# RNA-seq Data Analysis

**Lecture 1: Reference genome guided analysis**

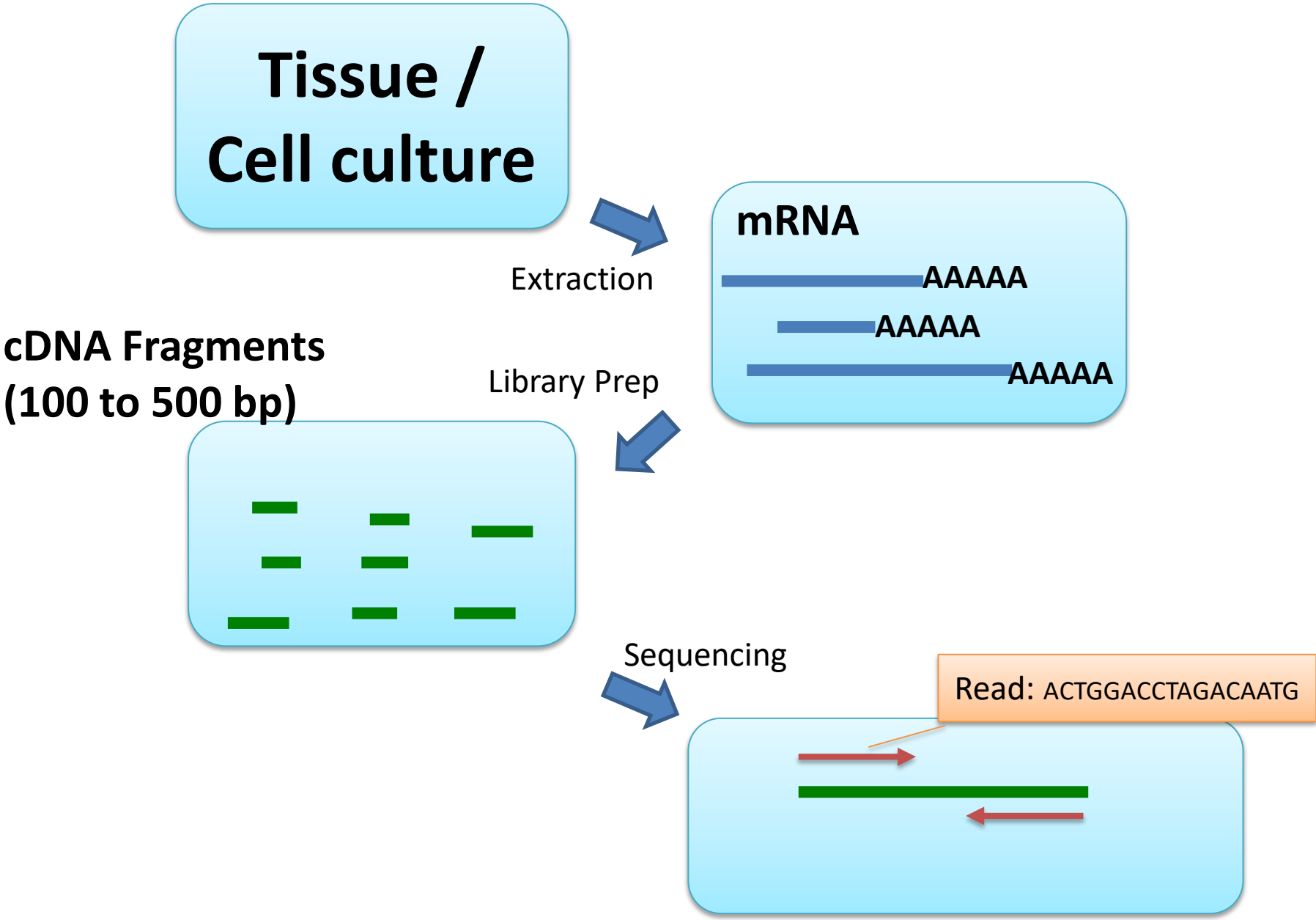
**Lecture 2: De novo assembly without reference**

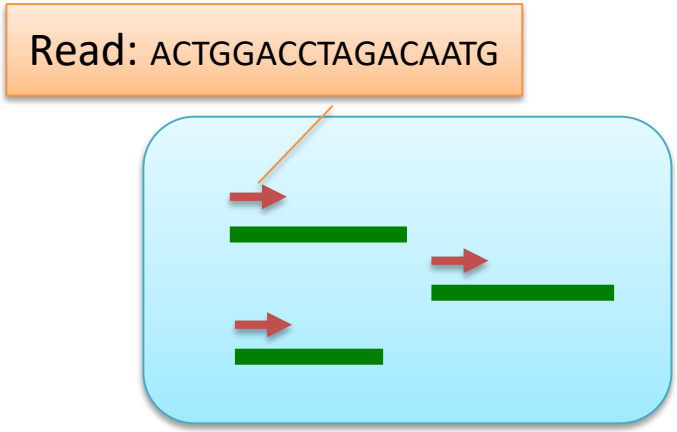
**Lecture 3: Statistics of RNA-seq data analysis**

**Qi Sun, Robert Bukowski, Minghui Wang**

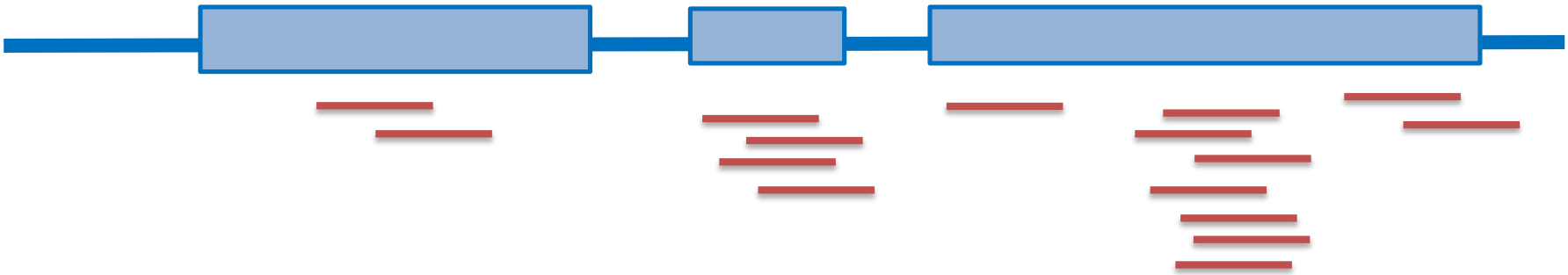
Bioinformatics Facility
Biotechnology Resource Center
Cornell University

# RNA-seq Experiment





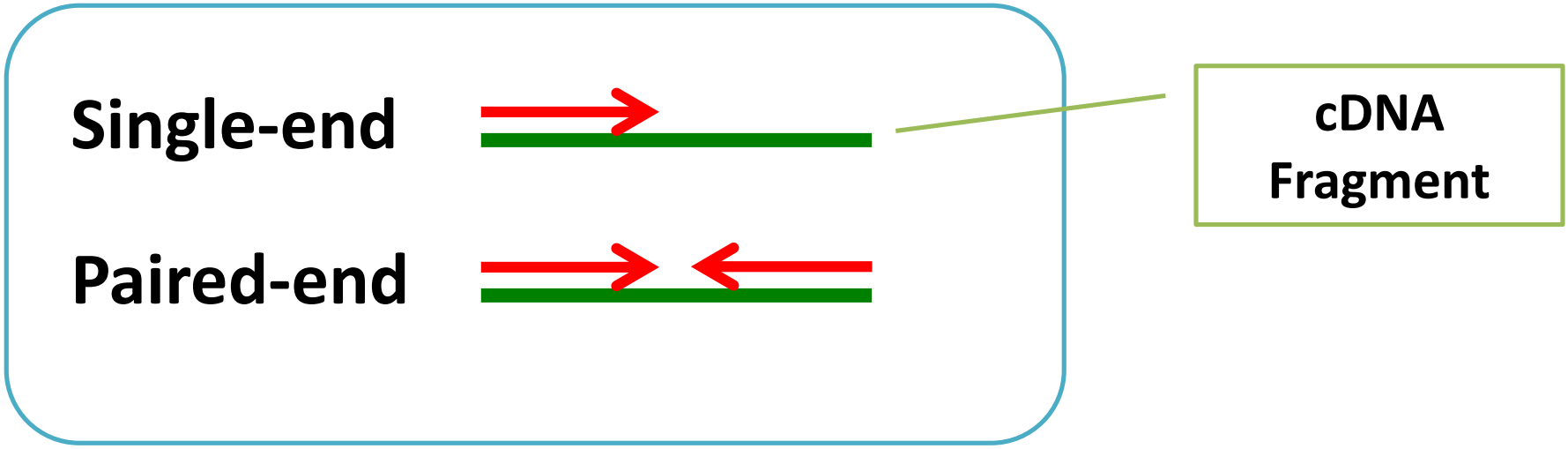

# Map reads to Gene



# Experimental Design

- **Single vs paired end;**
- **Read length (50bp, 75bp, … );**
- **Stranded vs non-stranded;**

