RNAseq Pipeline Construction

Overview:

- Last class:
  - Principles of RNAseq
  - Overview of practice data
- Today:
  - Build and test RNAseq pipeline
  - Preliminary analysis of gene expression

Overview:

- Last class:
  - Principles of RNAseq
  - Overview of practice data
- Today:
  - Build and test RNAseq pipeline
  - Preliminary analysis of gene expression

What students receive from to prep for class:

RNAseq_Pipeline.zip

- conda environment configuration: Demo_ENV.yml

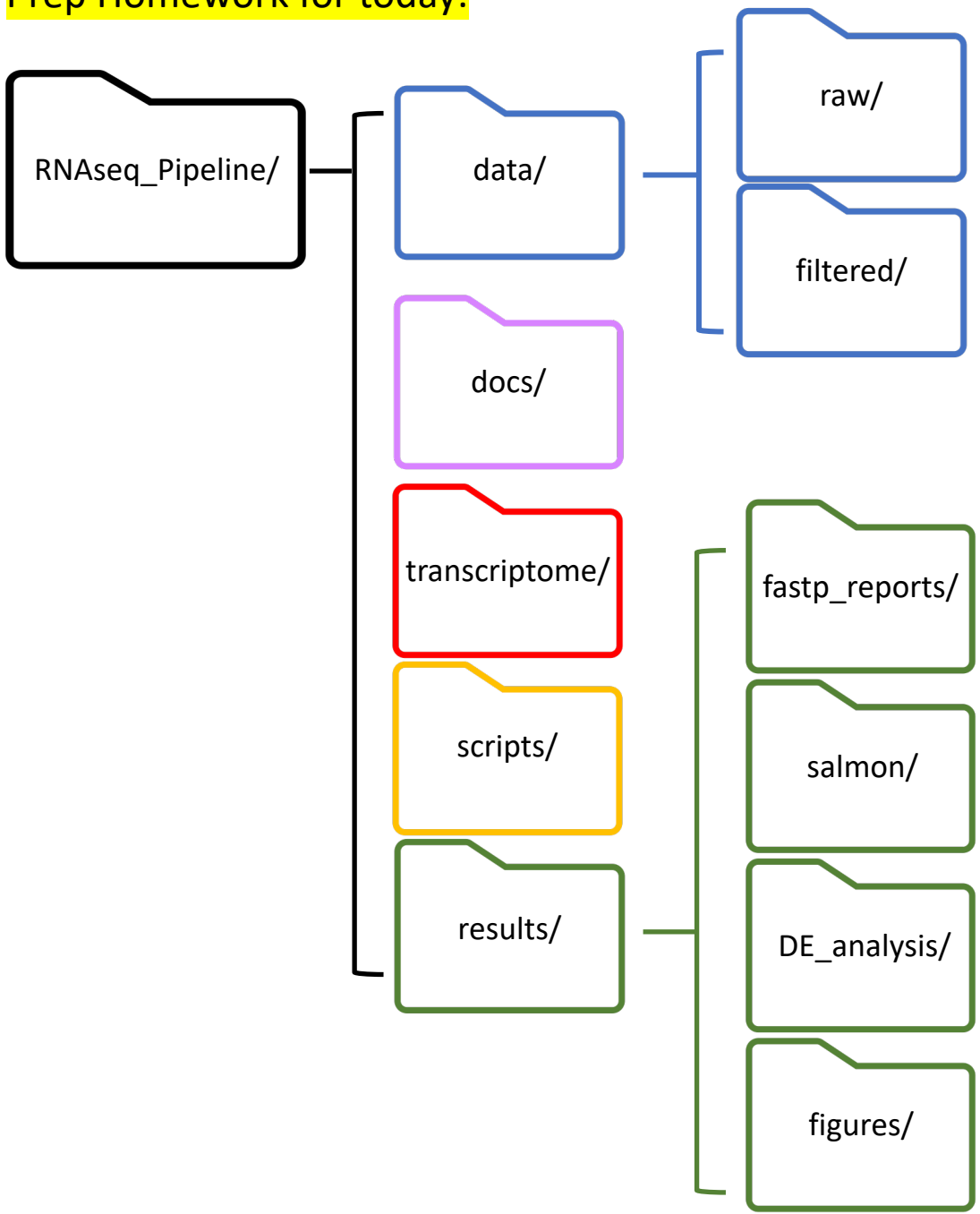
Prep Homework →

- **PARTIAL** STEP1_RNAseq_config.sh
- **PARTIAL** STEP2_RNAseq_Pipeline.sh

Build together in class →

- SampleList.txt

Prep Homework for today:



- **SET UP** a new project:
  - Edit STEP1_RNAseq_config.sh including directories and subdirectories
  - Run STEP1_RNAseq_config.sh
  - Link to Atom
- **CREATE** and **TEST** conda environment from Demo_ENV.yml:

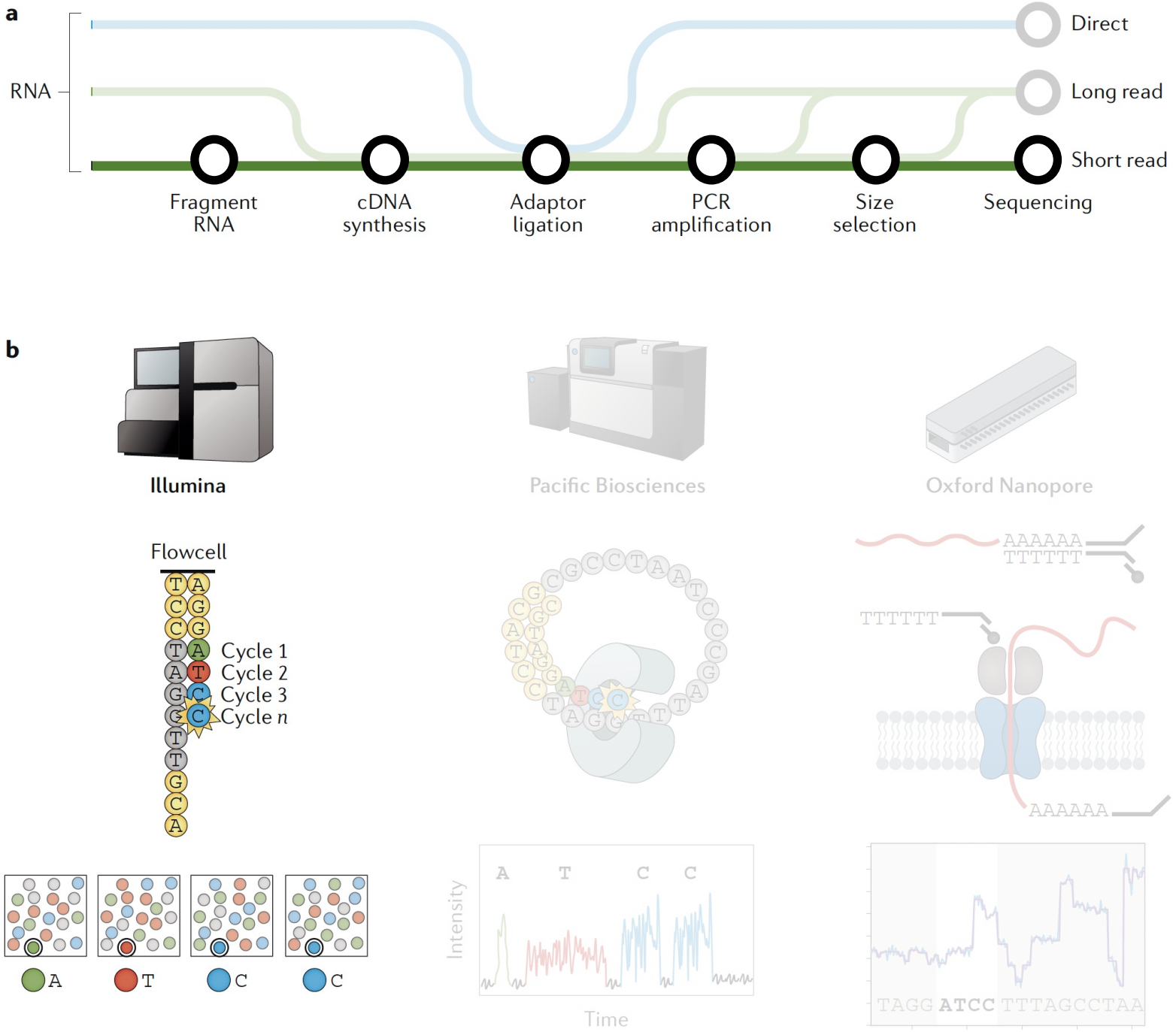
```
# create environment
conda env create –f Demo_ENV.yml

# activate environment
conda activate Demo_ENV

# test environment
python --version
multiqc --version
fastp --version
salmon --version
```

RNAseq Sample Data

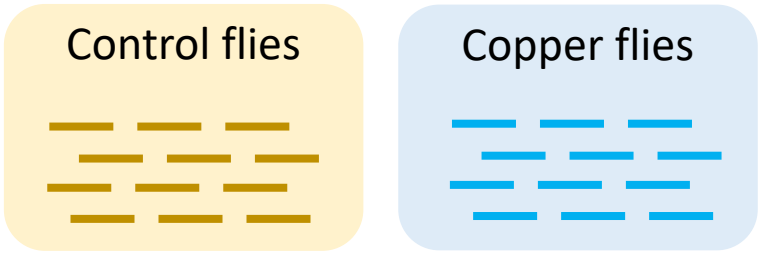
Data from:
Everman et al. 2021: SRA PRJNA633166
- 20 samples (subset to 10000 reads)

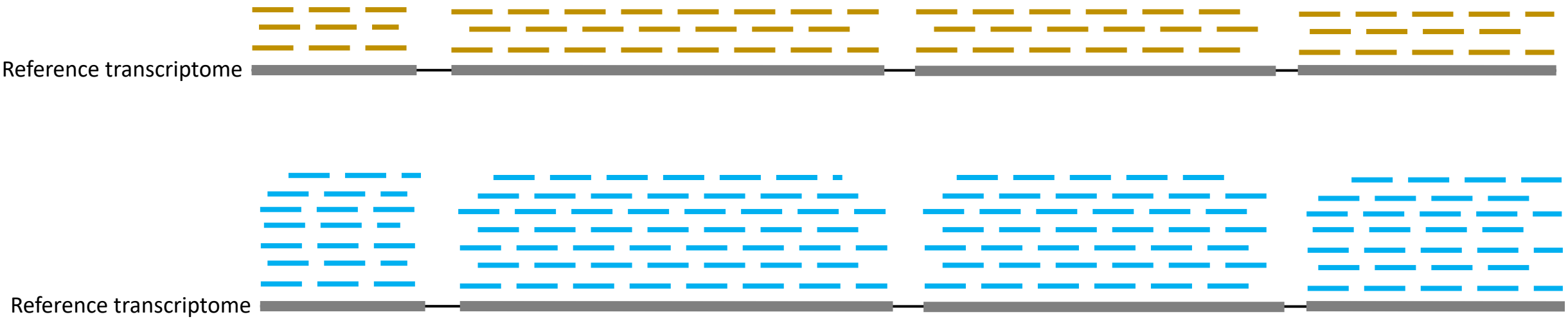
Experiment:
- 10 *D. melanogaster* strains
  - 6 strains are copper resistant
  - 4 strains are copper sensitive
- All females
- 2 treatment conditions
  - Control
  - 50mM $CuSO_4$

Question:
Does gene expression response to copper vary between high and low copper resistance strains?



