RNA seq: differential expression analysis

For INF-BIO 4121/9121
Fall semester 2016

Rebekah Oomen / Monica Hongrø Solbakken
r.a.oomen@ibv.uio.no / m.h.solbakken@ibv.uio.no

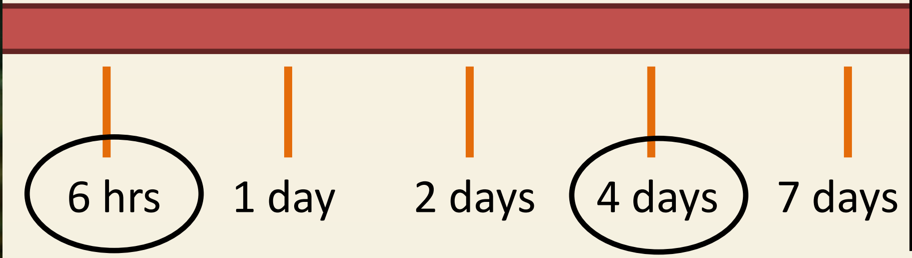
UiO : Centre for Ecological and Evolutionary Synthesis
University of Oslo

The INFBIO case

- Non-model organism – Atlantic cod
- Reference genome available
- A treatment to investigate immune responses
- Simple treatment-control setup over time

The INFBIO case

An infection over time



6 hrs

1 day

2 days

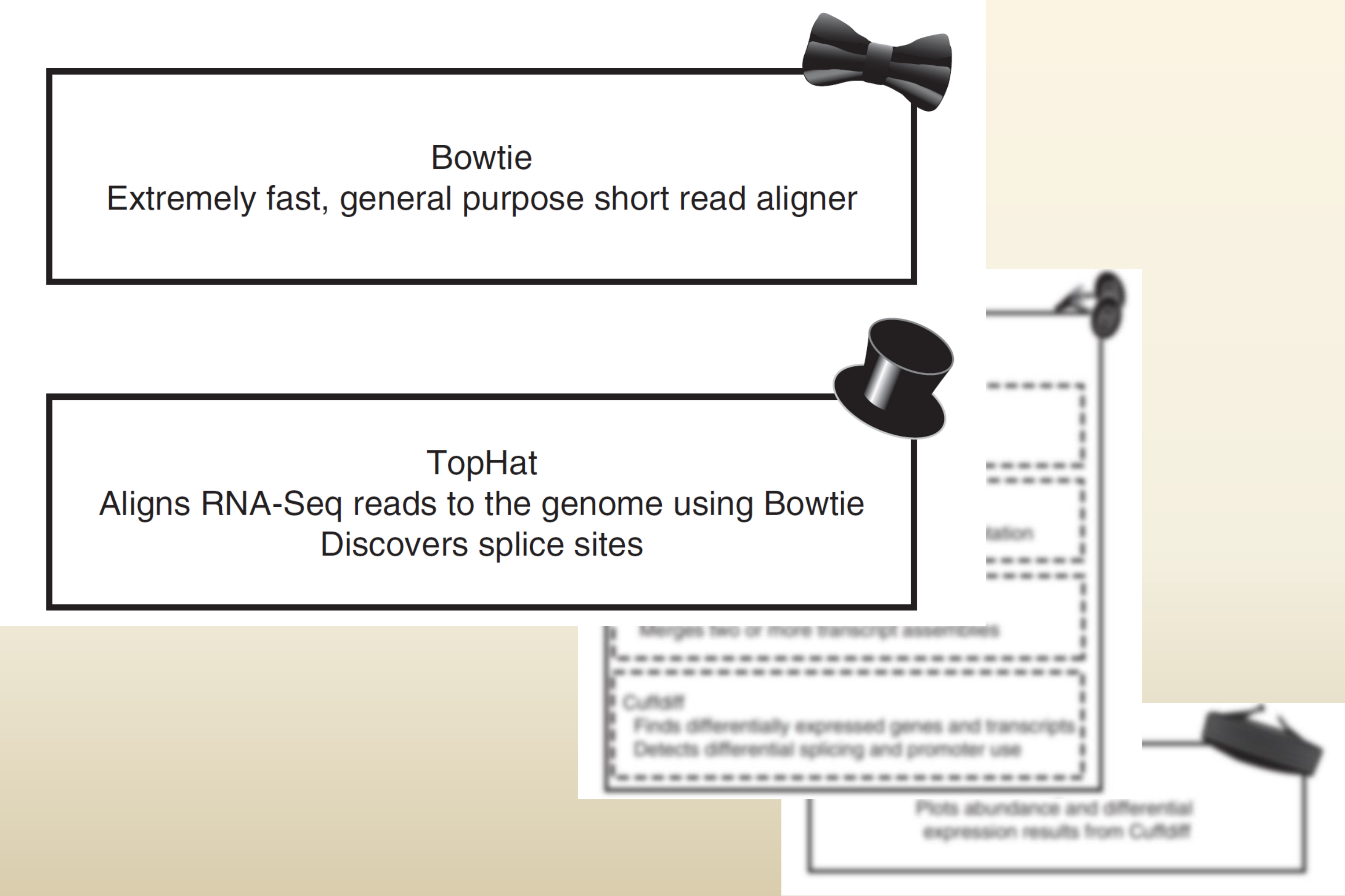
4 days

7 days

Two time-points, 6 treated and 6 controls

In total 12 samples

The TUXEDO pipeline



Bowtie
Extremely fast, general purpose short read aligner

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

lation

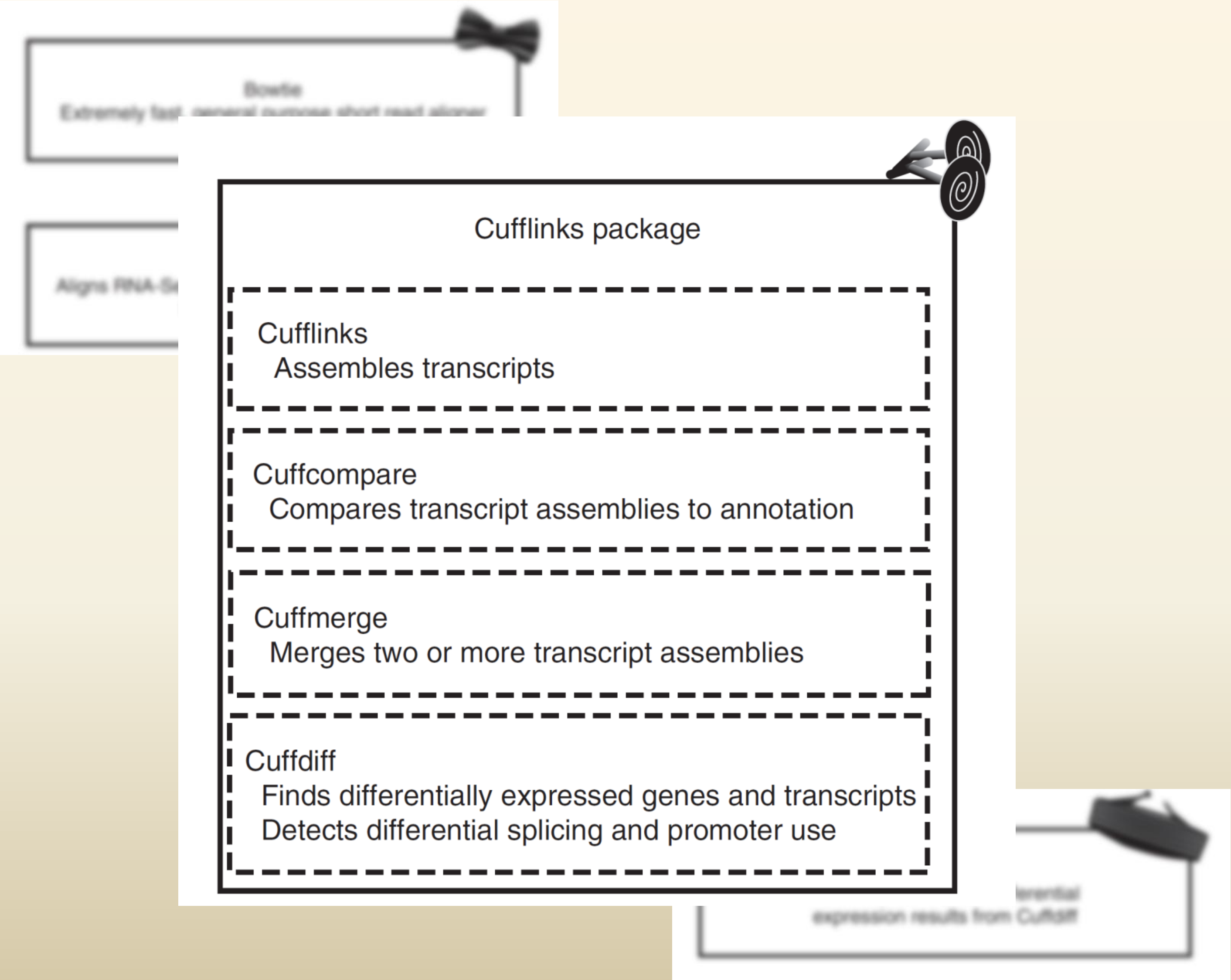
Merges two or more transcript assemblies

