RNA seq: differential expression analysis

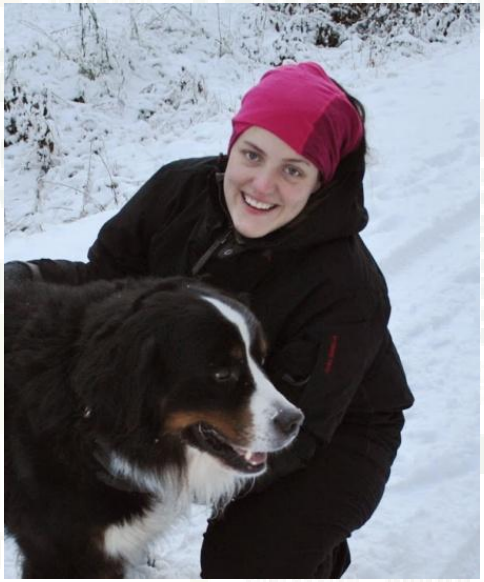
For INF-BIO 4121/9121
Fall semester 2015

Monica Hongrø Solbakken
m.h.solbakken@ibv.uio.no

UiO : Centre for Ecological and Evolutionary Synthesis
University of Oslo

About me

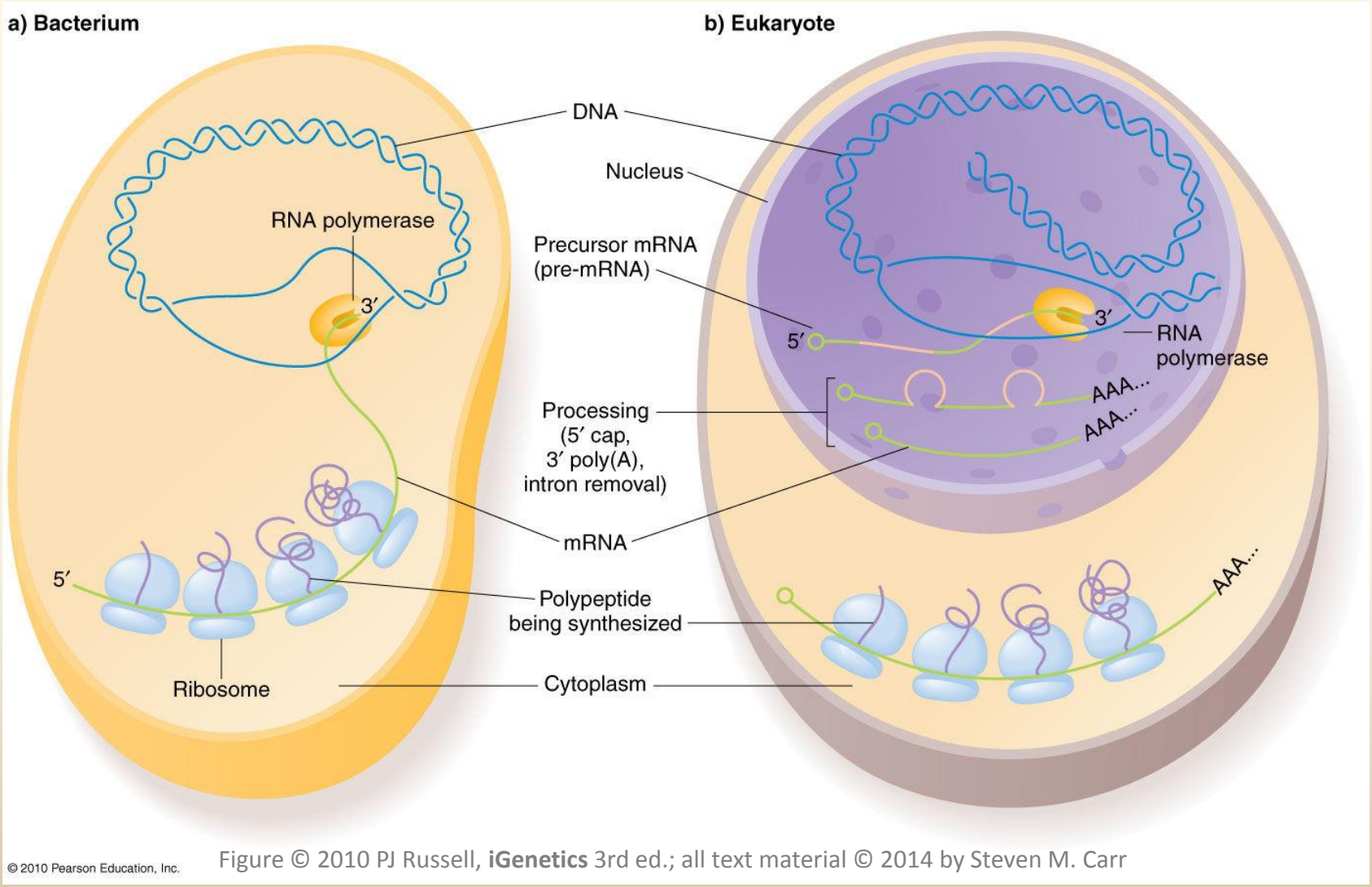
- PhD student @ CEES
- Works on the immune system of Atlantic cod
- From sampling through lab to computer
  - But not a bioinformatician ☺
- Uses RNAseq to investigate the development of immune responses during infection



