RNA-Seq Pre-processing

Lauren Vanderlinden

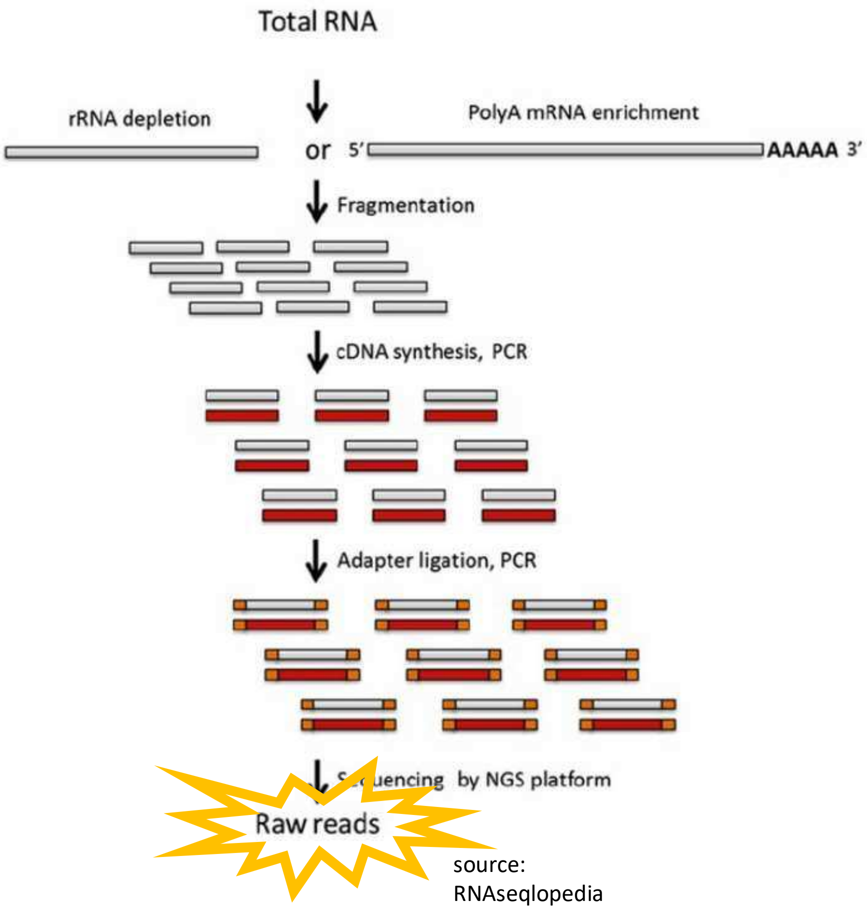
BIOS 6660

Spring 2019

Overview from Last Time

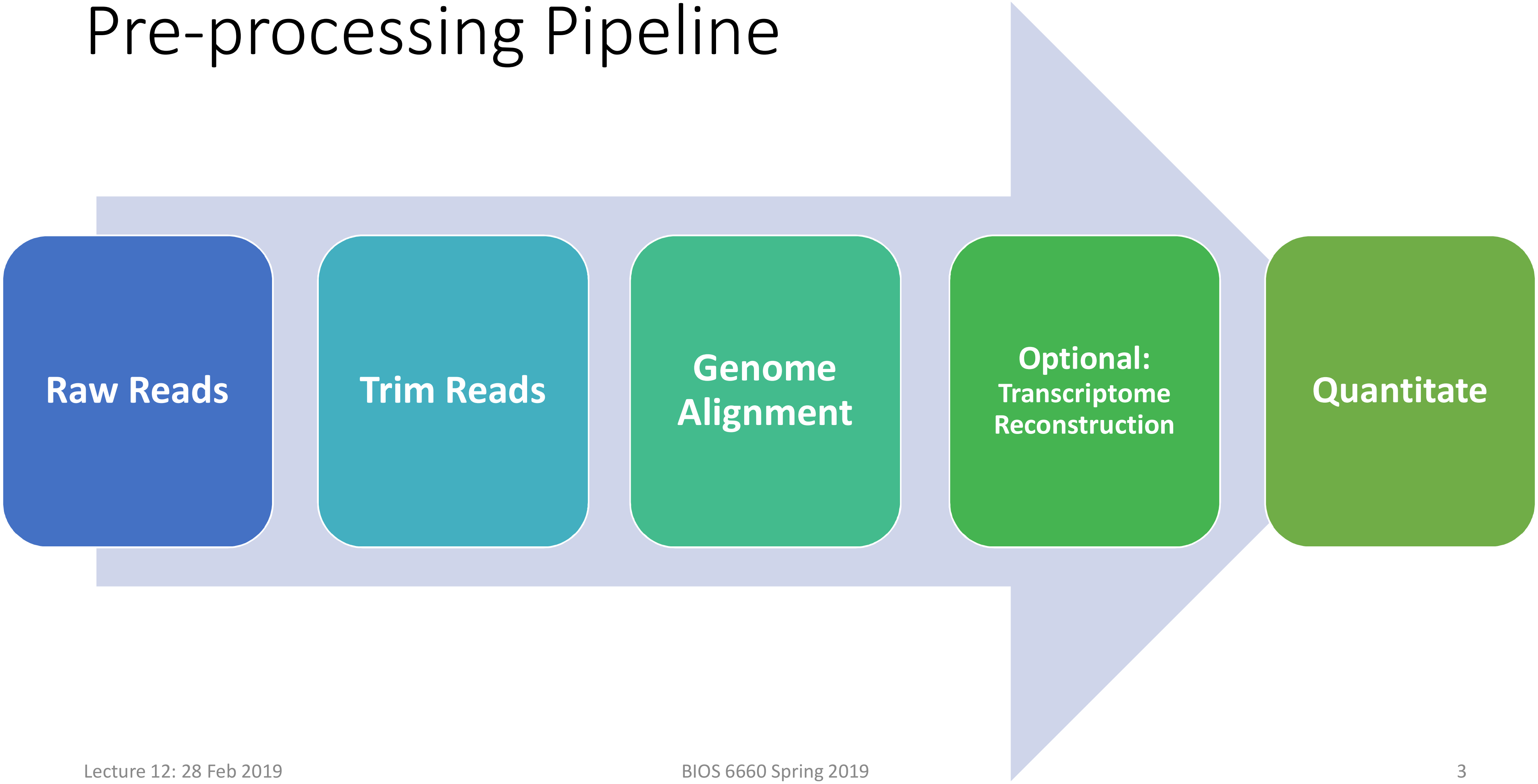
- Talked about RNA-Sequencing
- Importance of RNA expression
- Technology used to get sequences

- Now you have raw reads!



source: RNAseqlopedia

Pre-processing Pipeline


Raw Reads

Trim Reads

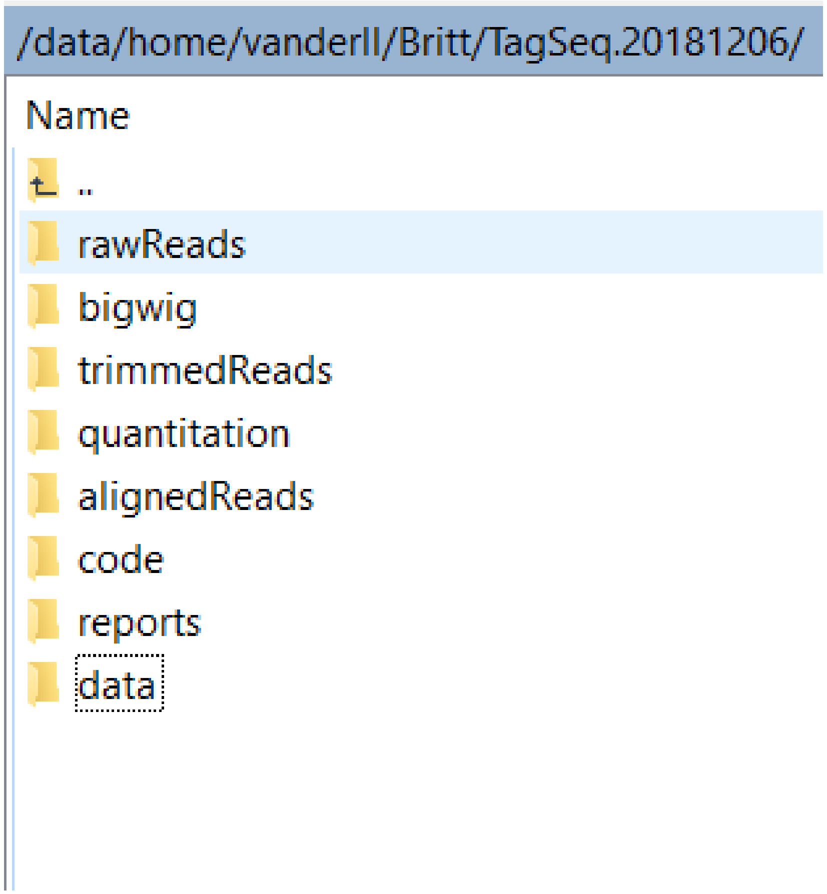
Genome Alignment

Optional: Transcriptome Reconstruction

Quantitate

Folder Structure

- For a typical coverage where the main goal is differential expression (25-30 million reads/sample) raw files range around 2-3.8 GB
- Find it useful to have these files in separate folders for each part of the process:
    - rawReads
    - trimmedReads
    - alignedReads
    - quantitation
    - data (matrix form finally)
- Suggestion only



/data/home/vanderll/Britt/TagSeq.20181206/

Name

..

rawReads

bigwig

trimmedReads

quantitation

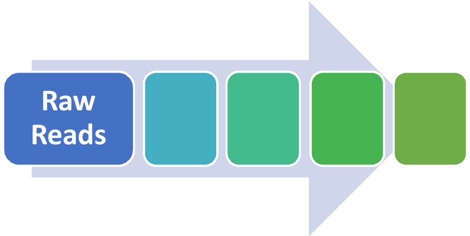
alignedReads

code

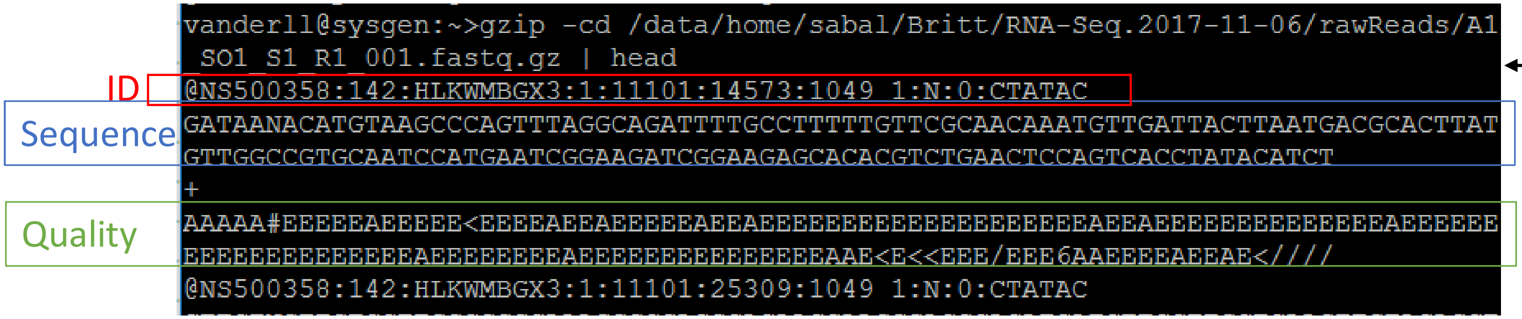
reports

data

FASTQ Format



Notice I’m using the pipping Pam talked about

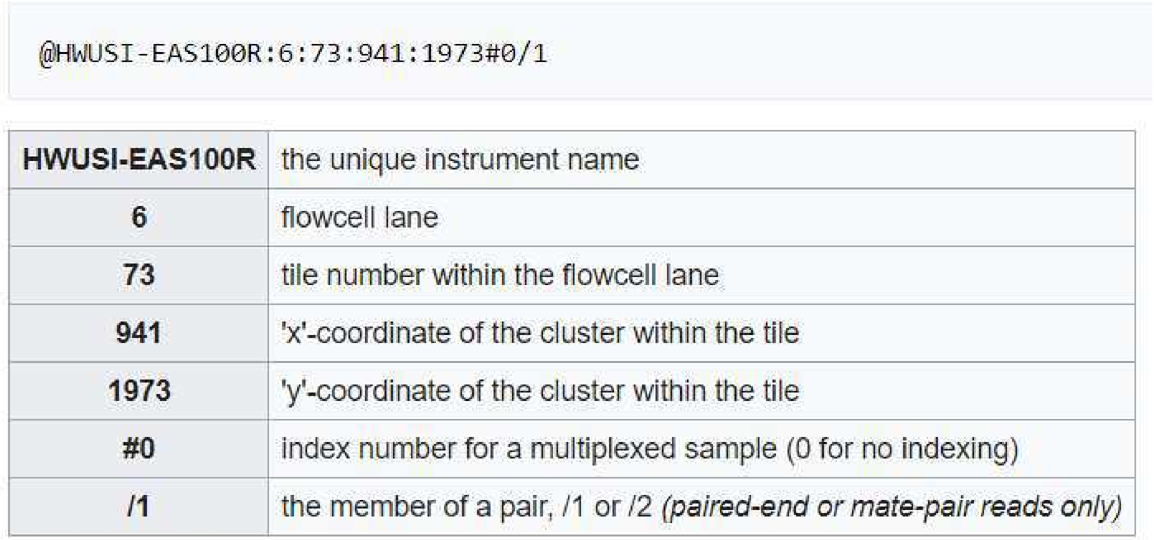


```
vanderll@sysgen:~>gzip -cd /data/home/sabal/Britt/RNA-Seq.2017-11-06/rawReads/A1
_SO1_S1_R1_001.fastq.gz | head
@NS500358:142:HLKWMBGX3:1:11101:14573:1049 1:N:0:CTATAC
GATAANACATGTAAGCCCAGTTTAGGCAGATTTTGCCTTTTTGTTCGCAACAAATGTTGATTACTTAATGACGCACTTAT
GTTGGCCGTGCAATCCATGAATCGGAAGATCGGAAGAGCACACGTCTGAACTCCAGTCACCTATACATCT
+
AAAAA#EEEEEAEEEEE<EEEEAEEAEEEEEAEEAEEEEEEEEEEEEEEEEEEEEAEEAEEEEEEEEEEEEEEAEEEEEE
EEEEEEEEEEEEEEAEEEEEEEEEAEEEEEEEEEEEEEEEAAE<E<<EEE/EEE6AAEEEEAEEAE<////
@NS500358:142:HLKWMBGX3:1:11101:25309:1049 1:N:0:CTATAC
```

- Single-end technology will give you 1 fastq file/sample
- Paired end technology will give you 2 fastq files/sample
- 4 lines for each read

1. Unique Read ID: begins with '@' character and is followed by a sequence identifier and an *optional* description (like a FASTA title line).
2. Raw sequence letters
3. Begins with a '+' character and is *optionally* followed by the same sequence identifier (and any description) again.
4. Encodes the quality values for the sequence in Line 2, and must contain the same number of symbols as letters in the sequence.