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

Plots abundance and differential
expression results from Cuffdiff

http://www.nature.com/nprot/journal/v7/n3/pdf/nprot.2012.016.pdf

The TUXEDO pipeline



Bowtie
Extremely fast, general purpose short read aligner

Aligns RNA-Se

Cufflinks package

Cufflinks
Assembles transcripts

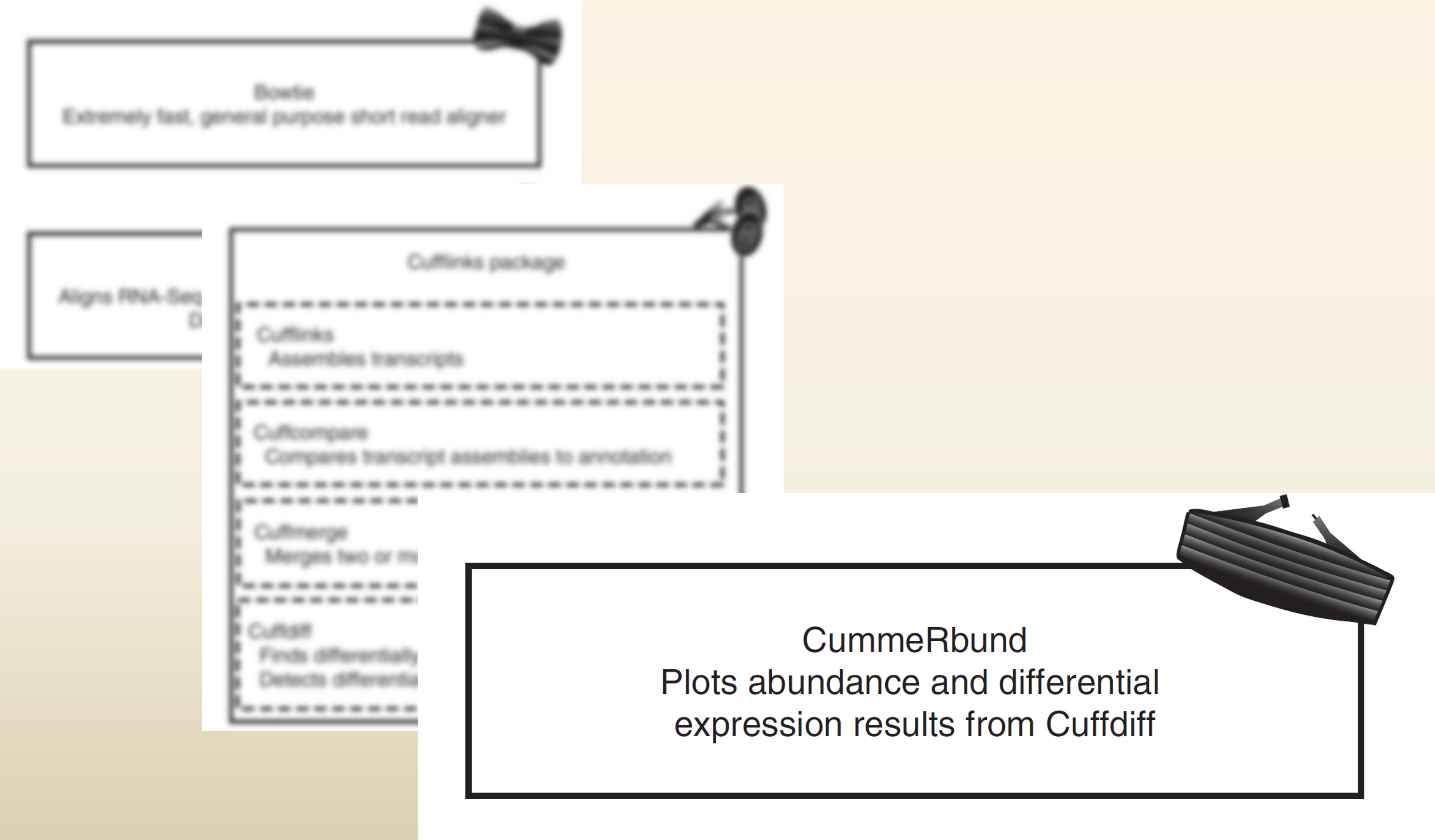
Cuffcompare
Compares transcript assemblies to annotation

Cuffmerge
Merges two or more transcript assemblies

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

ferential
expression results from Cuffdiff

The TUXEDO pipeline



Bowtie
Extremely fast, general purpose short read aligner

Aligns RNA-Seq
D

Cufflinks package

Cufflinks
Assembles transcripts

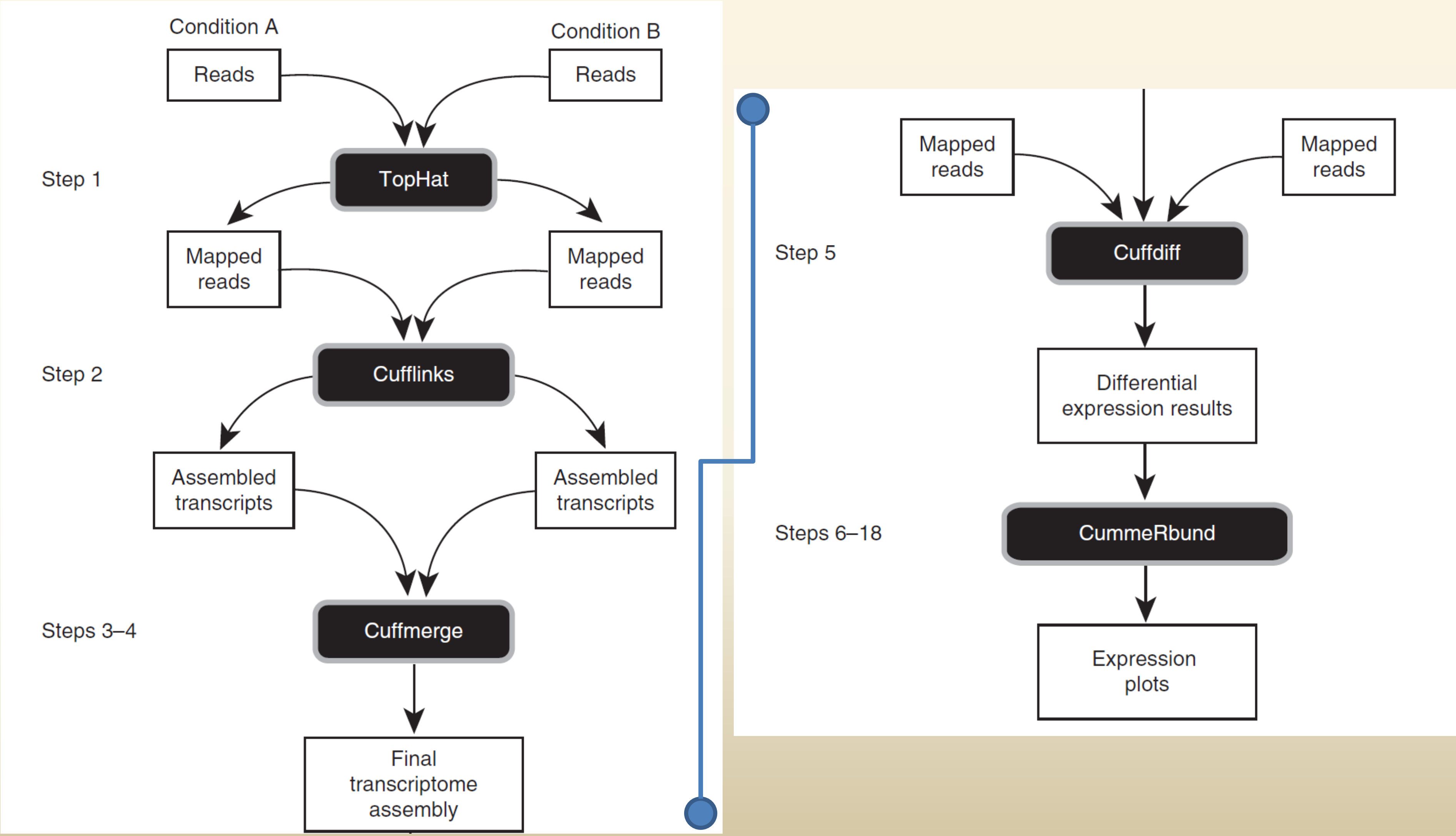
Cuffcompare
Compares transcript assemblies to annotation