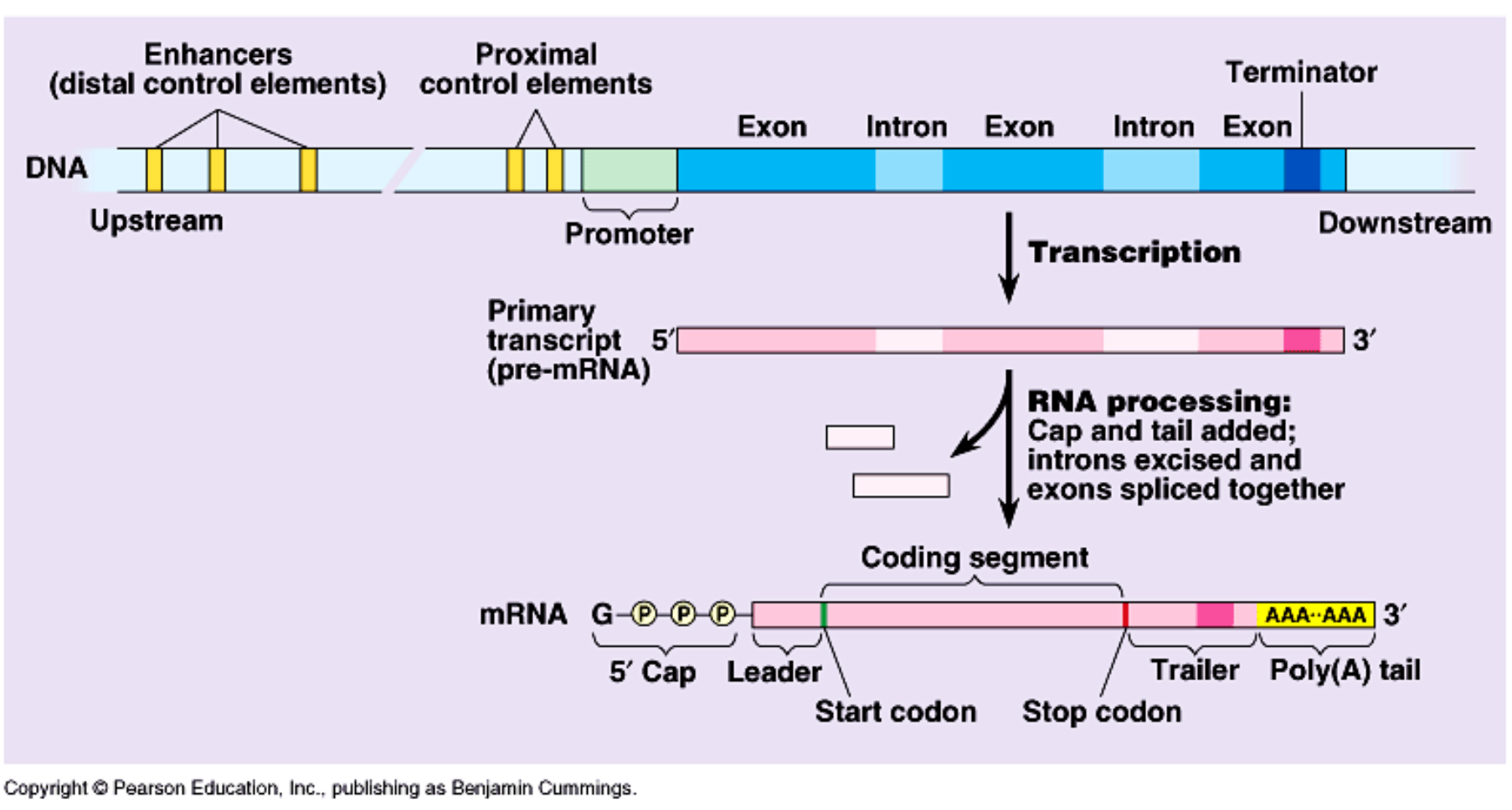
Outline I

- The RNAseq module day 1
  - Transcriptomics intro
  - Preparing for RNAseq and raw data evaluation
- Today we will cover
  - Experimental design and considerations
  - Evaluation of RNAseq data
  - Sequence trimming
  - Transcriptome assembly