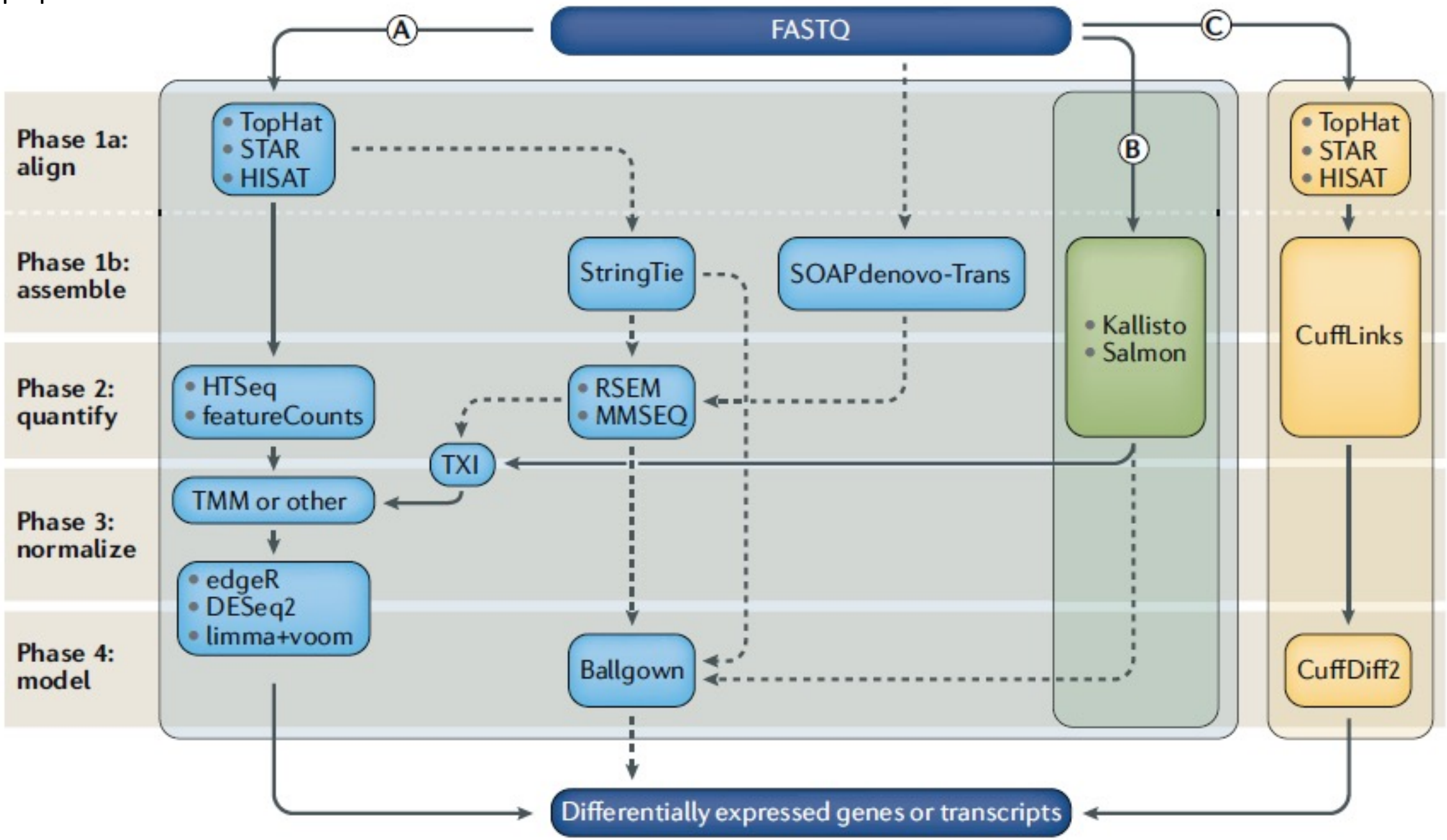
Differential Expression Analysis:



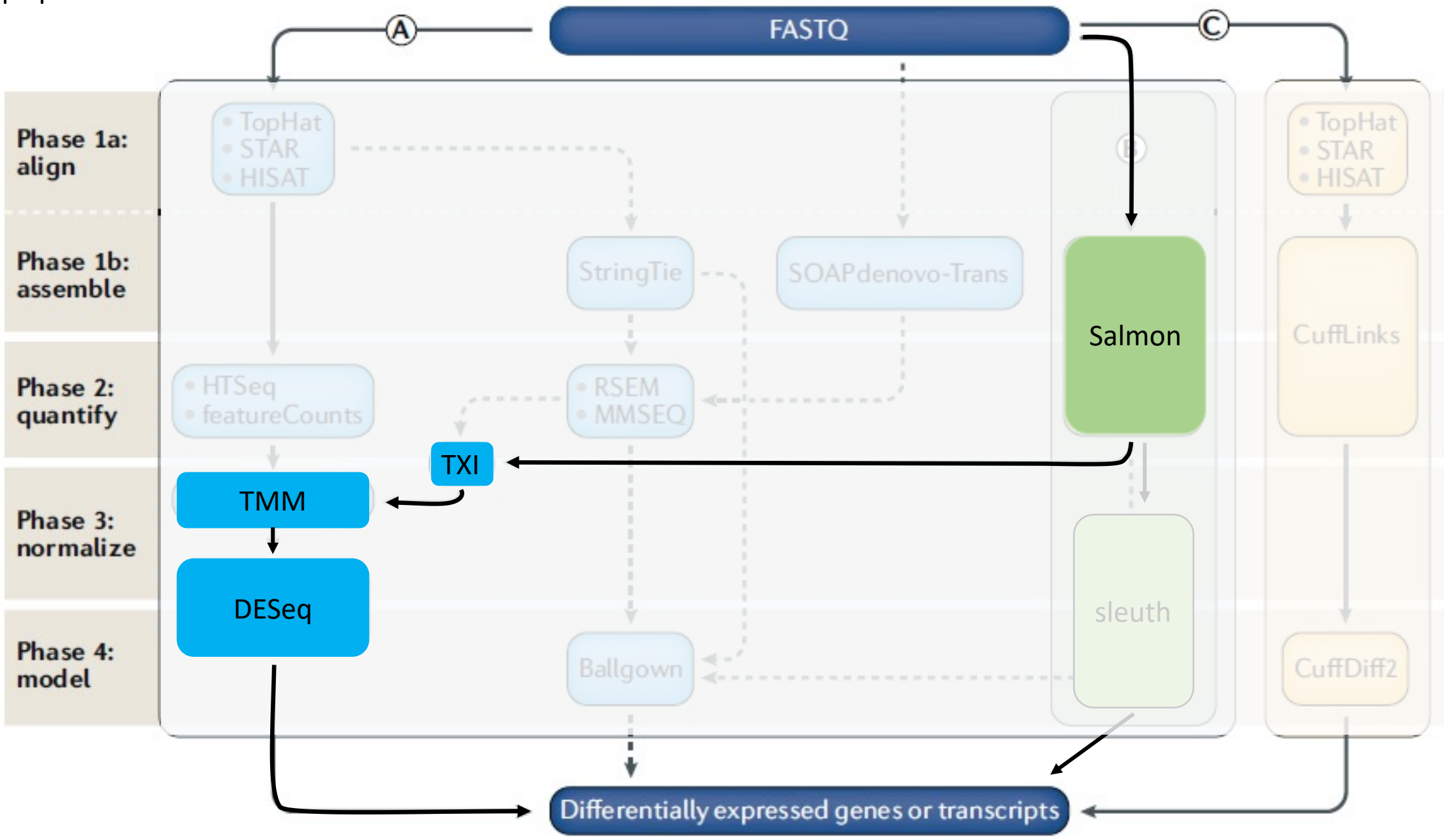


Question:
Does gene expression response to copper vary between high and low copper resistance strains?

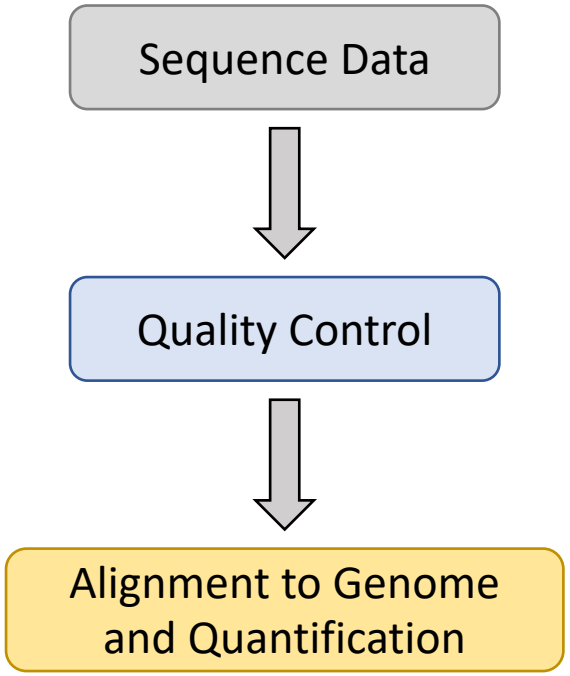
RNAseq Pipelines



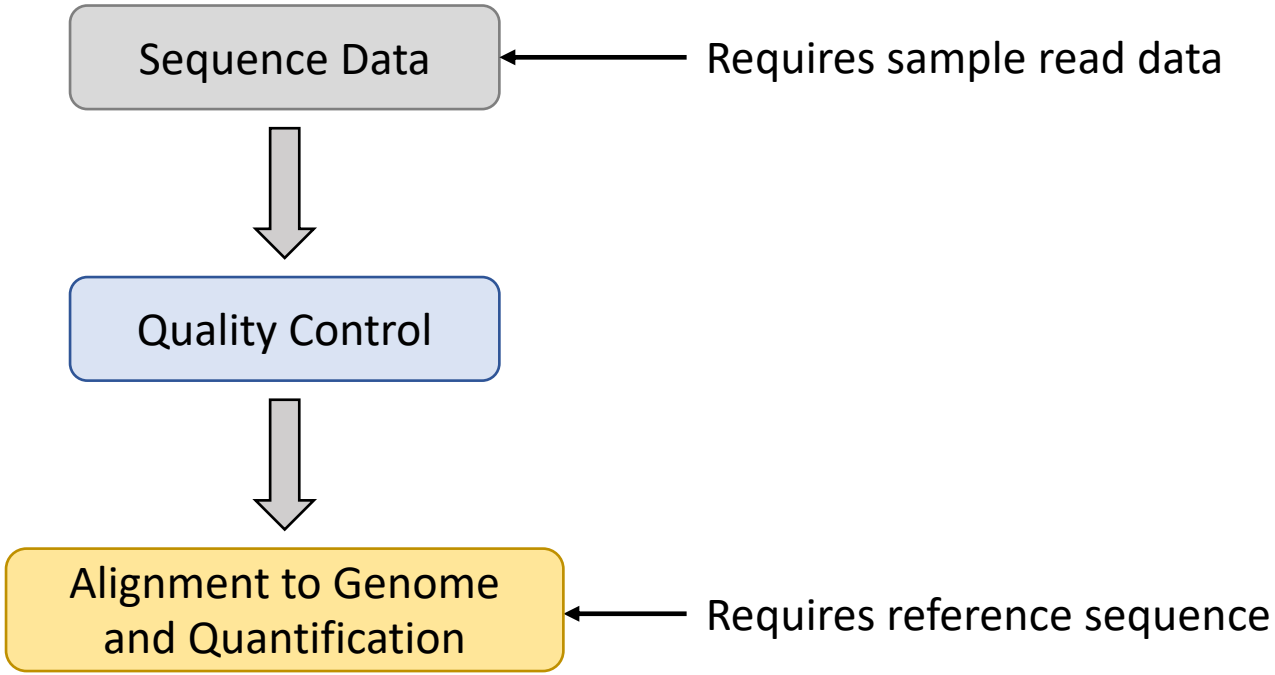
RNAseq Pipelines



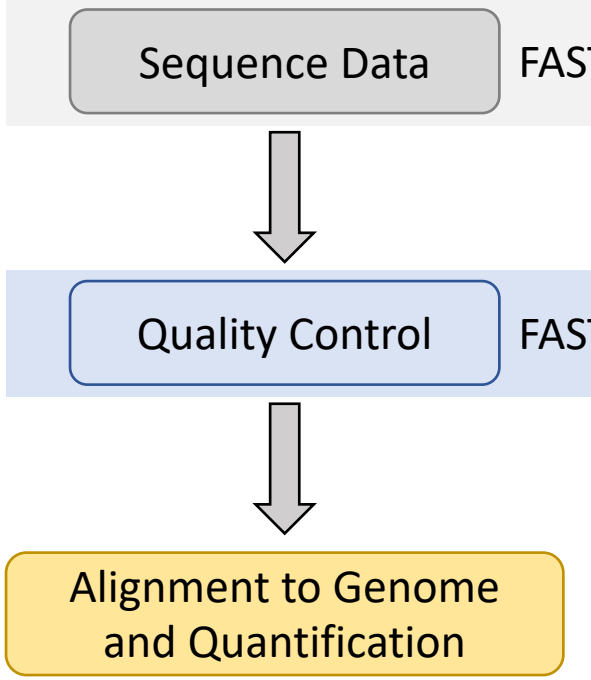
Building our Pipeline:



Building our Pipeline:



Building our Pipeline:

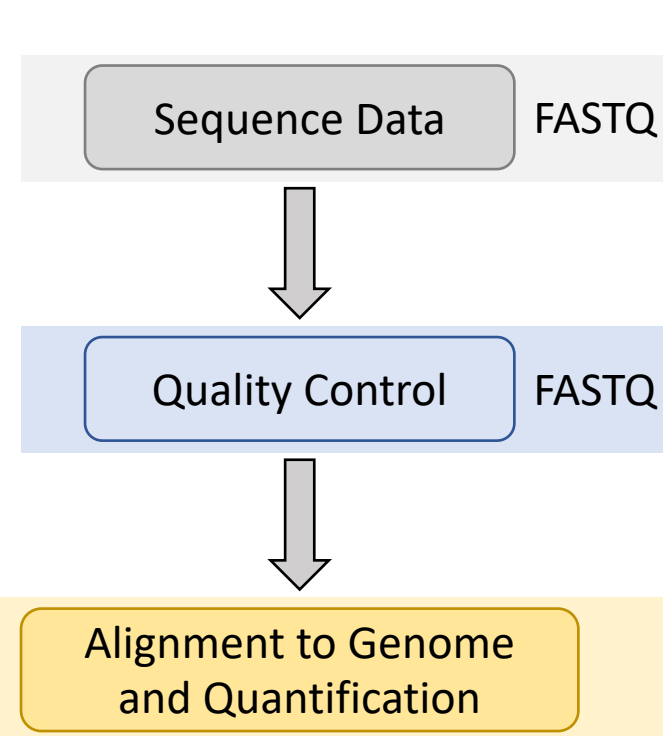



```
# Retrieve raw data and reference transcriptome

sh STEP2_RNAseq_Pipeline.sh
```

```
# TRIM and FILTER raw data

fastp...
multiqc...
```

fastp: https://github.com/OpenGene/fastp

Building our Pipeline:




```
# Retrieve raw data and reference transcriptome

STEP2_RNAseq_Pipeline.sh
```

```
# TRIM and FILTER raw data

fastp...
multiqc...
```

```
# INDEX reference and ALIGN trimmed data

salmon index...
salmon quant...
```

fastp: https://github.com/OpenGene/fastp

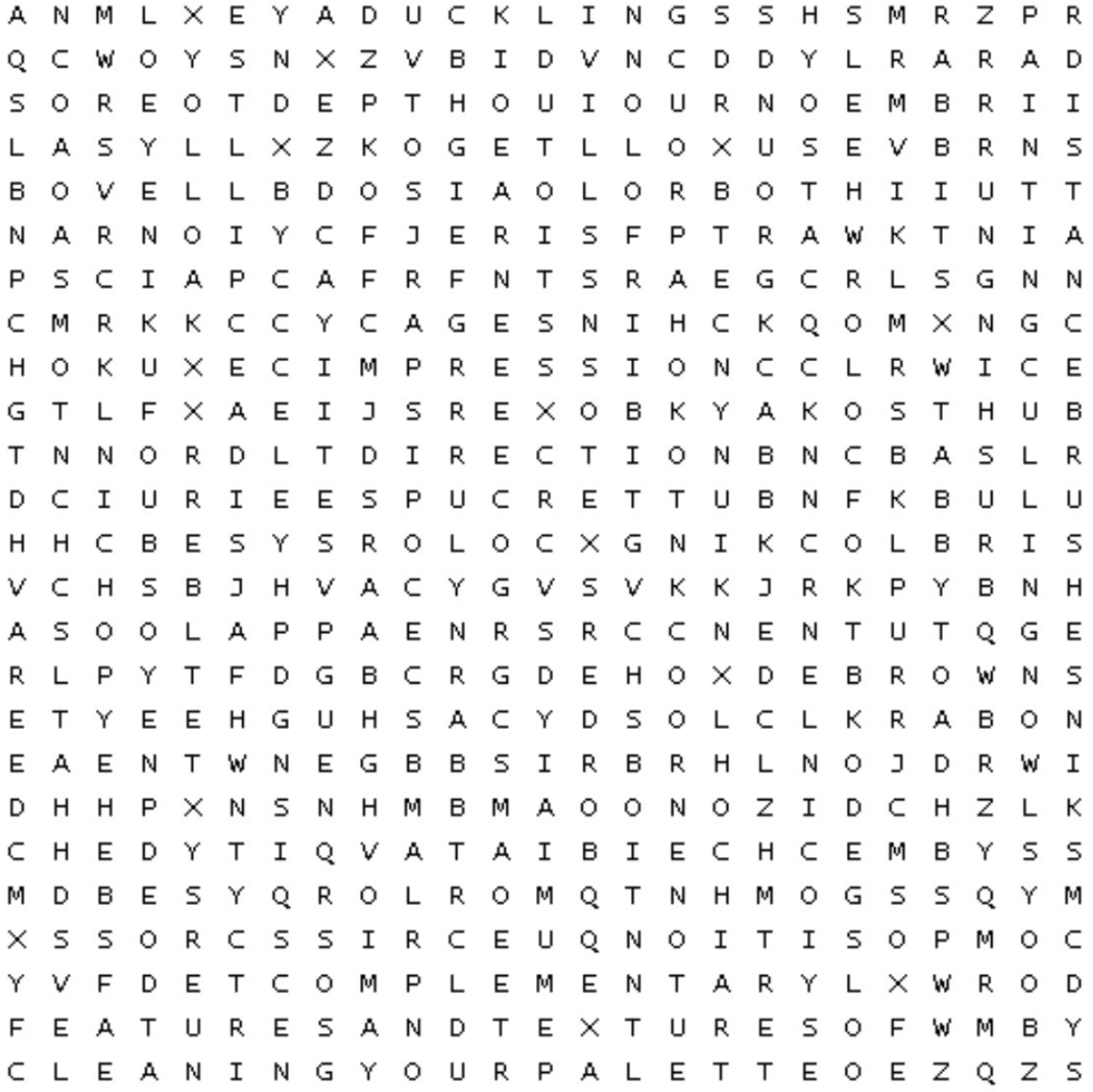
STEP 1: Indexing a reference genome:

- Indexing allows the aligner to quickly find potential alignment sites
- Indexing varies by aligner tool
- Indexing only needs to be run once

A N M L X E Y A D U C K L I N G S S H S M R Z P R
Q C W O Y S N X Z V B I D V N C D D Y L R A R A D
S O R E O T D E P T H O U I O U R N O E M B R I I
L A S Y L L X Z K O G E T L L O X U S E V B R N S
B O V E L L B D O S I A O L O R B O T H I I U T T
N A R N O I Y C F J E R I S F P T R A W K T N I A
P S C I A P C A F R F N T S R A E G C R L S G N N
C M R K K C C Y C A G E S N I H C K Q O M X N G C
H O K U X E C I M P R E S S I O N C C L R W I C E
G T L F X A E I J S R E X O B K Y A K O S T H U B
T N N O R D L T D I R E C T I O N B N C B A S L R
D C I U R I E E S P U C R E T T U B N F K B U L U
H H C B E S Y S R O L O C X G N I K C O L B R I S
V C H S B J H V A C Y G V S V K K J R K P Y B N H
A S O O L A P P A E N R S R C C N E N T U T Q G E
R L P Y T F D G B C R G D E H O X D E B R O W N S
E T Y E E H G U H S A C Y D S O L C L K R A B O N
E A E N T W N E G B B S I R B R H L N O J D R W I
D H H P X N S N H M B M A O O N O Z I D C H Z L K
C H E D Y T I Q V A T A I B I E C H C E M B Y S S
M D B E S Y Q R O L R O M Q T N H M O G S S Q Y M
X S S O R C S S I R C E U Q N O I T I S O P M O C
Y V F D E T C O M P L E M E N T A R Y L X W R O D
F E A T U R E S A N D T E X T U R E S O F W M B Y
C L E A N I N G Y O U R P A L E T T E O E Z Q Z S

STEP 1: Indexing a reference genome:

- Indexing allows the aligner to quickly find potential alignment sites
- Indexing varies by aligner tool
- Indexing only needs to be run once



Hash Table:

k_i	[interval)
CLE	[190, 199)
COL	[200, 205)
DEE	[206, 210)

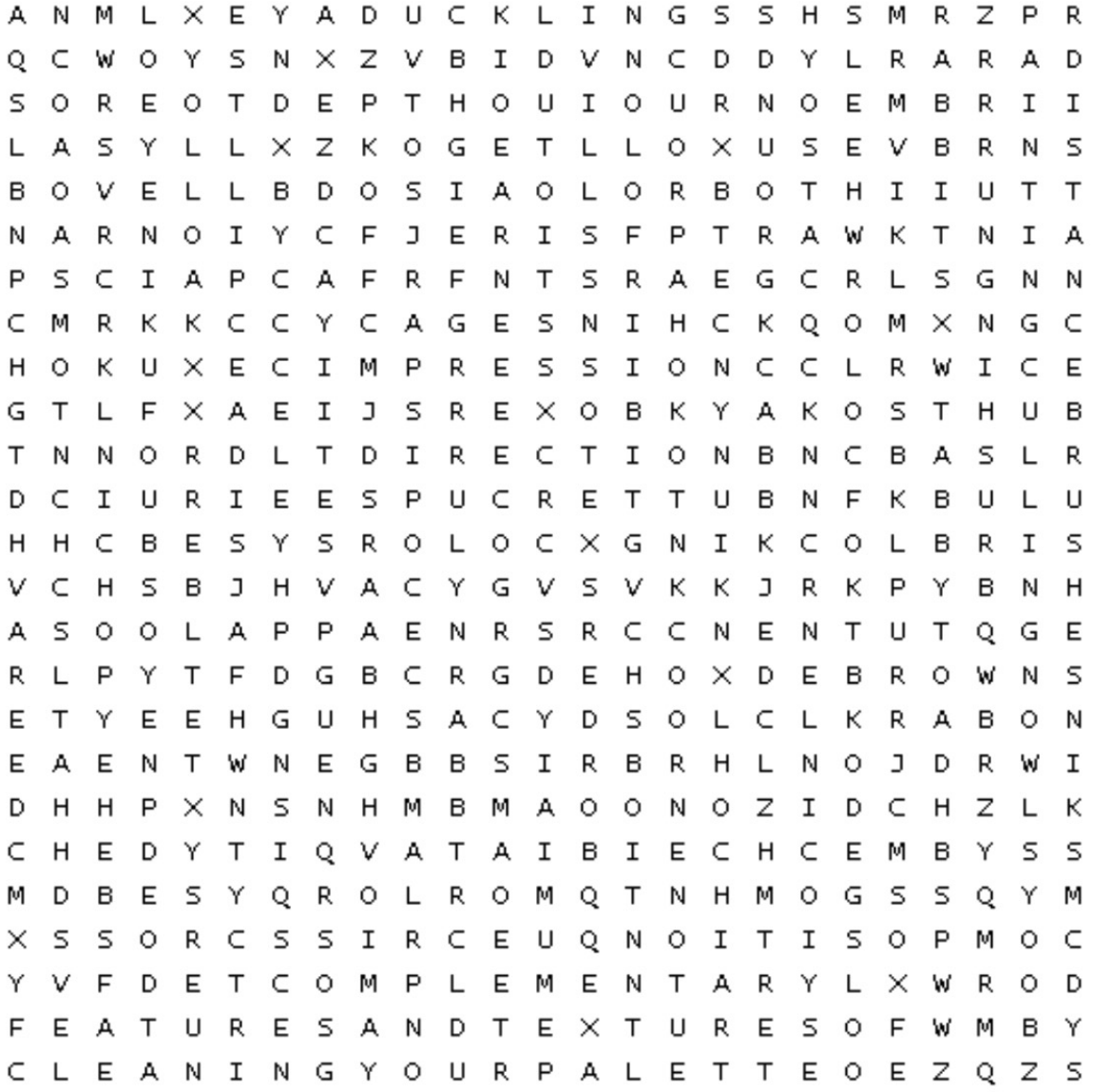
Suffix Array:

C	C	C	C	C	C	C	C	D	D
L	L	O	O	O	O	O	O	E	E
E	E	L	L	L	L	L	L	E	E
A	A	D	O	O	O	L	T	P	R
N	N		R	R	R	I			
E	I		S	F	W	E			
R	N			A	H	S			
	G			M	E				
				I	E				
				L	E				
				I	L				
				E					
				S					

- Groups suffixes alphabetically
- Stores information about position of the characters in the “genome”

STEP 1: Indexing a reference genome:

- Indexing allows the aligner to quickly find potential alignment sites
- Indexing varies by aligner tool
- Indexing only needs to be run once

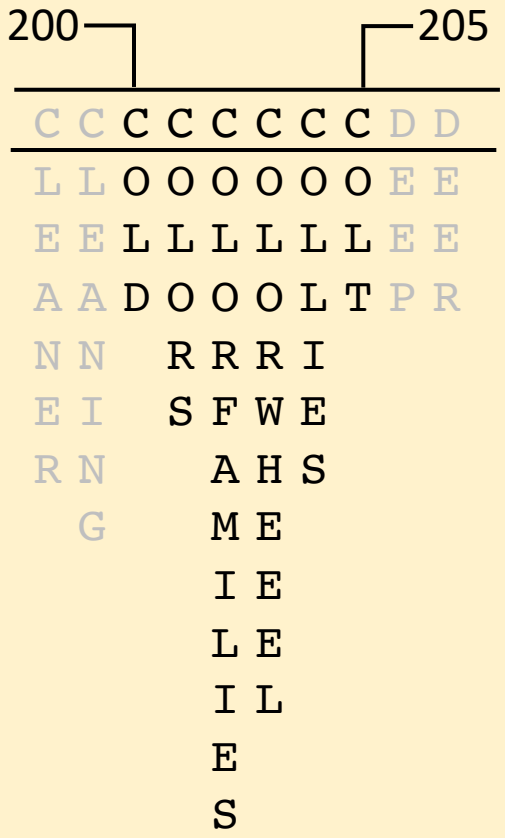


Hash Table:

k_i	[interval)
CLE	[190, 199)
COL	[200, 205)
DEE	[206, 210)

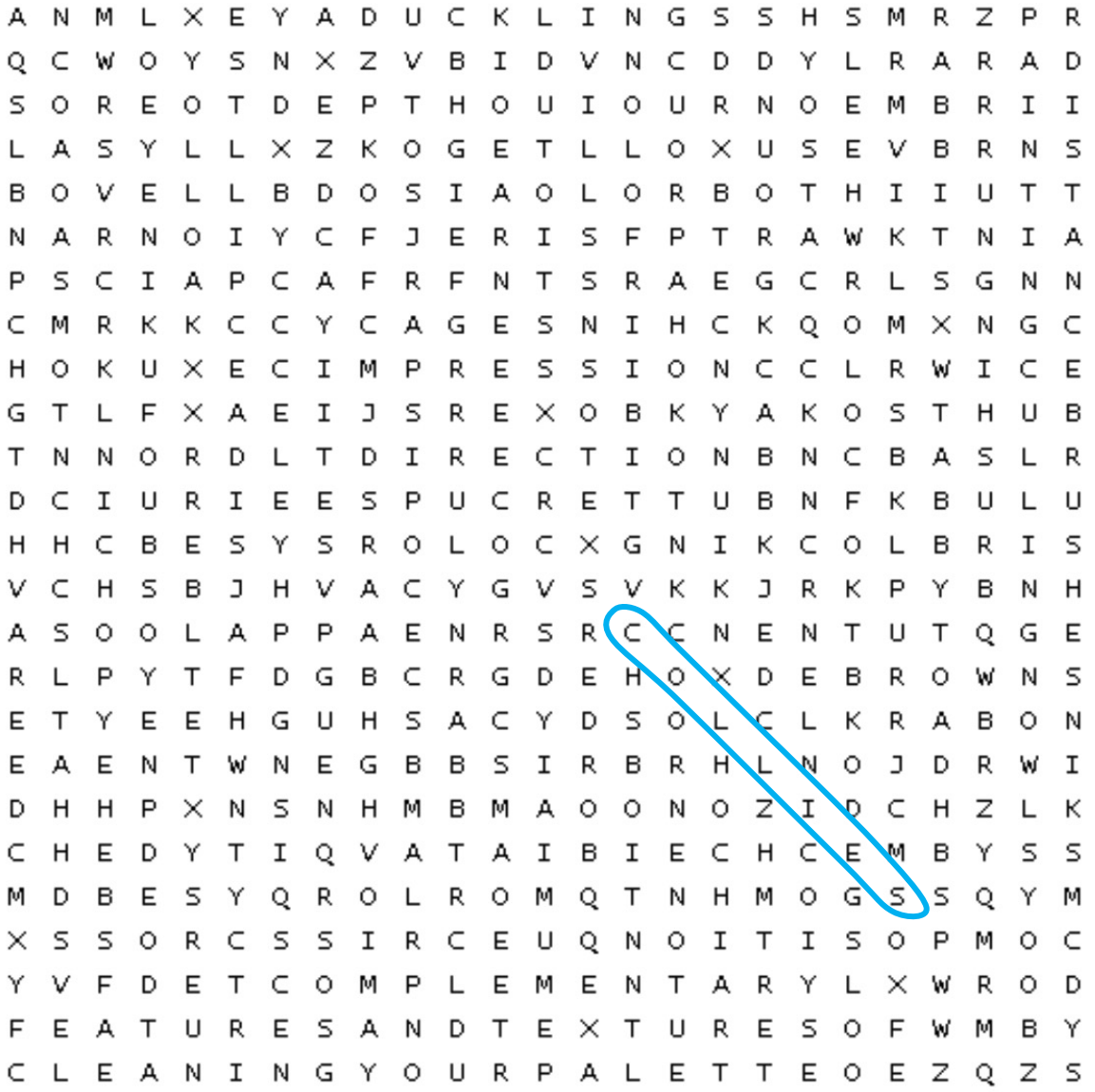
Query: COLLIES

Suffix Array:



STEP 1: Indexing a reference genome:

- Indexing allows the aligner to quickly find potential alignment sites
- Indexing varies by aligner tool
- Indexing only needs to be run once

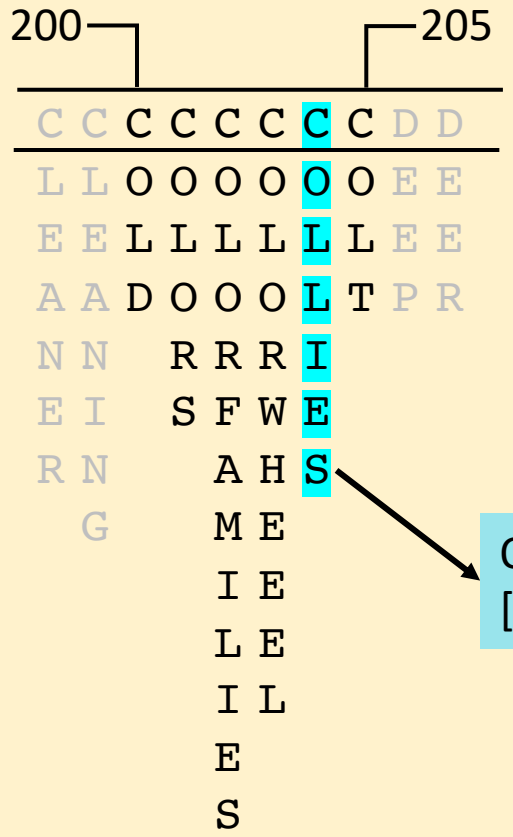


Hash Table:

k_i	[interval)
CLE	[190, 199)
COL	[200, 205)
DEE	[206, 210)

Query: COLLIES

Suffix Array:



Query Position:
[15,15; 21,21)

Salmon: fast and bias-aware quantification of transcript expression using dual-phase inference. Patro et al. 2017. Nat Methods

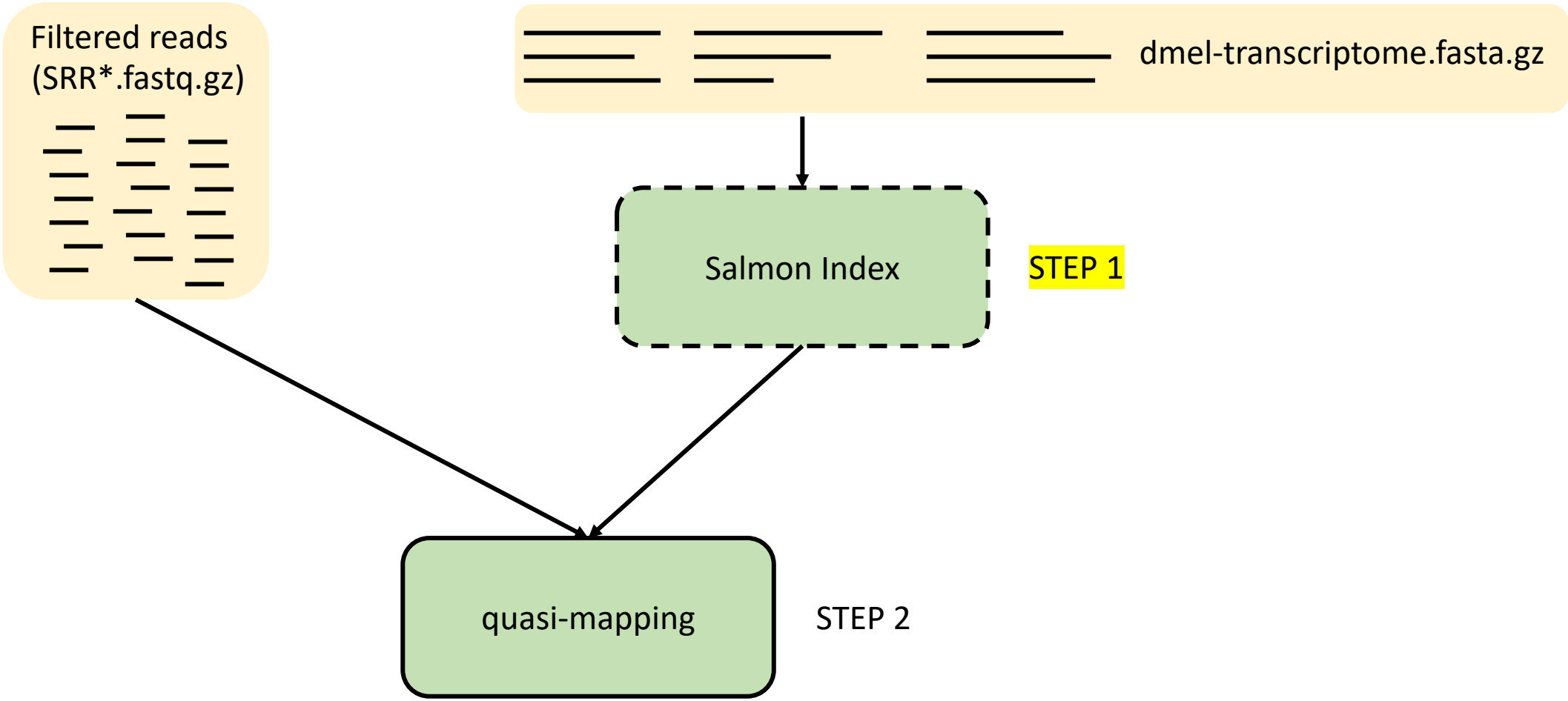


Image: RNA-Seq Blog

Salmon: fast and bias-aware quantification of transcript expression using dual-phase inference. Patro et al. 2017. Nat Methods