Cuffmerge
Merges two or m

Cuffdiff
Finds differentially
Detects differentia

CummeRbund
Plots abundance and differential
expression results from Cuffdiff

The TUXEDO pipeline



Condition A

Condition B

Reads

Reads

Step 1

TopHat

Mapped reads

Mapped reads

Step 2

Cufflinks

Assembled transcripts

Assembled transcripts

Steps 3–4

Cuffmerge

Final transcriptome assembly

Mapped reads

Mapped reads

Step 5

Cuffdiff

Differential expression results

Steps 6–18

CummeRbund

Expression plots

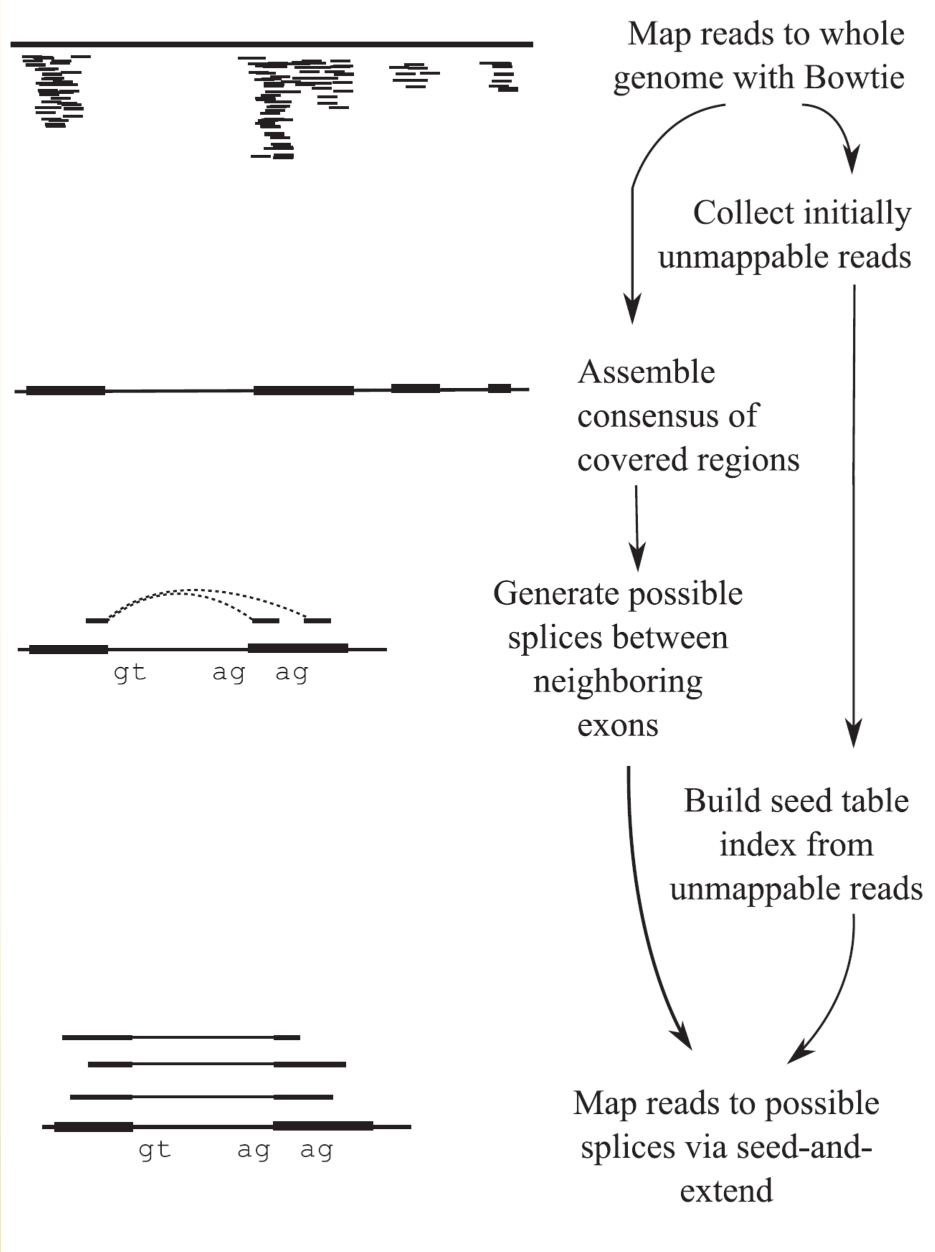
Input material for Tuxedo

- Raw reads
  - May be trimmed if f.ex mapping % is low
- A fairly good reference genome

Step 1 – mapping with Tophat

- Built upon the short read mapper Bowtie
  - – Burrows-Wheeler indexing
- Tophat identifies possible splice junctions in the Bowtie alignment
- New mapping to these splice juntions
- Thus, no annotation of reference needed

The details of Tophat



Map reads to whole genome with Bowtie

Collect initially unmappable reads

Assemble consensus of covered regions

Generate possible splices between neighboring exons

Build seed table index from unmappable reads

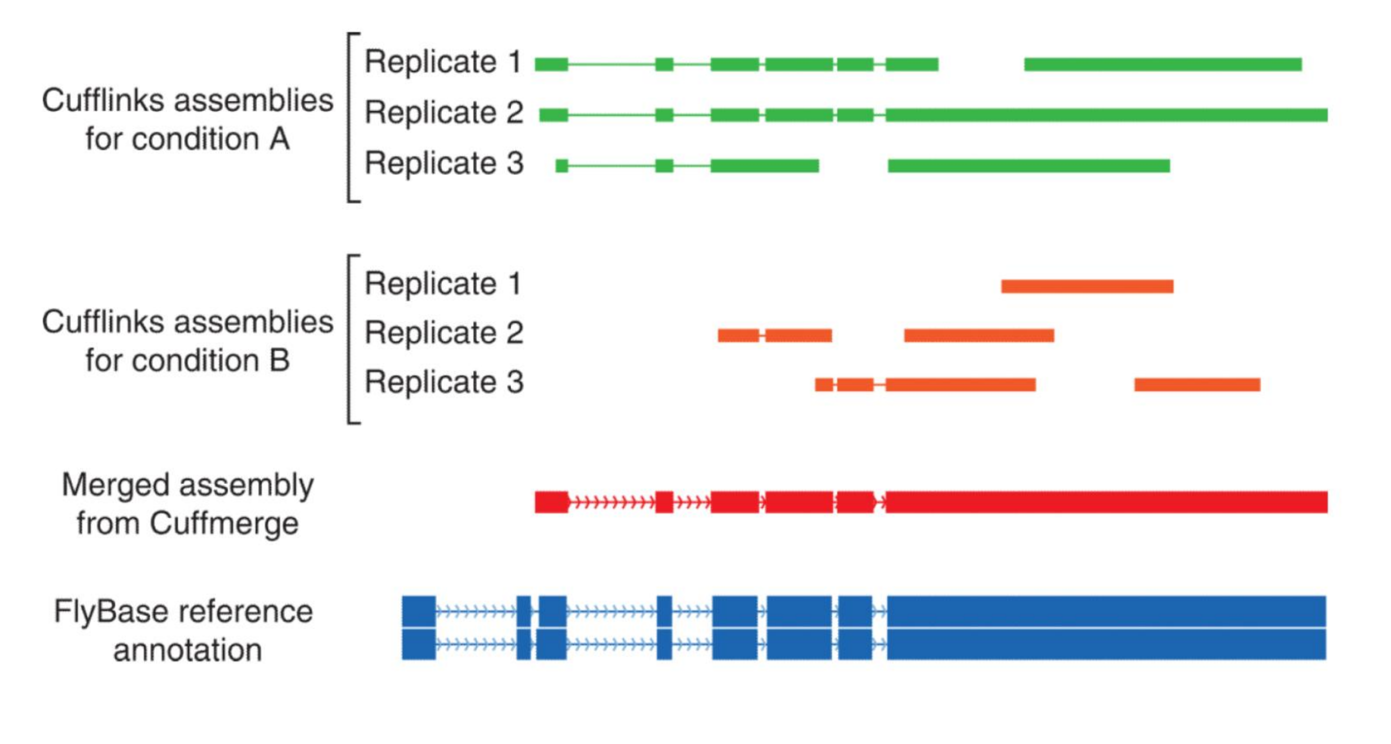
Map reads to possible splices via seed-and-extend