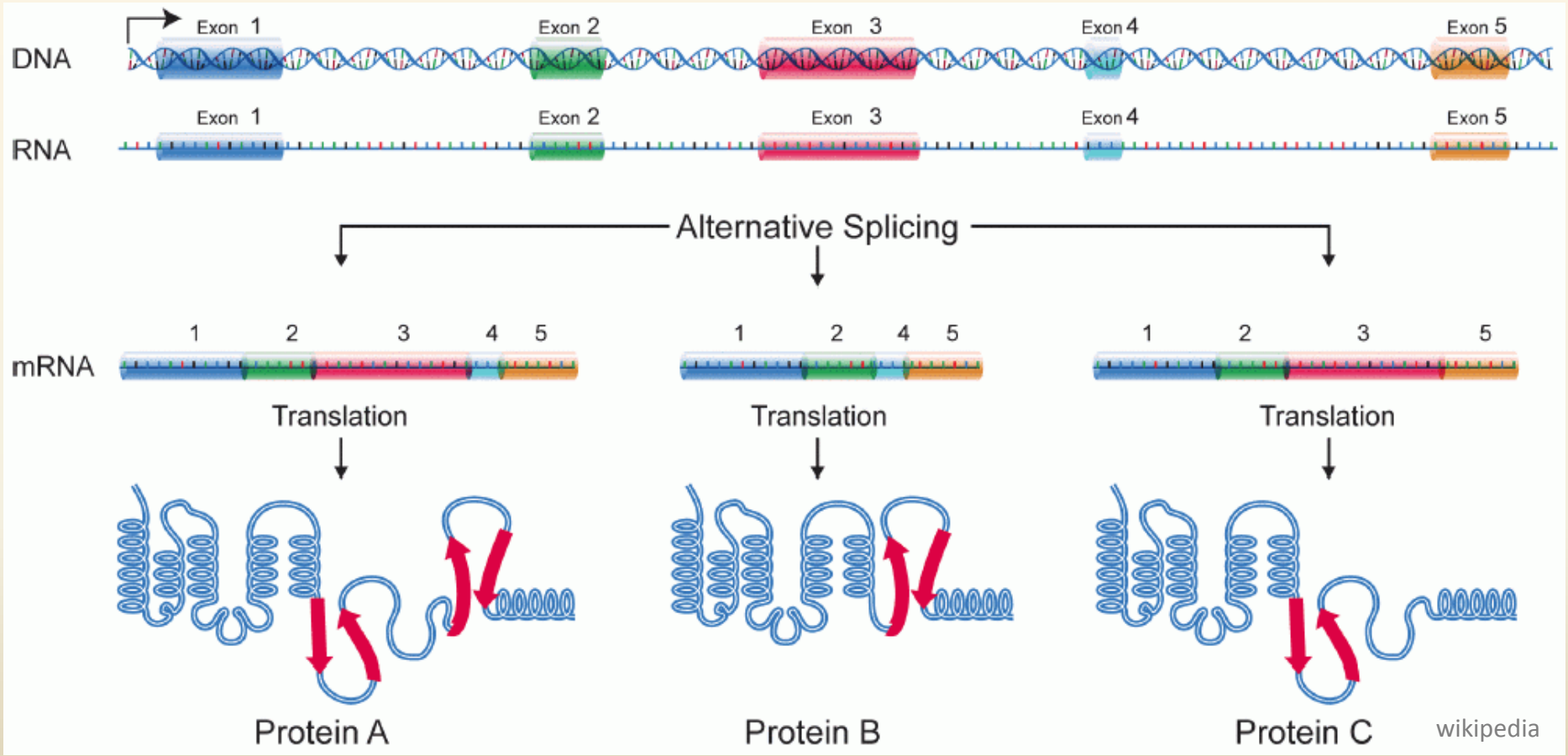
Transcription



Transcription eukaryote mRNA

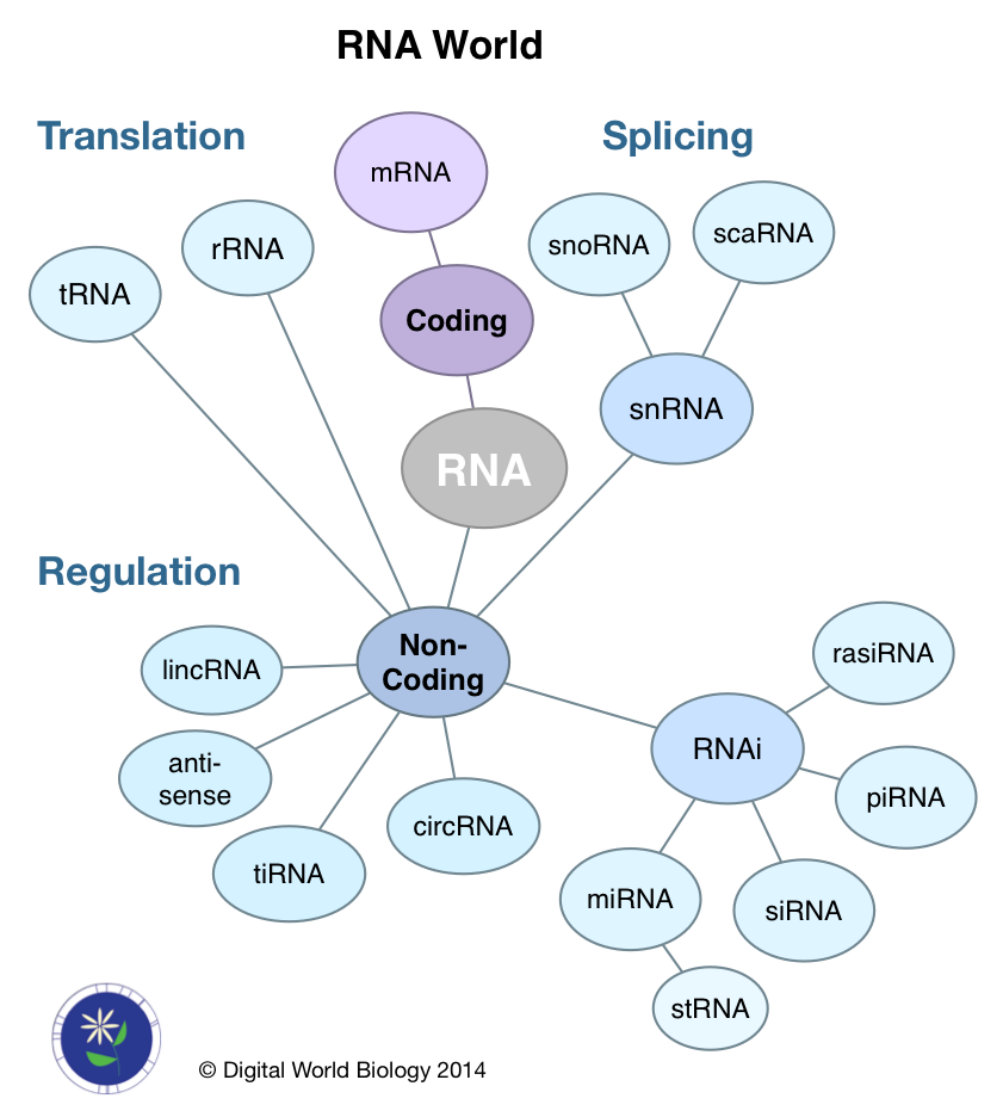


Splice variation eukaryotic mRNA



A transcriptome REALLY is

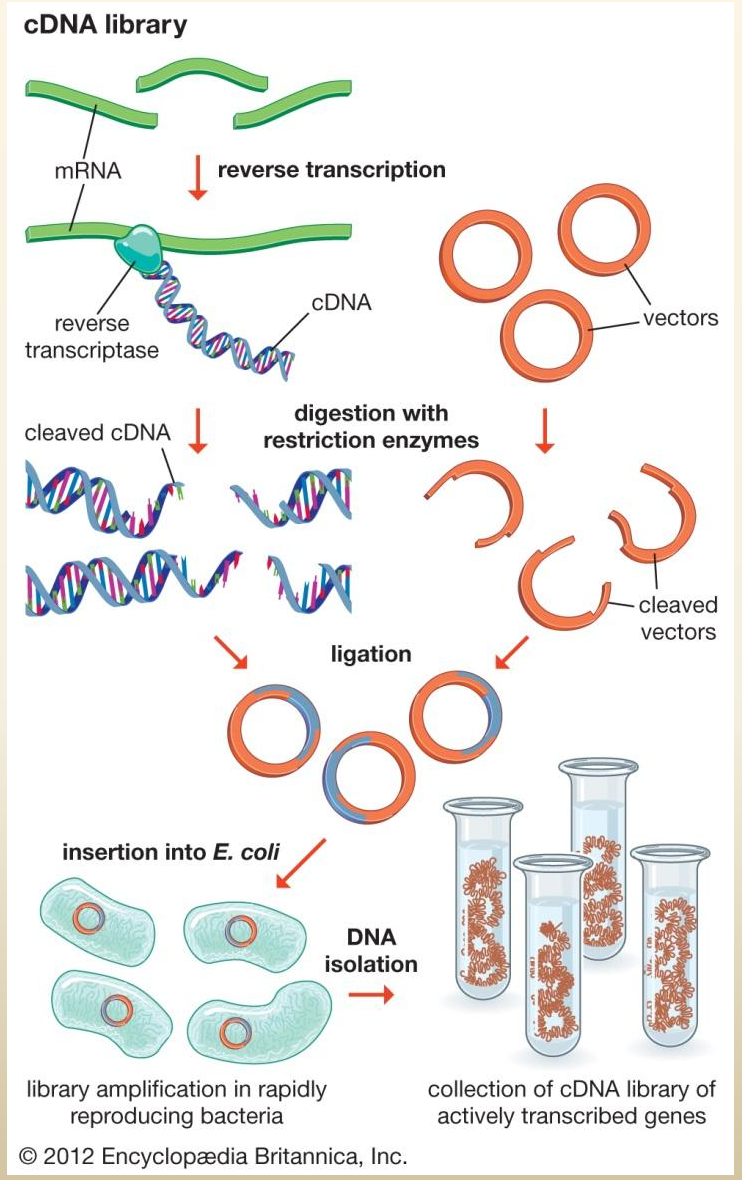