FASTQ Illumina ID Format



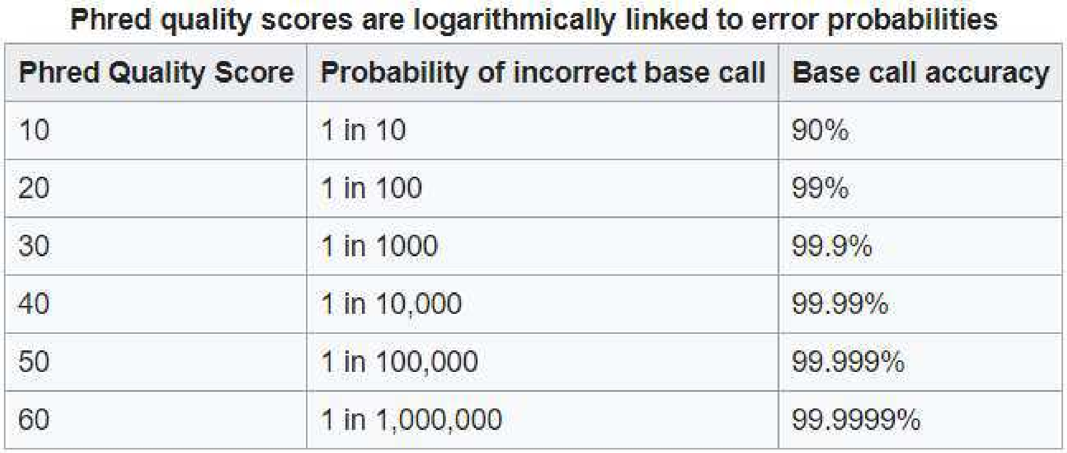
```
@HWUSI-EAS100R:6:73:941:1973#0/1
```

HWUSI-EAS100R	the unique instrument name
6	flowcell lane
73	tile number within the flowcell lane
941	'x'-coordinate of the cluster within the tile
1973	'y'-coordinate of the cluster within the tile
#0	index number for a multiplexed sample (0 for no indexing)
/1	the member of a pair, /1 or /2 *(paired-end or mate-pair reads only)*

Source: wikipedia

Phred Quality (Q) Score

- A score assigned to each individual base called
- Developed to help in the automation of DNA sequencing in the Human Genome Project
- Phred score is logarithmically related to the base-calling error probabilities



Phred quality scores are logarithmically linked to error probabilities

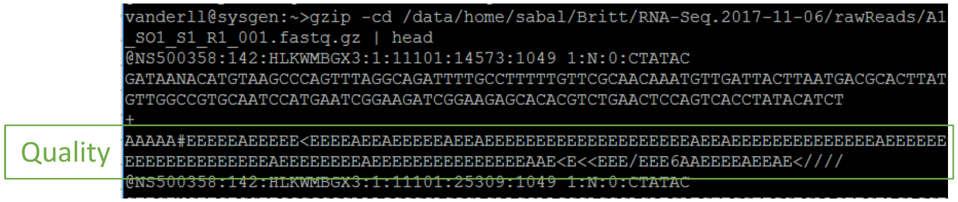
Phred Quality Score	Probability of incorrect base call	Base call accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.9%
40	1 in 10,000	99.99%
50	1 in 100,000	99.999%
60	1 in 1,000,000	99.9999%

Source: wikipedia

Illumina standards say Phred of 30 or higher considered “high quality”

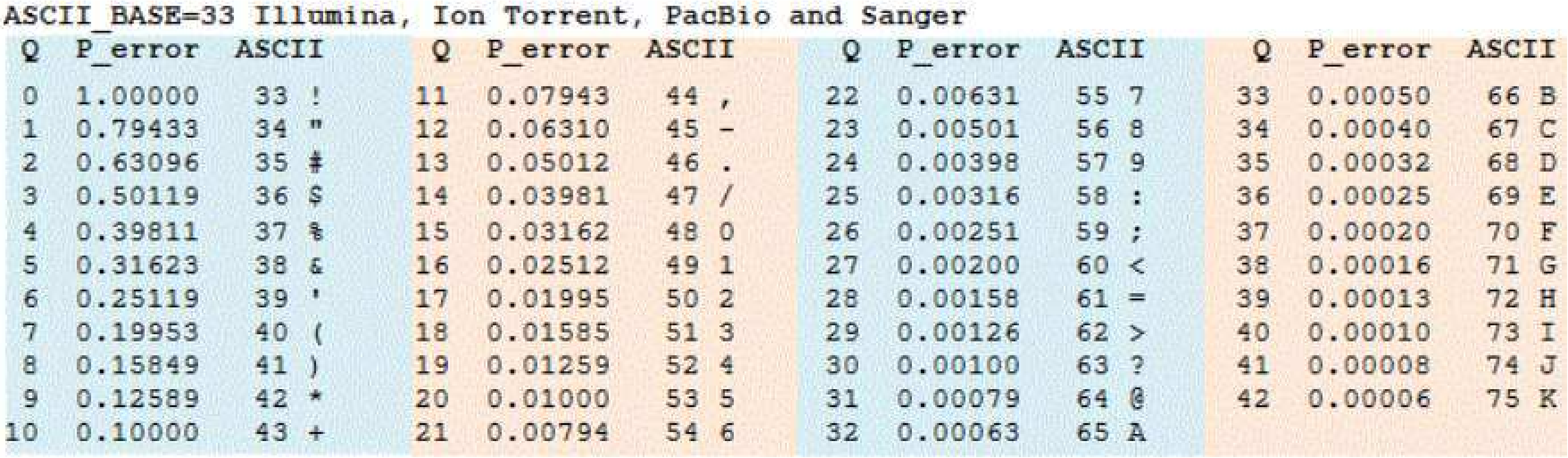
FASTQ and Phred Quality Score

- FASTQ score represented as ASCII-33 characters



```
vanderll@sysgen:~>gzip -cd /data/home/sabal/Britt/RNA-Seq.2017-11-06/rawReads/A1
_SO1_S1_R1_001.fastq.gz | head
@NS500358:142:HLKWMBGX3:1:11101:14573:1049 1:N:0:CTATAC
GATAANACATGTAAGCCCAGTTTAGGCAGATTTTGCCTTTTTGTTCGCAACAAATGTTGATTACTTAATGACGCACTTAT
GTTGGCCGTGCAATCCATGAATCGGAAGATCGGAAGAGCACACGTCTGAACTCCAGTCACCTATACATCT
+
AAAAA#EEEEEAEEEEE<EEEEAEEAEEEEEAEEAEEEEEEEEEEEEEEEEEEEEAEEAEEEEEEEEEEEEEEEAEEEEEE
EEEEEEEEEEEEEEAEEEEEEEEEEAEEEEEEEEEEEEEEEEEAAE<E<<EEE/EEE6AAEEEEAEEAE<////
@NS500358:142:HLKWMBGX3:1:11101:25309:1049 1:N:0:CTATAC
```

Quality



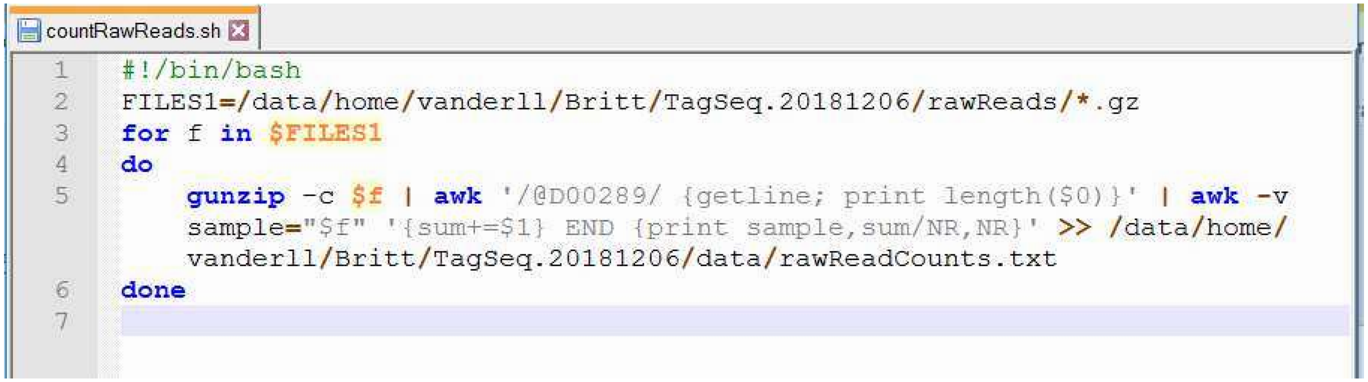
ASCII_BASE=33 Illumina, Ion Torrent, PacBio and Sanger

Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII
0	1.00000	33 !	11	0.07943	44 ,	22	0.00631	55 7	33	0.00050	66 B
1	0.79433	34 "	12	0.06310	45 -	23	0.00501	56 8	34	0.00040	67 C
2	0.63096	35 #	13	0.05012	46 .	24	0.00398	57 9	35	0.00032	68 D
3	0.50119	36 $	14	0.03981	47 /	25	0.00316	58 :	36	0.00025	69 E
4	0.39811	37 %	15	0.03162	48 0	26	0.00251	59 ;	37	0.00020	70 F
5	0.31623	38 &	16	0.02512	49 1	27	0.00200	60 <	38	0.00016	71 G
6	0.25119	39 '	17	0.01995	50 2	28	0.00158	61 =	39	0.00013	72 H
7	0.19953	40 (	18	0.01585	51 3	29	0.00126	62 >	40	0.00010	73 I
8	0.15849	41)	19	0.01259	52 4	30	0.00100	63 ?	41	0.00008	74 J
9	0.12589	42 *	20	0.01000	53 5	31	0.00079	64 @	42	0.00006	75 K
10	0.10000	43 +	21	0.00794	54 6	32	0.00063	65 A			

Source: USEARCH

How many reads do I have?

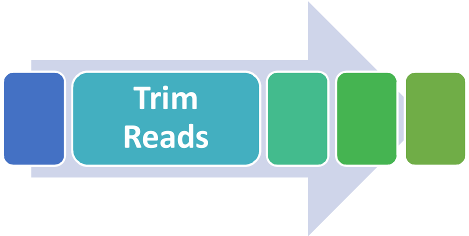
- One of the first questions before anything else is how many reads do my samples have?
- Make a bash script to count the number of reads you have



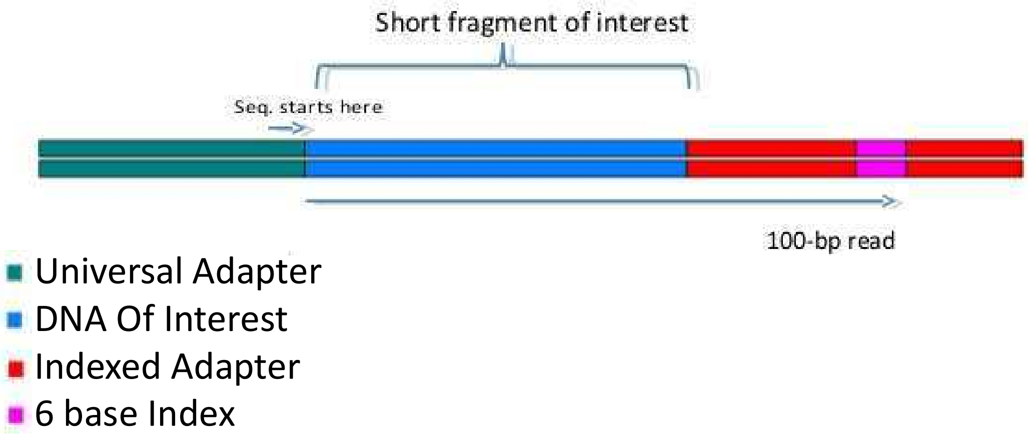
```
#!/bin/bash
FILES1=/data/home/vanderll/Britt/TagSeq.20181206/rawReads/*.gz
for f in $FILES1
do
    gunzip -c $f | awk '/@D00289/ {getline; print length($0)}' | awk -v
    sample="$f" '{sum+=$1} END {print sample,sum/NR,NR}' >> /data/home/
    vanderll/Britt/TagSeq.20181206/data/rawReadCounts.txt
done
```

- Notice I’m counting the number of lines which start with @D00289, not the total lines in the file
- Writes a text file with your fastq name and number of reads

Trim Reads



- Adapter Trimming
  - Trimming off the adapter sequences of reads
  - ABSOLUTE MUST for small RNA
  - Improves *de novo* assemblies
- Quality Trimming
  - May increase mapping rates
  - May also loose information
- Software Programs
  - BBDuk
  - Cutadapt
  - Trim Galore!
  - PRINSEQ
  - Trimmomatic
  - Sickle/Sythe
  - FASTX Toolkit

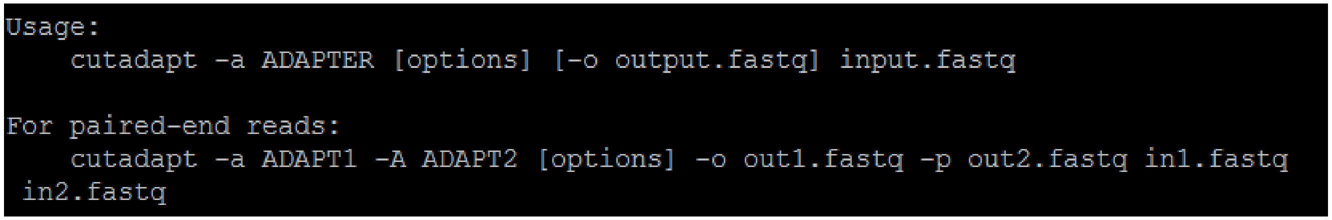


Source: SciLifeLab

Cutadapt code

Extra options I’ve used:
-q quality score cut-off before adapter removal
-m minimum length if read to keep after trimming
-u remove a specified length of bases from beginning