Cufflinks

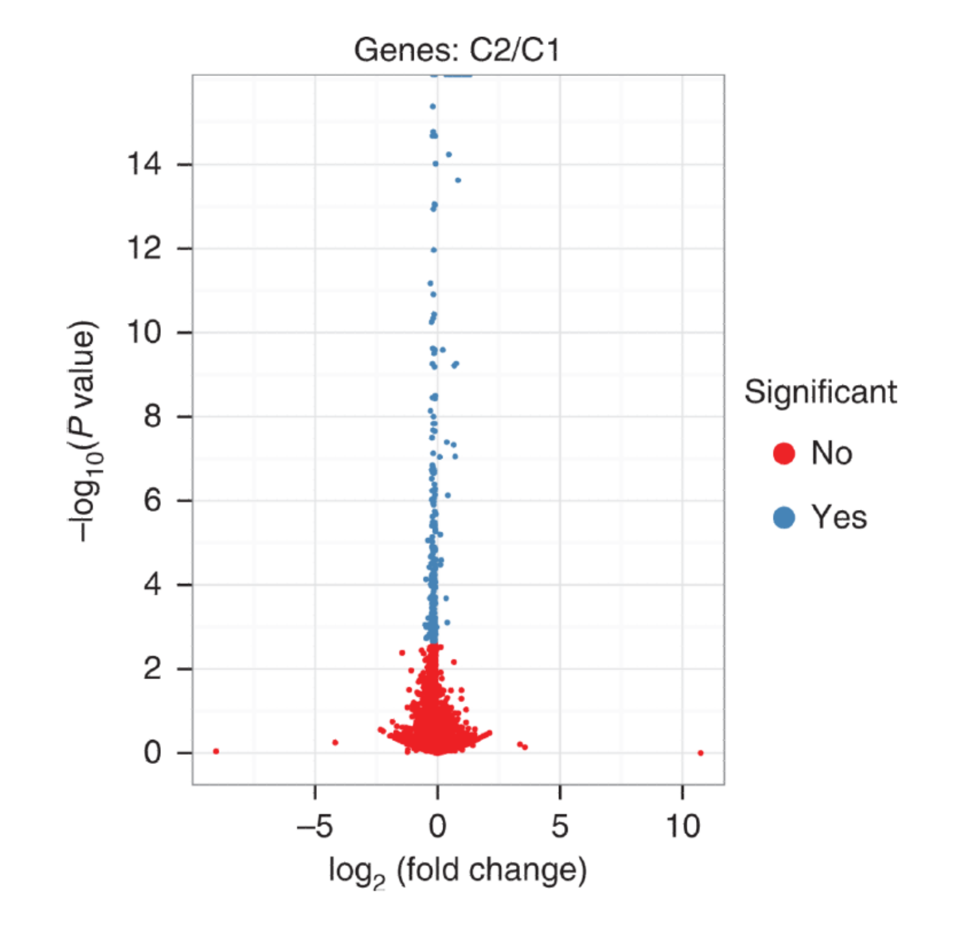
- Transcript assembly
  - A parsimonious strategy to resolve isoforms
- First level transcript quantification
  - Immature vs mature transcripts

Cuffmerge

- Pooling of cufflinks data per sample to ensure proper overall experiment ”present transcripts” overview

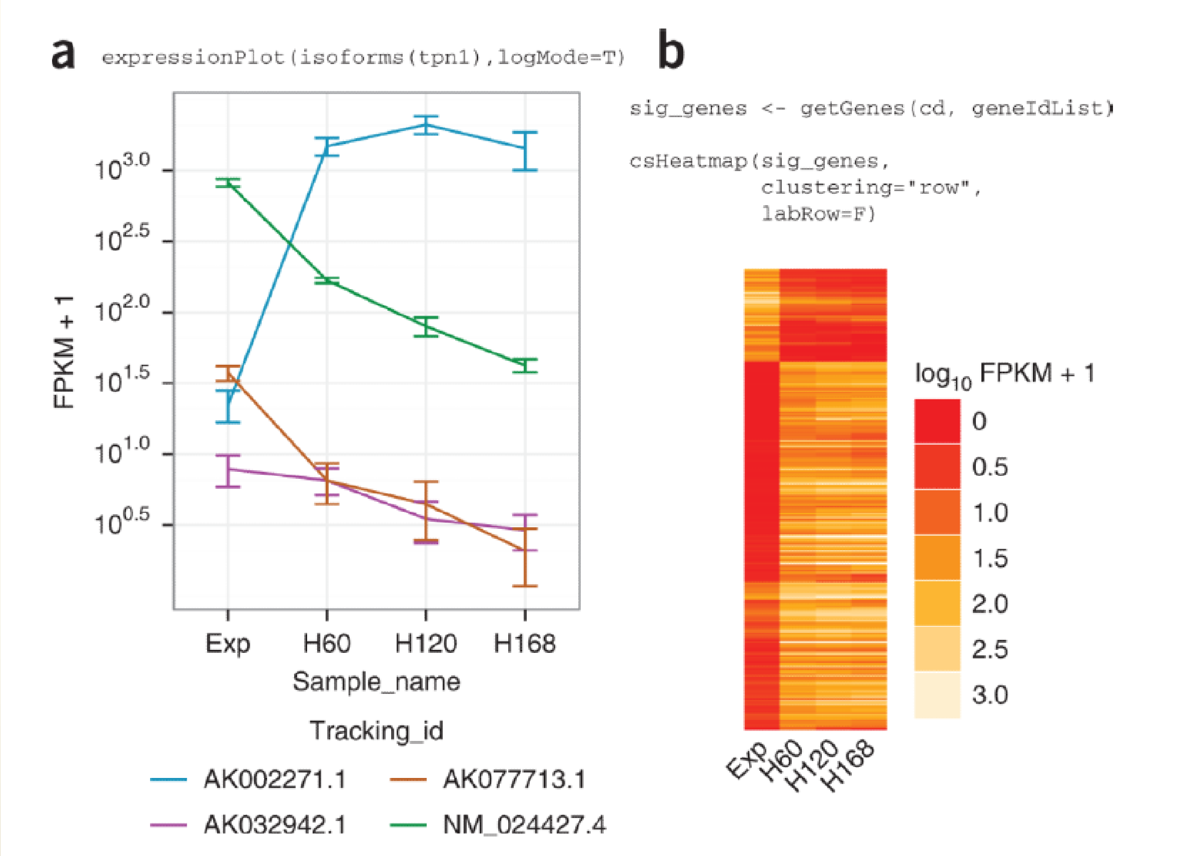

Cufflinks assemblies for condition A
Replicate 1
Replicate 2
Replicate 3
Cufflinks assemblies for condition B
Replicate 1
Replicate 2
Replicate 3
Merged assembly from Cuffmerge
FlyBase reference annotation

Cuffdiff



Cuffdiff ”learns the variation for each gene across replicates” to calculate differential expression

cummeRbund



```
expressionPlot(isoforms(tpn1),logMode=T)
```

```
sig_genes <- getGenes(cd, geneIdList)

csHeatmap(sig_genes,
          clustering="row",
          labRow=F)
```

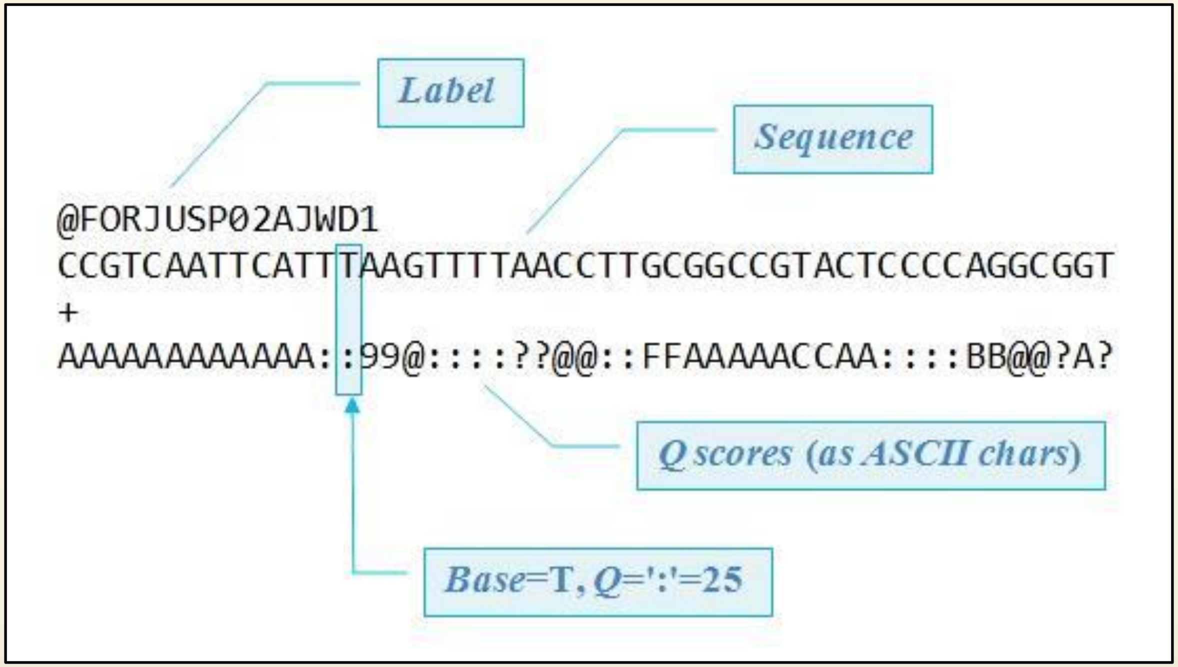
All about the visuals…

Sequence evaluation / trimming and their effects on transcriptome analyses

Sequencing facility provides:

- Your sequence data in fastq format
- Usually a sequencing run together with an overall data evaluation
- If the sequencing facility has performed library prep you can ask for library quality checks
- Some guarantee certain amounts of reads from a lane/flow cell/SMRTcell

The fastq format



```
@FORJUSP02AJWD1
CCGTCAATTCATTTAAGTTTTAACCTTGCGGCCGTACTCCCCAGGCGGT
+
AAAAAAAAAAAA::99@::::??@@::FFAAAAACCAA:::BB@@?A?
```

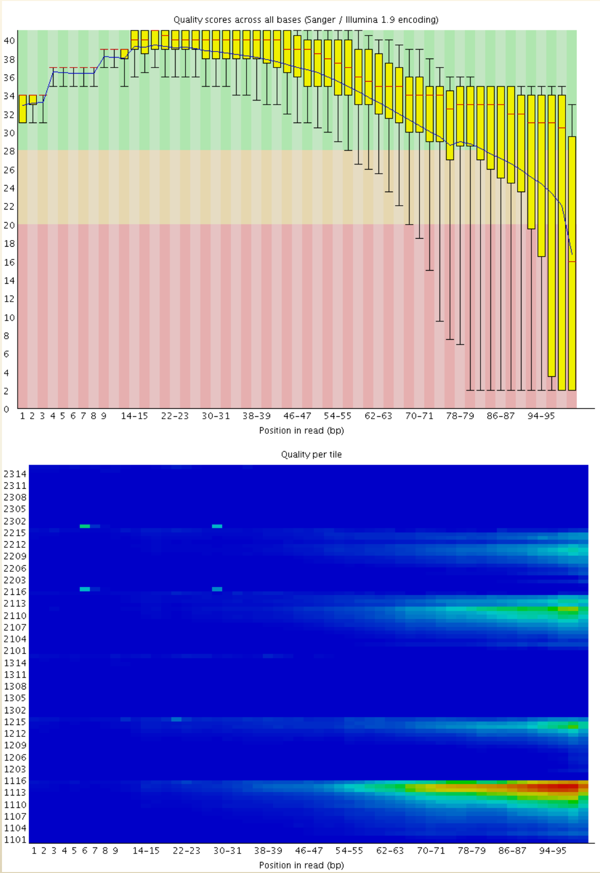
- Beware! Different sequencing technologies uses different quality encodings - ex : might correspond to different qualities

Raw sequence evaluation

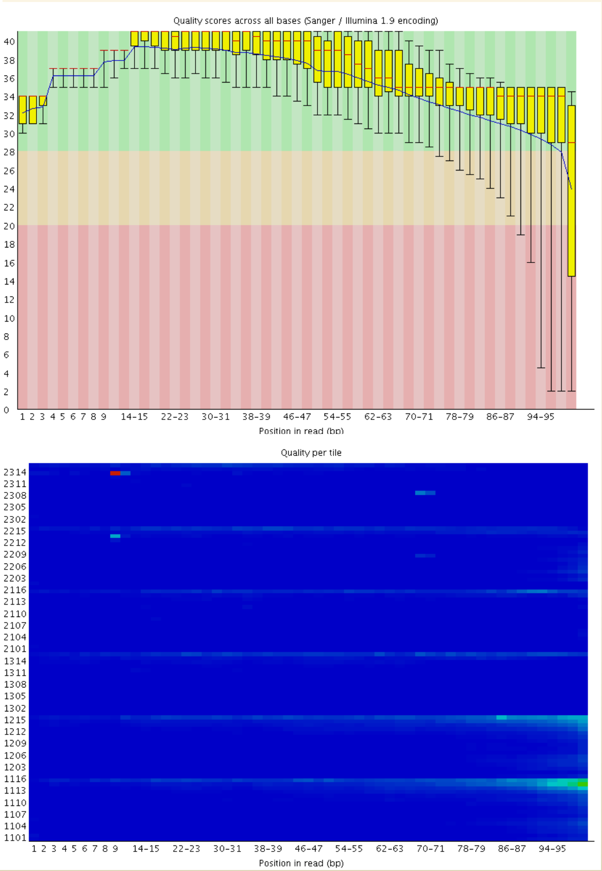
- Look for:
  - – Low sequence output from a certain lane(s)
  - – Several consecutive cycles with lower quality
  - – Very poor read 2 – danger of loosing pairing
  - – Very biased kmer profile and sequence content
- Some sources of bias are
  - – Instrument error
  - – Poor starting material
  - – Over-amplification of library