# single-end vs paired-end

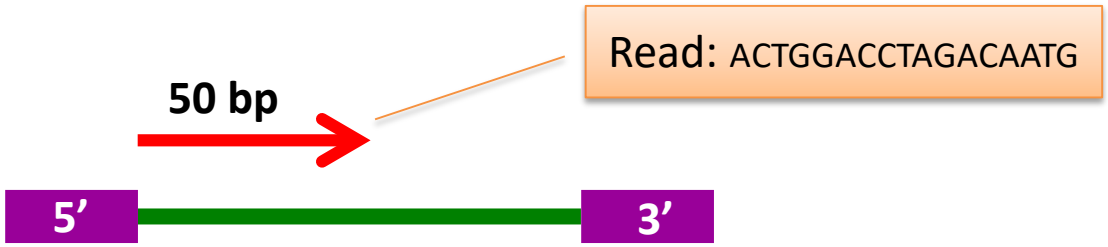



What you get from the facility:

**Single-end: one fastq file per sample**

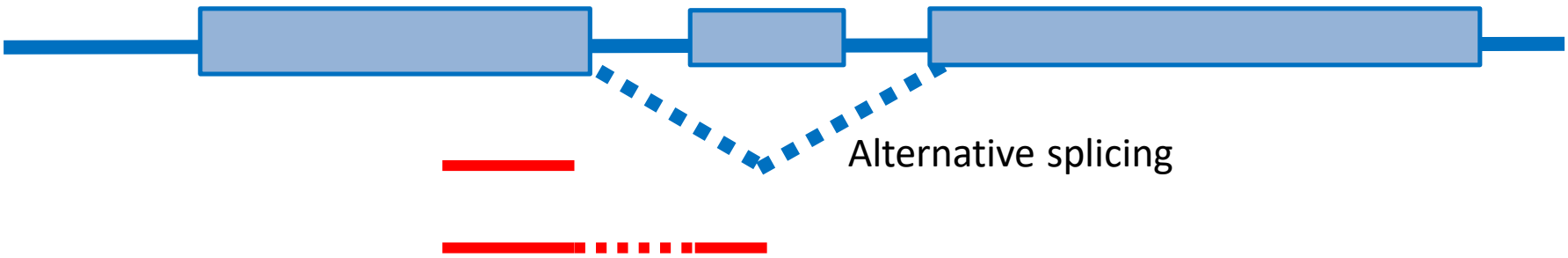
**Paired-end: two fastq files per sample**