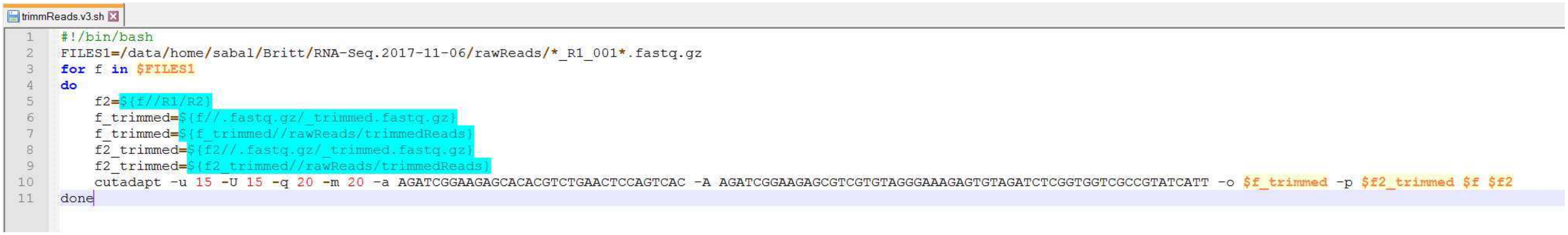
- Cutadapt is on yampa



```
Usage:
    cutadapt -a ADAPTER [options] [-o output.fastq] input.fastq

For paired-end reads:
    cutadapt -a ADAPT1 -A ADAPT2 [options] -o out1.fastq -p out2.fastq in1.fastq
 in2.fastq
```

- Example bash script



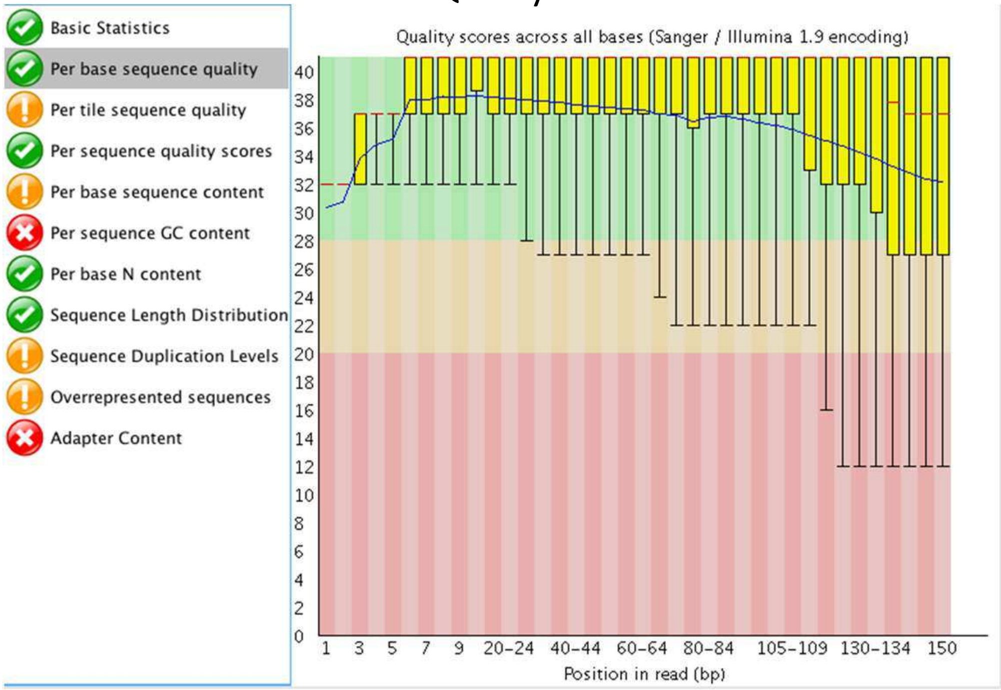
```
#!/bin/bash
FILES1=/data/home/sabal/Britt/RNA-Seq.2017-11-06/rawReads/*_R1_001*.fastq.gz
for f in $FILES1
do
    f2=${f//R1/R2}
    f_trimmed=${f//.fastq.gz/_trimmed.fastq.gz}
    f_trimmed=${f_trimmed//rawReads/trimmedReads}
    f2_trimmed=${f2//.fastq.gz/_trimmed.fastq.gz}
    f2_trimmed=${f2_trimmed//rawReads/trimmedReads}
    cutadapt -u 15 -U 15 -q 20 -m 20 -a AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC -A AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTAGATCTCGGTGGTCGCCGTATCATT -o $f_trimmed -p $f2_trimmed $f $f2
done
```

FastQC

```
fastqc /path/sample.fastq.gz
```

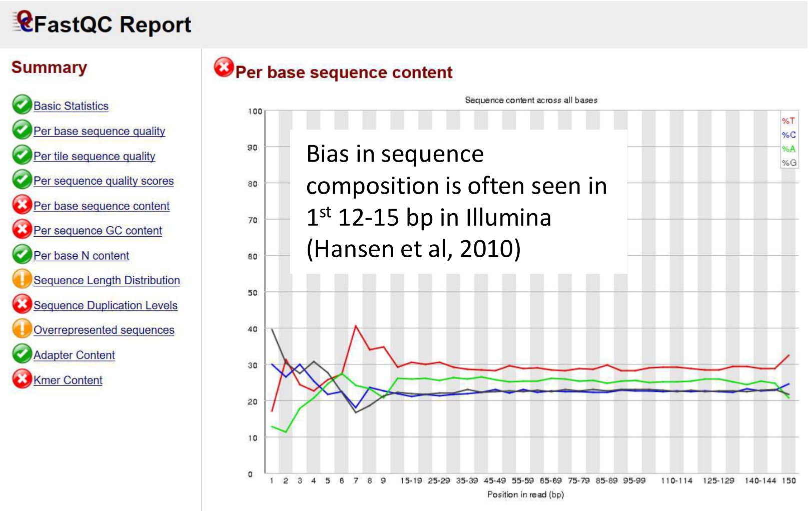
- FastQC installed on yampa
- Program which evaluates quality of reads
- Use this to check the trimming went well
- Want to stay in the green zone
- Really want green checks on:
  - Per sequence quality scores
  - Per base sequence content
- Look at example report

Quality Score Across Bases



- ✓ Basic Statistics
- ✓ Per base sequence quality
- ! Per tile sequence quality
- ✓ Per sequence quality scores
- ! Per base sequence content
- ✗ Per sequence GC content
- ✓ Per base N content
- ✓ Sequence Length Distribution
- ! Sequence Duplication Levels
- ! Overrepresented sequences
- ✗ Adapter Content

FastQC Pre and Post Quality Trimming



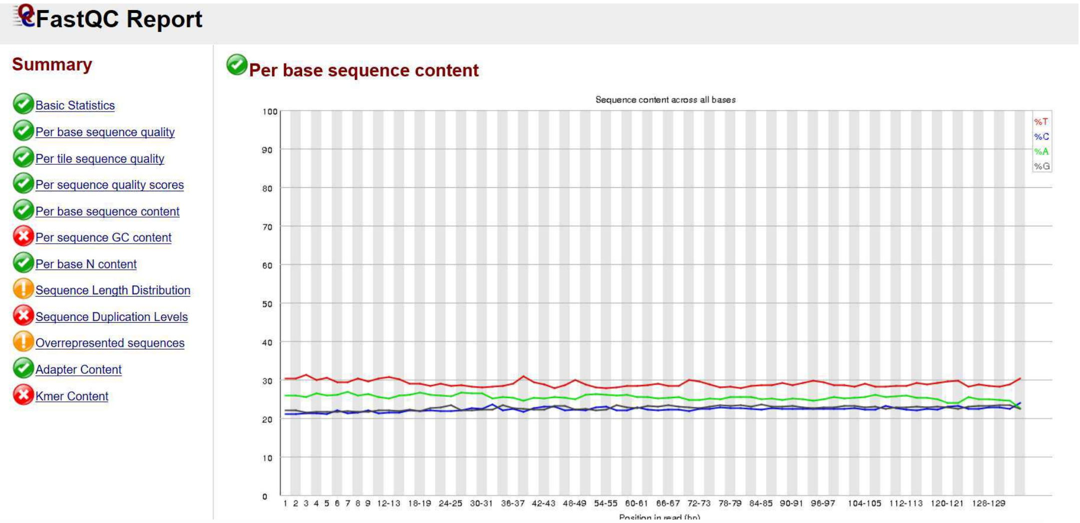
FastQC Report

Summary

- ✓ Basic Statistics
- ✓ Per base sequence quality
- ✓ Per tile sequence quality
- ✓ Per sequence quality scores
- ✗ Per base sequence content
- ✗ Per sequence GC content
- ✓ Per base N content
- ! Sequence Length Distribution
- ✗ Sequence Duplication Levels
- ! Overrepresented sequences
- ✓ Adapter Content
- ✗ Kmer Content

✗ Per base sequence content

Bias in sequence composition is often seen in 1st 12-15 bp in Illumina (Hansen et al, 2010)

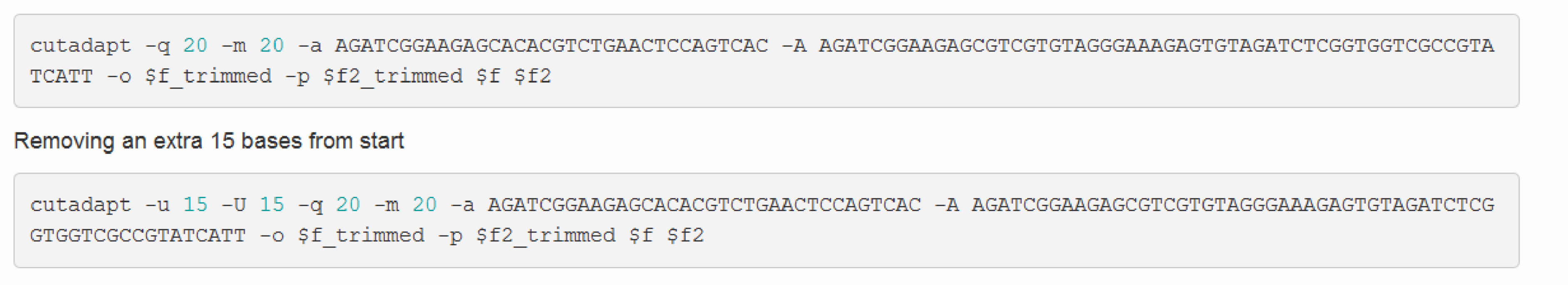


FastQC Report

Summary

- ✓ Basic Statistics
- ✓ Per base sequence quality
- ✓ Per tile sequence quality
- ✓ Per sequence quality scores
- ✓ Per base sequence content
- ✗ Per sequence GC content
- ✓ Per base N content
- ! Sequence Length Distribution
- ✗ Sequence Duplication Levels
- ! Overrepresented sequences
- ✓ Adapter Content
- ✗ Kmer Content

✓ Per base sequence content

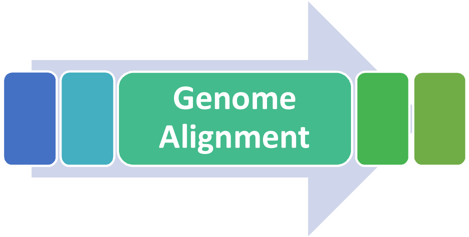


```
cutadapt -q 20 -m 20 -a AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC -A AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTAGATCTCGGTGGTCGCCGTA
TCATT -o $f_trimmed -p $f2_trimmed $f $f2
```

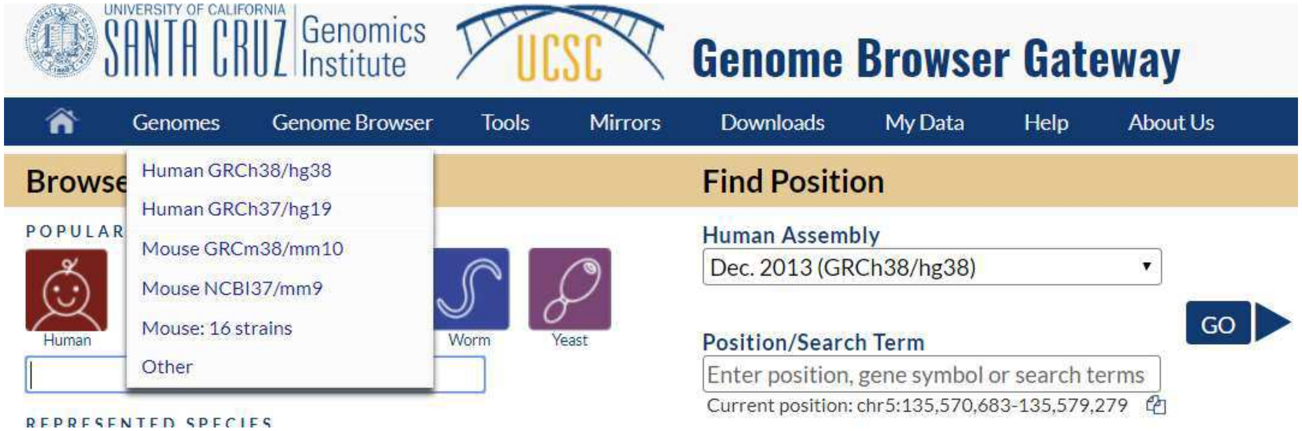
Removing an extra 15 bases from start

```
cutadapt -u 15 -U 15 -q 20 -m 20 -a AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC -A AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTAGATCTCG
GTGGTCGCCGTATCATT -o $f_trimmed -p $f2_trimmed $f $f2
```

Align Reads to Genome



- Align reads to a reference genome
  - Humans: Genome Reference Consortium human genome build 38 (GRCh38, aka hg38)
    - hg19 is still commonly used



Find Position

Human Assembly: Dec. 2013 (GRCh38/hg38)

Position/Search Term: Enter position, gene symbol or search terms

Current position: chr5:135,570,683-135,579,279

Species	Most recent reference genome
Fruit fly	dm6
Mouse	mm10
Rat	rn6
Human	hg38

Alignment Tools

1. Unspliced Aligners (No Gaps Allowed)
    - BWA
    - Bowtie/Bowtie2
    - NovoAlign, SeqMap, SEAL
2. Spliced Aligners (Allows Splice-Junctions)
    - Erange
    - SpliceSeq
    - BBMap
    - Hisat/Hisat2
    - STAR
    - TopHat

