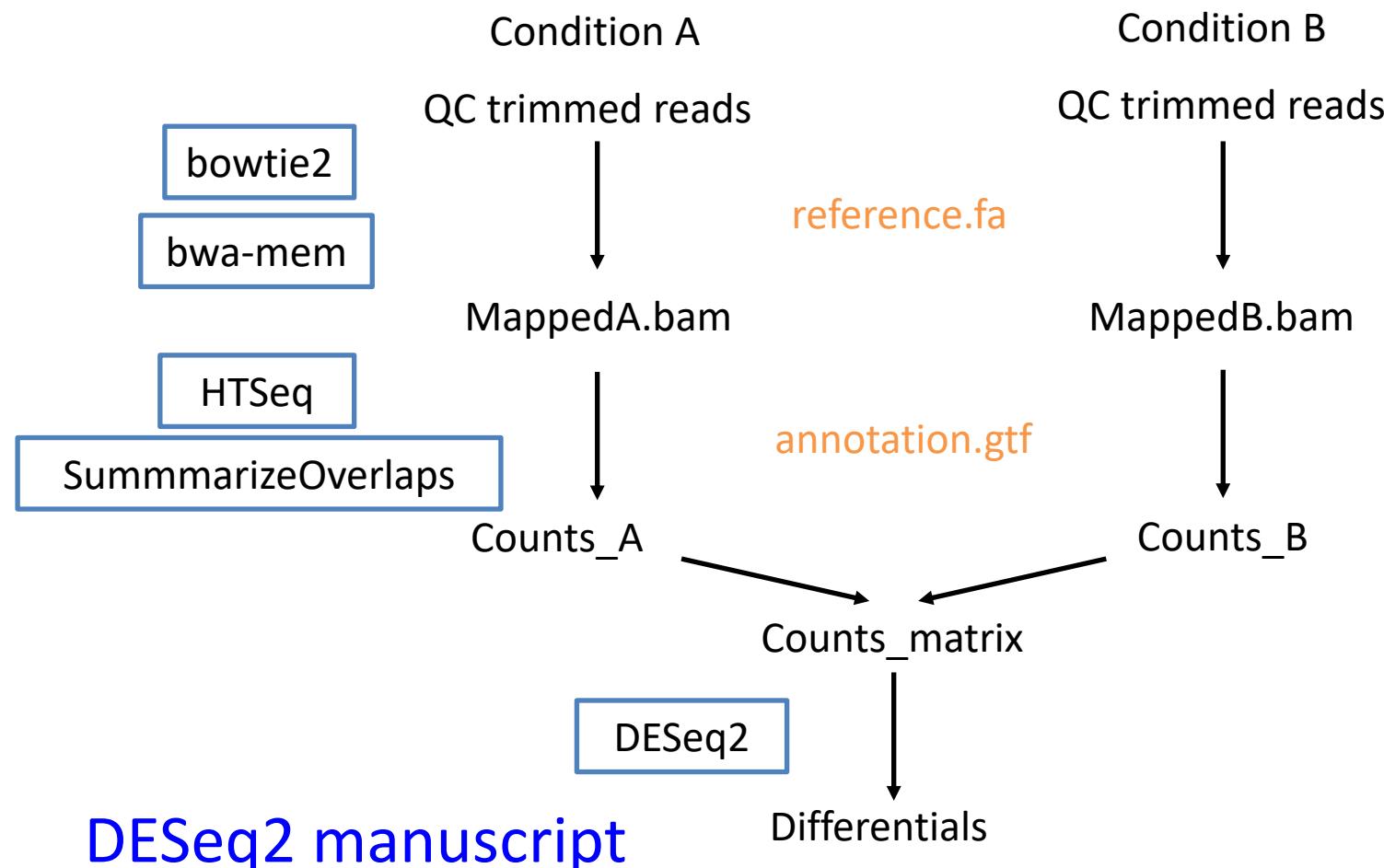
read_42 99 ref_genome 155 42 75M = 162 80 ATTTTGTTC <CCCCGGGFF AS:i:0 XN:i:0

Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	[!-?A-~]{1,254}	Query template NAME
2	FLAG	Int	[0,2 ¹⁶ -1]	bitwise FLAG
3	RNAME	String	* [!-()+-<>-~] [!-~]*	Reference sequence NAME
4	POS	Int	[0,2 ³¹ -1]	1-based leftmost mapping POSition
5	MAPQ	Int	[0,2 ⁸ -1]	MAPping Quality
6	CIGAR	String	* ([0-9]+[MIDNSHPX=])+	CIGAR string
7	RNEXT	String	* = [!-()+-<>-~] [!-~]*	Ref. name of the mate/next read
8	PNEXT	Int	[0,2 ³¹ -1]	Position of the mate/next read
9	TLEN	Int	[-2 ³¹ +1,2 ³¹ -1]	observed Template LENgth
10	SEQ	String	* [A-Za-z.=.]+	segment SEQuence
11	QUAL	String	[!-~]+	ASCII of Phred-scaled base QUALity+33