A snapshot in time of **all RNAs** present in a sample isolated from a given cell, tissue or organism



Obtaining transcriptomes I

Sanger cDNA library sequencing

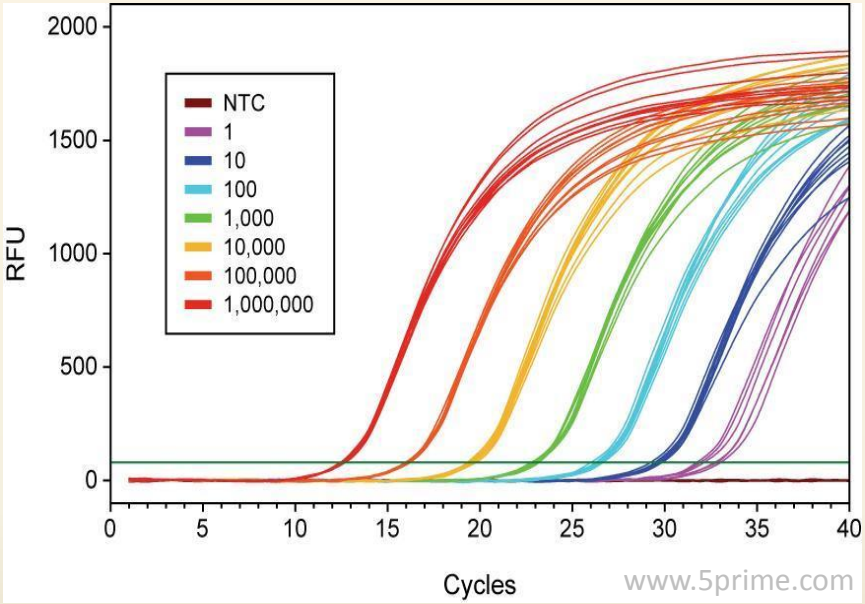
- mRNA converted to the more stable cDNA
- cDNA cleaved and ligated into vectors
- Vectors amplified (cloned) in *E. coli*
- DNA isolated = cDNA library
- Sequenced on Sanger
- Low throughput
- High accuracy