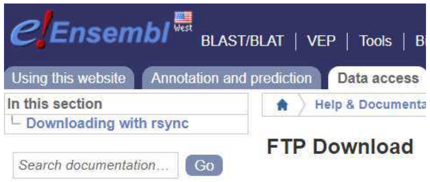
Annotated Guided Aligners

De Novo Splice Aligners

Hisat2 Reference Genome

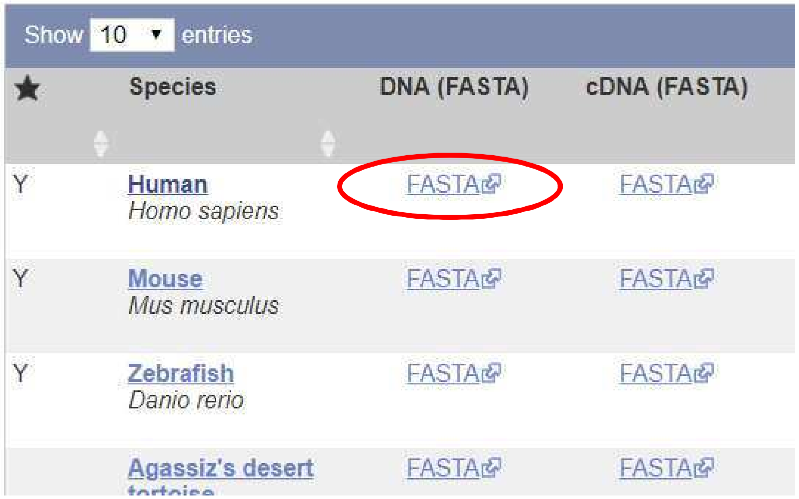
- Hisat2 is on yampa
- For any alignment tool, you need reference files

```
hisat2-build reference.fa reference.hisat
```

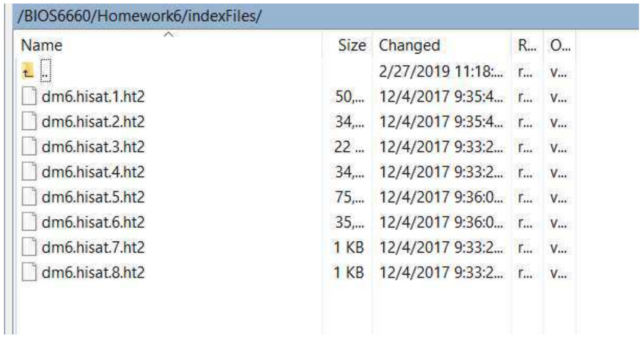


FTP Download

https://uswest.ensembl.org/info/data/ftp/index.html

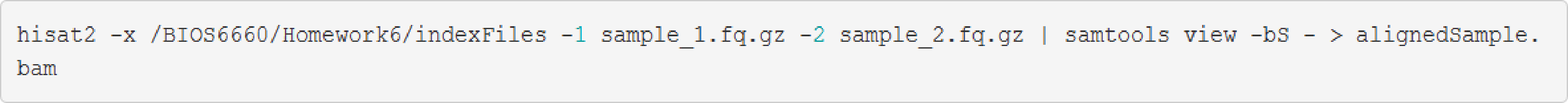


★	Species	DNA (FASTA)	cDNA (FASTA)
Y	Human *Homo sapiens*	FASTA	FASTA
Y	Mouse *Mus musculus*	FASTA	FASTA
Y	Zebrafish *Danio rerio*	FASTA	FASTA
	Agassiz's desert tortoise	FASTA	FASTA


/BIOS6660/Homework6/indexFiles/

Name	Size	Changed	R...	O...
..		2/27/2019 11:18:...	r...	v...
dm6.hisat.1.ht2	50,...	12/4/2017 9:35:4...	r...	v...
dm6.hisat.2.ht2	34,...	12/4/2017 9:35:4...	r...	v...
dm6.hisat.3.ht2	22 ...	12/4/2017 9:33:2...	r...	v...
dm6.hisat.4.ht2	34,...	12/4/2017 9:33:2...	r...	v...
dm6.hisat.5.ht2	75,...	12/4/2017 9:36:0...	r...	v...
dm6.hisat.6.ht2	35,...	12/4/2017 9:36:0...	r...	v...
dm6.hisat.7.ht2	1 KB	12/4/2017 9:33:2...	r...	v...
dm6.hisat.8.ht2	1 KB	12/4/2017 9:33:2...	r...	v...

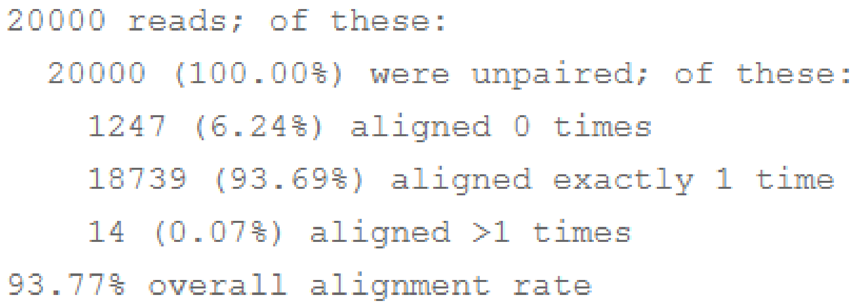
Hisat2 Alignment



```
hisat2 -x /BIOS6660/Homework6/indexFiles -1 sample_1.fq.gz -2 sample_2.fq.gz | samtools view -bS - > alignedSample.
bam
```

- Options you might want:
  - --un writing an output file for those that don't align
  - -N # mismatches you will allow (default 0)
- For paired-end reads, you will now go from 2 files/sample down to 1 file/sample
- BAM output file is a compressed binary version of a SAM file
- SAM = Sequence Alignment/Map format

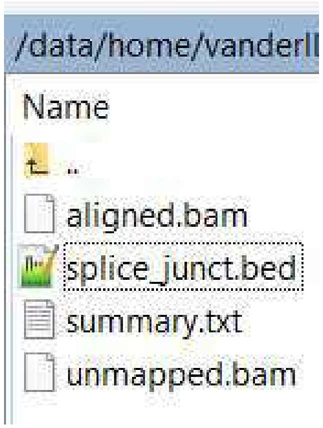
Message it prints after running:



```
20000 reads; of these:
  20000 (100.00%) were unpaired; of these:
    1247 (6.24%) aligned 0 times
    18739 (93.69%) aligned exactly 1 time
    14 (0.07%) aligned >1 times
93.77% overall alignment rate
```

Batch Alignment

- Python script called: alignBatch.genome.hisat.py
- Written by Spencer Mahaffey (thank you!)
- This gives you 4 files:
  - Aligned.bam
  - Splice_junct.bed
  - Summary.txt
  - Unmapped.bam
- Look at py script



/data/home/vanderll

Name

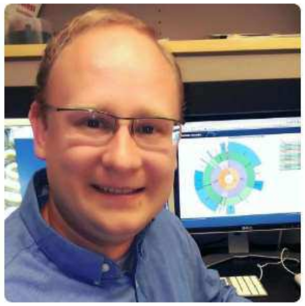
..

aligned.bam

splice_junct.bed

summary.txt

unmapped.bam



Spencer Mahaffey

smahaffey

```
python /data/home/vanderll/teaching/BIOS6660_spring2019/programs/alignBatch.genome.hisat.py --input-s
uffix .fastq.gz --index-dir /data/home/vanderll/annotation/dm6.hisat2.reference -P /data/home/vanderl
l/teaching/BIOS6660_spring2019/data/hw6/subsetFASTQpairs
```

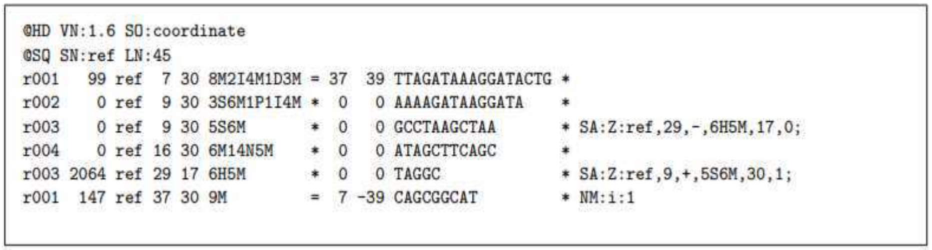
Alignment Error Example

- I got this following error after alignment one time
- Saying that there is a read that has more bases than scores assigned to it
- It did output a bam file up to this point
- What the issue was that the core uploaded the fastq files to the web and at some point their connection was interrupted so the fastq file was only partially uploaded

```
[samopen] SAM header is present: 1870 sequences.
Error: Read D00289:29:CCT5JANXX:5:2316:4056:36315
1:N:0:CAGGCG+CTTGTA has more read characters than
quality values.
terminate called after throwing an instance of 'int'
(ERR): hisat2-align died with signal 6 (ABRT) (core
dumped)
```

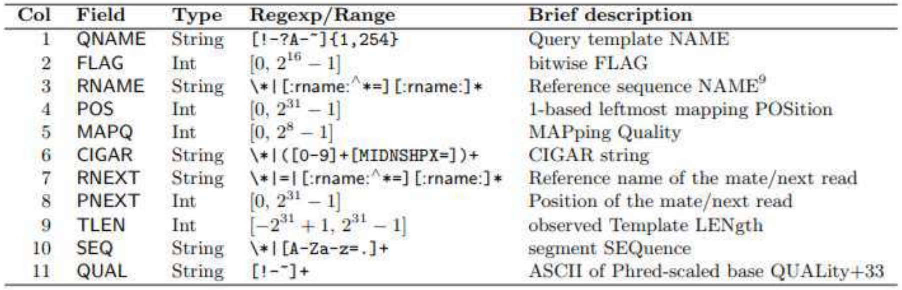
BAM Files: Alignment Output

```
samtools view sample.sorted.bam | head
```



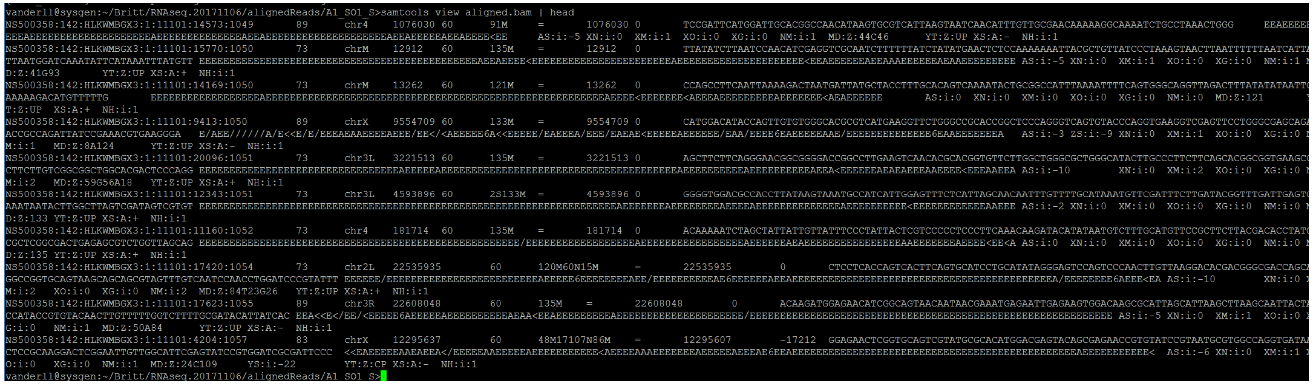
```
@HD VN:1.6 SO:coordinate
@SQ SN:ref LN:45
r001   99 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r002    0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
r003    0 ref  9 30 5S6M       *  0   0 GCCTAAGCTAA       * SA:Z:ref,29,-,6H5M,17,0;
r004    0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
r003 2064 ref 29 17 6H5M       *  0   0 TAGGC             * SA:Z:ref,9,+,5S6M,30,1;
r001  147 ref 37 30 9M         =  7 -39 CAGCGGCAT         * NM:i:1
```

Source: samtools manual



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	`\*|[:rname:^*=][:rname:]*`	Reference sequence NAME[9]
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	`\*|=|[:rname:^*=][:rname:]*`	Reference 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