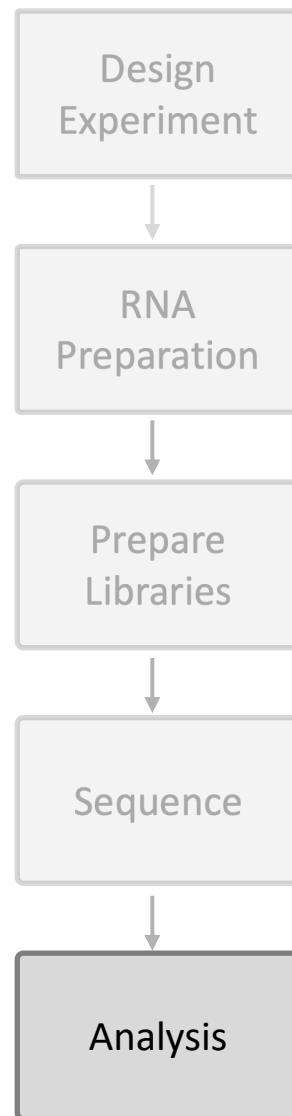
RNAseq – General Analysis Workflow



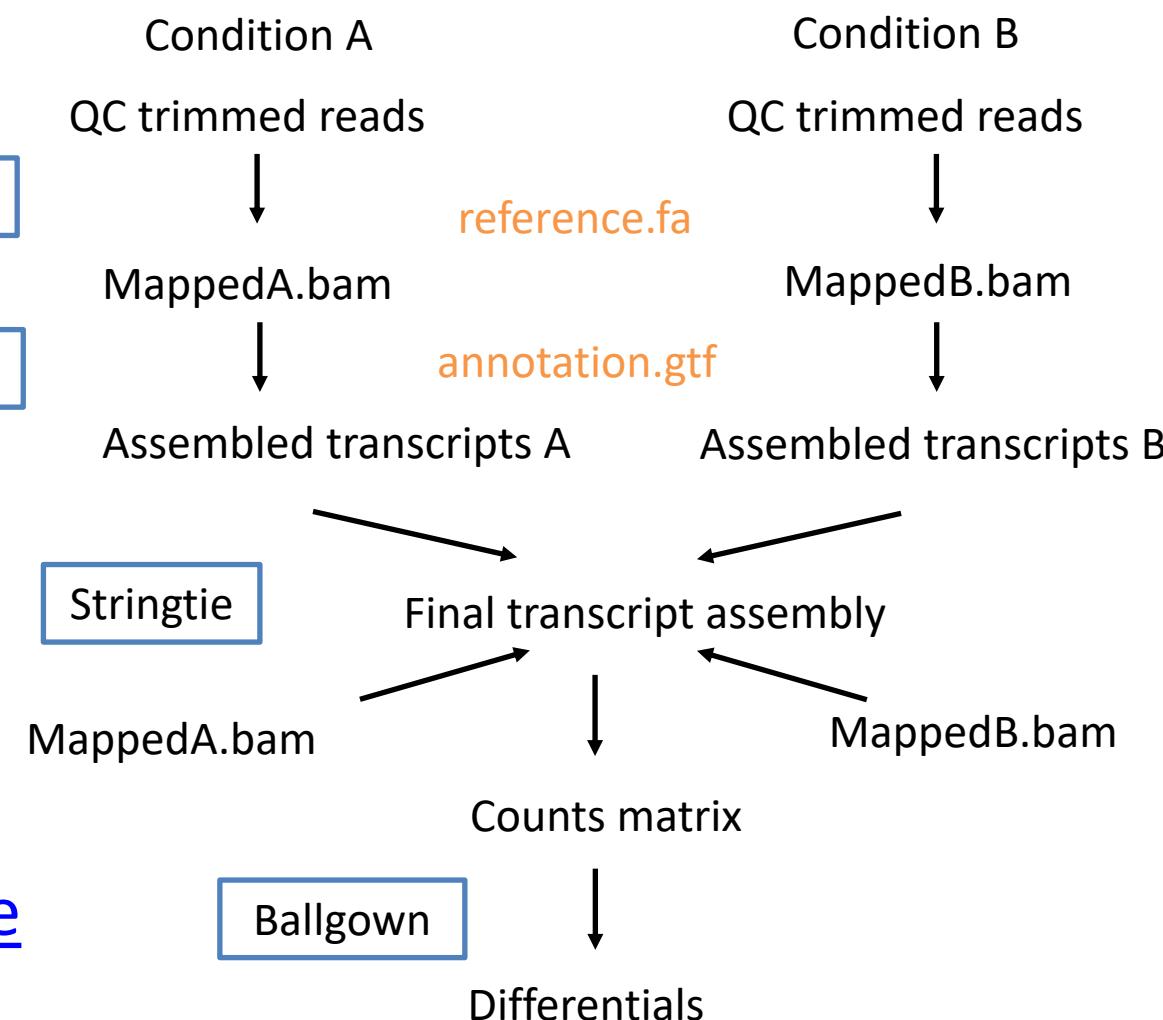
Prokaryote/Eukaryote – gene level



RNAseq – General Analysis Workflow

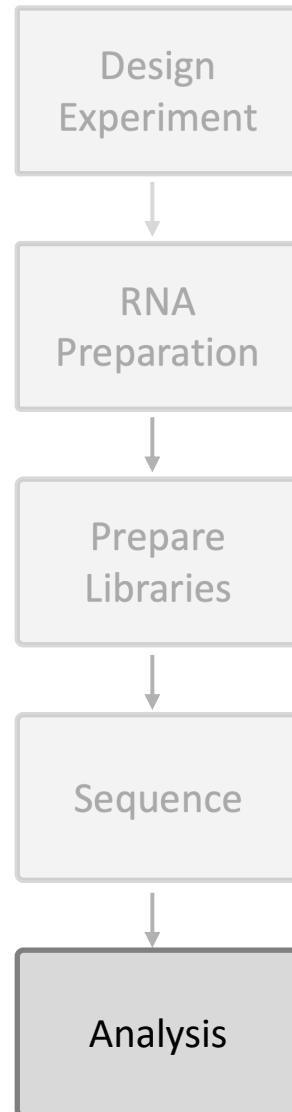