# Read length (50 bp, 100 bp, …)

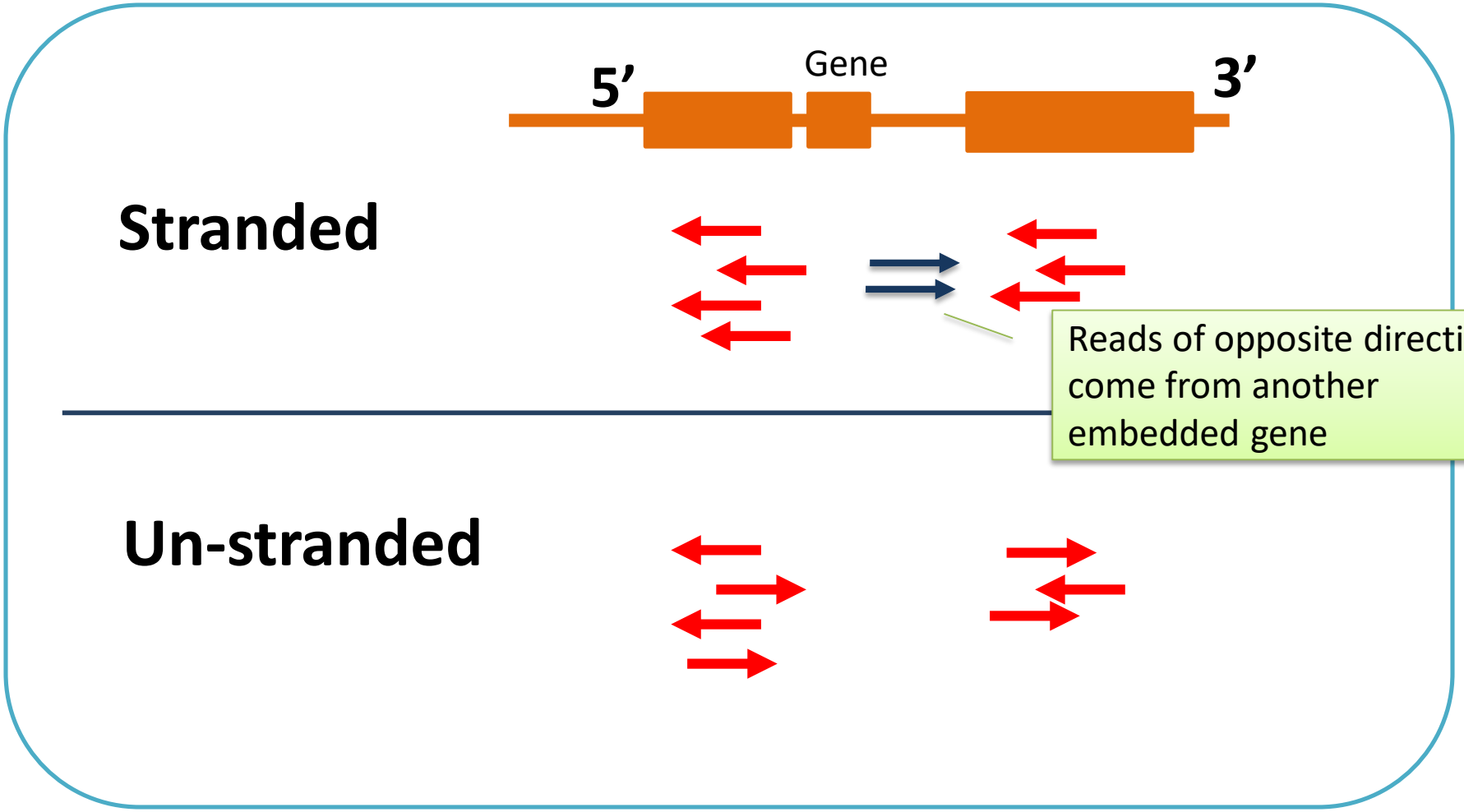



1. For gene expression level, 50 bp is good enough;

2. In some cases, longer reads are desired
   - Isoforms;
   - Distinguish alleles/paralogs;



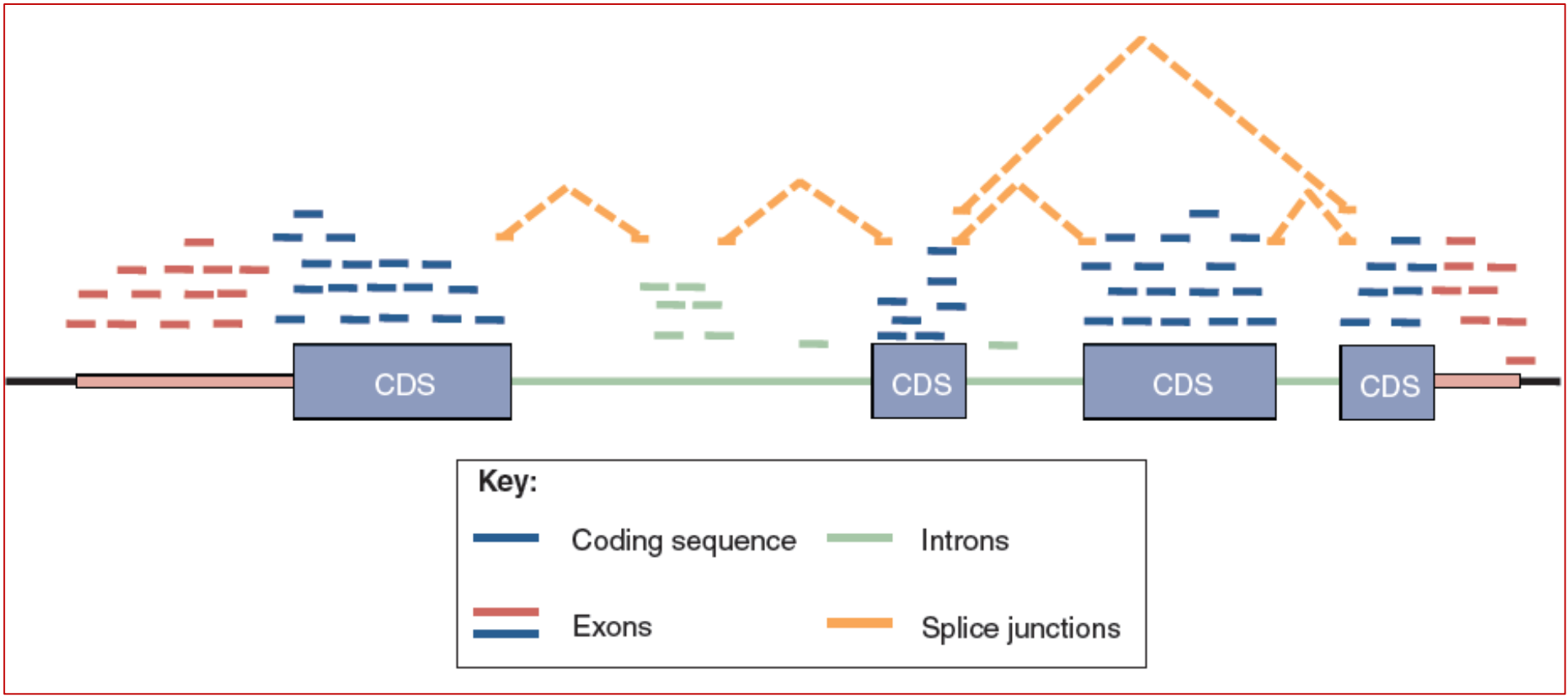
# Strand vs un-stranded



## For quantification of gene expression

- **Read length:** 50 to 100 bp
- **Paired vs single ends:** Single end
- **Number of reads:** >5 million per sample
- **Replicates:** 3 replicates

# RNA-seq Data Analysis



# Data analysis procedures

**Step 1.** Check quality of the reads (optional);

**Step 2.** Map reads to the genome;

**Step 3.** Count reads per gene.

# About the files

1. Reference genome (FASTA)
2. FASTQ
3. GFF3/GTF
4. SAM/BAM

```
>chr1
TTCTAGGTCTGCGATATTTCCTGCCTATCCATTTTGTTAACTCTTCAATG
CATTCCACAAATACCTAAGTATTCTTTAATAATGGTGGTTTTTTTTTTTT
TTTGCATCTATGAAGTTTTTTCAAATTCTTTTTAAGTGACAAAACTTGTA
CATGTGTATCGCTCAATATTTCTAGTCGACAGCACTGCTTTCGAGAATGT
AAACCGTGCACTCCCAGGAAAATGCAGACACAGCACGCCTCTTTGGGACC
GCGGTTTATACTTTCGAAGTGCTCGGAGCCCTTCCTCCAGACCGTTCTCC
CACACCCCGCTCCAGGGTCTCTCCCGGAGTTACAAGCCTCGCTGTAGGCC
CCGGGAACCCAACGCGGTGTCAGAGAAGTGGGGTCCCCTACGAGGGACCA
GGAGCTCCGGGCGGGCAGCAGCTGCGGAAGAGCCGCGCGAGGCTTCCCAG
AACCCGGCAGGGGCGGGAAGACGCAGGAGTGGGGAGGCGGAACCGGGACC
CCGCAGAGCCCGGGTCCCTGCGCCCCACAAGCCTTGGCTTCCCTGCTAGG
GCCGGGCAAGGCCGGGTGCAGGGCGCGGCTCCAGGGAGGAAGCTCCGGGG
CGAGCCCAAGACGCCTCCCGGGCGGTCGGGGCCCAGCGGCGGCGTTCGCA
GTGGAGCCGGGCACCGGGCAGCGGCCGCGGAACACCAGCTTGGCGCAGGC
TTCTCGGTCAGGAACGGTCCCGGGCCTCCCGCCCGCCTCCCTCCAGCCCC
TCCGGGTCCCCTACTTCGCCCCGCCAGGCCCCCACGACCCTACTTCCCGC
GGCCCCGGACGCCTCCTCACCTGCGAGCCGCCCTCCCGGAAGCTCCCGCC
GCCGCTTCCGCTCTGCCGGAGCCGCTGGGTCCTAGCCCCGCCGCCCCCAG
TCCGCCCGCGCCTCCGGGTCCTAACGCCGCCGCTCGCCCTCCACTGCGCC
CTCCCCGAGCGCGGCTCCAGGACCCCGTCGACCCGGAGCGCTGTCCTGTC
GGGCCGAGTCGCGGGCCTGGGCACGGAACTCACGCTCACTCCGAGCTCCC
GACGTGCACACGGCTCCCATGCGTTGTCTTCCGAGCGTCAGGCCGCCCCT
ACCCGTGCTTTCTGCTCTGCAGACCCTCTTCCTAGACCTCCGTCCTTTGT
```

# About the files

1. FASTA
2. **RNA-seq data (FASTQ)**
3. GFF3/GTF
4. SAM/BAM

```
@HWUSI-EAS525:2:1:13336:1129#0/1
GTTGGAGCCGGCGAGCGGGACAAGGCCCTTGTCCA
+
ccacacccacccccccccc[[cccc_ccaccbbb_
@HWUSI-EAS525:2:1:14101:1126#0/1
GCCGGGACAGCGTGTTGGTTGGCGCGCGGTCCCTC
+
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
@HWUSI-EAS525:2:1:15408:1129#0/1
CGGCCTCATTCTTGGCCAGGTTCTGGTCCAGCGAG
+
cghhchhgchehhdffccgdgh]gcchhcahWcea
@HWUSI-EAS525:2:1:15457:1127#0/1
CGGAGGCCCCCGCTCCTCTCCCCCGCGCCCGCGCC
+
^BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
@HWUSI-EAS525:2:1:15941:1125#0/1
TTGGGCCCTCCTGATTTCATCGGTTCTGAAGGCTG
+
SUIF\_XYWW]VaOZZZ\V\bYbb_]ZXTZbbb_b
@HWUSI-EAS525:2:1:16426:1127#0/1
GCCCGTCCTTAGAGGCTAGGGGACCTGCCCGCCGG
```

# About the files

1. FASTA
2. **RNA-seq data (FASTQ)**
3. GFF3/GTF
4. SAM/BAM

```
@HWUSI-EAS525:2:1:13336:1129#0/1
GTTGGAGCCGGCGAGCGGGACAAGGCCCTTGTCCA
+
ccacacccacccccccccc[[cccc_ccaccbbb_
@HWUSI-EAS525:2:1:14101:1126#0/1
GCCGGGACAGCGTGTTGGTTGGCGCGCGGTCCCTC
+
```

**Single-end data: one file per sample**

**Paired-end data: two files per sample**

```
+
SUIF\_XYWW]VaOZZZ\V\bYbb_]ZXTZbbb_b
@HWUSI-EAS525:2:1:16426:1127#0/1
GCCCGTCCTTAGAGGCTAGGGGACCTGCCCGCCGG
```

# About the files

1. FASTA
2. FASTQ
3. **Annotation (GFF3/GTF)**
4. SAM/BAM

```
chr12   unknown exon    96066054        96067770
.       +       .       gene_id "PGAM1P5"; gene_name
"PGAM1P5"; transcript_id "NR_077225"; tss_id "TSS14770";
chr12   unknown CDS     96076483        96076598
.       -       1       gene_id "NTN4"; gene_name
"NTN4"; p_id "P12149"; transcript_id "NM_021229"; tss_id
"TSS6395";
chr12   unknown exon    96076483        96076598
.       -       .       gene_id "NTN4"; gene_name
"NTN4"; p_id "P12149"; transcript_id "NM_021229"; tss_id
"TSS6395";
chr12   unknown CDS     96077274        96077487
.       -       2       gene_id "NTN4"; gene_name
"NTN4"; p_id "P12149"; transcript_id "NM_021229"; tss_id
"TSS6395";
...
```