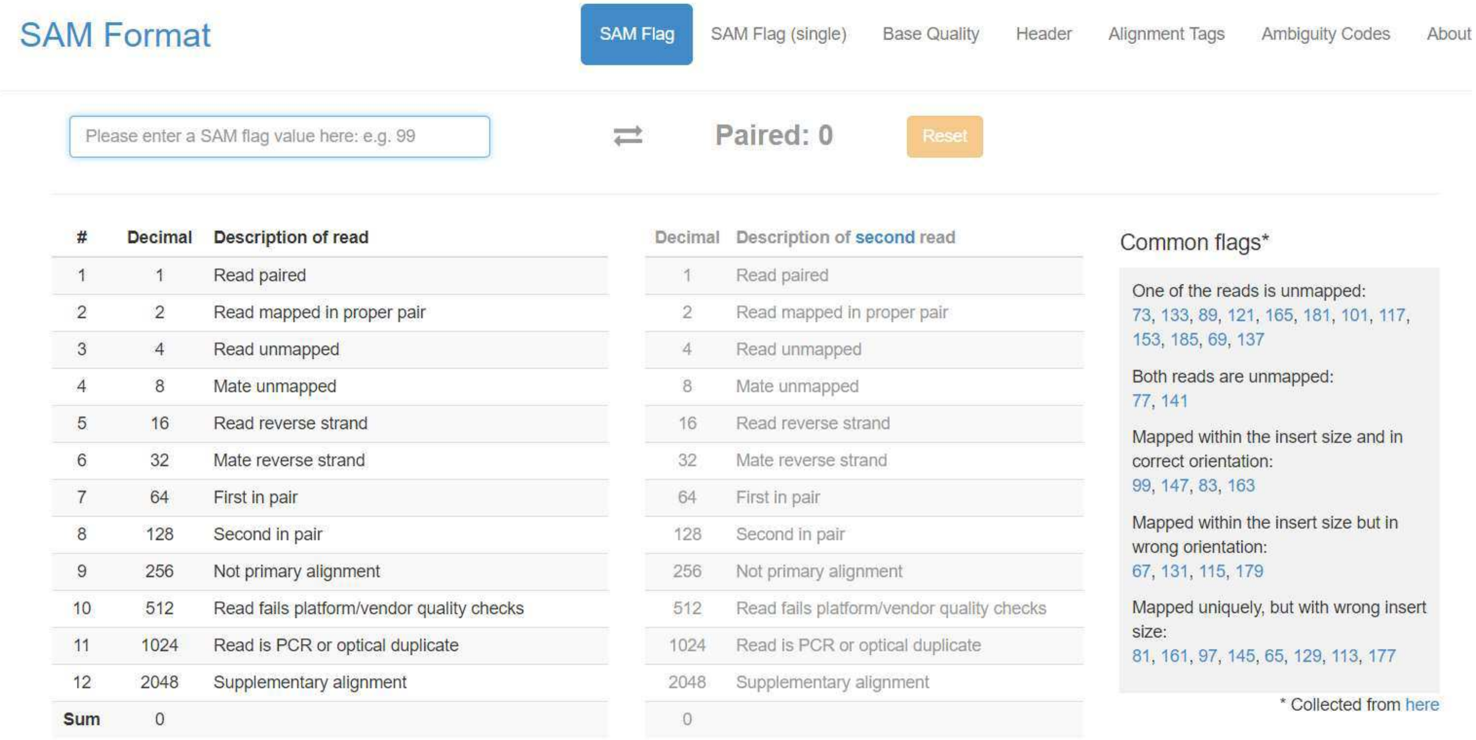
- Optional header '@'
- Alignment info
- Example: first read starts at chr4: 1,076,030



```
vanderll@sysgen:~/Britt/RNAseq.20171106/alignedReads/A1_SO1_S>samtools view aligned.bam | head
NS500358:142:HLKWMBGX3:1:11101:14573:1049       89      chr4    1076030 60      91M     =       1076030 0       TCCGATTCATGGATTGCACGGCCAACATAAGTGCGTCATTAAGTAATCAACATTTGTTGCGAACAAAAAGGCAAAATCTGCCTAAACTGGG     EEAEEEEE
EEEAEEEEEEEEEEEEEEEEEEEEEEAEEEEEEEEEEEEEEEAEEAEEEEEEEEEEEEEEEEEEEEEEEAEEAEEEEEEAEEAEEEE<EE     AS:i:-5 XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:44C46      YT:Z:UP XS:A:-  NH:i:1
NS500358:142:HLKWMBGX3:1:11101:15770:1050       73      chrM    12912   60      135M    =       12912   0       TTATATCTTAATCCAACATCGAGGTCGCAATCTTTTTTATCTATATGAACTCTCCAAAAAAATTACGCTGTTATCCCTAAAGTAACTTAATTTTTTAATCATTA
TTAATGGATCAAATATTCATAAATTTATGTT EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEAEEAEEEE<EEEEEEEEEEEEEEEEEEEEEEEEEEEAEEEEEEEEEEEEEEAEEEEEEEEEEEEEEEEE<EEAEEEEEEEAEEAAAEEEEEEEAEAAEEEEEEEEEE AS:i:-5 XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1 M
D:Z:41G93       YT:Z:UP XS:A:+  NH:i:1
NS500358:142:HLKWMBGX3:1:11101:14169:1050       73      chrM    13262   60      121M    =       13262   0       CCAGCCTTCAATTAAAAGACTAATGATTATGCTACCTTTGCACAGTCAAAATACTGCGGCCATTTAAAATTTTCAGTGGCAGGTTAGACTTTATATATAATT
AAAAAGACATGTTTTTG       EEEEEEEEEEEEEEEEEEAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEAEEEE<EEEEEEE<AEEEAEEEEEEEEEEEAEEEEEEE<AEAEEEEEEE       AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:121        Y
T:Z:UP  XS:A:+  NH:i:1
NS500358:142:HLKWMBGX3:1:11101:9413:1050        89      chrX    9554709 60      133M    =       9554709 0       CATGGACATACCAGTTGTGTGGGCACGCGTCATGAAGGTTCTGGGCCGCACCGGCTCCCAGGGTCAGTGTACCCAGGTGAAGGTCGAGTTCCTGGGCGAGCAGA
ACCGCCAGATTATCCGAAACGTGAAGGGA   E/AEE//////A/E<<E/E/EEEAEAAEEEEAEEE/EE</<AEEEEE6A<<EEEEE/EAEEEA/EEE/EAEAE<EEEEEAEEEEEE/EAA/EEEE6EAEEEEEEAAE/EEEEEEEEEEEEE6EAAEEEEEEEEA   AS:i:-3 ZS:i:-9 XN:i:0  XM:i:1  XO:i:0  XG:i:0 N
M:i:1   MD:Z:8A124      YT:Z:UP XS:A:-  NH:i:1
NS500358:142:HLKWMBGX3:1:11101:20096:1051       73      chr3L   3221513 60      135M    =       3221513 0       AGCTTCTTCAGGGAACGGCGGGGACCGGCCTTGAAGTCAACACGCACGGTGTTCTTGGCTGGGCGCTGGGCATACTTGCCCTTCTTCAGCACGGCGGTGAAGCC
CTTCTTGTCGGCGGCTGGCACGACTCCCAGG EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEAEEEEEAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEAEEEEEEEEEEEEAEEEEEEEEEEEAEEA<EEEEEEAEAEAEEEEAAEEEEE<EEEAAEEA AS:i:-10        XN:i:0  XM:i:2  XO:i:0  XG:i:0 N
M:i:2   MD:Z:59G56A18   YT:Z:UP XS:A:+  NH:i:1
NS500358:142:HLKWMBGX3:1:11101:12343:1051       73      chr3L   4593896 60      2S133M  =       4593896 0       GGGGTGGACGCCACCTTATAAGTAAATGCCATCATTGGAGTTTCTCATTAGCAACAATTTGTTTTGCATAAATGTTCGATTTCTTGATACGGTTTGATTGAGTC
AAATAATACTTGGCTTAGTCGATAGTCGTGT EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEAEEEEEEEEEAEEEEEEEEAEEEEEEEEEAEEEEAEEEEEEEEEEEEEEEEAEEEEEEEEEEE<EEEEEEEEEEEEEAAEEE AS:i:-2 XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0 M
D:Z:133 YT:Z:UP XS:A:+  NH:i:1
NS500358:142:HLKWMBGX3:1:11101:11160:1052       73      chr4    181714  60      135M    =       181714  0       ACAAAAATCTAGCTATTATTGTTATTTCCCTATTACTCGTCCCCCTCCCTTCAAACAAGATACATATAATGTCTTTGCATGTTCCGCTTCTTACGACACCTATC
CGCTCGGCGACTGAGAGCGTCTGGTTAGCAG EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE/EEEEEEEEEEEEEEEEEAEEEEEEEEEEEEAEEEEEEEEEEEEEEEEEEAEEEEEEAEAEEEEEEEEEEEEEEEEEAAEEEEEEEEEAEEEE<EE<A AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0 M
D:Z:135 YT:Z:UP XS:A:+  NH:i:1
NS500358:142:HLKWMBGX3:1:11101:17420:1054       73      chr2L   22535935        60      120M60N15M      =       22535935        0       CTCCTCACCAGTCACTTCAGTGCATCCTGCATATAGGGAGTCCAGTCCCAACTTGTTAAGGACACGACGGGCGACCAGCA
GGCCGGTGCAGTAAGCAGCAGCGTAGTTTGTCAATCCAACCTGGATCCCGTATTT EEEEEE/EEEEEEEEEEEEEEEEEEEEEEEEEEEEAEEEEE6EEEEEEEEEAEE/EEEEEEEEEEEAE6EEEEEEAEEAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEA/EEEEEEEE6AEEE<EA AS:i:-10        XN:i:0 X
M:i:2   XO:i:0  XG:i:0  NM:i:2  MD:Z:84T23G26   YT:Z:UP XS:A:+  NH:i:1
NS500358:142:HLKWMBGX3:1:11101:17623:1055       89      chr3R   22608048        60      135M    =       22608048        0       ACAAGATGGAGAACATCGGCAGTAACAATAACGAAATGAGAATTGAGAAGTGGACAAGCGCATTAGCATTAAGCTTAAGCAATTACTA
CCATACCGTGTACAACTTGTTTTTGGTCTTTTGCGATACATTATCAC EEA<<E</EE/<EEEEE6AEEEEEAEEEEEEEEEEEAEAA<EEAEEEEEEEEEEAEEEEEEEEEEEEEEEEEEEEEEEE/EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE AS:i:-5 XN:i:0  XM:i:1  XO:i:0 X
G:i:0   NM:i:1  MD:Z:50A84      YT:Z:UP XS:A:-  NH:i:1
NS500358:142:HLKWMBGX3:1:11101:4204:1057        83      chrX    12295637        60      48M17107N86M    =       12295607        -17212  GGAGAACTCGGTGCAGTCGTATGCGCACATGGACGAGTACAGCGAGAACCGTGTATCCGTAATGCGTGGCCAGGTATAA
CTCCGCAAGGACTCGGAATTGTTGGCATTCGAGTATCCGTGGATCGCGATTCCC  <<EAEEEEEAAEAEEEA</EEEEEAAEEEEEAEEEEEEEEEEE<AEEEEAAAEEEEEEEEEAEEEEEAEEEAE6EEAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEAEEEEEEEEEEE<  AS:i:-6 XN:i:0  XM:i:1 X
O:i:0   XG:i:0  NM:i:1  MD:Z:24C109     YS:i:-22        YT:Z:CP XS:A:-  NH:i:1
vanderll@sysgen:~/Britt/RNAseq.20171106/alignedReads/A1_SO1_S>
```

SAM Flags

https://www.samformat.info/sam-format-flag



Paired: 0

#	Decimal	Description of read
1	1	Read paired
2	2	Read mapped in proper pair
3	4	Read unmapped
4	8	Mate unmapped
5	16	Read reverse strand
6	32	Mate reverse strand
7	64	First in pair
8	128	Second in pair
9	256	Not primary alignment
10	512	Read fails platform/vendor quality checks
11	1024	Read is PCR or optical duplicate
12	2048	Supplementary alignment
Sum	0	

Decimal	Description of second read
1	Read paired
2	Read mapped in proper pair
4	Read unmapped
8	Mate unmapped
16	Read reverse strand
32	Mate reverse strand
64	First in pair
128	Second in pair
256	Not primary alignment
512	Read fails platform/vendor quality checks
1024	Read is PCR or optical duplicate
2048	Supplementary alignment
0	

Common flags*

One of the reads is unmapped:
73, 133, 89, 121, 165, 181, 101, 117, 153, 185, 69, 137

Both reads are unmapped:
77, 141

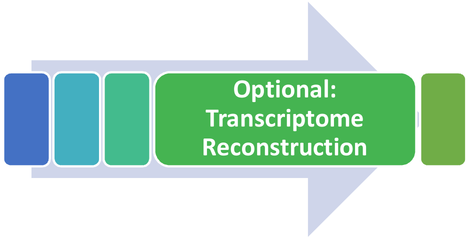
Mapped within the insert size and in correct orientation:
99, 147, 83, 163

Mapped within the insert size but in wrong orientation:
67, 131, 115, 179

Mapped uniquely, but with wrong insert size:
81, 161, 97, 145, 65, 129, 113, 177

* Collected from here

Transcriptome Reconstruction



Optional: Transcriptome Reconstruction

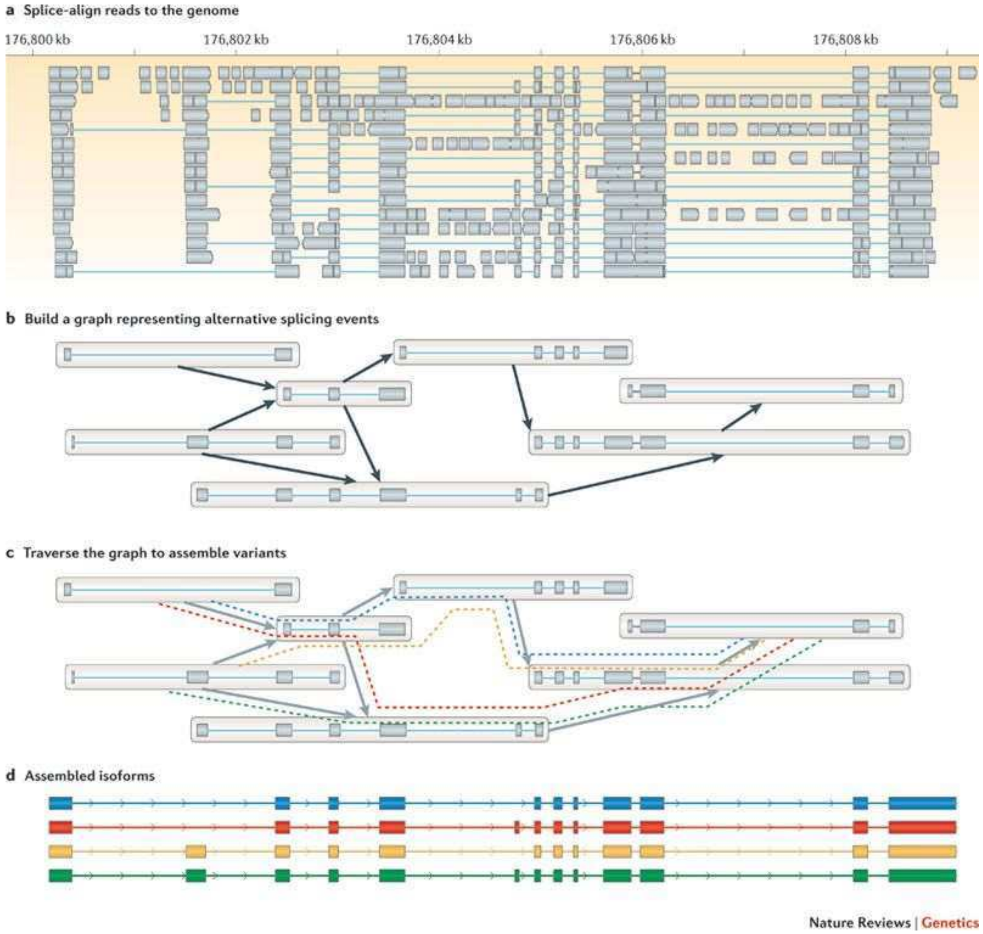
- This is an optional step in the process and I actually don’t perform it very often
- You need paired-end reads and deep coverage
- Software:
    - Cufflinks
    - Stringtie
    - Scripture
    - Traph
    - Trinity
    - TransAbyss
    - Velvet