Eukaryote – transcript level



[Tuxedo suite](#)

RNAseq – Differential Gene Expression



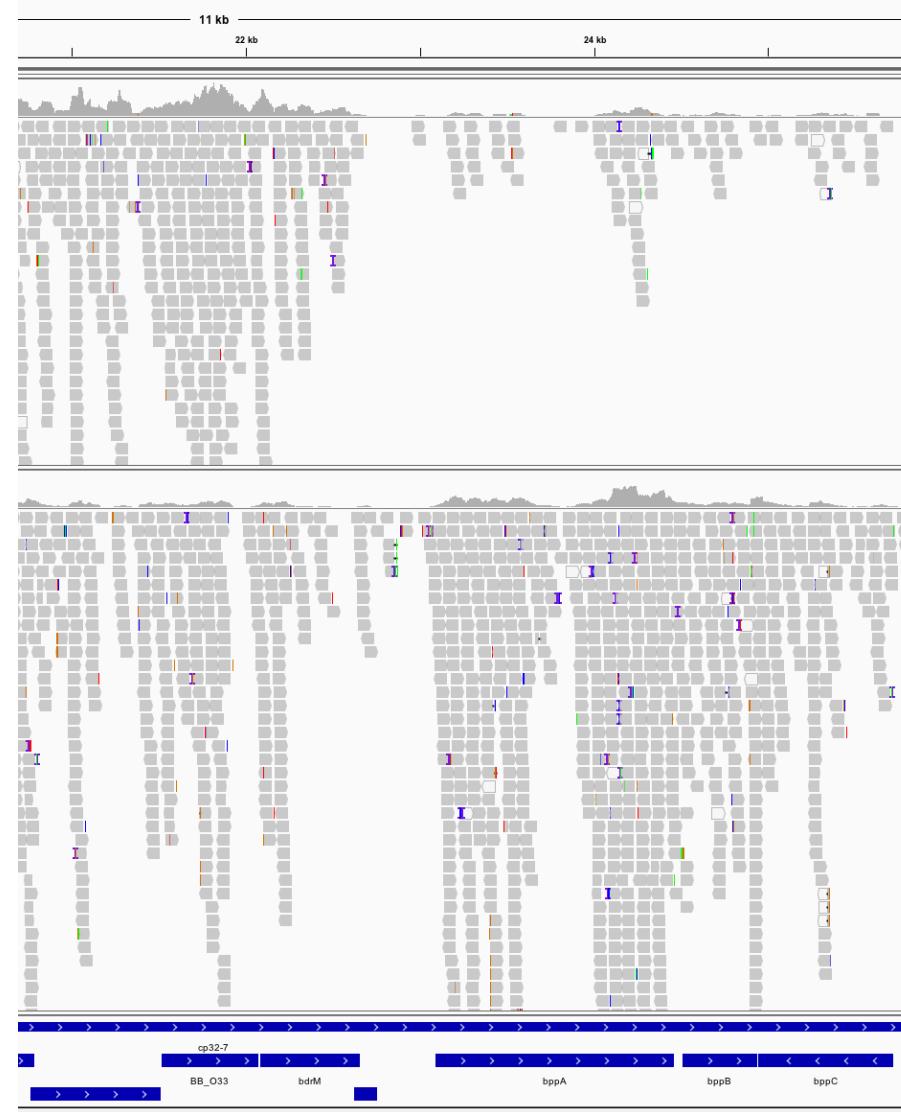
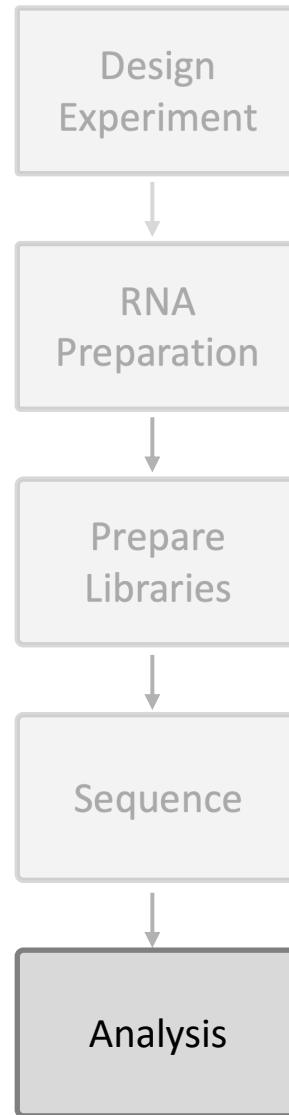
example RNAseq analysis pipeline with R