Check the fastqc reports I

Read 1



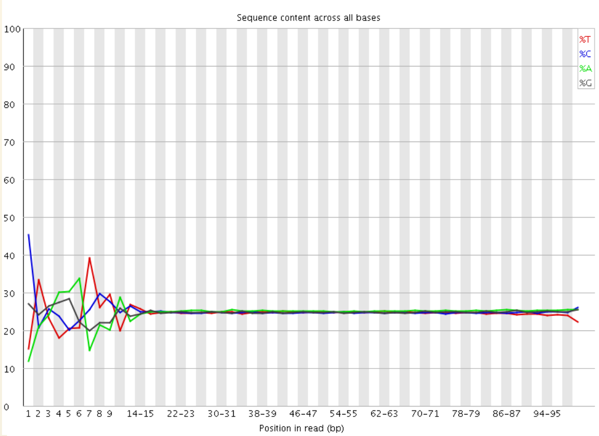
Read 2

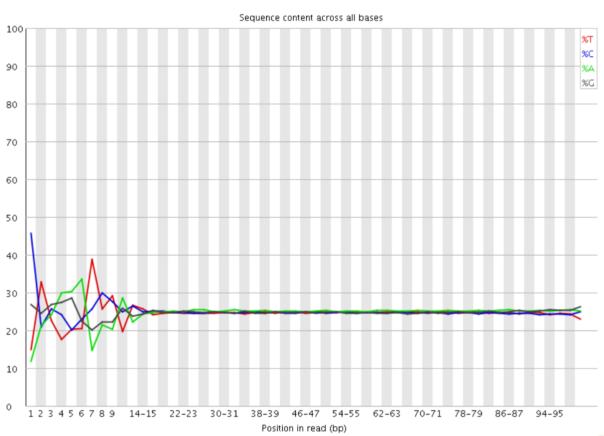


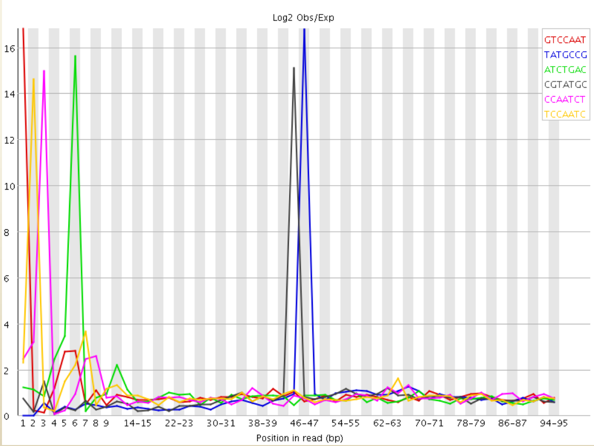
Check the fastqc reports II

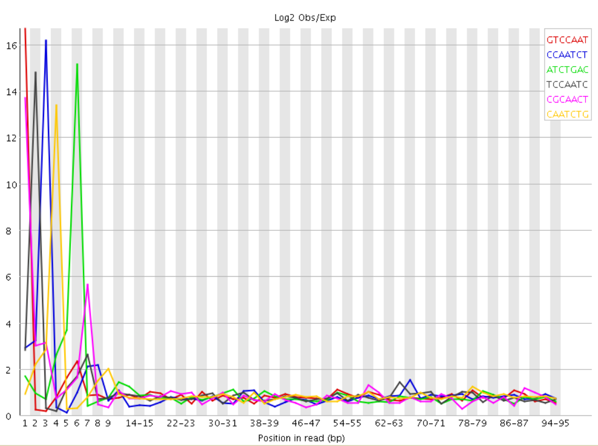
Read 1

Read 2









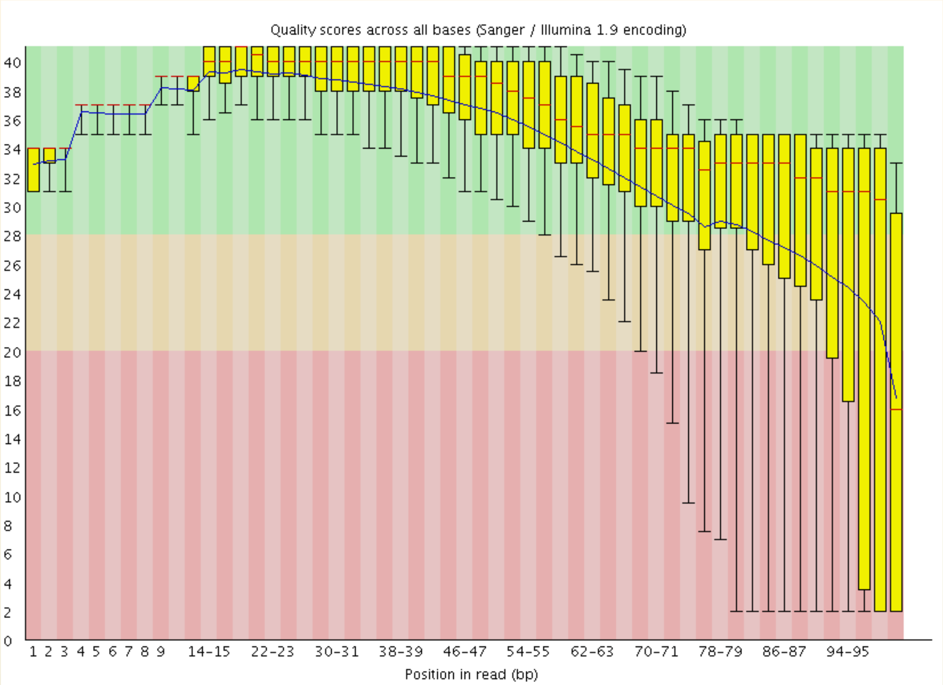
Raw read trimming

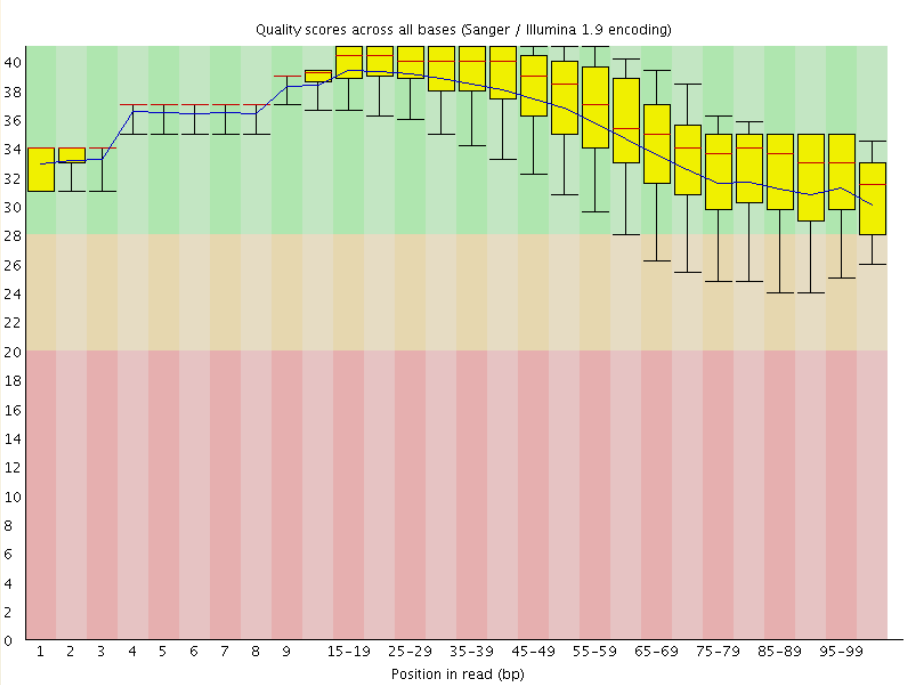
- Adapter trim: based on sequence similarity
  - Adapter/sequencing primer removal
- Hard trim: set number of bases
  - Certain primers
  - Known bias
- Soft trim: set quality threshold
  - Quality trimming
  - May be modified to trim on other criteria for special applications

What to trim I

- Depending on the aim of your project
  - Library adapters
  - Sequencing primers
  - Poor quality sequence beginning/end of read
  - Un-randomness at beginning/end of read

What to trim II

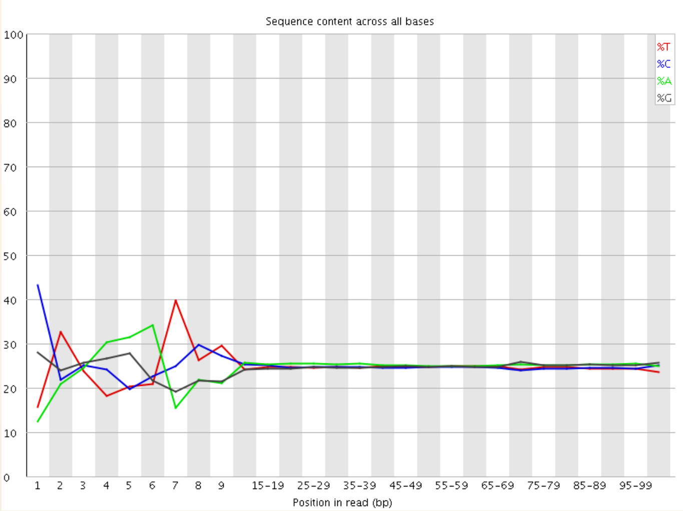


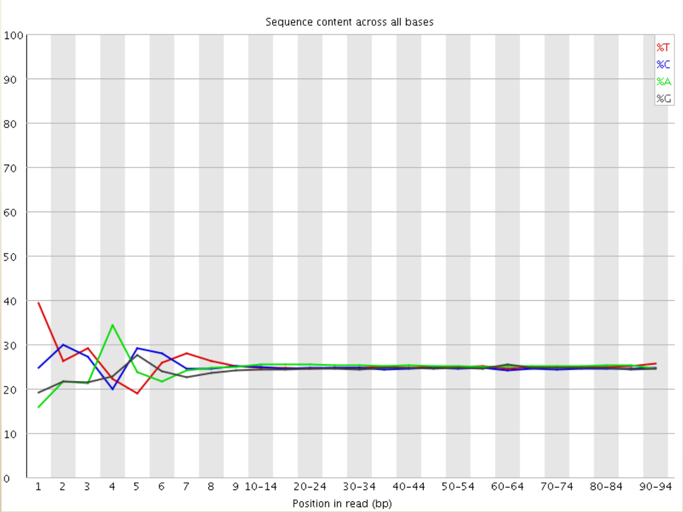


- *De novo*: Improves transcriptome assembly accuracy and efficiency
- Reference: shorter mapping time

Random hexamers in mRNA seq

- Hexamers are not completely random
- Hexamer hard trim is an option
- Might loose more data
- Might improve assembly
- Consider hard trim if your assembly/mapping stats are poor





What is enough trimming?

- Recommended to do adapter and quality trim
- Expect to loose between 10 and 15 % of your sequence data (more with suboptimal libraries)
- A stringent and/or global trimming setup leads to more data loss
  - This works well if making a transcriptome assembly
  - Differential expression analysis will suffer

Trimming of uninformative reads?