Reference guided assemblies

De novo assemblies

Reference Guided Assemblies

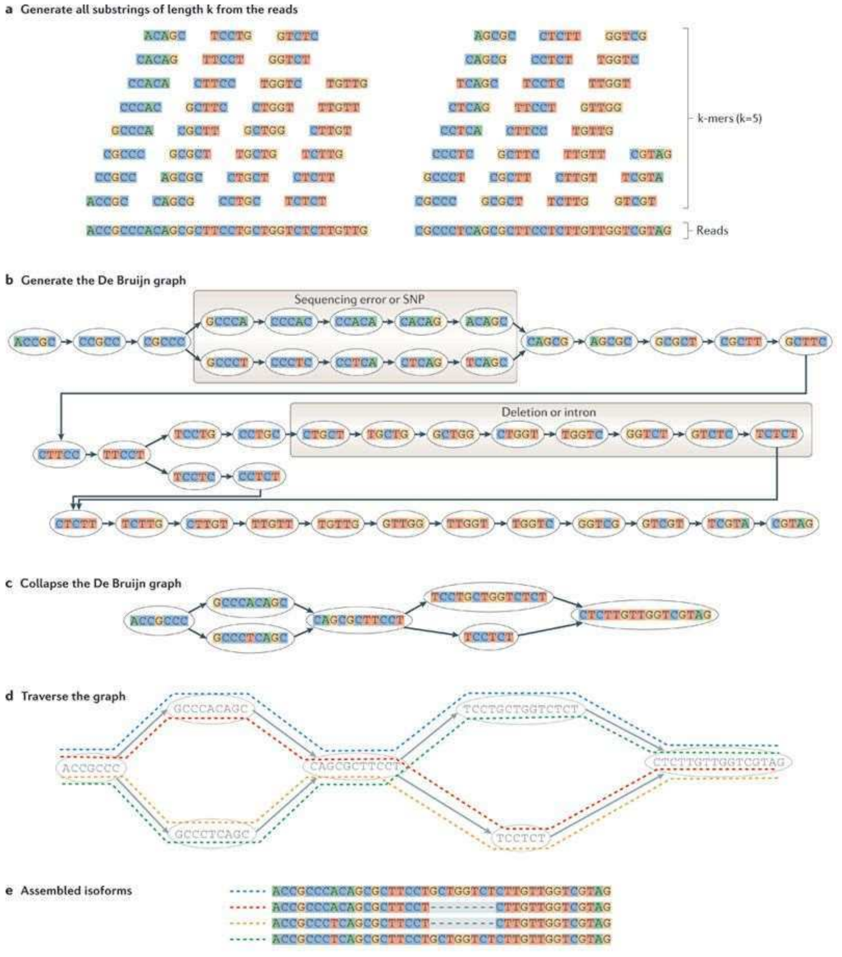
- Splice junctions are the main guiding force in the reconstruction
- See which exons of these splice junctions overlap
- Estimate your isoforms


a Splice-align reads to the genome
176,800 kb
176,802 kb
176,804 kb
176,806 kb
176,808 kb
b Build a graph representing alternative splicing events
c Traverse the graph to assemble variants
d Assembled isoforms
Nature Reviews | Genetics

Source: Martin & Wang 2015

De Novo Assemblies

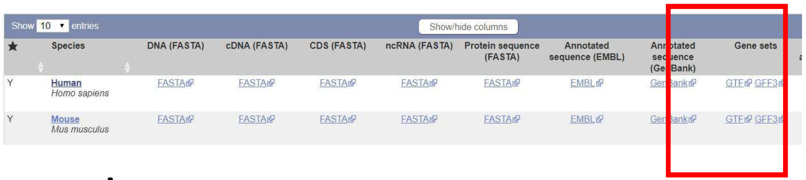
- Depends on De Brujin graph
- Combinatorial mathematics, creates every possible combinations of length k
- Identify indels and SNPs/errors
- Estimate your isoforms
- Computationally much more time than reference based approach



Nature Reviews | Genetics

Source: Martin & Wang 2015

Gene Transfer Format (GTF)

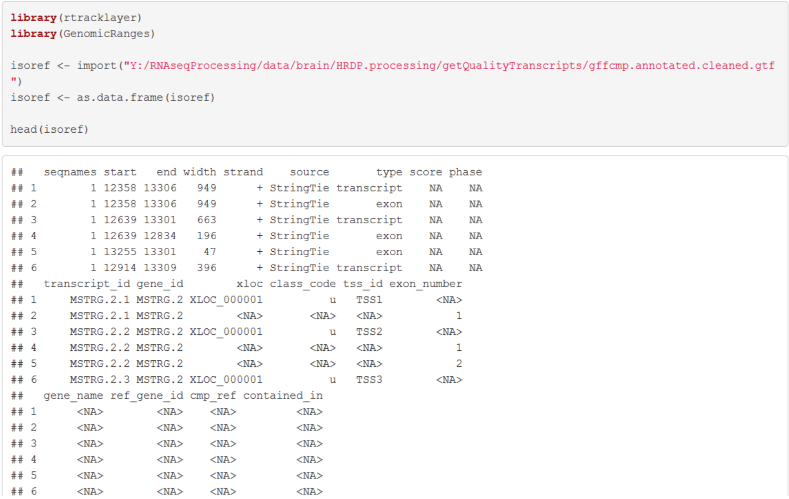


Show 10 ▾ entries

Show/hide columns

★	Species	DNA (FASTA)	cDNA (FASTA)	CDS (FASTA)	ncRNA (FASTA)	Protein sequence (FASTA)	Annotated sequence (EMBL)	Annotated sequence (GenBank)	Gene sets
Y	**Human** *Homo sapiens*	FASTA	FASTA	FASTA	FASTA	FASTA	EMBL	GenBank	GTF GFF3
Y	**Mouse** *Mus musculus*	FASTA	FASTA	FASTA	FASTA	FASTA	EMBL	GenBank	GTF GFF3

- Transcriptome file format
- Can get references from ENSEMBL: Gene sets
- Easy to look at using R/rtracklayer

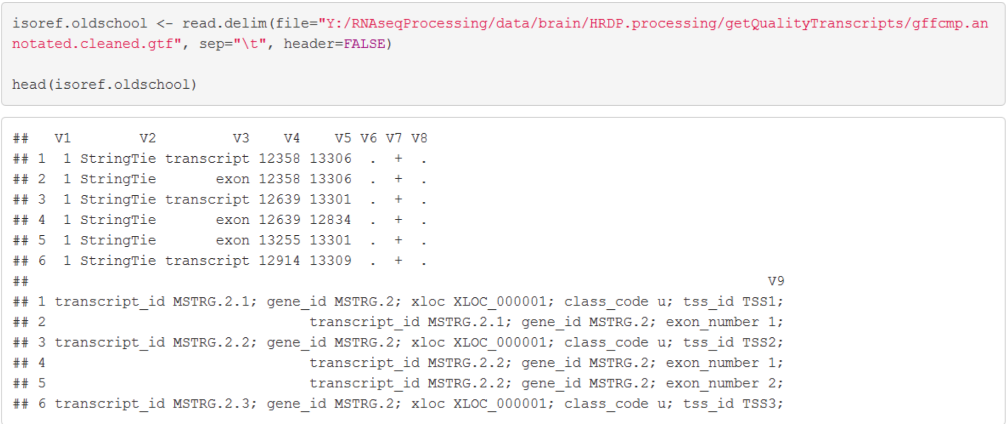


```r
library(rtracklayer)
library(GenomicRanges)

isoref <- import("Y:/RNAseqProcessing/data/brain/HRDP.processing/getQualityTranscripts/gffcmp.annotated.cleaned.gtf
")
isoref <- as.data.frame(isoref)

head(isoref)
```

```
##   seqnames start   end width strand    source       type score phase
## 1        1 12358 13306   949      + StringTie transcript    NA    NA
## 2        1 12358 13306   949      + StringTie       exon    NA    NA
## 3        1 12639 13301   663      + StringTie transcript    NA    NA
## 4        1 12639 12834   196      + StringTie       exon    NA    NA
## 5        1 13255 13301    47      + StringTie       exon    NA    NA
## 6        1 12914 13309   396      + StringTie transcript    NA    NA
##   transcript_id gene_id        xloc class_code tss_id exon_number
## 1     MSTRG.2.1 MSTRG.2 XLOC_000001          u   TSS1        <NA>
## 2     MSTRG.2.1 MSTRG.2        <NA>       <NA>   <NA>           1
## 3     MSTRG.2.2 MSTRG.2 XLOC_000001          u   TSS2        <NA>
## 4     MSTRG.2.2 MSTRG.2        <NA>       <NA>   <NA>           1
## 5     MSTRG.2.2 MSTRG.2        <NA>       <NA>   <NA>           2
## 6     MSTRG.2.3 MSTRG.2 XLOC_000001          u   TSS3        <NA>
##   gene_name ref_gene_id cmp_ref contained_in
## 1      <NA>        <NA>    <NA>         <NA>
## 2      <NA>        <NA>    <NA>         <NA>
## 3      <NA>        <NA>    <NA>         <NA>
## 4      <NA>        <NA>    <NA>         <NA>
## 5      <NA>        <NA>    <NA>         <NA>
## 6      <NA>        <NA>    <NA>         <NA>
```

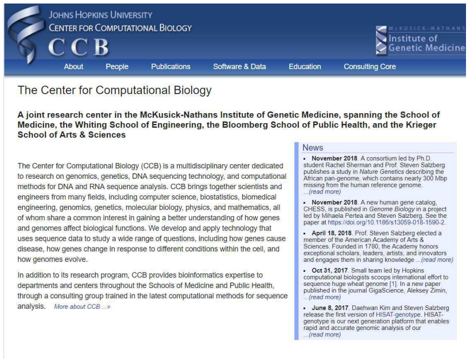


```r
isoref.oldschool <- read.delim(file="Y:/RNAseqProcessing/data/brain/HRDP.processing/getQualityTranscripts/gffcmp.an
notated.cleaned.gtf", sep="\t", header=FALSE)

head(isoref.oldschool)
```

```
##   V1        V2         V3    V4    V5 V6 V7 V8
## 1  1 StringTie transcript 12358 13306  .  +  .
## 2  1 StringTie       exon 12358 13306  .  +  .
## 3  1 StringTie transcript 12639 13301  .  +  .
## 4  1 StringTie       exon 12639 12834  .  +  .
## 5  1 StringTie       exon 13255 13301  .  +  .
## 6  1 StringTie transcript 12914 13309  .  +  .
##                                                                                        V9
## 1 transcript_id MSTRG.2.1; gene_id MSTRG.2; xloc XLOC_000001; class_code u; tss_id TSS1;
## 2                               transcript_id MSTRG.2.1; gene_id MSTRG.2; exon_number 1;
## 3 transcript_id MSTRG.2.2; gene_id MSTRG.2; xloc XLOC_000001; class_code u; tss_id TSS2;
## 4                               transcript_id MSTRG.2.2; gene_id MSTRG.2; exon_number 1;
## 5                               transcript_id MSTRG.2.2; gene_id MSTRG.2; exon_number 2;
## 6 transcript_id MSTRG.2.3; gene_id MSTRG.2; xloc XLOC_000001; class_code u; tss_id TSS3;
```

Tuxedo Suite



JOHNS HOPKINS UNIVERSITY
CENTER FOR COMPUTATIONAL BIOLOGY
CCB

McKUSICK-NATHANS Institute of Genetic Medicine

About People Publications Software & Data Education Consulting Core

The Center for Computational Biology

A joint research center in the McKusick-Nathans Institute of Genetic Medicine, spanning the School of Medicine, the Whiting School of Engineering, the Bloomberg School of Public Health, and the Krieger School of Arts & Sciences

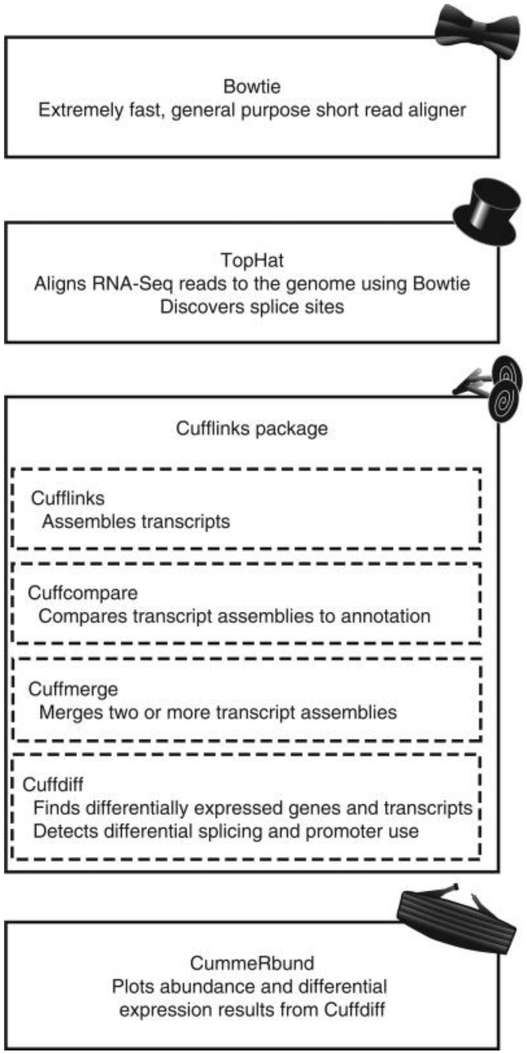
The Center for Computational Biology (CCB) is a multidisciplinary center dedicated to research on genomics, genetics, DNA sequencing technology, and computational methods for DNA and RNA sequence analysis. CCB brings together scientists and engineers from many fields, including computer science, biostatistics, biomedical engineering, genomics, genetics, molecular biology, physics, and mathematics, all of whom share a common interest in gaining a better understanding of how genes and genomes affect biological functions. We develop and apply technology that uses sequence data to study a wide range of questions, including how genes cause disease, how genes change in response to different conditions within the cell, and how genomes evolve.