R is a free software environment for statistical computing and graphics

Rstudio is an integrated development environment (IDE) for R

Similar DESeq2 tutorial [here](#)

RNAseq – Differential Gene Expression w/ DESeq2



Simplest form, asking if ‘count’ between two treatments for a gene is not zero, reject null hypothesis

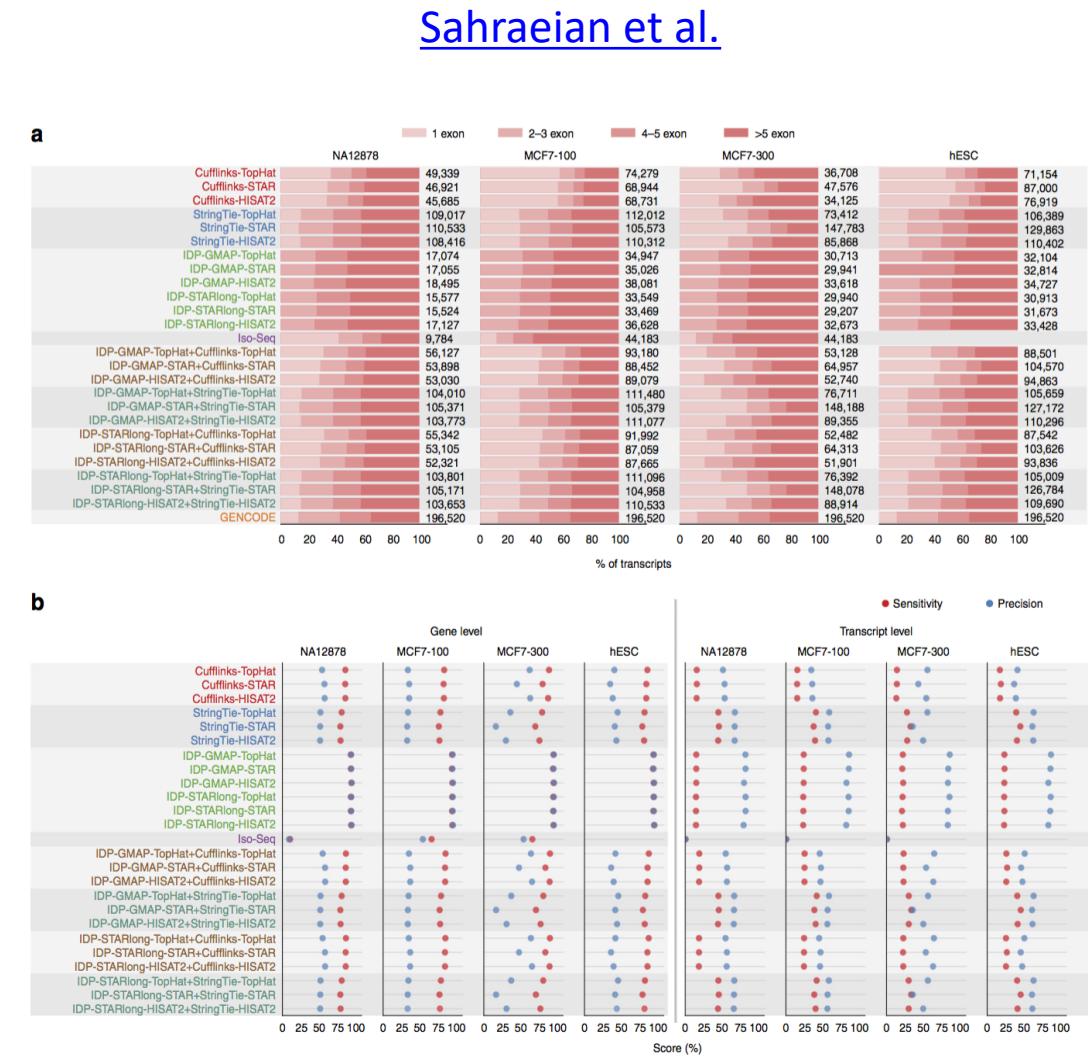
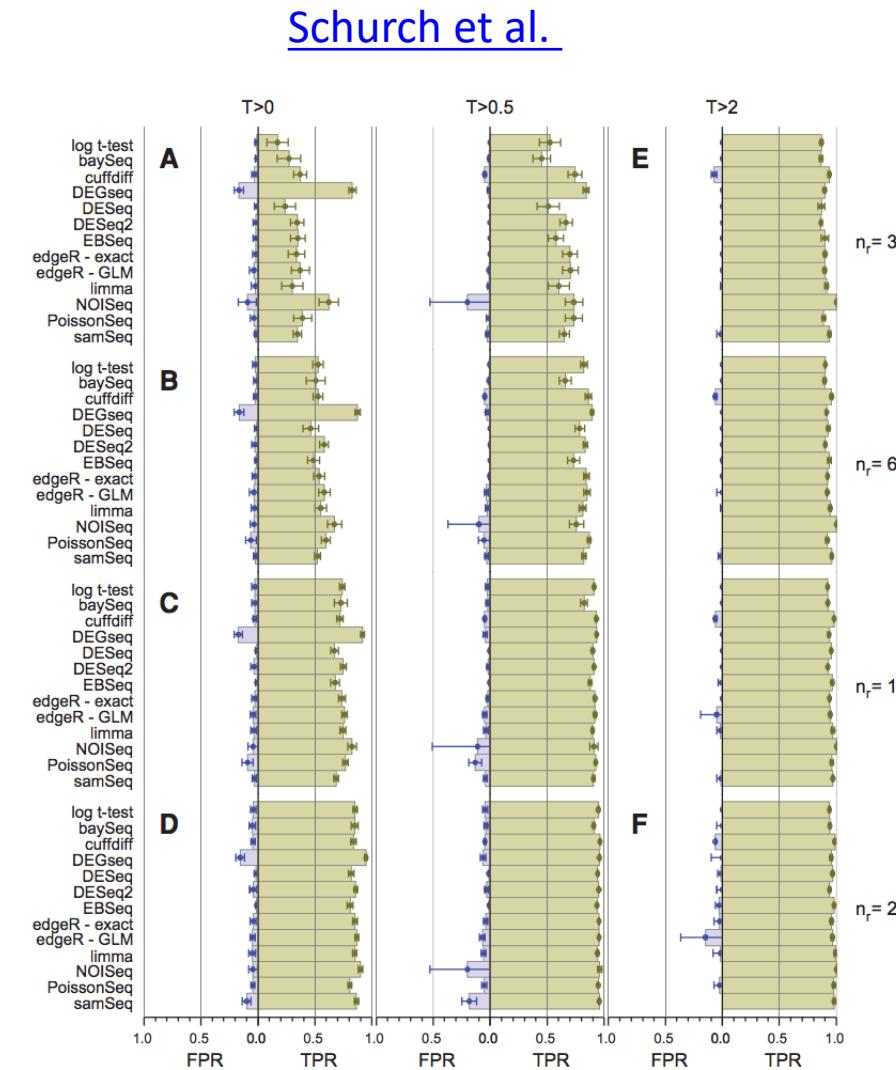
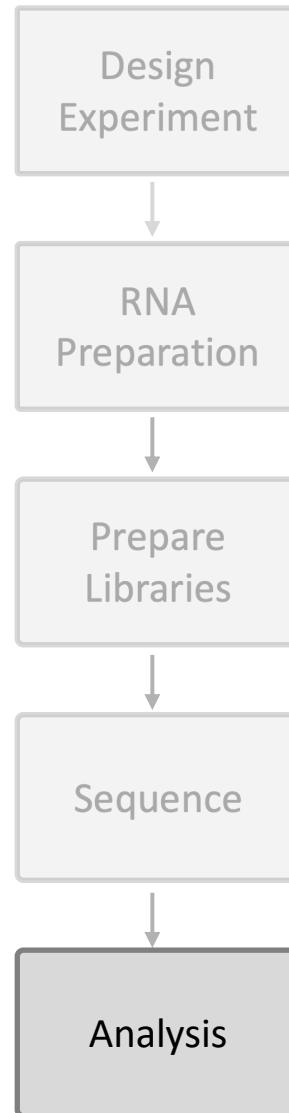
DESeq2 uses a generalized linear model where read counts are modeled using a Negative Binomial Distribution