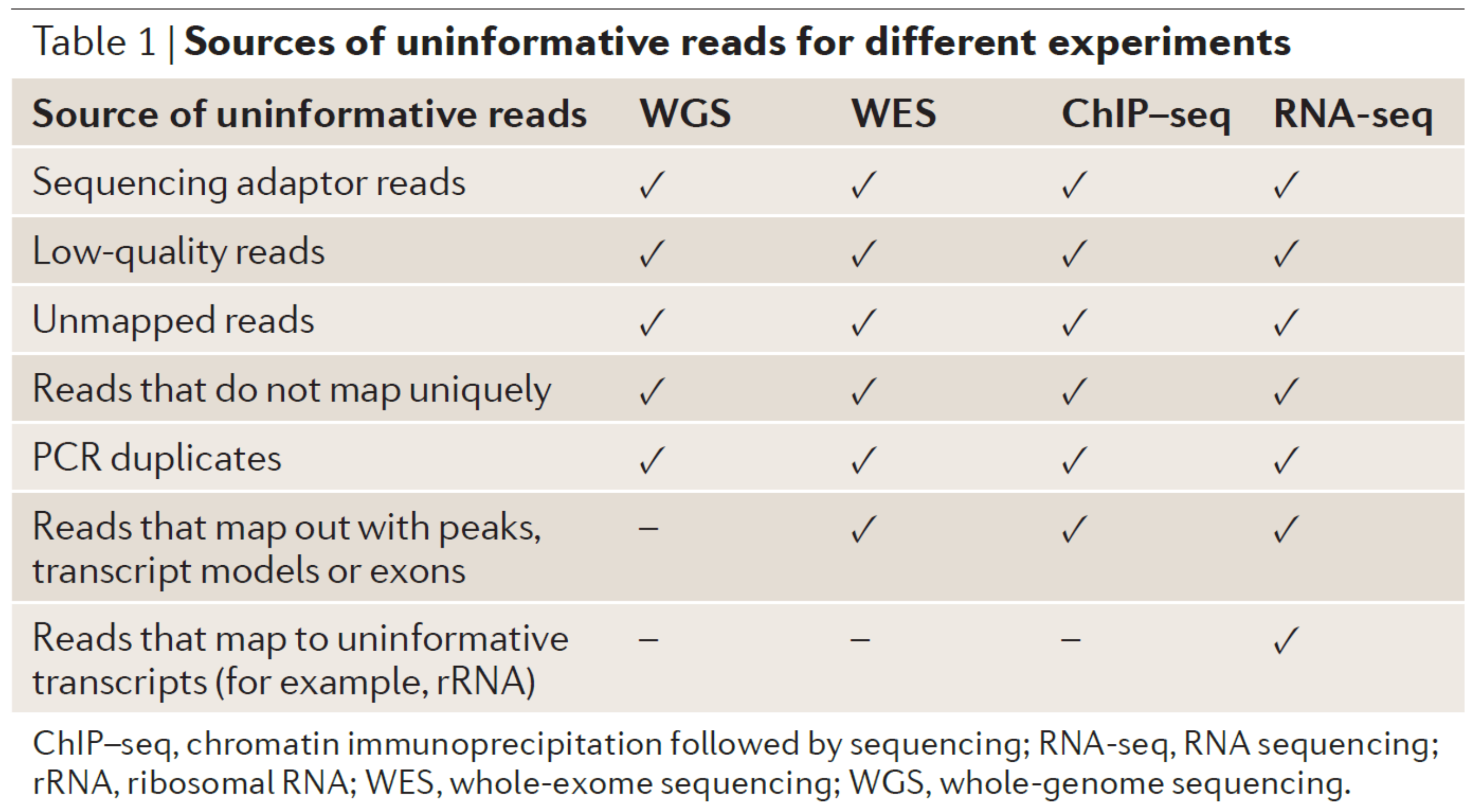


Table 1 | **Sources of uninformative reads for different experiments**

Source of uninformative reads	WGS	WES	ChIP–seq	RNA-seq
Sequencing adaptor reads	✓	✓	✓	✓
Low-quality reads	✓	✓	✓	✓
Unmapped reads	✓	✓	✓	✓
Reads that do not map uniquely	✓	✓	✓	✓
PCR duplicates	✓	✓	✓	✓
Reads that map out with peaks, transcript models or exons	–	✓	✓	✓
Reads that map to uninformative transcripts (for example, rRNA)	–	–	–	✓

ChIP–seq, chromatin immunoprecipitation followed by sequencing; RNA-seq, RNA sequencing; rRNA, ribosomal RNA; WES, whole-exome sequencing; WGS, whole-genome sequencing.

http://www.nature.com/nrg/journal/v15/n2/pdf/nrg3642.pdf

Low complexity / “identical” reads?

- Assembly
  - May slow down the assembly process
  - Do not contribute to increased resolution
  - Handled in various ways be different software
    - May lead to misassembled transcripts
- Differential expression
  - Do not remove low complex reads!
  - Normalization / removing reads affects read count

Effects of read trimming

Assembly

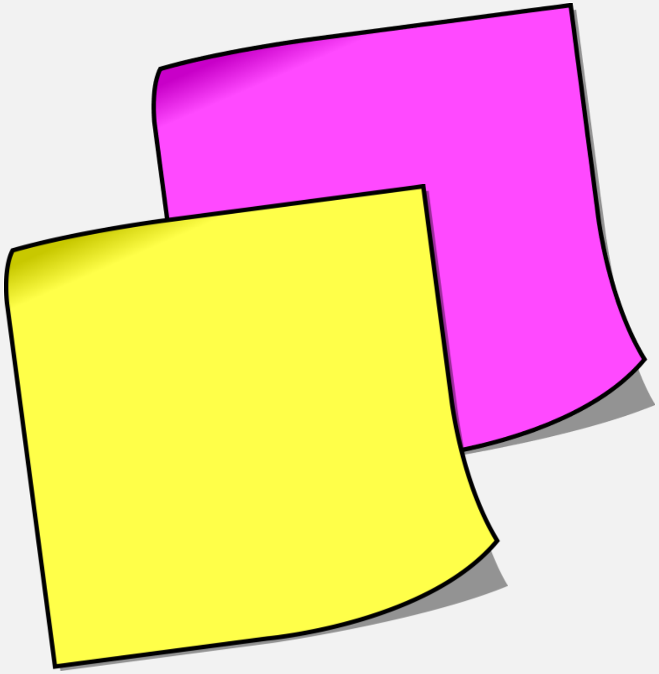
- Trimming-transcriptome completeness trade-off
- Trade-off between computation time and lower precision
- Trimming w/ sequence correction will lead to loss of rare transcripts

DE analysis

- Reduced dataset but higher % in mapback towards reference
- The trade-off is between Q20 and Q30
- Extensive trimming reduces information about lowly expressed genes

The sticky notes!

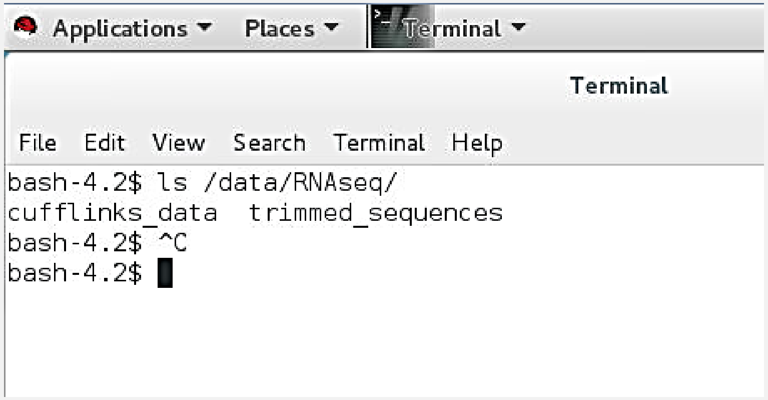
- Put up YELLOW if command is running nicely
- Put up PINK if error or other issues



Where is what?

https://login.tl.uio.no/

`/data/RNAseq/<various_folders>`

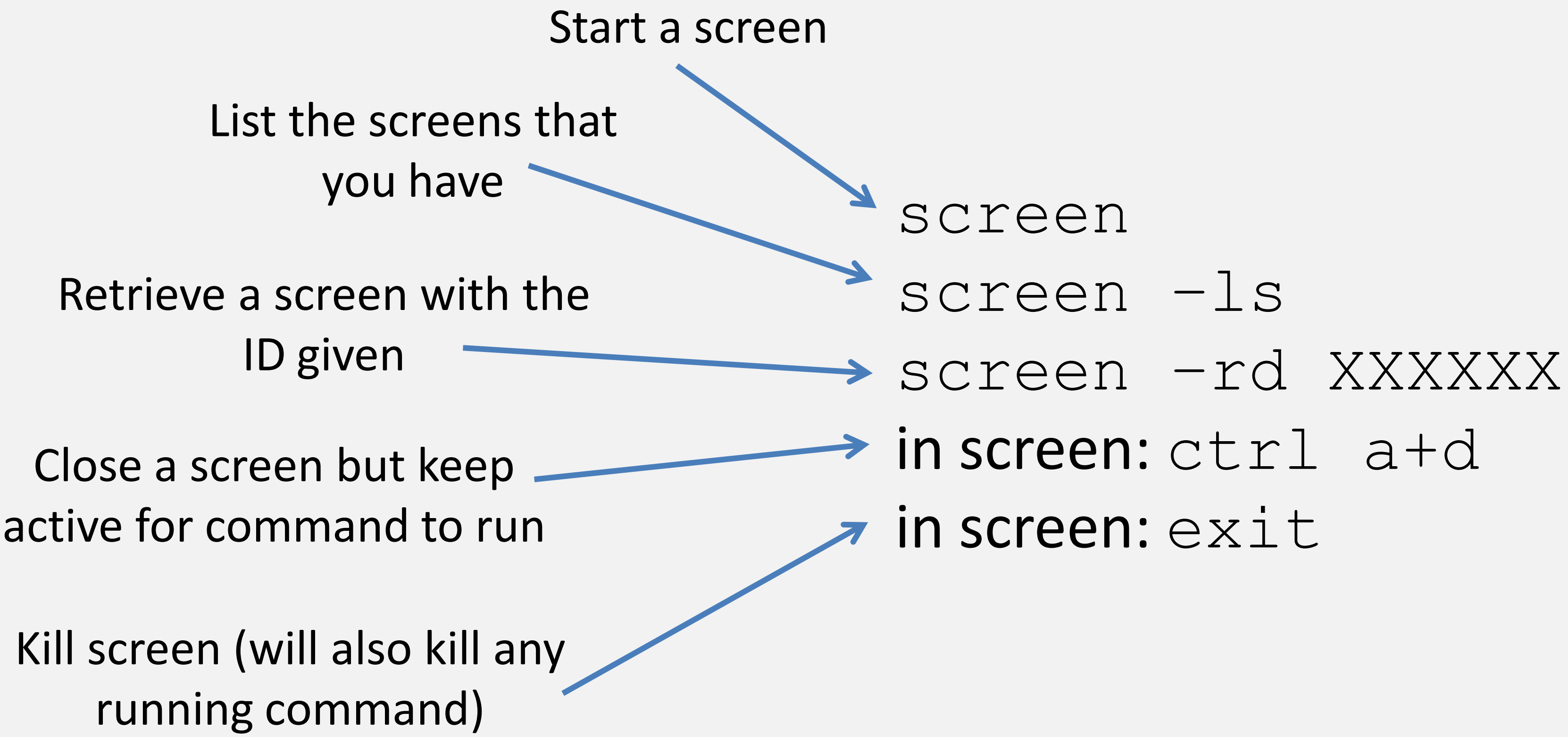


```
bash-4.2$ ls /data/RNAseq/
cufflinks_data  trimmed_sequences
bash-4.2$ ^C
bash-4.2$ 
```