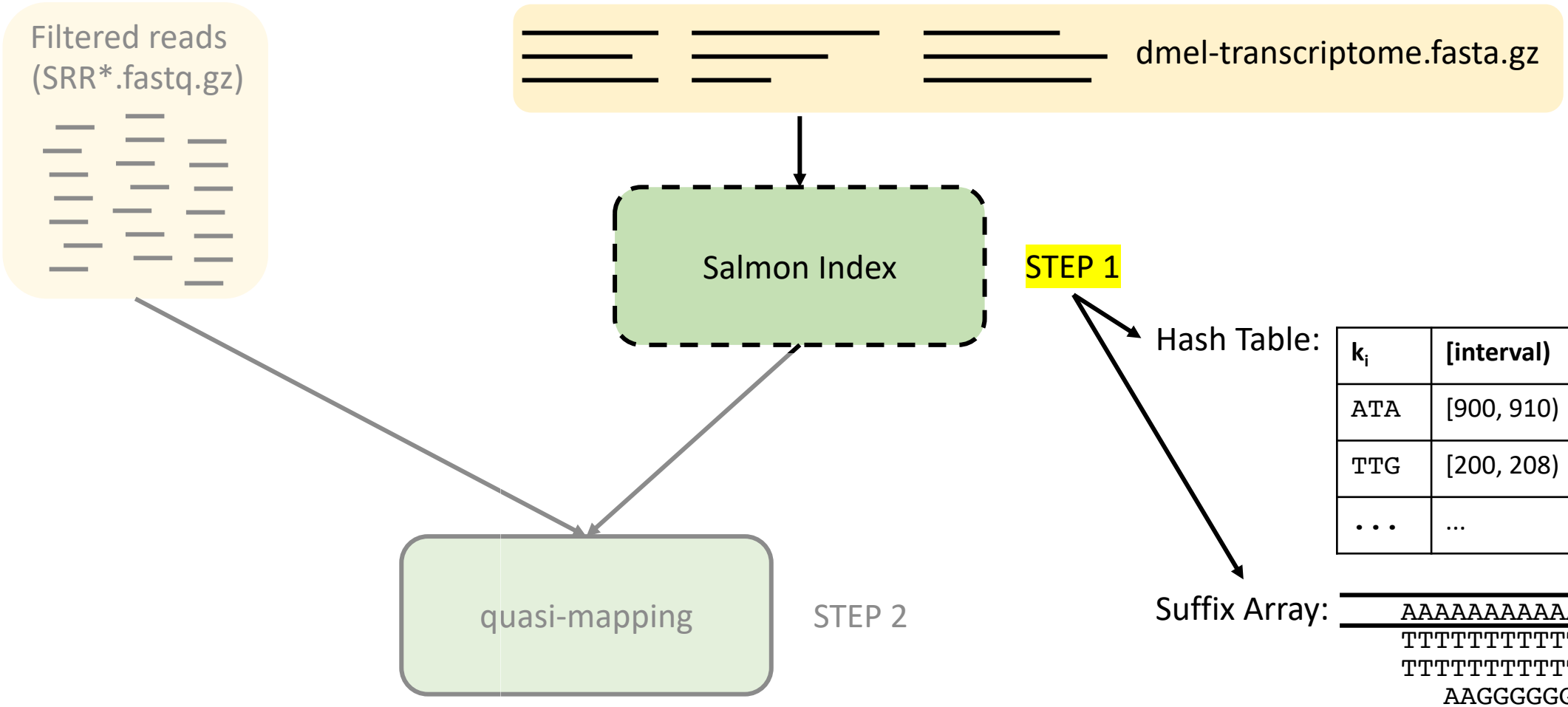


Image: RNA-Seq Blog

Index the transcriptome:

Very similar to other aligners we have used (BWA, Variant Calling Pipeline)

```
bwa index PATH/ref_genome/ref_genome.fasta
```

```
salmon index -t ${REF_GENOME} -i ${REFS_DIR}/dmel_idx
```

transcriptome

index

Output (directory)

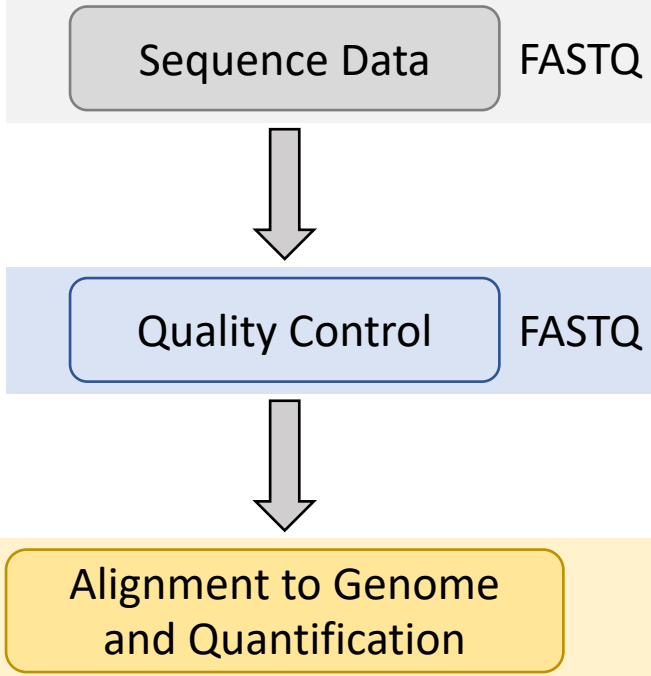
Add salmon indexing step to STEP2_RNAseq_Pipeline.sh

Index the transcriptome:

Check for a DIRECTORY (-f = file)

```
if [[ ! -d "${REFS_DIR}/dmel_idx" ]]; then
  echo "Indexing Reference Transcriptome"

  salmon index -t ${REF_GENOME} -i ${REFS_DIR}/dmel_idx || { echo "indexing failed" ; exit 1; }

else
  echo "Index found; skipping indexing step"
fi
```

Building our Pipeline:



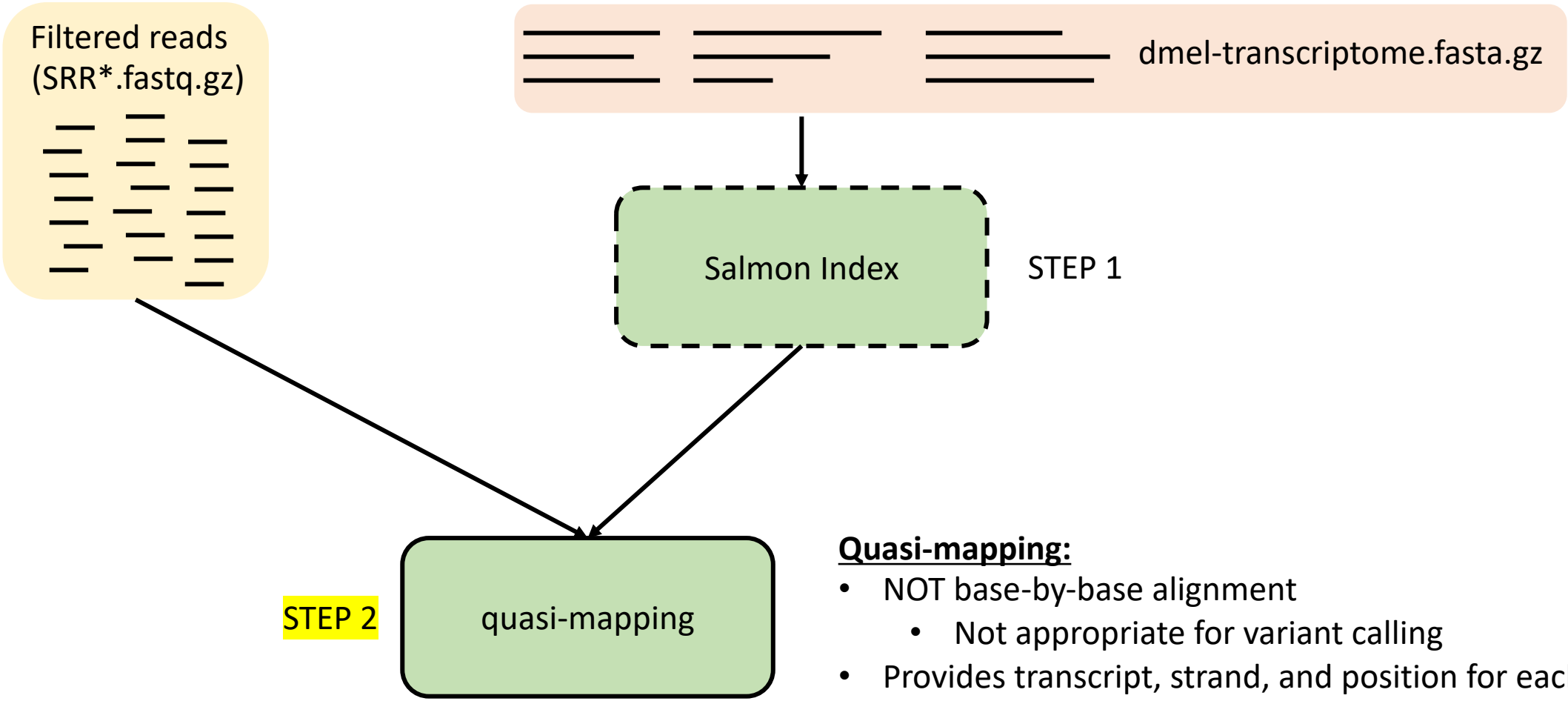

```
# Retrieve raw data and reference transcriptome

STEP2_RNAseq_Pipeline.sh
```

```
# TRIM and FILTER raw data

fastp...
multiqc...
```

```
# INDEX reference and ALIGN trimmed data

salmon index -t ${REF_GENOME} -i ${REFS_DIR}/dmel_idx
salmon quant...
```

fastp: https://github.com/OpenGene/fastp

Salmon: fast and bia-aware quantification of transcript expression using dual-phase inference. Patro et al. 2017. Nat Methods



Quasi-mapping:

- NOT base-by-base alignment
  - Not appropriate for variant calling
- Provides transcript, strand, and position for each read (enough info for counting transcripts)
- Searches by matching MMP to the index and then SKIPPING FORWARD in the read to the next MMP that has a match in the index