- mean and variance are not equal (poisson)
- NB uses a gene-specific dispersion parameter

Wald test - estimated standard error of a log₂ fold change to test if it is equal to zero

Likelihood Ratio Test (LRT) - useful for testing multiple terms at once. Conceptually similar to ANOVA

RNAseq – Differential Gene Expression Tool Choice



Pulling project code from GitHub



Type the following in your favorite browser

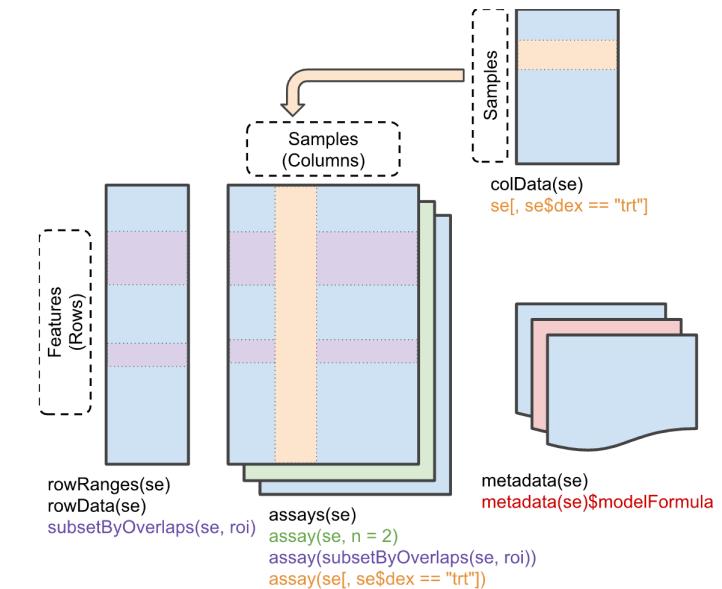
github.com/niaid/ACE_Uganda_2021_RNAseq

Click on 'Rstudio_server_instructions.txt'

RNAseq – R SummarizedExperiment

Goals for example RNAseq analysis

1. Generate a matrix of counts of reads mapped to each gene for each sample
 1. [SummarizedExperiment](#)
 2. Coordination of metadata (experimental design) with the count matrix
2. Sample exploration and QC
 1. Look at potential batch effects
3. Differential Expression analysis

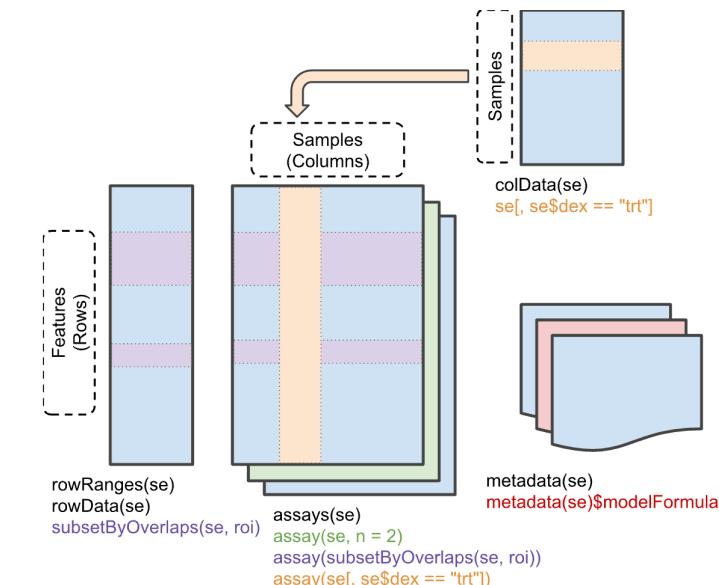


RNAseq – R SummarizedExperiment

Goals for example RNAseq analysis

1. Generate a matrix of counts of reads mapped to each gene for each sample
 1. [SummarizedExperiment](#)
 2. Coordination of metadata (experimental design) with the count matrix