This file can be opened in Excel

# About the files

1. FASTA

2. FASTQ

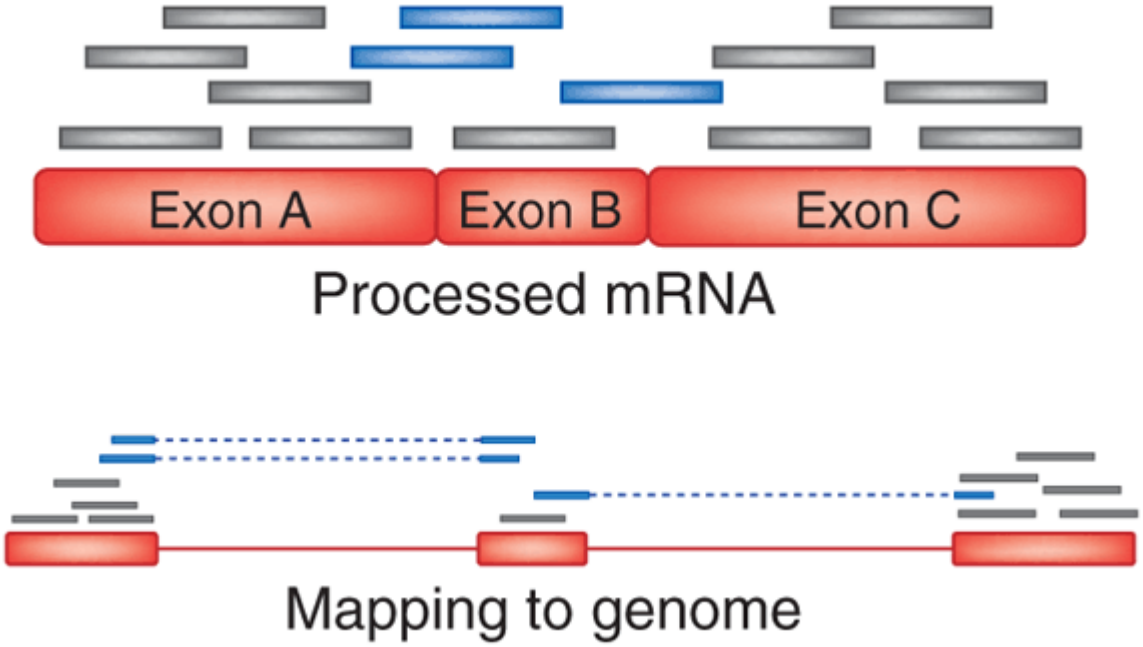
3. GFF3/GTF

4. **Alignment (SAM/BAM)**

```
HWUSI-EAS525_0042_FC:6:23:10200:18582#0/1	16	1	10	40	35M	*	0	0	AGCCAAAGATTGCATCAGTTCTGCTGCTATTTCCT	agafgfaffcfdf[fdcffcggggccfdffagggg	MD:Z:35 NH:i:1 HI:i:1 NM:i:0 SM:i:40 XQ:i:40 X2:i:0
HWUSI-EAS525_0042_FC:3:28:18734:20197#0/1	16	1	10	40	35M	*	0	0	AGCCAAAGATTGCATCAGTTCTGCTGCTATTTCCT	hghhghhhhhhhhhhhhhhhhghhhhhghhfhhh	MD:Z:35 NH:i:1 HI:i:1 NM:i:0 SM:i:40 XQ:i:40 X2:i:0
HWUSI-EAS525_0042_FC:3:94:1587:14299#0/1	16	1	10	40	35M	*	0	0	AGCCAAAGATTGCATCAGTTCTGCTGCTATTTCCT	hfhghhhhhhhhhhghhhhhhhhhhhhhhhhhhhg	MD:Z:35 NH:i:1 HI:i:1 NM:i:0 SM:i:40 XQ:i:40 X2:i:0
D3B4KKQ1:227:D0NE9ACXX:3:1305:14212:73591	0	1	11	40	51M	*	0	0	GCCAAAGATTGCATCAGTTCTGCTGCTATTTCCTCCTATCATTCTTTCTGA	CCCFFFFFFGFFHHJGIHHJJJFGGJJGIIIIIIGJJJJJJJJJJIJJJJE	MD:Z:51 NH:i:1 HI:i:1 NM:i:0 SM:i:40 XQ:i:40 X2:i:0
HWUSI-EAS525_0038_FC:5:35:11725:5663#0/1	16	1	11	40	35M	*	0	0	GCCAAAGATTGCATCAGTTCTGCTGCTATTTCCTC	hhehhhhhhhhhghghhhhhhhhhhhhhhhhhhh	MD:Z:35 NH:i:1 HI:i:1 NM:i:0 SM:i:40 XQ:i:40 X2:i:0
```

# Map reads to genome: Tophat or STAR

- Alignment of genomic sequencing vs RNA-seq




Cole Trapnell & Steven L Salzberg, *Nature Biotechnology* **27**, 455 - 457 (2009)

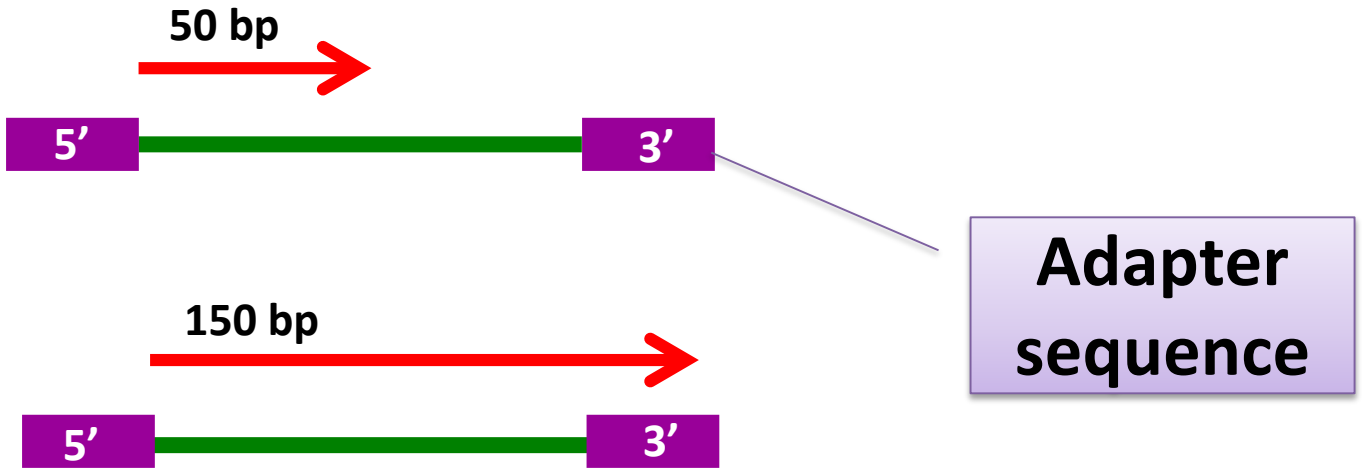
# Issues in alignment

- Trim or not trim reads (adapters and low quality reads)
- Novel Splicing junction discovery
- Remove PCR duplicates?
- Remove rRNA tRNA?
- Ambiguity in alignment

# Read Trimming: Low Quality and Adapter

- Not needed in most cases, especially for reads <=100bp;
- STAR soft clipping can remove some adapters;
- Avoid aggressive quality clipping which could cause miss-mapping

**Long sequence reads could read into the adapter:**



# Read Trimming: Low Quality and Adapter

## Trimming software:

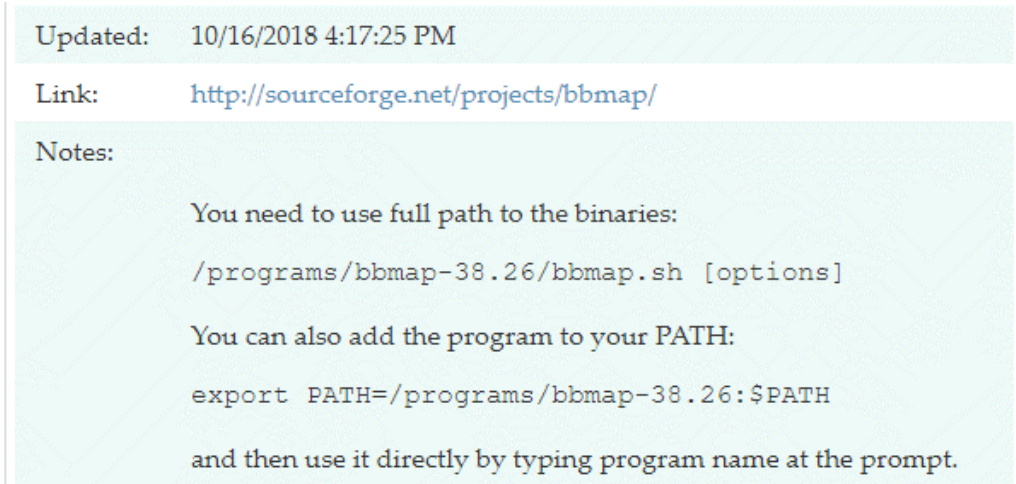
- BBDuk
- Trimmomatics
- Cutadapt

```
bbduk.sh in=reads.fq out=clean.fq
ref=adapters.fa t=8 ktrim=r k=23
mink=11 hdist=1 tpe tbo
```

If you want to run a software:

1. Read software manual;
2. Read instruction on BiopHPC

https://biohpc.cornell.edu/lab/userguide.aspx?a=software&i=248#c



| | |
|---|---|
| Updated: | 10/16/2018 4:17:25 PM |
| Link: | http://sourceforge.net/projects/bbmap/ |
| Notes: | |

You need to use full path to the binaries:

```
/programs/bbmap-38.26/bbmap.sh [options]
```

You can also add the program to your PATH:

```
export PATH=/programs/bbmap-38.26:$PATH
```

and then use it directly by typing program name at the prompt.