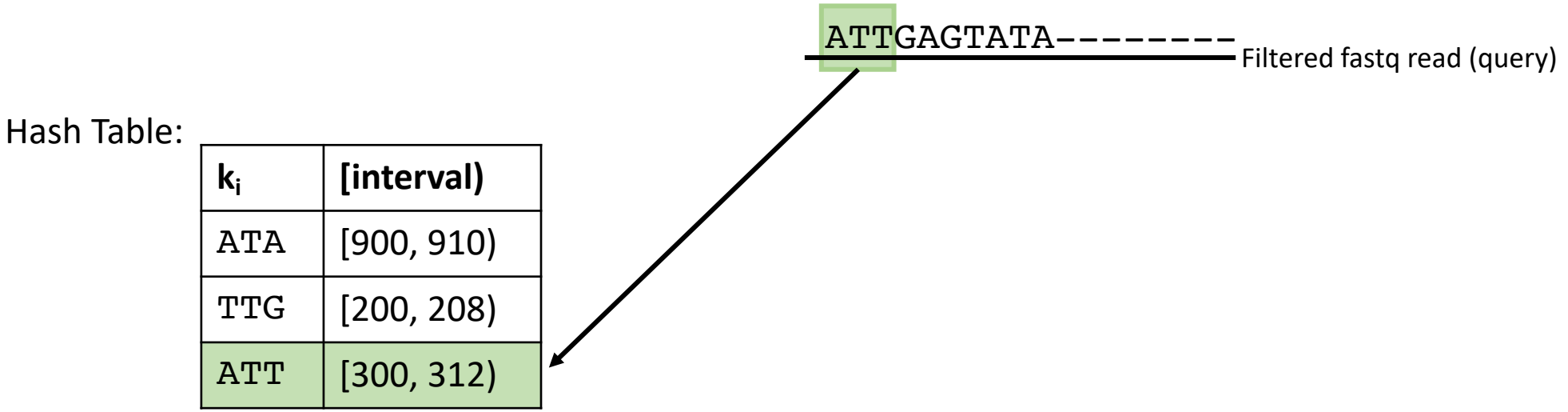
Image: RNA-Seq Blog

for loop, iterate through all filtered samples

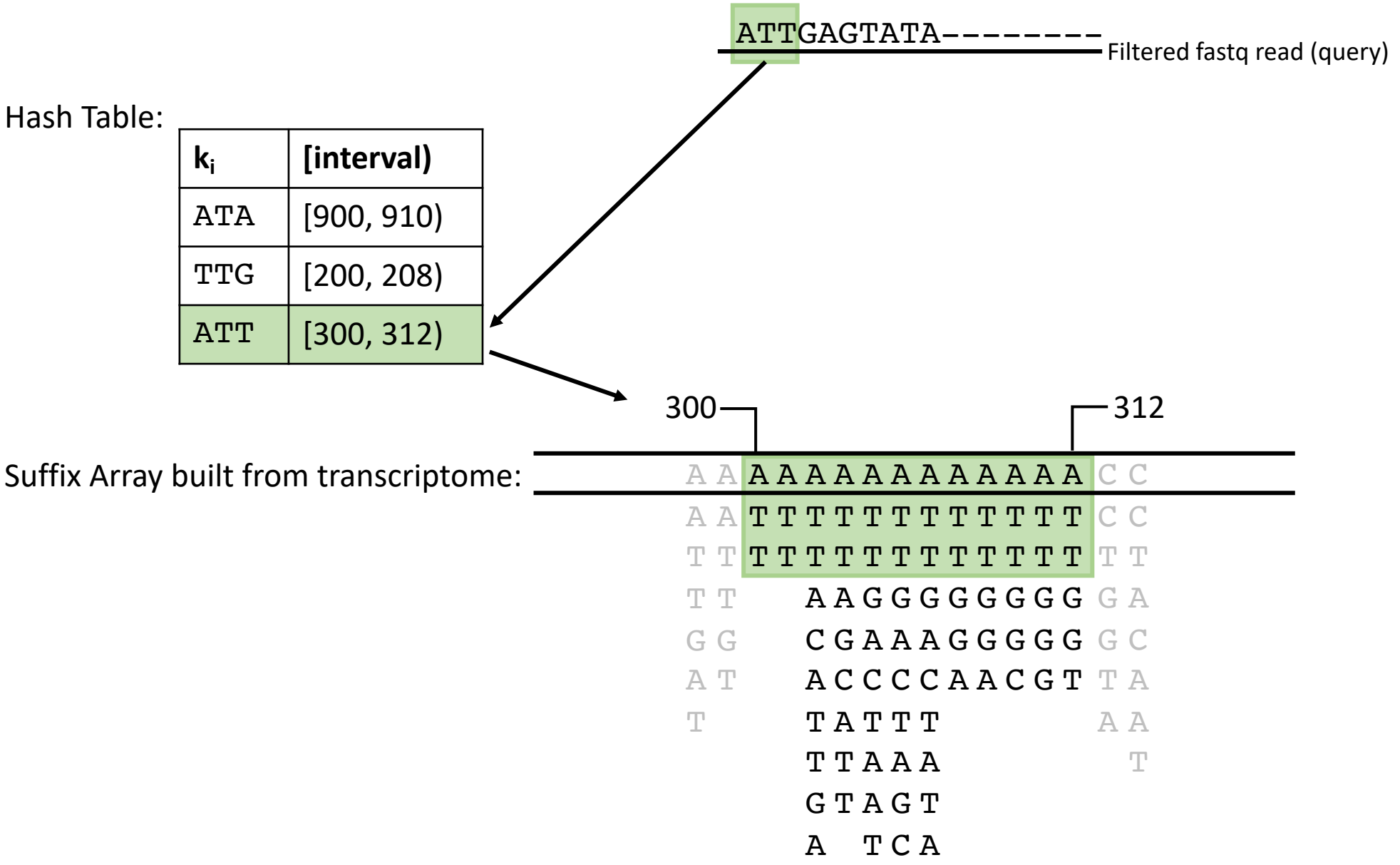
Assess whether alignment has already been completed

Code Chunk

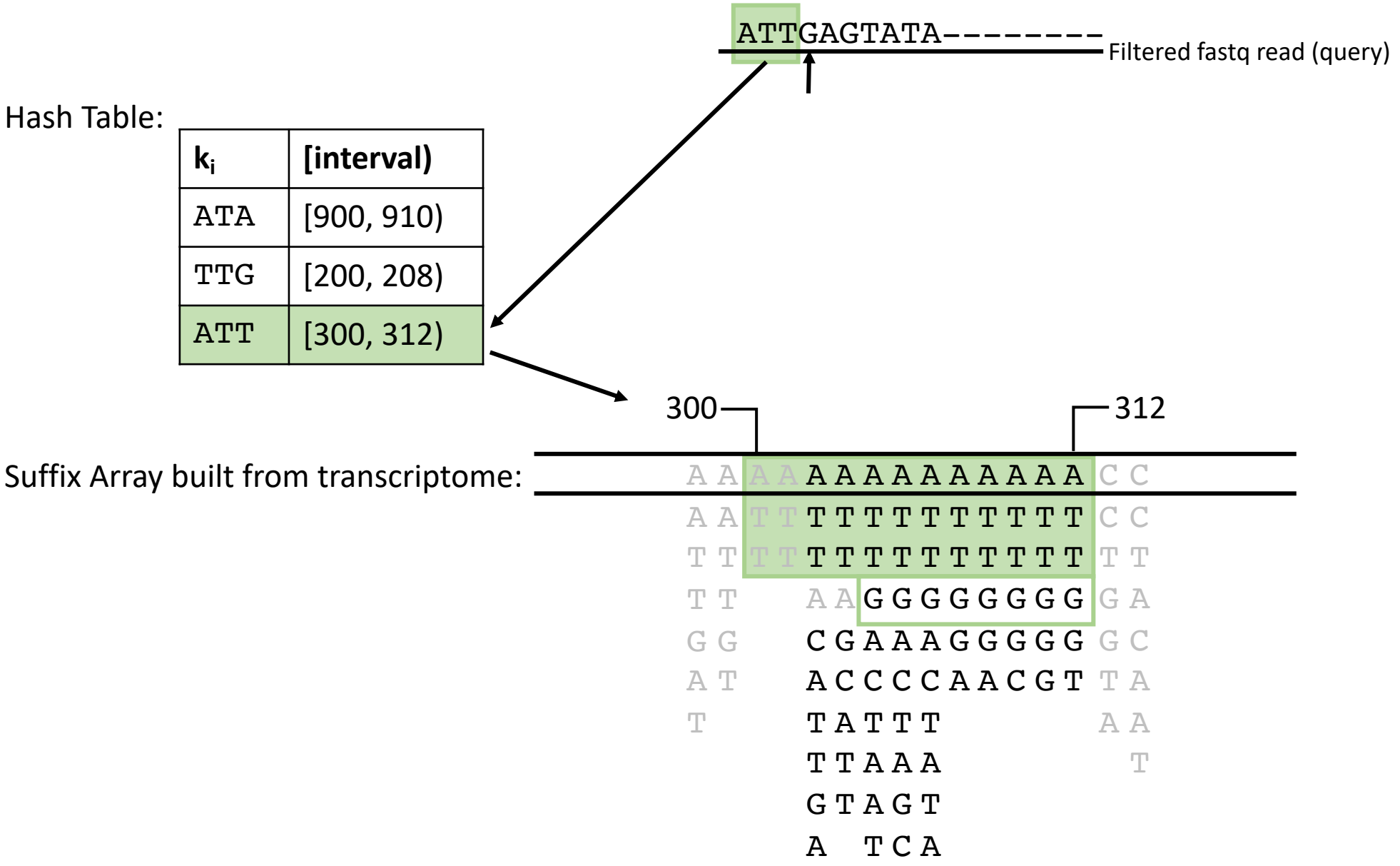
```
for FQ1 in ${DATA_FILT}/*.1.filt.fastq.gz
  do

    sample=$(basename ${FQ1} .1.filt.fastq.gz)

    if [[ ! -d "${SALMON_DIR}/${sample}_quant" ]]; then
     echo "Quantifying sample ${sample}"

     salmon quant -i ${INDEX} \
                  -l A \ # automatically determine library type
                  -1 ${DATA_FILT}/${sample}.1.filt.fastq.gz \
                  -2 ${DATA_FILT}/${sample}.2.filt.fastq.gz \
                  -p 8 \
                  --validateMappings \ # uses selective alignment
                  -o ${SALMON_DIR}/${sample}_quant

     else
      echo "Quantified file found; skipping quant step"
     fi

  done
```