In addition to its research program, CCB provides bioinformatics expertise to departments and centers throughout the Schools of Medicine and Public Health, through a consulting group trained in the latest computational methods for sequence analysis. *More about CCB ...»*

News

- **November 2018**. A consortium led by Ph.D. student Rachel Sherman and Prof. Steven Salzberg publishes a study in *Nature Genetics* describing the African pan-genome, which contains nearly 300 Mbp missing from the human reference genome. *...(read more)*
- **November 2018**. A new human gene catalog, CHESS, is published in *Genome Biology* in a project led by Mihaela Pertea and Steven Salzberg. See the paper at https://doi.org/10.1186/s13059-018-1590-2.
- **April 18, 2018**. Prof. Steven Salzberg elected a member of the American Academy of Arts & Sciences. Founded in 1780, the Academy honors exceptional scholars, leaders, artists, and innovators and engages them in sharing knowledge *...(read more)*
- **Oct 31, 2017**. Small team led by Hopkins computational biologists scoops international effort to sequence huge wheat genome [1]. In a new paper published in the journal GigaScience, Aleksey Zimin, *...(read more)*
- **June 8, 2017**. Daehwan Kim and Steven Salzberg release the first version of HISAT-genotype. HISAT-genotype is our next generation platform that enables rapid and accurate genomic analysis of our *...(read more)*

This Johns Hopkins group produces all the "Tuxedo Tools"



Bowtie
Extremely fast, general purpose short read aligner

TopHat
Aligns RNA-Seq reads to the genome using Bowtie
Discovers splice sites

Cufflinks package

Cufflinks
Assembles transcripts

Cuffcompare
Compares transcript assemblies to annotation

Cuffmerge
Merges two or more transcript assemblies

Cuffdiff
Finds differentially expressed genes and transcripts
Detects differential splicing and promoter use

CummeRbund
Plots abundance and differential expression results from Cuffdiff

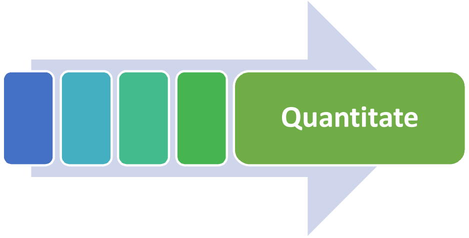
Source: Trapnell et al, 2012

New version of tophat is hisat2

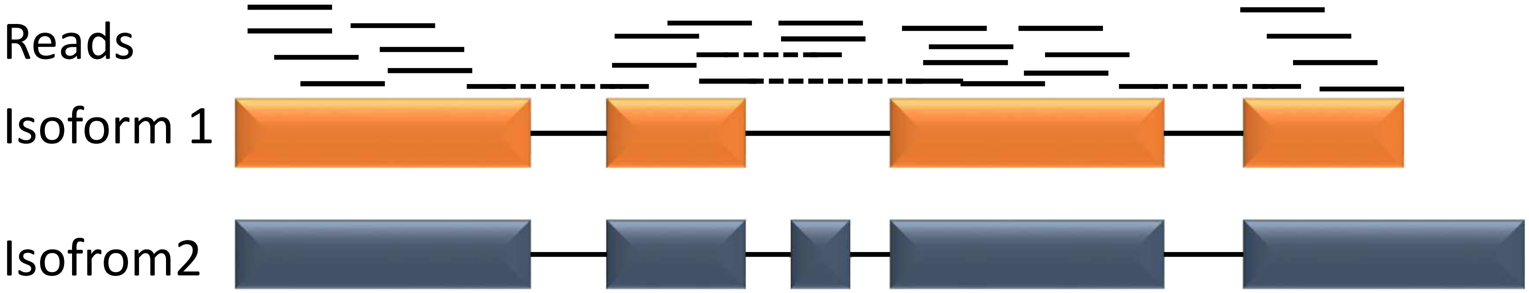
Newer version of cufflinks is stringtie

This suite is extremely popular

Quantitate Reads

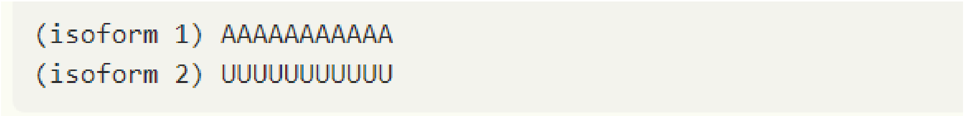


- Summarize on either the gene level or isoform level
- HTSeq (python module) or RSEM
- HTSeq not good for overlapping reads (i.e. not good for isoforms)



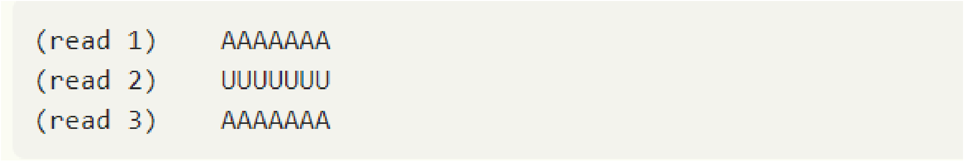
RSEM Quantitation

- RNA-Seq Expectation Maximization (RSEM). Li et al, 2010
- Say you have a gene with 2 isoforms. Each has it’s own unique exon



```
(isoform 1) AAAAAAAAAAA
(isoform 2) UUUUUUUUUUU
```

- Say you have 3 reads for these



```
(read 1)    AAAAAAA
(read 2)    UUUUUUU
(read 3)    AAAAAAA
```

- Would say 2/3 of your reads for this gene come from isoform 1

RSEM Quantitation 2

- However, example was extremely simplified as isoforms are highly similar and many don’t have completely unique exons.
  - Isoform 1: Exon1-Exon2-Exon3-Exon4
  - Isoform 2: Exon1-Exon3-Exon4
  - Isoform 3: Exon2-Exon3-Exon4
- Notation
  - N = total # of reads (library size)
  - M = # known isoforms
  - L = read length
  - Li = length isoform
  - τi = TPM (fraction of transcripts belong to isoform i out of all transcripts in sample * million)
  - Ɵi = prior probability any single read derived from isoform i
- Take away θ has a constant (uniform) probability density

$$\theta_i = \frac{\tau_i * l_i}{\sum_{k=1}^{M} \tau_k * l_i}$$

$$\tau_i = \frac{\theta_i / l_i}{\sum_{k=1}^{M} \theta_k / l_i}$$

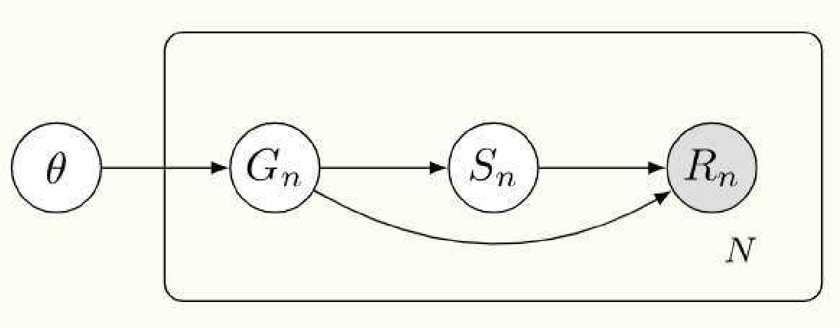
$$\theta_i \propto \tau_i * l_i$$

$$\sum_i \theta_i = \sum_i \tau_i = 1$$

$$p(\theta) \propto 1$$

RSEM Quantitation 3

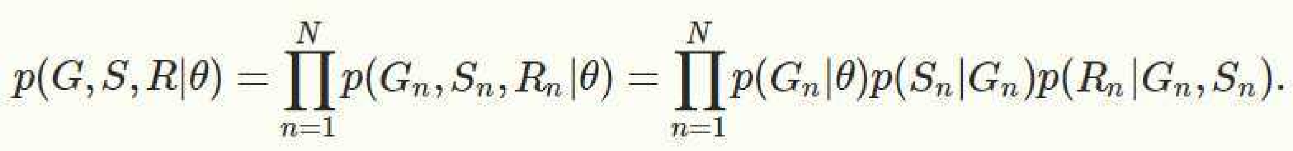
- What does an estimate of ϴ (or τ) look like?
- Full posterior distribution is p(ϴ|R) or p(τ|R)
- Li et al 2010, describe following Bayesian network (a probabilistic graphical model):



- Gn is isoform and Sn is starting position
- Probability of a read coming from a specific gene with influence the number of reads you see in your data

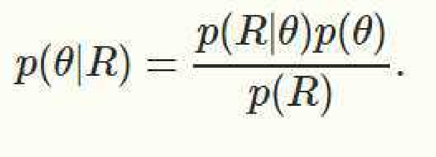
RSEM Quantitation 4

Start with a joint probability:



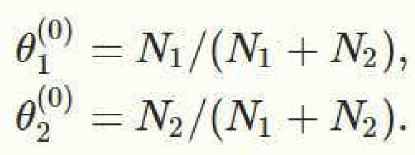
$$p(G,S,R|\theta) = \prod_{n=1}^{N} p(G_n, S_n, R_n|\theta) = \prod_{n=1}^{N} p(G_n|\theta)p(S_n|G_n)p(R_n|G_n, S_n).$$

With lots of math:



$$p(\theta|R) = \frac{p(R|\theta)p(\theta)}{p(R)}.$$

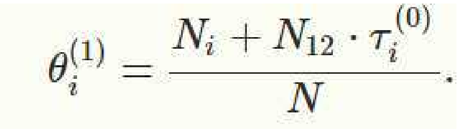
Start an estimate of Θ1 and Θ2 (assuming a simple 2 isoform dataset with N1 # reads aligning uniquely to isoform 1 and N2 # reads aligning uniquely to isoform 2



$$\theta_1^{(0)} = N_1/(N_1 + N_2),$$
$$\theta_2^{(0)} = N_2/(N_1 + N_2).$$

RSEM Quantitation 5

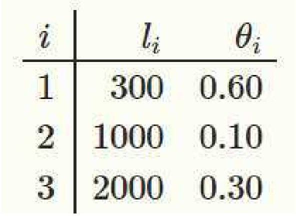
- Start doing expectation maximization for finding the maximum a posteriori estimate of Ɵ
  - N12 = # reads overlapping both isoforms
  - N = # read total (N1 N2 + N12)



$$\theta_i^{(1)} = \frac{N_i + N_{12} \cdot \tau_i^{(0)}}{N}.$$

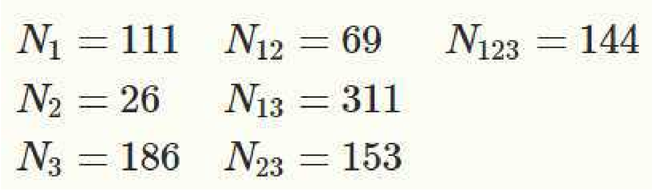
- Repeat this cycle till $\theta^{(r+1)}$ does differ too much from $\theta^{(r)}$
- Simulation example:

Truth



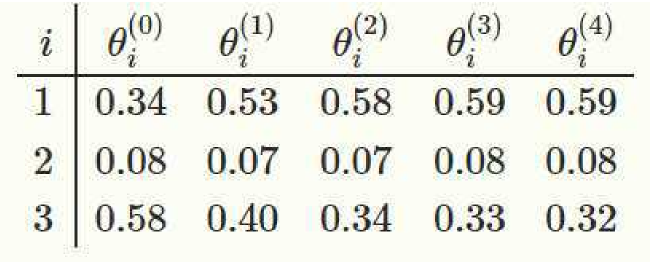
i	l_i	θ_i
1	300	0.60
2	1000	0.10
3	2000	0.30

Counts



$N_1 = 111$	$N_{12} = 69$	$N_{123} = 144$
$N_2 = 26$	$N_{13} = 311$	
$N_3 = 186$	$N_{23} = 153$	

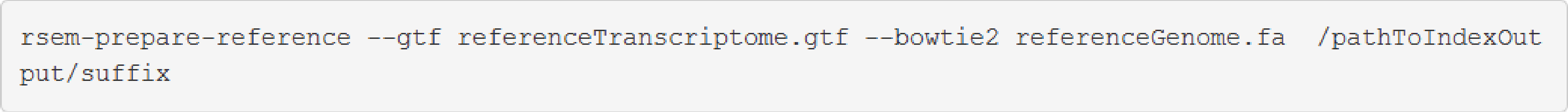
RSEM iterations



i	$\theta_i^{(0)}$	$\theta_i^{(1)}$	$\theta_i^{(2)}$	$\theta_i^{(3)}$	$\theta_i^{(4)}$
1	0.34	0.53	0.58	0.59	0.59
2	0.08	0.07	0.07	0.08	0.08
3	0.58	0.40	0.34	0.33	0.32

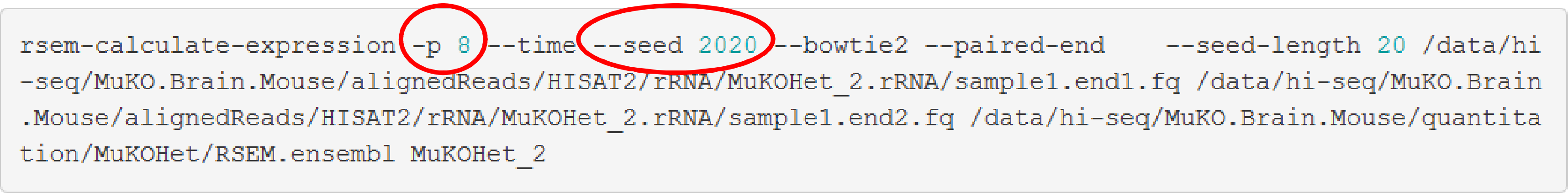
RSEM Code

- Like alignment tools, you need to build a reference
- It is actually aligning to the **transcriptome** within this process (bowtie)



```
rsem-prepare-reference --gtf referenceTranscriptome.gtf --bowtie2 referenceGenome.fa  /pathToIndexOut
put/suffix
```

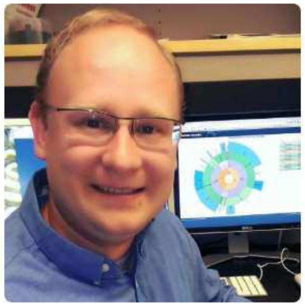
- RSEM on sample:



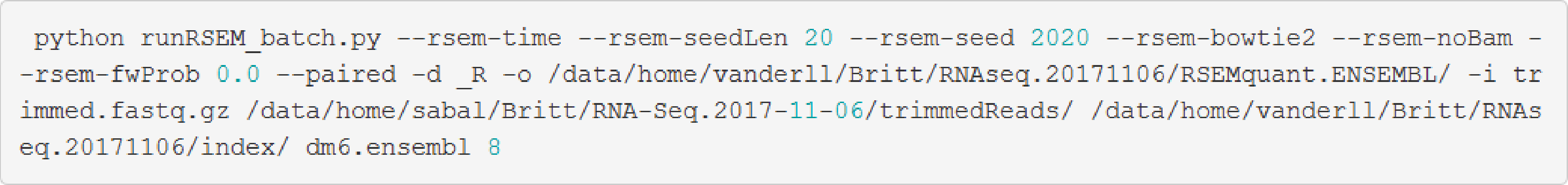
```
rsem-calculate-expression -p 8 --time --seed 2020 --bowtie2 --paired-end    --seed-length 20 /data/hi
-seq/MuKO.Brain.Mouse/alignedReads/HISAT2/rRNA/MuKOHet_2.rRNA/sample1.end1.fq /data/hi-seq/MuKO.Brain
.Mouse/alignedReads/HISAT2/rRNA/MuKOHet_2.rRNA/sample1.end2.fq /data/hi-seq/MuKO.Brain.Mouse/quantita
tion/MuKOHet/RSEM.ensembl MuKOHet_2
```

RSEM Batch

- Python script called: runRSEM_batch.py
- Written by Spencer Mahaffey (thank you again!)