Obtaining expression I

Quantitative RT-PCR

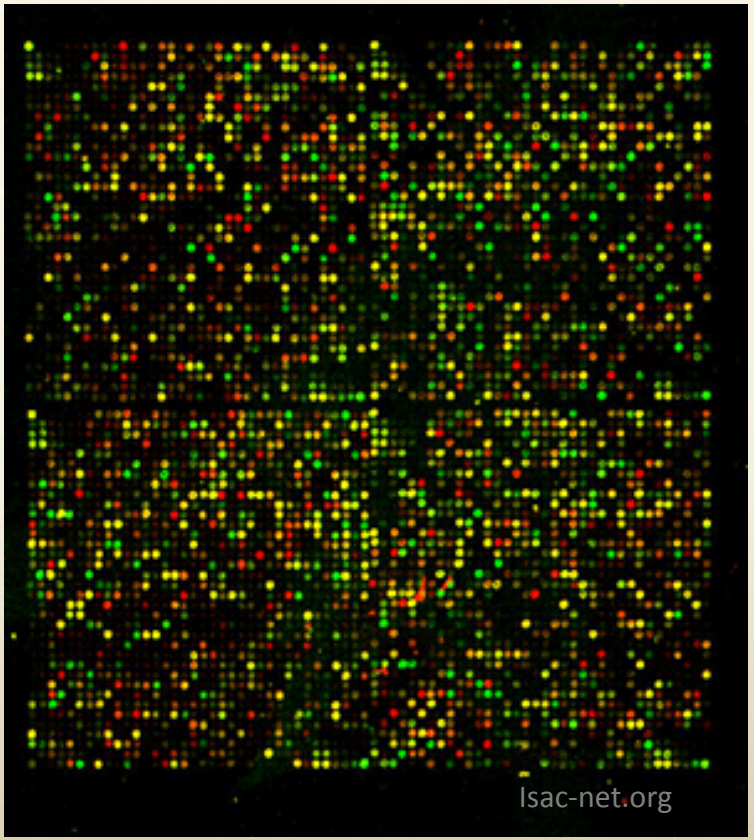
- cDNA library sequencing does not provide expression levels
- qRT-PCR requires knowledge of gene sequence
- Hard manual work
- Low throughput
- Expression level relative to control



Obtaining expression II

Microarray - expression determination

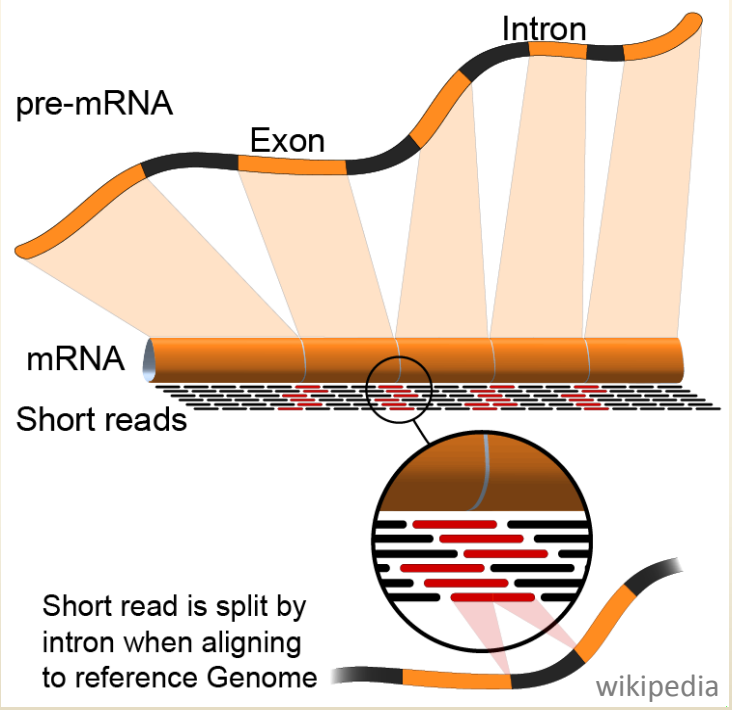
- Requires gene sequences for probe design
- High throughput compared to qRT-PCR
- Possibility of outsourcing
- Expression results relative to all probes