# Novel splicing junction

* very slow in TOPHAT

- STAR always perform novel junction detection;
- Use two-pass if novel junctions are critical for your project;

**PCR Duplicates: Not needed;**

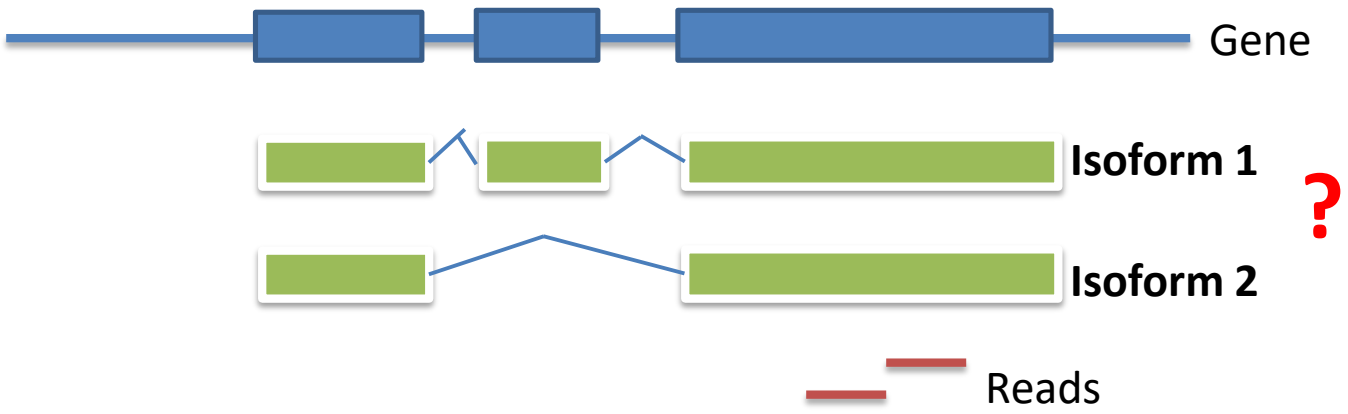
Not possible to detect true PCR duplicates in RNA-seq. Many of the identical reads are not PCR duplicates.

**rRNA tRNA removal: Not needed.**

They are mapped to genome, but not used in gene counts.

# Short reads caused ambiguity in mapping

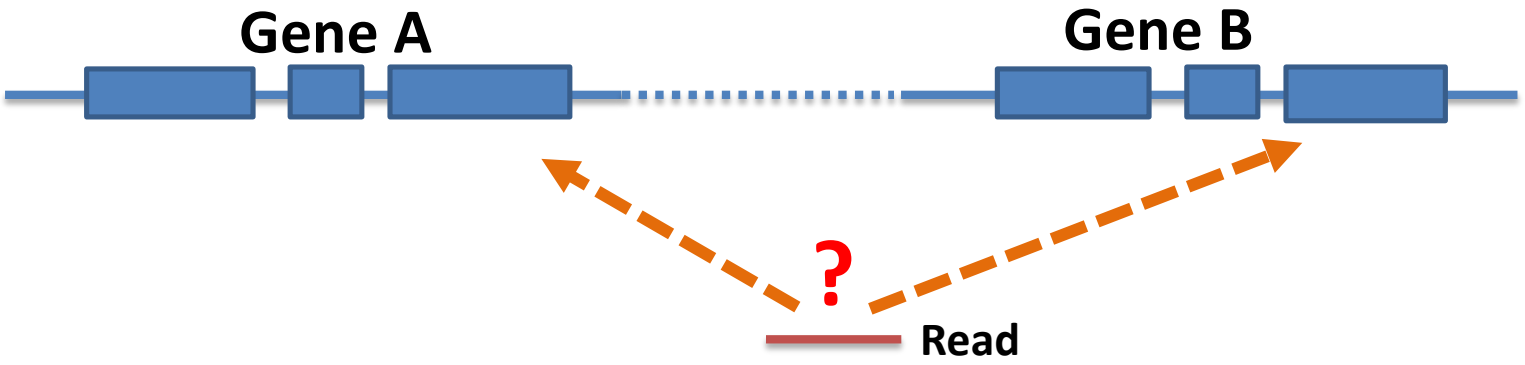
## 1. Ambiguity in splicing isoform;




# Use gene level quantification

# Short reads caused ambiguity in mapping

## 2. Ambiguity in paralogs




**STAR and HTSeq**

Discard multi-mapped reads

* This might not be desirable for some genes, e.g. duplicated genes in reference

# Diagnose low mapping rate

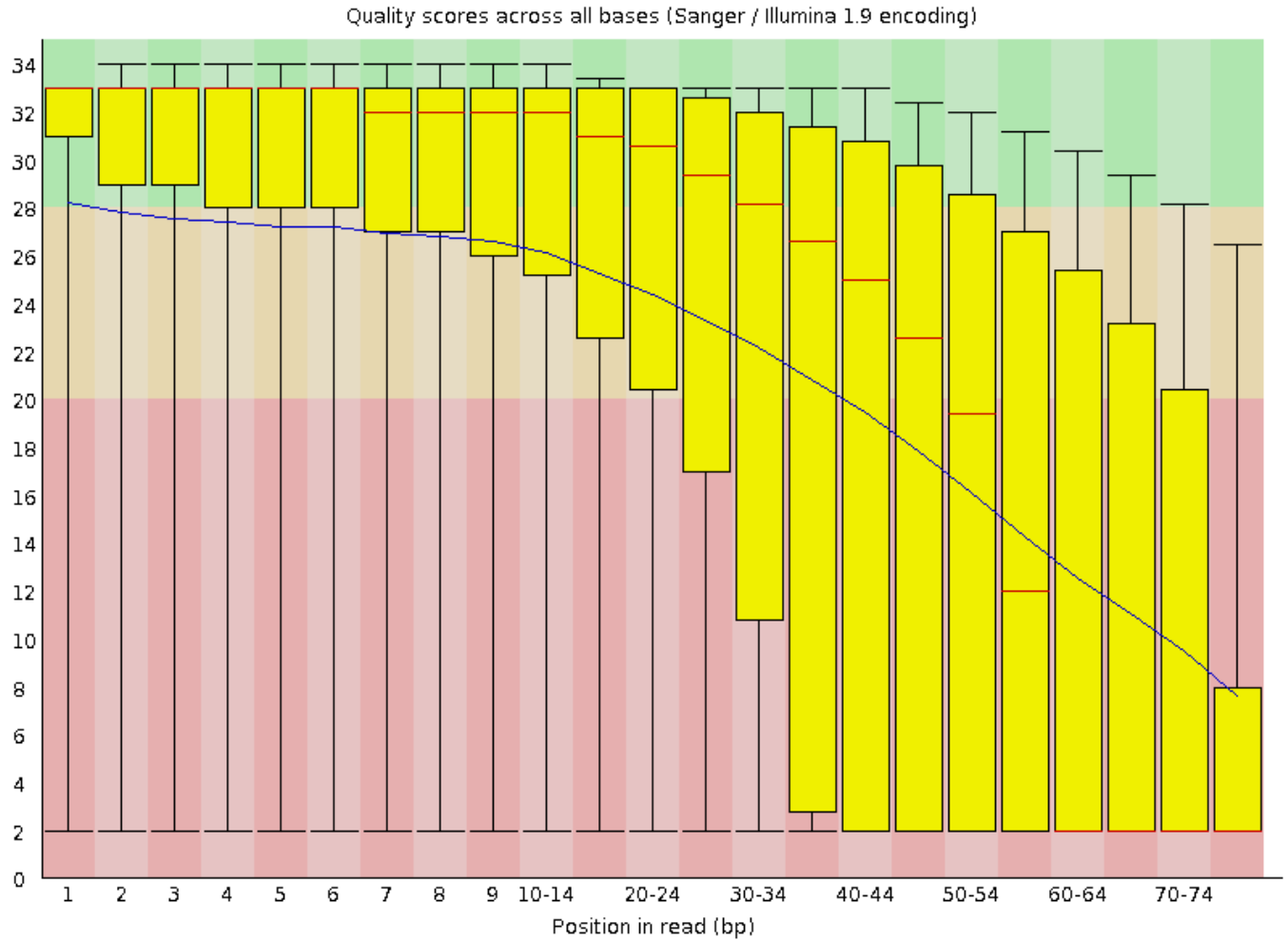
1. Low quality reads or reads with adapters *
   - Trimming tools (FASTX, Trimmomatic, et al.)