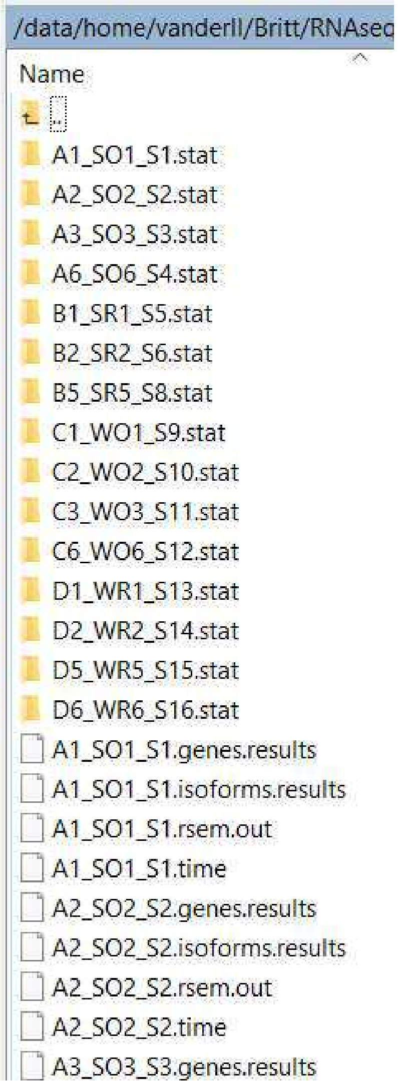
Spencer Mahaffey
smahaffey



```
 python runRSEM_batch.py --rsem-time --rsem-seedLen 20 --rsem-seed 2020 --rsem-bowtie2 --rsem-noBam -
-rsem-fwProb 0.0 --paired -d _R -o /data/home/vanderll/Britt/RNAseq.20171106/RSEMquant.ENSEMBL/ -i tr
immed.fastq.gz /data/home/sabal/Britt/RNA-Seq.2017-11-06/trimmedReads/ /data/home/vanderll/Britt/RNAs
eq.20171106/index/ dm6.ensembl 8
```

RSEM Output

- Folder with sample name:
  - 3 files regarding mathematical files
  - 1 files for count estimated at each iteration
  - 1 file for other mathematical estimates in model
- 4 Files per sample:
  - sampleName.genes.results
  - sampleName.isoforms.results
  - sampleName.rsem.out
  - sampleName.time



/data/home/vanderll/Britt/RNAseq

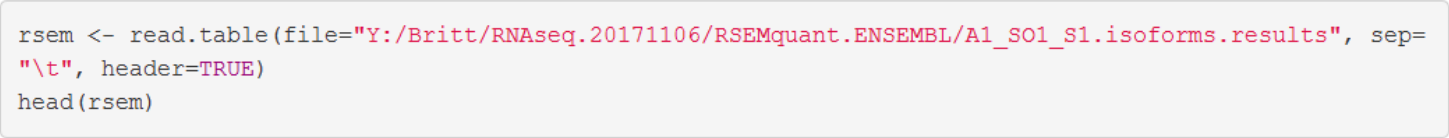
Name

..

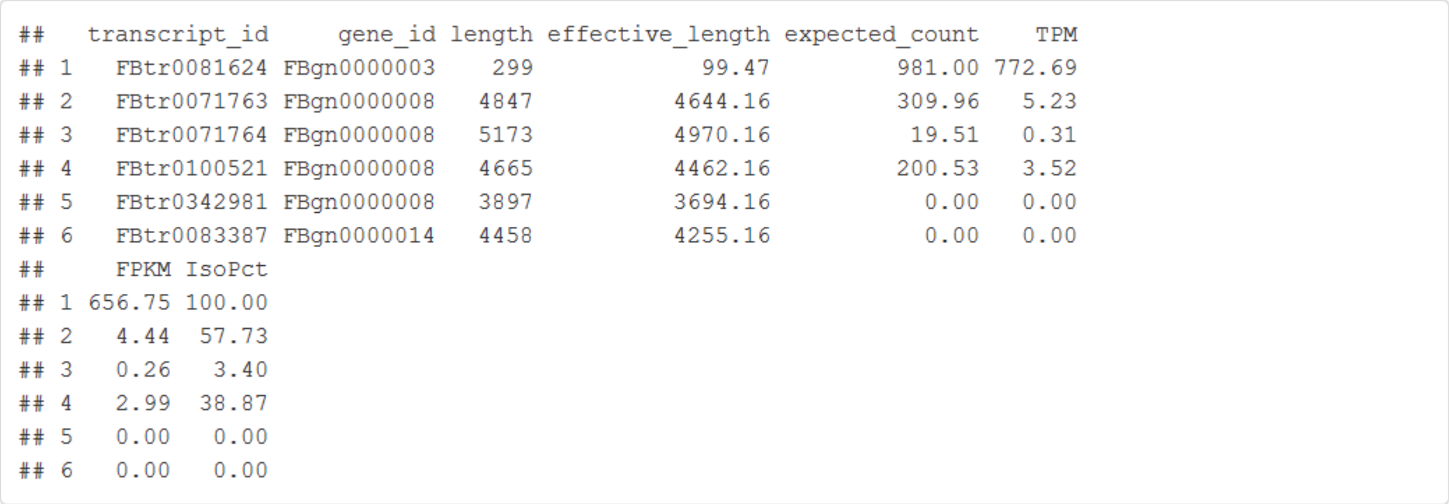
A1_SO1_S1.stat
A2_SO2_S2.stat
A3_SO3_S3.stat
A6_SO6_S4.stat
B1_SR1_S5.stat
B2_SR2_S6.stat
B5_SR5_S8.stat
C1_WO1_S9.stat
C2_WO2_S10.stat
C3_WO3_S11.stat
C6_WO6_S12.stat
D1_WR1_S13.stat
D2_WR2_S14.stat
D5_WR5_S15.stat
D6_WR6_S16.stat
A1_SO1_S1.genes.results
A1_SO1_S1.isoforms.results
A1_SO1_S1.rsem.out
A1_SO1_S1.time
A2_SO2_S2.genes.results
A2_SO2_S2.isoforms.results
A2_SO2_S2.rsem.out
A2_SO2_S2.time
A3_SO3_S3.genes.results

RSEM Output 2

- The genes.results and isoforms.results files have the same format

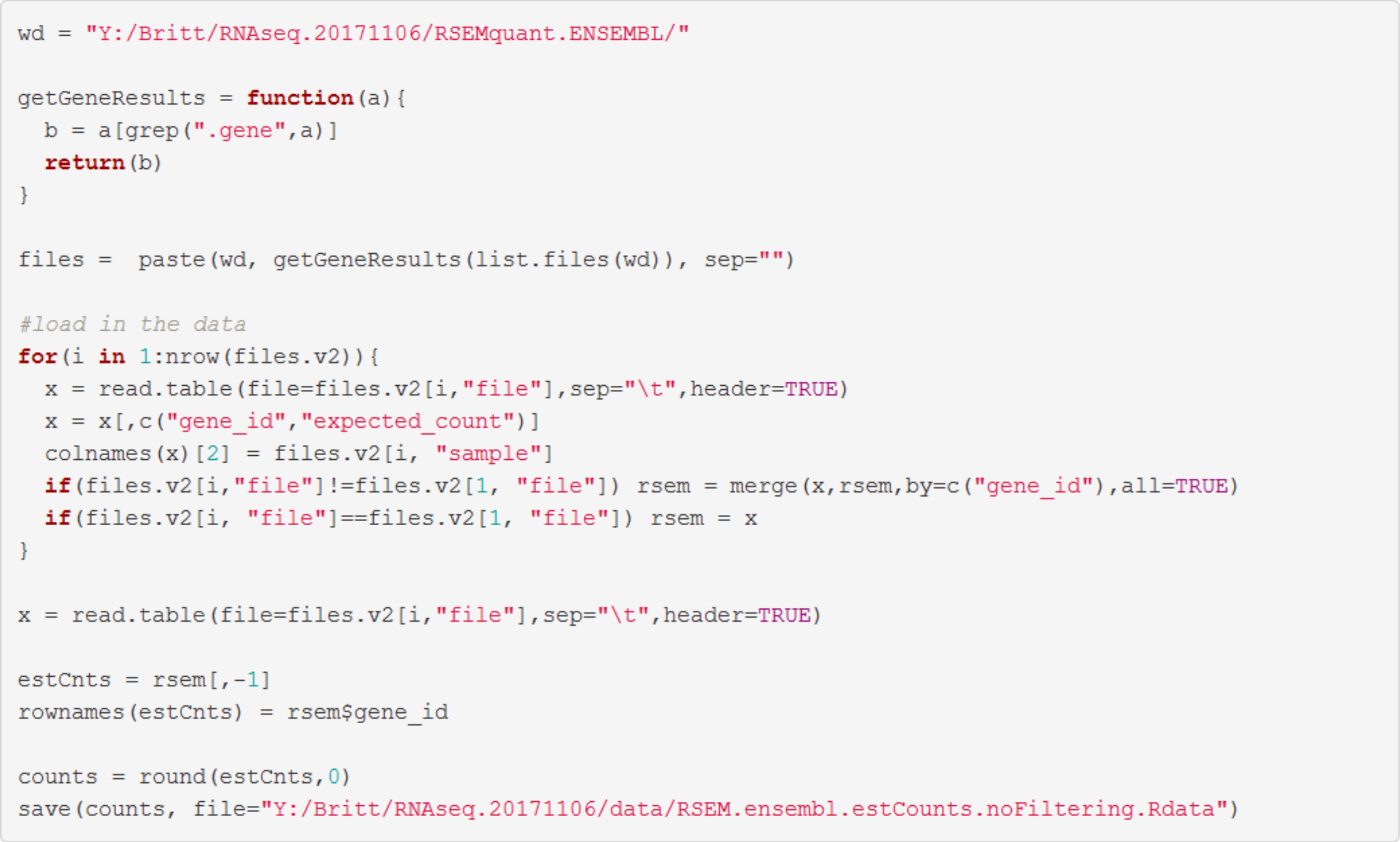


```
rsem <- read.table(file="Y:/Britt/RNAseq.20171106/RSEMquant.ENSEMBL/A1_SO1_S1.isoforms.results", sep=
"\t", header=TRUE)
head(rsem)
```



```
##   transcript_id     gene_id length effective_length expected_count    TPM
## 1   FBtr0081624 FBgn0000003    299            99.47         981.00 772.69
## 2   FBtr0071763 FBgn0000008   4847          4644.16         309.96   5.23
## 3   FBtr0071764 FBgn0000008   5173          4970.16          19.51   0.31
## 4   FBtr0100521 FBgn0000008   4665          4462.16         200.53   3.52
## 5   FBtr0342981 FBgn0000008   3897          3694.16           0.00   0.00
## 6   FBtr0083387 FBgn0000014   4458          4255.16           0.00   0.00
##     FPKM IsoPct
## 1 656.75 100.00
## 2   4.44  57.73
## 3   0.26   3.40
## 4   2.99  38.87
## 5   0.00   0.00
## 6   0.00   0.00
```

RSEM Merge Data



```r
wd = "Y:/Britt/RNAseq.20171106/RSEMquant.ENSEMBL/"

getGeneResults = function(a){
  b = a[grep(".gene",a)]
  return(b)
}

files =  paste(wd, getGeneResults(list.files(wd)), sep="")

#load in the data
for(i in 1:nrow(files.v2)){
  x = read.table(file=files.v2[i,"file"],sep="\t",header=TRUE)
  x = x[,c("gene_id","expected_count")]
  colnames(x)[2] = files.v2[i, "sample"]
  if(files.v2[i,"file"]!=files.v2[1, "file"]) rsem = merge(x,rsem,by=c("gene_id"),all=TRUE)
  if(files.v2[i, "file"]==files.v2[1, "file"]) rsem = x
}

x = read.table(file=files.v2[i,"file"],sep="\t",header=TRUE)

estCnts = rsem[,-1]
rownames(estCnts) = rsem$gene_id

counts = round(estCnts,0)
save(counts, file="Y:/Britt/RNAseq.20171106/data/RSEM.ensembl.estCounts.noFiltering.Rdata")
```

Reporting Preprocessing

- Data
  - When downloaded
  - Where saved
- Processing code
  - Just referencing where or what was run
  - Versions of programs
- Summary of reads at each step
- If quantitating to a known reference transcriptome, like to check out percentage Ensembl “biotype” aligned to
- Look at Rmarkdown report
- The Rmd and html will be available on yampa under:

References

Hansen KD, Brenner SE, Dudoit S (2010). *Biases in Illumina transcriptome sequencing caused by random hexamer priming.* Nucleic Acids Res. 38(12):e131.

Pertea M, Pertea GM, Antonescu CM, Chang TC, Mendell JT, Salzberg SL. (2015) *StringTie enables improved reconstruction of a transcriptome from RNA-seq reads.* Nat Biotechnol. 33(3):290-5.

Martin JA, Wang Z. (2011) *Next-generation transcriptome assembly.* Nat Rev Genet. 12(10):671-82.

Cole Trapnell, Adam Roberts, Loyal Goff, Geo Pertea, Daehwan Kim, David R Kelley, Harold Pimentel, Steven L Salzberg, John L Rinn, and Lior Pachter (2012) *Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks*. Nat Protoc. 2012 Mar 1; 7(3): 562–578.

Li, B., Ruotti, V., Stewart, R.M., Thomson, J.A. & Dewey, C.N. (2010). *RNA-Seq gene expression estimation with read mapping uncertainty*. Bioinformatics **26**, 493–500.

https://www.biostat.wisc.edu/bmi776/lectures/rnaseq.pdf