Assay(se)											
rowRanges(se)			gene	con-1	con-2	con-3	con-4	trt-1	trt-2	trt-3	trt-4
gene feature start stop			BB_0002	136	145	131	101	260	169	217	322
BB_002 exon1 2357 2987			BB_0004	459	287	391	118	295	289	206	331
BB_002 exon2 3045 4012			BB_0005	780	591	724	237	471	450	338	470
BB_002 exon3 4098 5765			BB_0006	1427	1393	1712	680	1330	1164	842	1280
BB_003 exon1 12987 13456			BB_0007	959	603	775	420	708	777	402	483
BB_003 exon2 14556 16777			BB_0008	113	84	85	27	88	100	75	82
			BB_0009	92	64	56	14	49	53	35	52
			BB_0010	244	210	271	214	437	367	283	308
			BB_0011	543	372	515	170	423	544	264	265
			BB_0012	187	112	125	64	106	142	90	114
			BB_0013	81	30	39	15	46	35	31	54
			BB_0014	330	243	247	78	154	164	117	209
			BB_0015	1520	1239	1534	271	559	495	383	501
			BB_0016	373	102	158	74	128	151	70	95
			BB_0017	116	114	122	42	113	112	82	167



replicate	condition	time
con-1	ph76_C	2
con-2	ph76_C	2
con-3	ph76_C	4
con-4	ph76_C	4
trt-1	ph68_N	2
trt-2	ph68_N	2
trt-3	ph68_N	4
trt-4	ph68_N	4

RNAseq – R SummarizedExperiment

Example GFF

```
##gff-version 3.2.1
LmjF.04 TriTrypDB exon 1 1000 .
LmjF.04 TriTrypDB CDS 1 1000 .
LmjF.04 TriTrypDB exon 1001 2000 .
LmjF.04 TriTrypDB CDS 1001 2000 .
LmjF.04 TriTrypDB exon 2001 3000 .
LmjF.04 TriTrypDB CDS 2001 3000 .
LmjF.04 TriTrypDB exon 3001 4000 .
LmjF.04 TriTrypDB CDS 3001 4000 .
LmjF.04 TriTrypDB exon 4001 5000 .
LmjF.04 TriTrypDB CDS 4001 5000 .
LmjF.04 TriTrypDB exon 5001 6000 .
LmjF.04 TriTrypDB CDS 5001 6000 .
LmjF.04 TriTrypDB exon 6001 7000 .
LmjF.04 TriTrypDB CDS 6001 7000 .
LmjF.04 TriTrypDB exon 7001 8000 .
LmjF.04 TriTrypDB CDS 7001 8000 .
LmjF.04 TriTrypDB exon 8001 9000 .
LmjF.04 TriTrypDB CDS 8001 9000 .
LmjF.04 TriTrypDB exon 9001 10000 .
LmjF.04 TriTrypDB CDS 9001 10000 .

transcript_id "rna_LmjF.04.1"; gene_id "LmjF.04.1"; gene_name "LmjF.04.1";
transcript_id "rna_LmjF.04.1"; gene_id "LmjF.04.1"; gene_name "LmjF.04.1";
transcript_id "rna_LmjF.04.1001"; gene_id "LmjF.04.1001"; gene_name "LmjF.04.1001";
transcript_id "rna_LmjF.04.1001"; gene_id "LmjF.04.1001"; gene_name "LmjF.04.1001";
transcript_id "rna_LmjF.04.2001"; gene_id "LmjF.04.2001"; gene_name "LmjF.04.2001";
transcript_id "rna_LmjF.04.2001"; gene_id "LmjF.04.2001"; gene_name "LmjF.04.2001";
transcript_id "rna_LmjF.04.3001"; gene_id "LmjF.04.3001"; gene_name "LmjF.04.3001";
transcript_id "rna_LmjF.04.3001"; gene_id "LmjF.04.3001"; gene_name "LmjF.04.3001";
transcript_id "rna_LmjF.04.4001"; gene_id "LmjF.04.4001"; gene_name "LmjF.04.4001";
transcript_id "rna_LmjF.04.4001"; gene_id "LmjF.04.4001"; gene_name "LmjF.04.4001";
transcript_id "rna_LmjF.04.4001"; gene_id "LmjF.04.4001"; gene_name "LmjF.04.4001";
transcript_id "rna_LmjF.04.5001"; gene_id "LmjF.04.5001"; gene_name "LmjF.04.5001";
transcript_id "rna_LmjF.04.5001"; gene_id "LmjF.04.5001"; gene_name "LmjF.04.5001";
transcript_id "rna_LmjF.04.6001"; gene_id "LmjF.04.6001"; gene_name "LmjF.04.6001";
transcript_id "rna_LmjF.04.6001"; gene_id "LmjF.04.6001"; gene_name "LmjF.04.6001";
transcript_id "rna_LmjF.04.7001"; gene_id "LmjF.04.7001"; gene_name "LmjF.04.7001";
transcript_id "rna_LmjF.04.7001"; gene_id "LmjF.04.7001"; gene_name "LmjF.04.7001";
transcript_id "rna_LmjF.04.8001"; gene_id "LmjF.04.8001"; gene_name "LmjF.04.8001";
transcript_id "rna_LmjF.04.8001"; gene_id "LmjF.04.8001"; gene_name "LmjF.04.8001";
transcript_id "rna_LmjF.04.9001"; gene_id "LmjF.04.9001"; gene_name "LmjF.04.9001";
transcript_id "rna_LmjF.04.9001"; gene_id "LmjF.04.9001"; gene_name "LmjF.04.9001";
```