2. Contamination?
   - fastq_species_detector (Available on BioHPC Lab. It identifies species for reads by blast against Genbank)

* Trimming is not needed in majority of RNA-seq experiments except for de novo assembly

# Step 1. Quality Control (QC) using FASTQC Software

## 1. Sequencing quality score



# A BioHPC tool for detecting read contamination

# fastq_species_detector

**Commands**:

```
mkdir /workdir/my_db
cp /shared_data/genome_db/BLAST_NCBI/nt*  /workdir/my_db
cp /shared_data/genome_db/BLAST_NCBI/taxdb.*  /workdir/my_db
/programs/fastq_species_detector/fastq_species_detector.sh  my_file.fastq.gz  /workdir/my_db
```

**Sample output:**

**Read distribution over species:**

| Species | #Reads | %Reads |
|---|---|---|
| Drosophila melagaster | 254 | 35.234 |
| Cyprinus carpio | 74 | 10.529 |
| Triticum aestivum | 12 | 2.059 |
| Microtus ochrogaster | 3 | 1.765 |
| Dyella jiangningensis | 3 | 1.765 |

# STAR is becoming more commonly used than TOPHAT

- Much faster;
- Requires more memory
  - 30G for human genome;
  - 10G for 500GB genome.

## Index the genome:

```
STAR --runMode genomeGenerate \
--runThreadN 2 \
--genomeDir STARgenome \
--genomeFastaFiles testgenome.fa \
--sjdbGTFfile testgenome.gff3  \
--sjdbGTFtagExonParentTranscript Parent \
--sjdbOverhang 49
```

## Map reads:

```
STAR --genomeDir STARgenome \
--runThreadN 2 \
--readFilesIn a.fastq.gz \
--readFilesCommand zcat \
--outFileNamePrefix a_ \
--outFilterMultimapNmax 1 \
--outReadsUnmapped unmapped_a  \
--outSAMtype BAM SortedByCoordinate
```

## Index the genome:

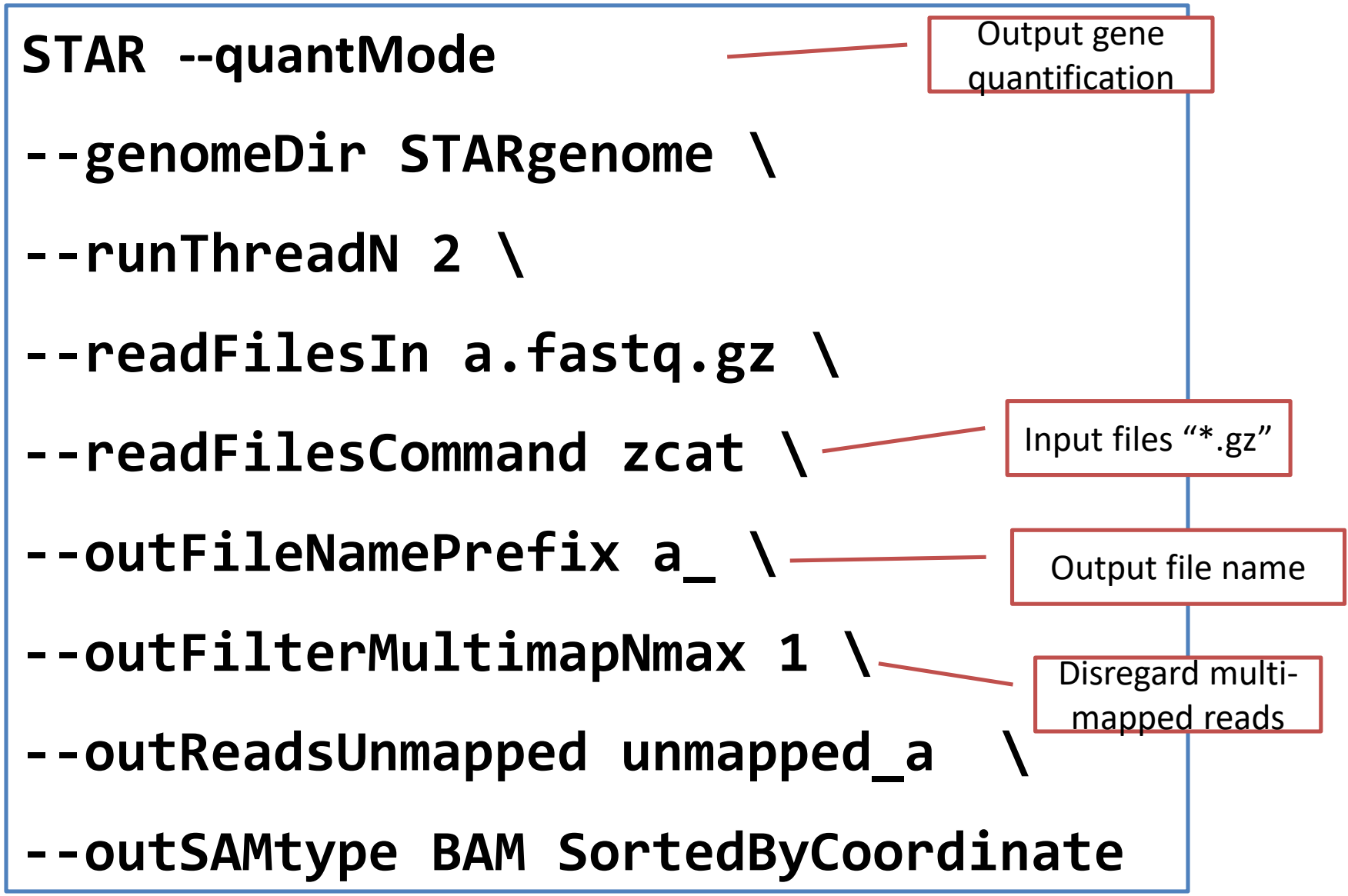
```
STAR --runMode genomeGenerate \
--runThreadN 2 \
--genomeDir STARgenome \
--genomeFastaFiles testgenome.fa \
--sjdbGTFfile testgenome.gtf  \
--sjdbOverhang 49
```

Use GTF, not gff3.
The STAR manual offers an option to use gff3, but in our experience, it is better to convert gff3 to gtf first with "gffread" tool.

Read length - 1

## Map reads:

```
STAR --quantMode
--genomeDir STARgenome \
--runThreadN 2 \
--readFilesIn a.fastq.gz \
--readFilesCommand zcat \
--outFileNamePrefix a_ \
--outFilterMultimapNmax 1 \
--outReadsUnmapped unmapped_a  \
--outSAMtype BAM SortedByCoordinate
```



# Setting parameters

```
STAR --quantMode GeneCounts --genomeDir genomedb  --
runThreadN 2 --outFilterMismatchNmax 2 --readFilesIn
WTa.fastq.gz --readFilesCommand zcat --outFileNamePrefix
WTa --outFilterMultimapNmax 1 --outSAMtype BAM
SortedByCoordinate
```

Some other parameters:

**--outFilterMismatchNmax :** max number of mismatch (Default 10)

**--outReadsUnmapped:** output unmapped reads

**Manual:**https://github.com/alexdobin/STAR/blob/master/doc/STARmanual.pdf

# Three Output files from STAR -1

***Log.final.out**

```
                         Number of input reads |       13547152
                     Average input read length |       49
                                   UNIQUE READS:
                    Uniquely mapped reads number |       12970876
                         Uniquely mapped reads % |       95.75%
                           Average mapped length |       49.32
                        Number of splices: Total |       1891468
             Number of splices: Annotated (sjdb) |       1882547
                        Number of splices: GT/AG |       1873713
                        Number of splices: GC/AG |       15843
                        Number of splices: AT/AC |       943
               Number of splices: Non-canonical |       969
```