- Don’t write to these folders – use your home area (~)
- When in doubt – copy needed files to ~ and run command
- Protect long-running commands with screen, nohup or similar.

Do you know screen?

- Protect long-running commands with screen, nohup or similar.



Start a screen

List the screens that you have

Retrieve a screen with the ID given

Close a screen but keep active for command to run

Kill screen (will also kill any running command)

```
screen
screen -ls
screen -rd XXXXXX
```

in screen: `ctrl a+d`

in screen: `exit`

Want to kill a running job?

- `Ctrl c` if the command is running on your command line
- `top` if the command is running in the background
  - In `top` press `k` and then the related PID number to kill it
  - Press `q` to quit `top`

Want to save output printing to the terminal (on screen)?

- To save standard out and standard err (stdout, stderr) attach the following to any command you like:
  - `1>file.out 2>file.err`
- Give the files names that make sense for you

```
cufflinks -p 30 -o 6hrs_A_cuff 6hrs_A/accepted_hits.bam

cufflinks -p 30 -o 6hrs_A_cuff 6hrs_A/accepted_hits.bam \
1>6hrsA_cufflinks.out 2>6hrsA_cufflinks.err
```

Short on syntax used on the slides

```
trim_galore \
--fastqc \
--gzip \
--length 40 \
--paired \
<~/Sample1_R1.gz> \
<~/Sample1_R2.gz> &
```

\ breaks up the command to make it more readable. Can also be used in the terminal

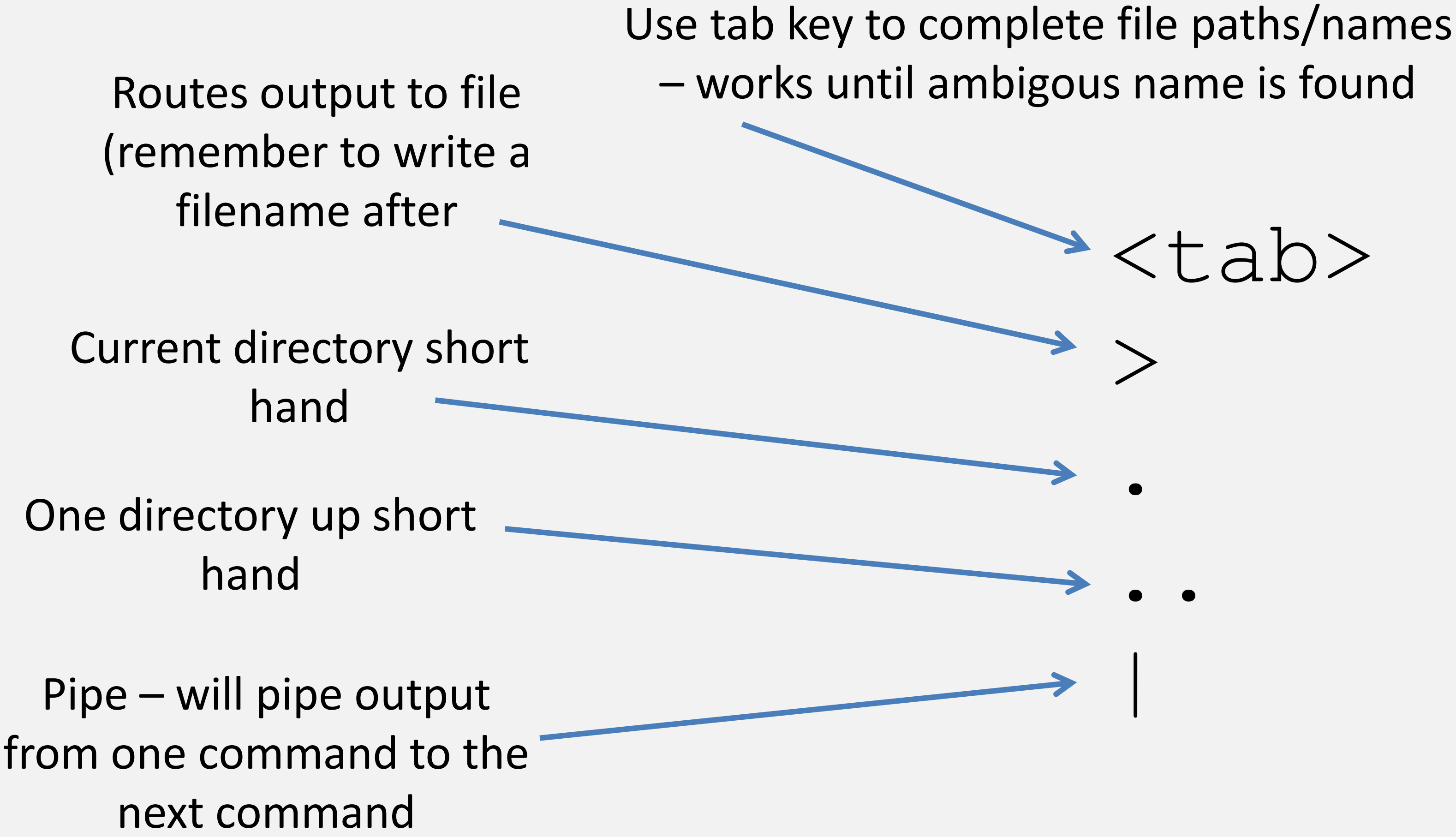
<...> fill in the true filename and do not use <>

& sends the command to the background to free the command line.

~ (tilde) is a shortcut referring to your home area

Short hands and short cuts

- Making life easier on the command line



Use tab key to complete file paths/names – works until ambigous name is found

`<tab>`

Routes output to file (remember to write a filename after

`>`

Current directory short hand

`.`

One directory up short hand

`..`

Pipe – will pipe output from one command to the next command

`|`

Clean up your home area regularly

- Delete any non-usable files in your home area

```
rm file
rm -r directory
```

PS: this cannot be reversed!

Your allocated resources

- Memory will not be an issue in this module
- Each student has 2 CPUs available
- Unless otherwise stated a command will use 1 CPU

Exercise 1

- Run fastqc on a full sample and a subsample
- Compare outputs

Run fastqc

```
pwd
mkdir <sample_name>
which fastqc

fastqc -o <sample_name> \
--noextract \
/data/RNAseq/trimmed_sequences/sample.fastq \
/data/RNAseq/trimmed_sequences/sample.fastq
```

`pwd` - see where you are
`mkdir` – make the folder you want the output in
`which` – to make sure fastqc is in your path

Evaluate the fastqc output

- What does per base sequence quality tell you?
- What does per tile sequence quality tell you?
- Do you see signs of adapters/hexamers in the per base sequence content?
- Is the GC content reasonable for a vertebrate?
- Any overrepresented sequences?
- What differes between the two sample sets?

Mapping

- Make directory for index in ~
- Copy index there (.tb2 files + gff file)
- Run tophat2 on subsample (if time full sample also)

```
tophat -o <name_dir> \
-p 2 \
--transcriptome-index=/data/RNAseq/gamo_index/ gadMor2_ena  gadMor2_ena \
< path to read 1> \
< path to read 2> \
1>~/file.out 2 >~/file.err
```

Evaluate the mapping results