Reads that map to a gene in GTF file are added to a tally of read counts for that gene

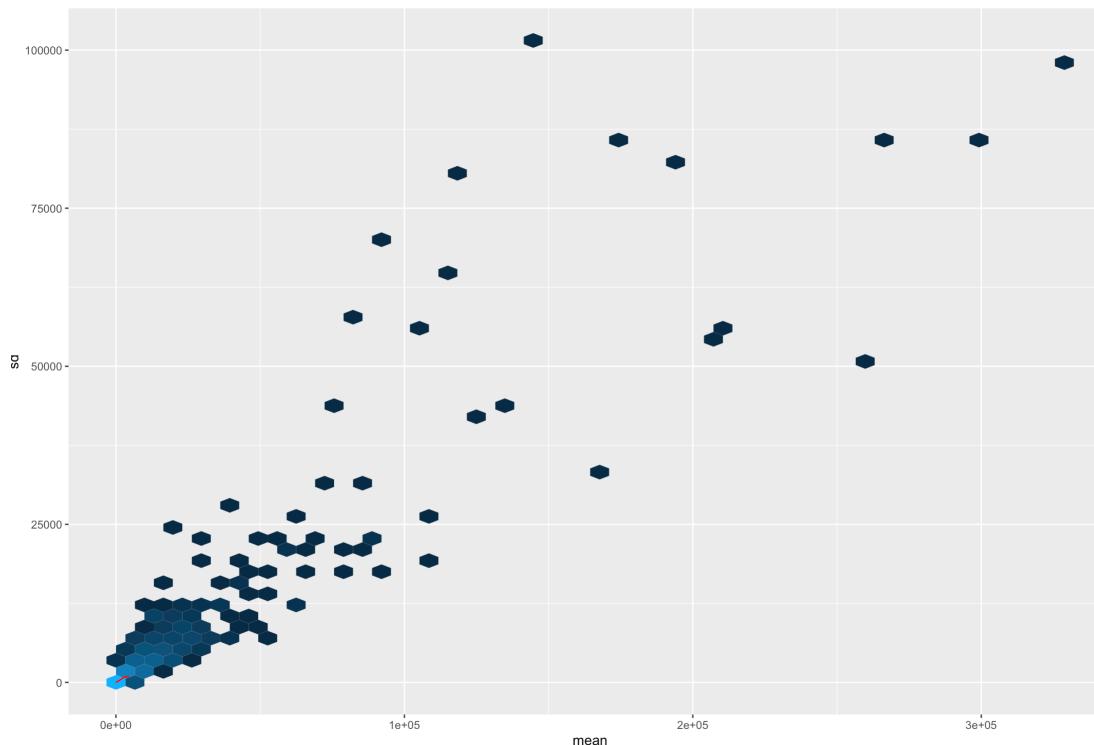
read_42 99 LmjF.04 4378 42 75M = 162 80 ATTTTGTTTT <CCCCGGGFF AS:i:0 XN:i:0



Coordinate read mapped to

RNAseq – R SummarizedExperiment

Mean read counts vs SD - RAW



Mean read counts vs SD - VST