# Three Output files from STAR -2

***ReadsPerGene.out.tab**

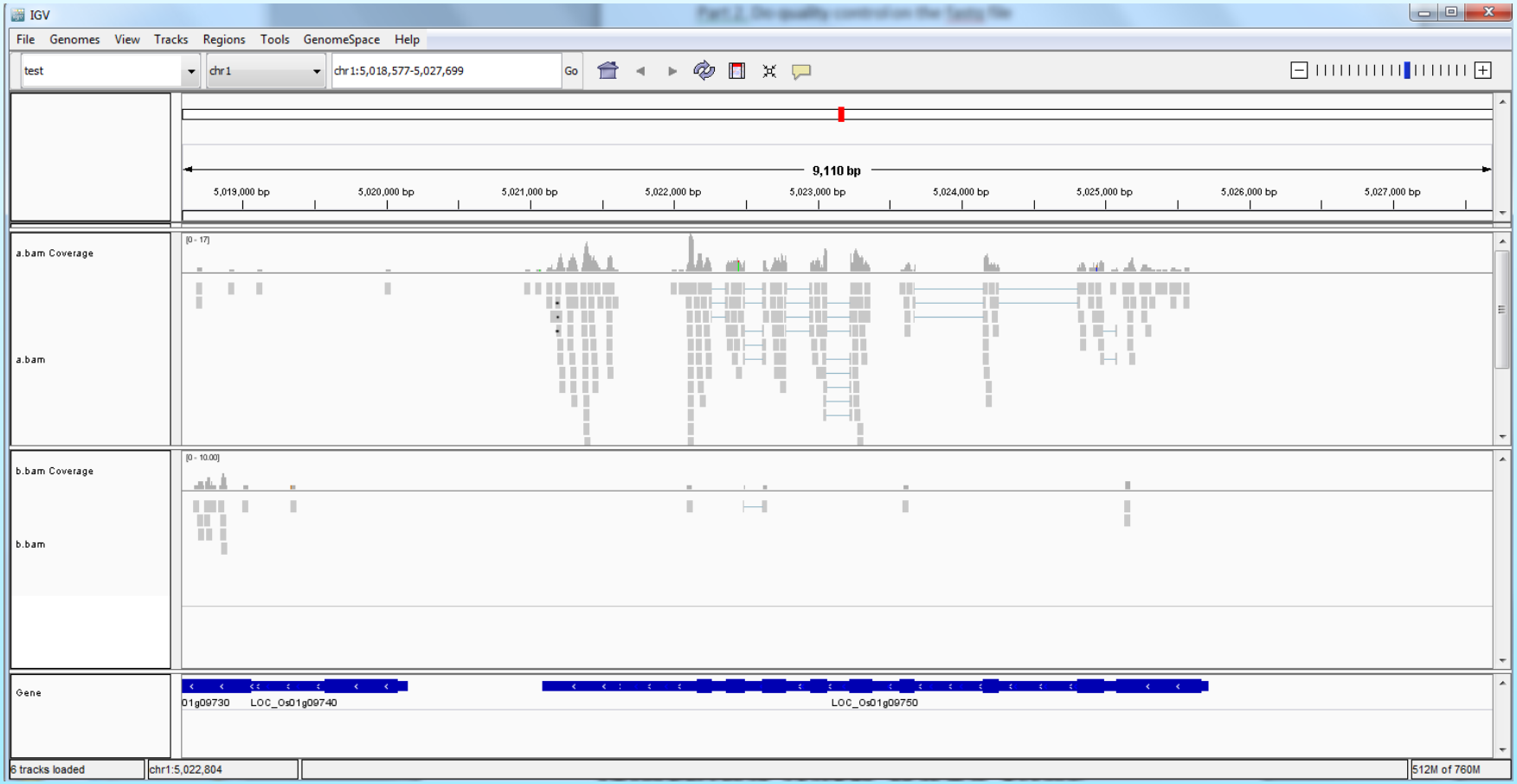
| | | | |
|---|---|---|---|
| N_unmapped | 1860780 | 1860780 | 1860780 |
| N_multimapping | 0 | 0 | 0 |
| N_noFeature | 258263 | 13241682 | 375703 |
| N_ambiguous | 461631 | 9210 | 17159 |
| gene:AT1G01010 | 50 | 1 | 49 |
| gene:AT1G01020 | 149 | 1 | 148 |
| gene:AT1G03987 | 0 | 0 | 0 |
| gene:AT1G01030 | 77 | 0 | 77 |
| gene:AT1G01040 | 583 | 41 | 669 |
| … | | | |

*column 1: gene ID*
*column 2: counts for unstranded RNA-seq*
*column 3: counts for the 1st read strand aligned with RNA (htseq-count option -s yes)*
*column 4: counts for the 2nd read strand aligned with RNA (htseq-count option -s reverse)*

# Three Output files from STAR -3

# Visualizing BAM files with IGV

* Before using IGV, the BAM files need to be indexed with “samtools index”, which creates a .bai file.



# Connection between software

**STAR output:**

Sample1

| | | | |
|---|---|---|---|
| N_unmapped | 1860780 | 1860780 | 1860780 |
| N_multimapping | 0 | 0 | 0 |
| N_noFeature | 258263 | 13241682 | 375703 |
| N_ambiguous | 461631 | 9210 | 17159 |
| gene:AT1G01010 | 50 | 1 | 49 |
| gene:AT1G01020 | 149 | 1 | 148 |
| gene:AT1G03987 | 0 | 0 | 0 |
| gene:AT1G01030 | 77 | 0 | 77 |
| gene:AT1G01040 | 583 | 41 | 669 |
| … | | | |

Sample2

| | | | |
|---|---|---|---|
| N_unmapped | 1637879 | 1637879 | 1637879 |
| N_multimapping | 0 | 0 | 0 |
| N_noFeature | 224759 | 11828019 | 354396 |
| N_ambiguous | 445882 | 8133 | 14924 |
| gene:AT1G01010 | 57 | 0 | 57 |
| gene:AT1G01020 | 174 | 2 | 172 |
| gene:AT1G03987 | 1 | 1 | 0 |
| gene:AT1G01030 | 91 | 3 | 88 |
| gene:AT1G01040 | 516 | 27 | 594 |
| gene:AT1G03993 | 0 | 81 | 2 |

**EdgeR input:**

| gene | Sample1 | Sample2 | Sample3 | Sample4 |
|---|---|---|---|---|
| AT1G01010 | 57 | 49 | 36 | 40 |
| AT1G01020 | 172 | 148 | 197 | 187 |
| AT1G03987 | 0 | 0 | 0 | 0 |
| AT1G01030 | 88 | 77 | 74 | 101 |
| AT1G01040 | 594 | 669 | 504 | 633 |
| AT1G03993 | 2 | 1 | 0 | 0 |
| … | … | … | … | … |

```
paste file1 file2 file3 file4 | \
cut -f1,4,8,12,16 | \
tail -n +5 \
> tmpfile

cat tmpfile | \
sed "s/^gene://" \
>gene_count.txt
```

## Connection between software

# Reading file into R

| | | | | |
|---|---|---|---|---|
| AT1G01010 | 57 | 49 | 36 | 40 |
| AT1G01020 | 172 | 148 | 197 | 187 |
| AT1G03987 | 0 | 0 | 0 | 0 |
| AT1G01030 | 88 | 77 | 74 | 101 |
| AT1G01040 | 594 | 669 | 504 | 633 |
| AT1G03993 | 2 | 1 | 0 | 0 |
| … | … | … | … | … |

```
x <- read.delim("gene_count.txt", header=F, row.names=1)

colnames(x)<-c("WTa","WTb","MUa","MUb")
```