Next generation transcriptomics

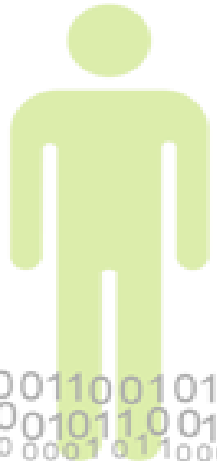
RNA sequencing

- Transcriptome and expression in one go
- No need for gene sequence information
- High throughput
- Can be outsourced
- Costly, but effective
- Expression results relative to all transcripts



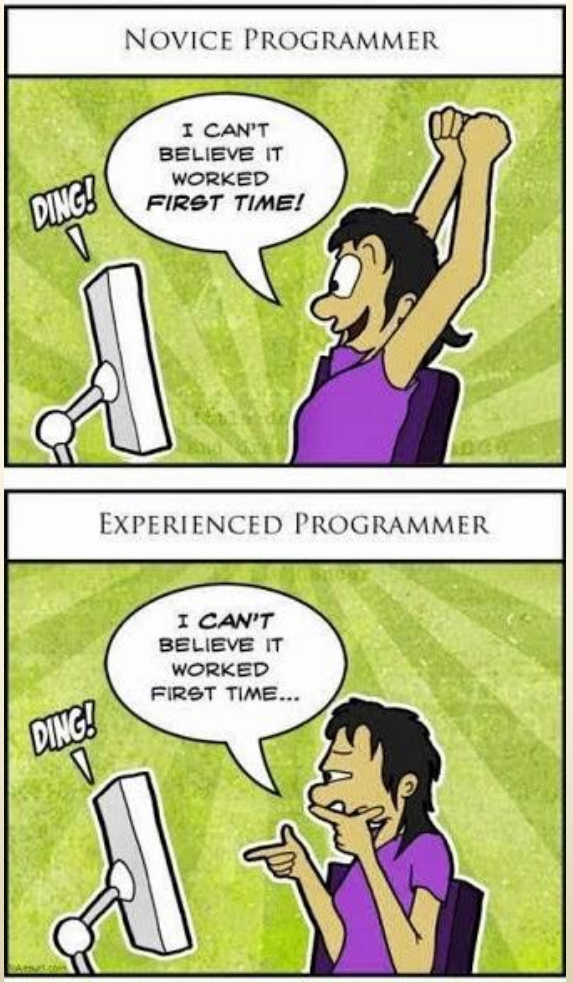
Cons

- Heavily dependent on proper experimental design
- Enormous amounts of data
- No straight forward analysis
- Usually no clear-cut story from individual gene expressions



Pros

- Others have traversed the path you now set upon
- There are pipelines to help you manage the data
- Careful design will highlight your hypothesis beautifully
- The data you possess when you are finished are immense and really cool