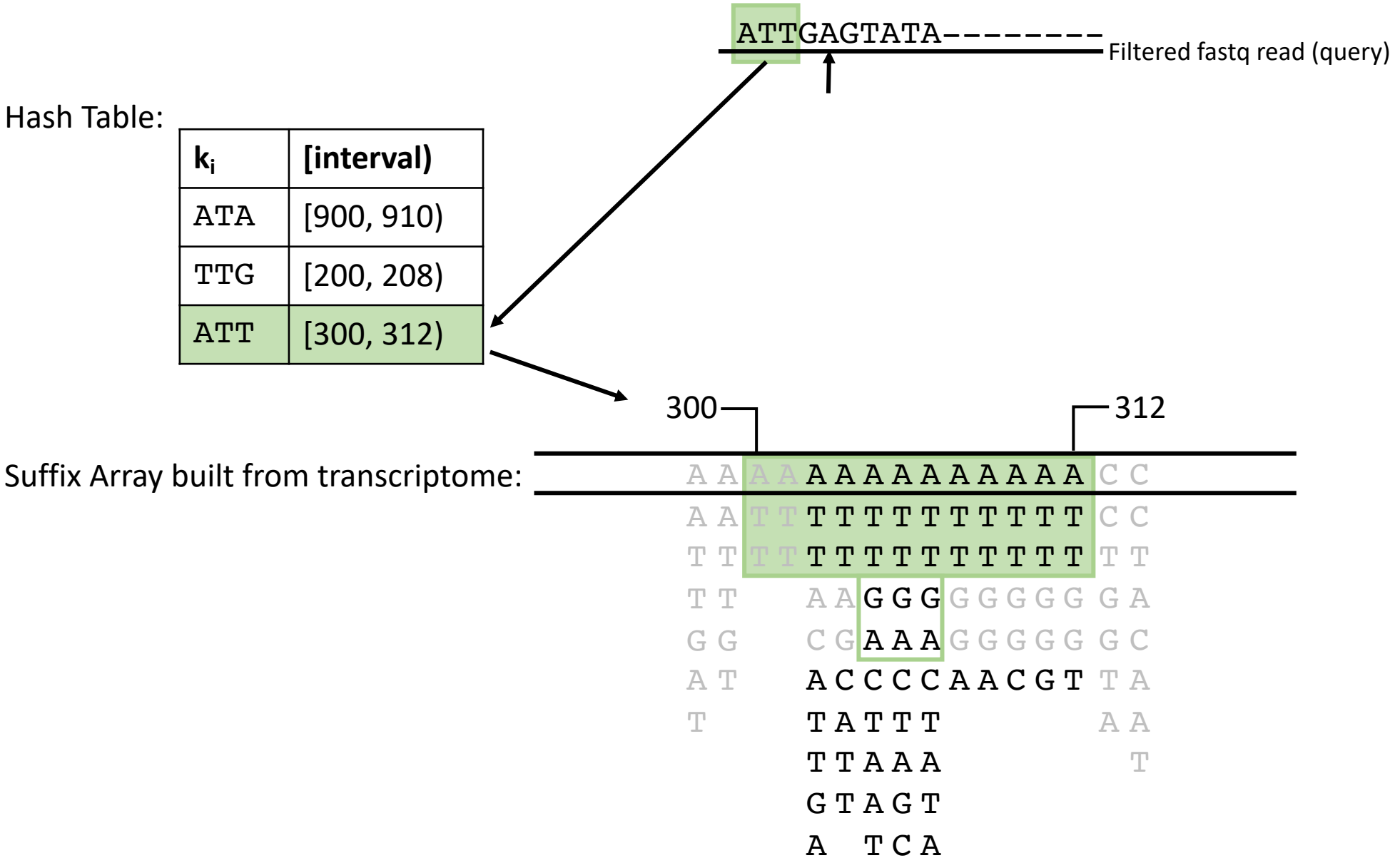
Adapted from Srivastrava et al. 2016



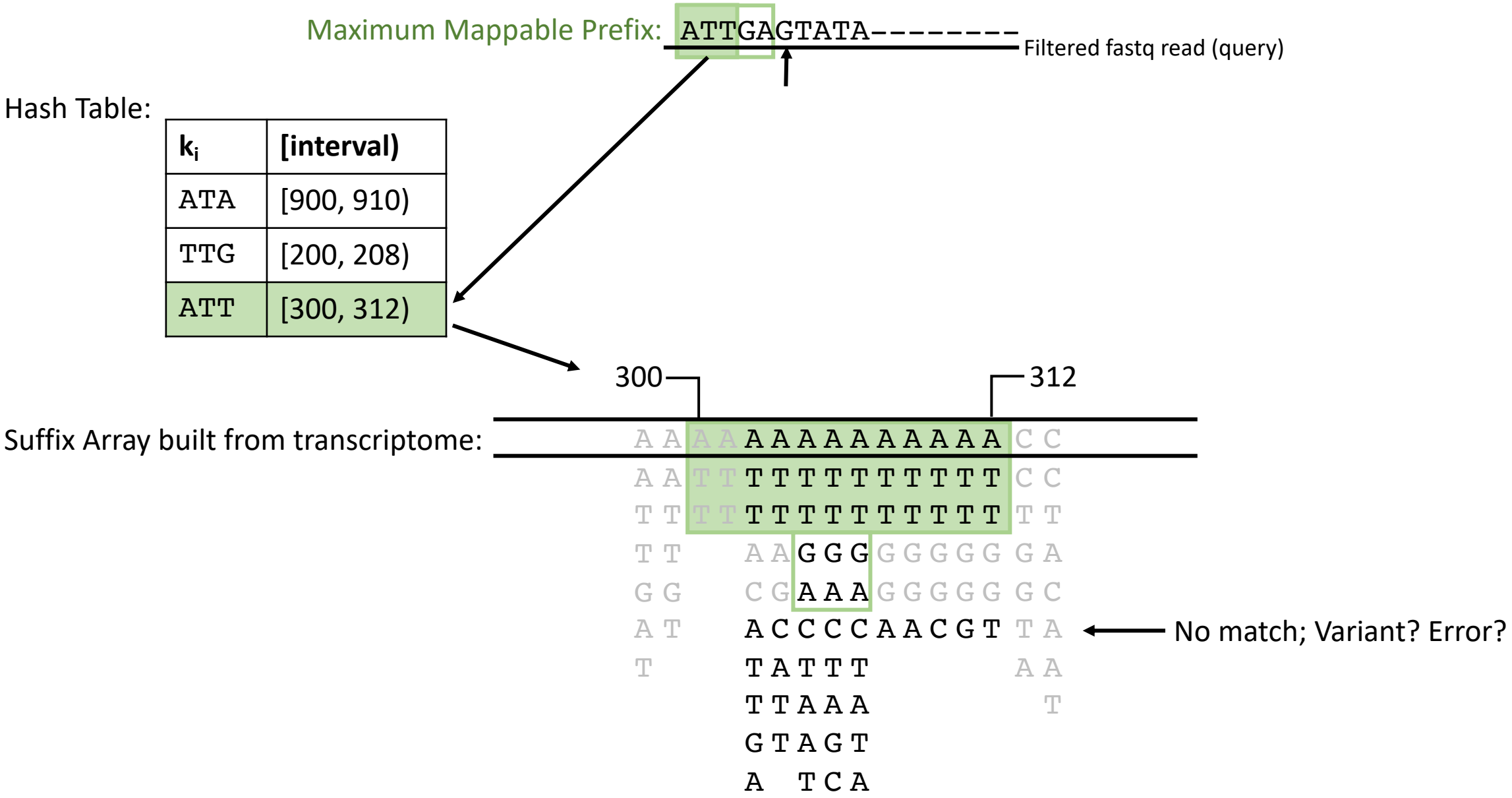
Adapted from Srivastrava et al. 2016



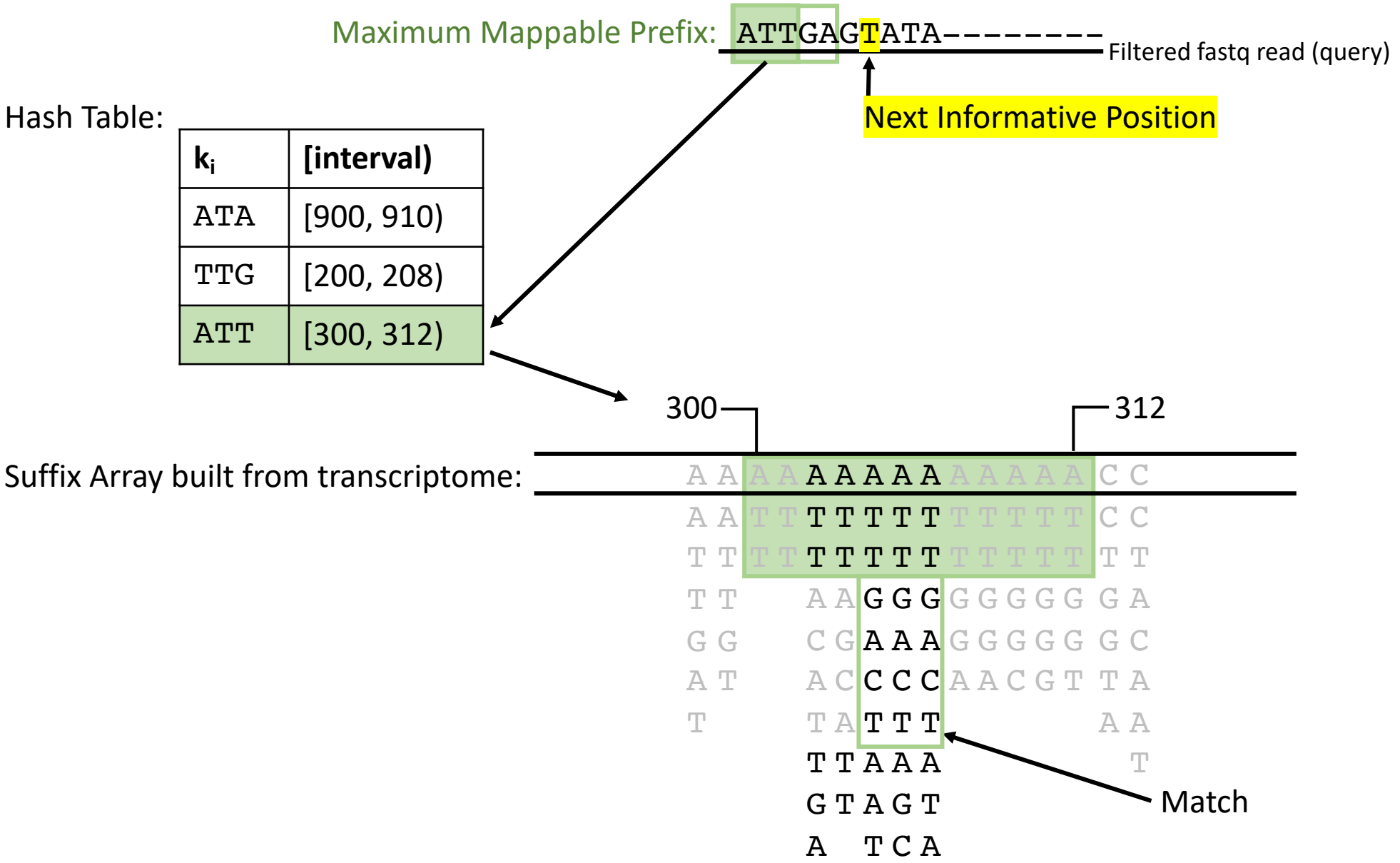
Adapted from Srivastrava et al. 2016



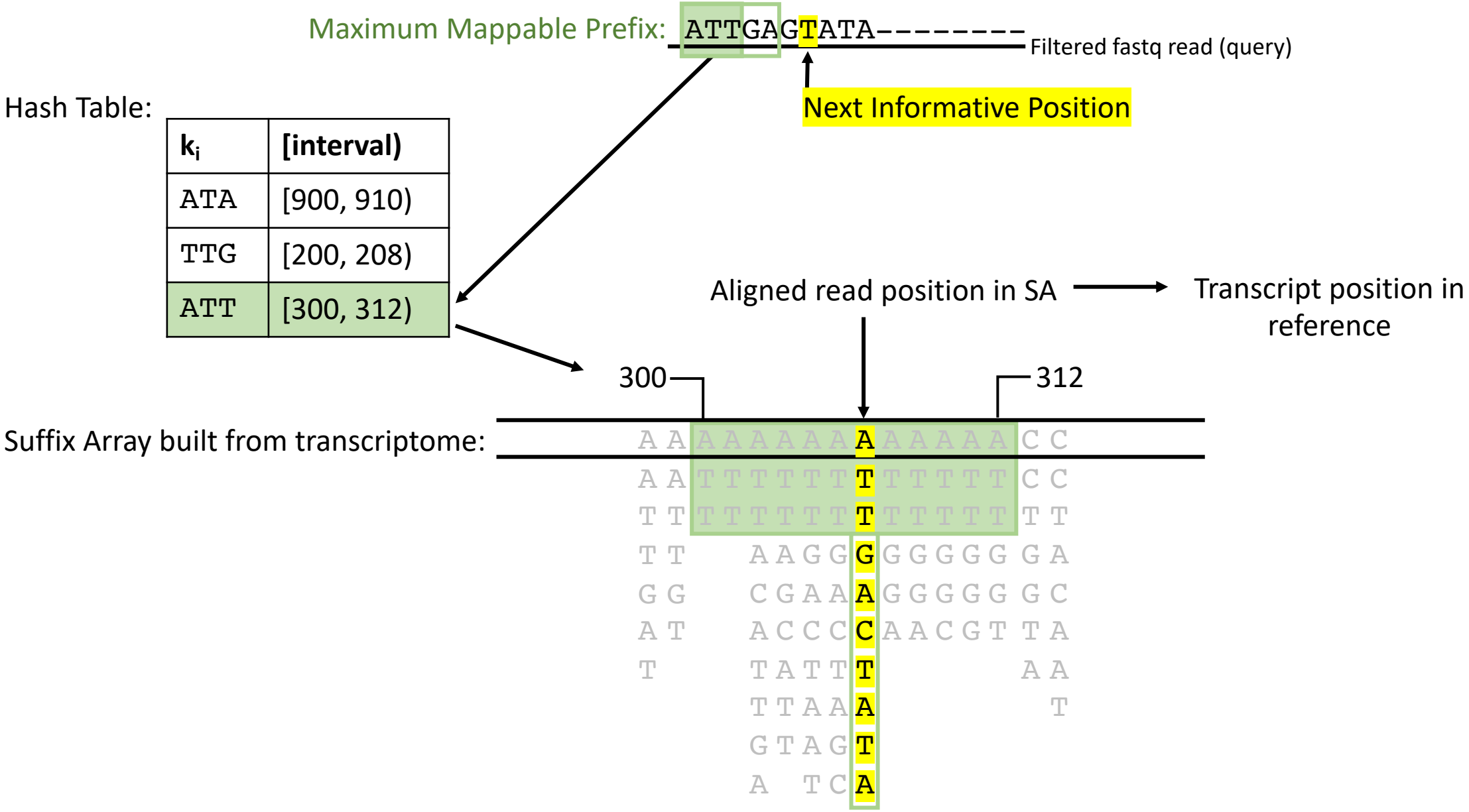
Adapted from Srivastrava et al. 2016



Adapted from Srivastrava et al. 2016

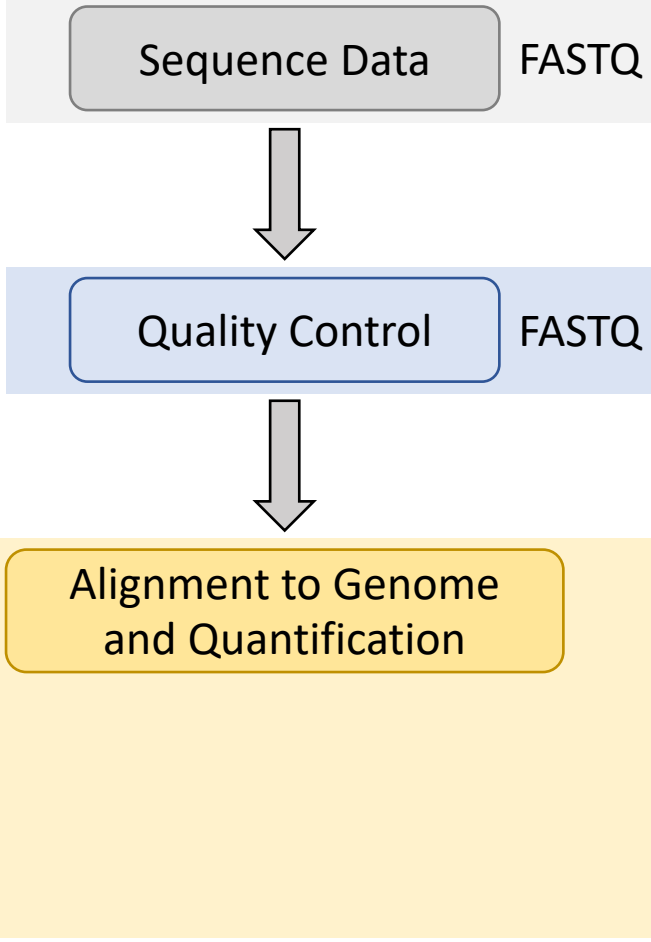


Adapted from Srivastrava et al. 2016



Adapted from Srivastrava et al. 2016

Building our Pipeline:

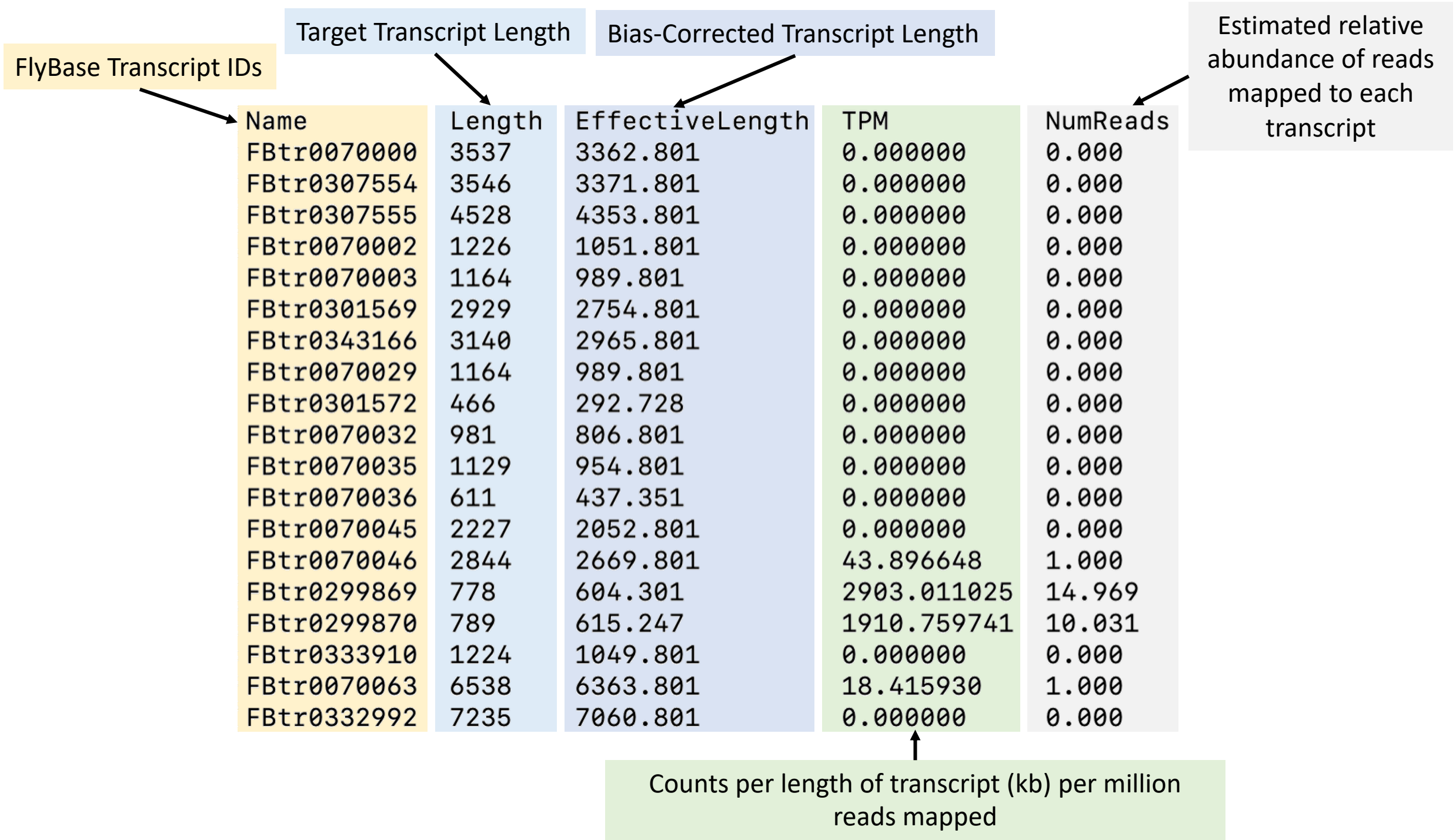



```
# Retrieve raw data and reference transcriptome

STEP2_RNAseq_Pipeline.sh
```

```
# STEP 3: TRIM and FILTER raw data

fastp...
multiqc...
```

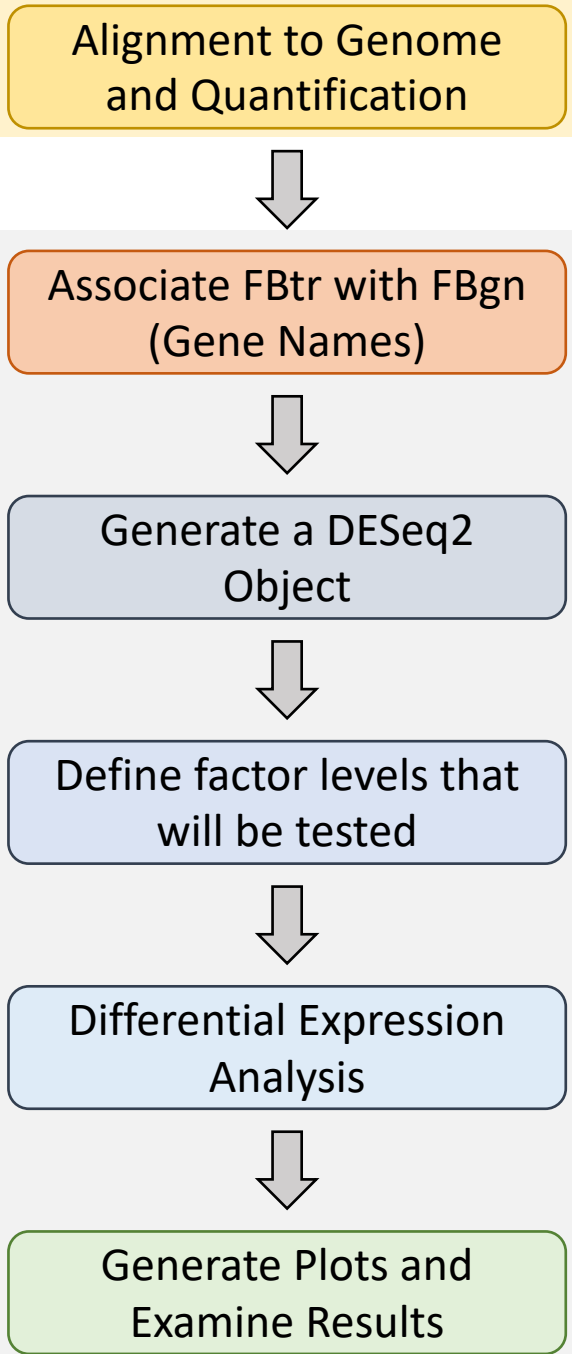
```
# STEP 3 con't: INDEX reference and ALIGN trimmed data

salmon index -t ${REF_GENOME} -i ${REFS_DIR}/dmel_idx
salmon quant -i ${INDEX} \
                    -l A \
                    -1 ${DATA_FILT}/${sample}.1.filt.fastq.gz \
                    -2 ${DATA_FILT}/${sample}.2.filt.fastq.gz \
                    -p 8 \
                    --validateMappings \
                    -o ${SALMON_DIR}/${sample}_quant
```

fastp: https://github.com/OpenGene/fastp



Name	Length	EffectiveLength	TPM	NumReads
FBtr0070000	3537	3362.801	0.000000	0.000
FBtr0307554	3546	3371.801	0.000000	0.000
FBtr0307555	4528	4353.801	0.000000	0.000
FBtr0070002	1226	1051.801	0.000000	0.000
FBtr0070003	1164	989.801	0.000000	0.000
FBtr0301569	2929	2754.801	0.000000	0.000
FBtr0343166	3140	2965.801	0.000000	0.000
FBtr0070029	1164	989.801	0.000000	0.000
FBtr0301572	466	292.728	0.000000	0.000
FBtr0070032	981	806.801	0.000000	0.000
FBtr0070035	1129	954.801	0.000000	0.000
FBtr0070036	611	437.351	0.000000	0.000
FBtr0070045	2227	2052.801	0.000000	0.000
FBtr0070046	2844	2669.801	43.896648	1.000
FBtr0299869	778	604.301	2903.011025	14.969
FBtr0299870	789	615.247	1910.759741	10.031
FBtr0333910	1224	1049.801	0.000000	0.000
FBtr0070063	6538	6363.801	18.415930	1.000
FBtr0332992	7235	7060.801	0.000000	0.000

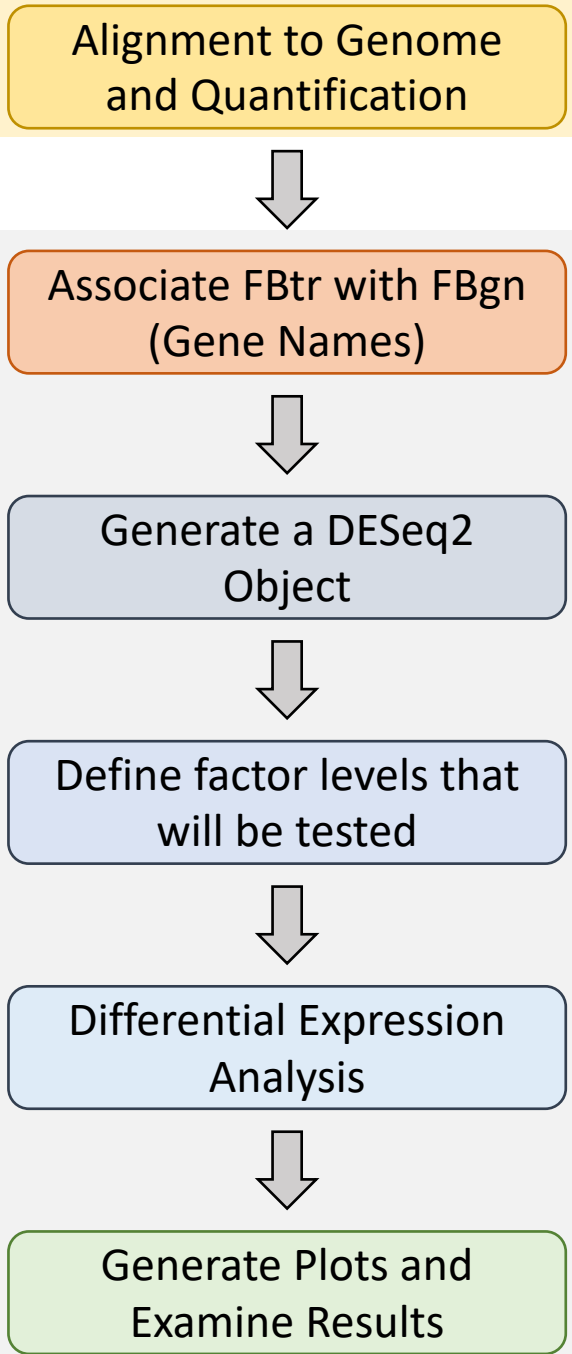
Differential Expression Workflow:




```
sh ${SCRIPTS_DIR}/STEP2_RNAseq_Pipeline.sh
```

```
RNAseq_Analysis.R
```

Differential Expression Workflow:




```
sh ${SCRIPTS_DIR}/STEP2_RNAseq_Pipeline.sh
```

```
RNAseq_Analysis.R
```

Next Steps and Prep for Next Class:

We will use DESeq to normalize the raw count data and run differential expression analysis. Look up the following functions and write code that creates a DESeq object and generates normalized counts:

- DESeqDatasetFromTximport()
- estimateSizeFactors()
- counts()