# Making Shell Script

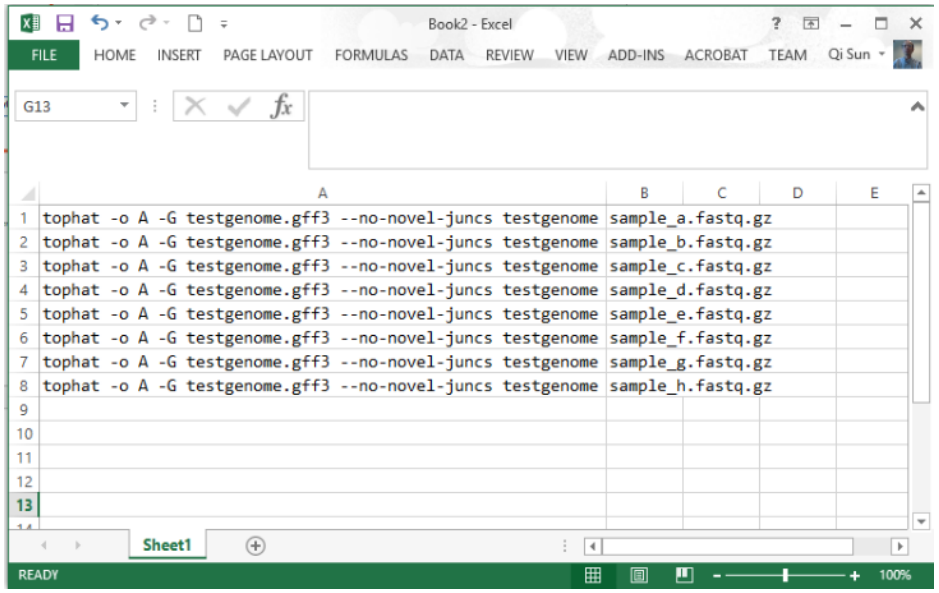
1. You can use Excel to make a shell script, and copy to the Notepad++/Text Wrangler, and remove tab characters

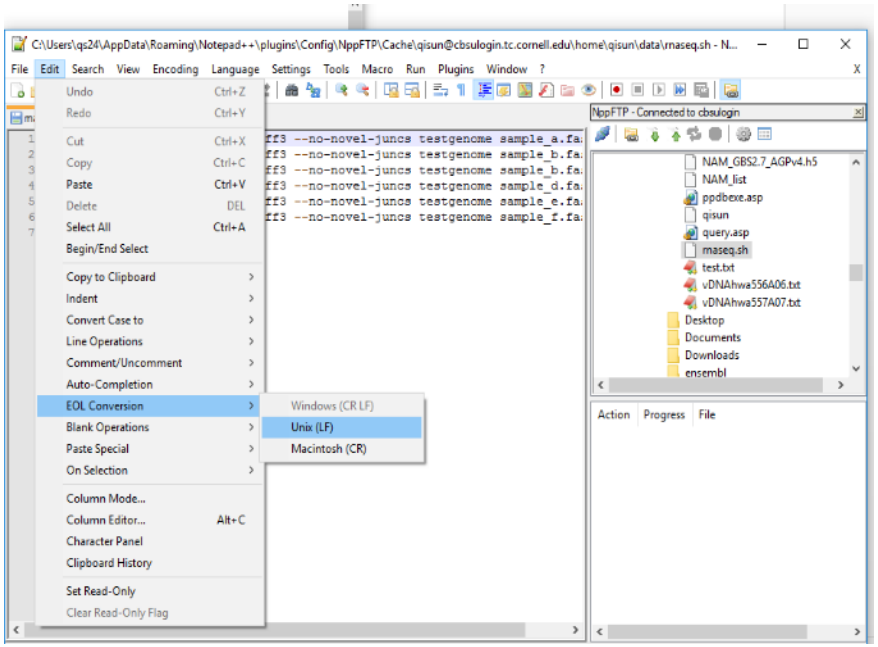
2. Mac Excel user:

Make sure to use “mac2unix myfile” command to convert it to Linux file.

3. Windows user

Make sure to save as UNIX file in NotePad++. Or use the “dos2unix myfile” command to convert it to Linux file.




**End of line in Text file**

Linux: /n
Win: /r/n
Mac (9): /r
Mac (x): /n (Excel still use OS 9 style)

## Running Shell Script (run it in “screen”)

```
sh ~/runtophat.sh >& mylog &
```

**Monitoring a job**
top
top -o %MEM
ps -fu myUserID
ps -fu myUserID | grep STAR

**Kill a job:**
kill PID ## you need to kill both shell script and STAR alignment that is still running
kill -9 PID
killall userID

**Run multiple jobs:**
nohup perl_fork_univ.pl script.sh 5 >& runlog &

# Parallelization (run in “screen” )

```
perl_fork_univ.pl ~/runSTAR.sh 5>& mylog &
```

5 jobs at a time

# QuantSeq 3’ mRNA sequencing for RNA quantification

1. Remove 12bp from 5’, trim adapter;
2. Alignment with STAR;
3. Quantification using forward strand counts;
4. If annotation is poor, you might need to extend 3’ UTR ;

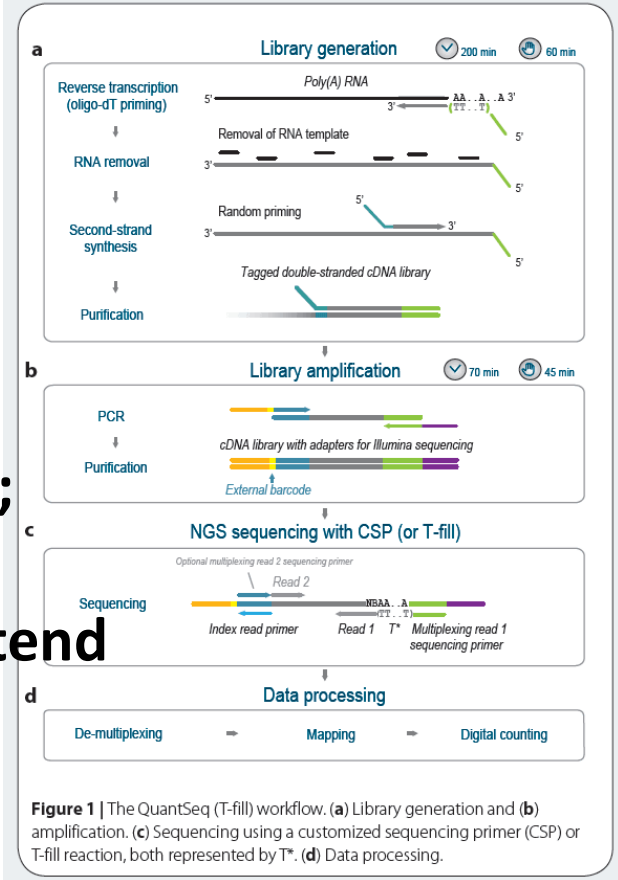



**Figure 1 |** The QuantSeq (T-fill) workflow. (**a**) Library generation and (**b**) amplification. (**c**) Sequencing using a customized sequencing primer (CSP) or T-fill reaction, both represented by T*. (**d**) Data processing.

# https://www.lexogen.com/quantseq-data-analysis/

```
#### use fastqc to check your data; please adjust the number of threads to your machine
fastqc --outdir qualitycheck --format fastq --threads 8 fastq/runID*
#### check the result with a browser
######## preparation for mapping
###go to fastq directory
cd fastq
### remove the adapter contamination, polyA read through, and low quality tails
for sample in runID*R1_001.fastq; do cat $i | bbduk.sh in=stdin.fq out=${i}_trimmed_clean
ref=/data/resources/polyA.fa.gz,/data/resources/truseq_rna.fa.gz k=13 ktrim=r useshortkmers=t mink=5 qtrim=r
trimq=10 minlength=20; done
### create symbolic links for better handling
### for-loops can be used according to name and structure
ln -s runID_control1_S1_L001_R1_001.fastq_trimmed_clean control1_R1.fastq
...
ln -s runID_treatment2_S4_L001_R12_001.fastq_trimmed_clean treatment2_R1.fastq
######### mapping
################################
# create for each sample a folder in star_out/
cd ..
mkdir star_out
mkdir star_out/control1
mkdir star_out/control2
...
### run star
for sample in control1 control2 treatment1 treatment2 ; do \
STAR --runThreadN 8 --genomeDir /data/star/human --readFilesIn fastq/${sample}_R1.fastq \
--outFilterType BySJout --outFilterMultimapNmax 20 --alignSJoverhangMin 8 --alignSJDBoverhangMin 1 \
--outFilterMismatchNmax 999 --outFilterMismatchNoverLmax 0.1 --alignIntronMin 20 \
--alignIntronMax 1000000 --alignMatesGapMax 1000000 --outSAMattributes NH HI NM MD
--outSAMtype BAM SortedByCoordinate --outFileNamePrefix star_out/${sample} ;\
done
#Indexed bam files are necessary for many visualization and downstream analysis tools
cd star_out
for bamfile in */starAligned.sortedByCoord.out.bam ; do samtools index ${bamfile}; done
```

# BioHPC Lab office hours

Time: 1-3 pm, every Monday & Thursday

Office: 618 Rhodes Hall

Sign-up: https://biohpc.cornell.edu/lab/office1.aspx

- General bioinformatics consultation/training is provided;
- Available throughout the year;

# Exercise 1

- STAR to align RNA-seq reads to genome and get read quantification

- Learn to use Linux shell script and parallelization