http://erika-vilches.com/

One last thing…

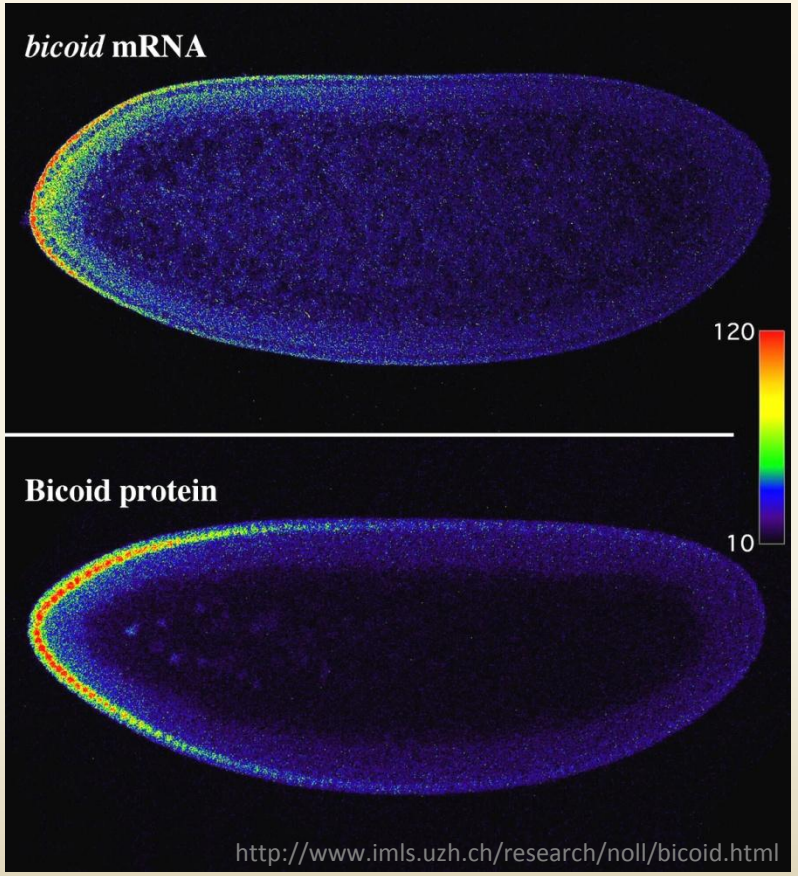
We are used to the single gene/few genes study

We interpret findings as ”biological significant”

We tend to think…

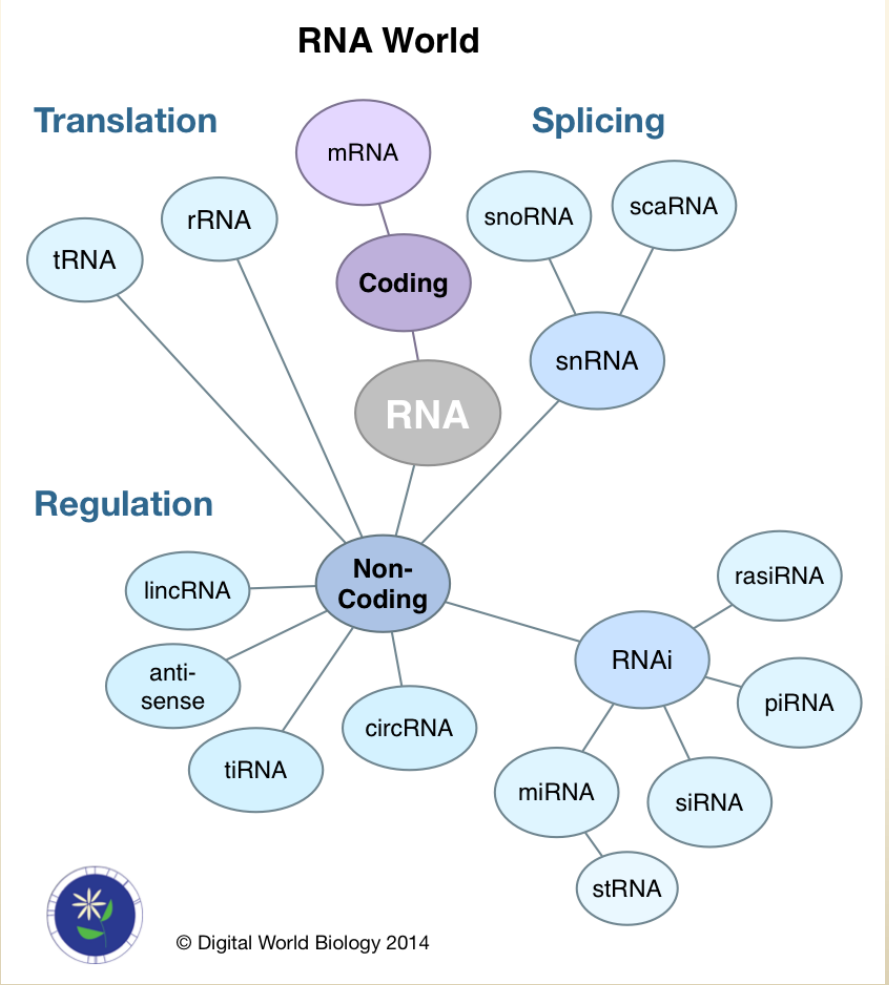
- Transcriptome = mRNA
- mRNA = Protein
- Protein = Biological relevance

- Things are seldom as simple as the bicod mRNA / bicoid protein relationship!

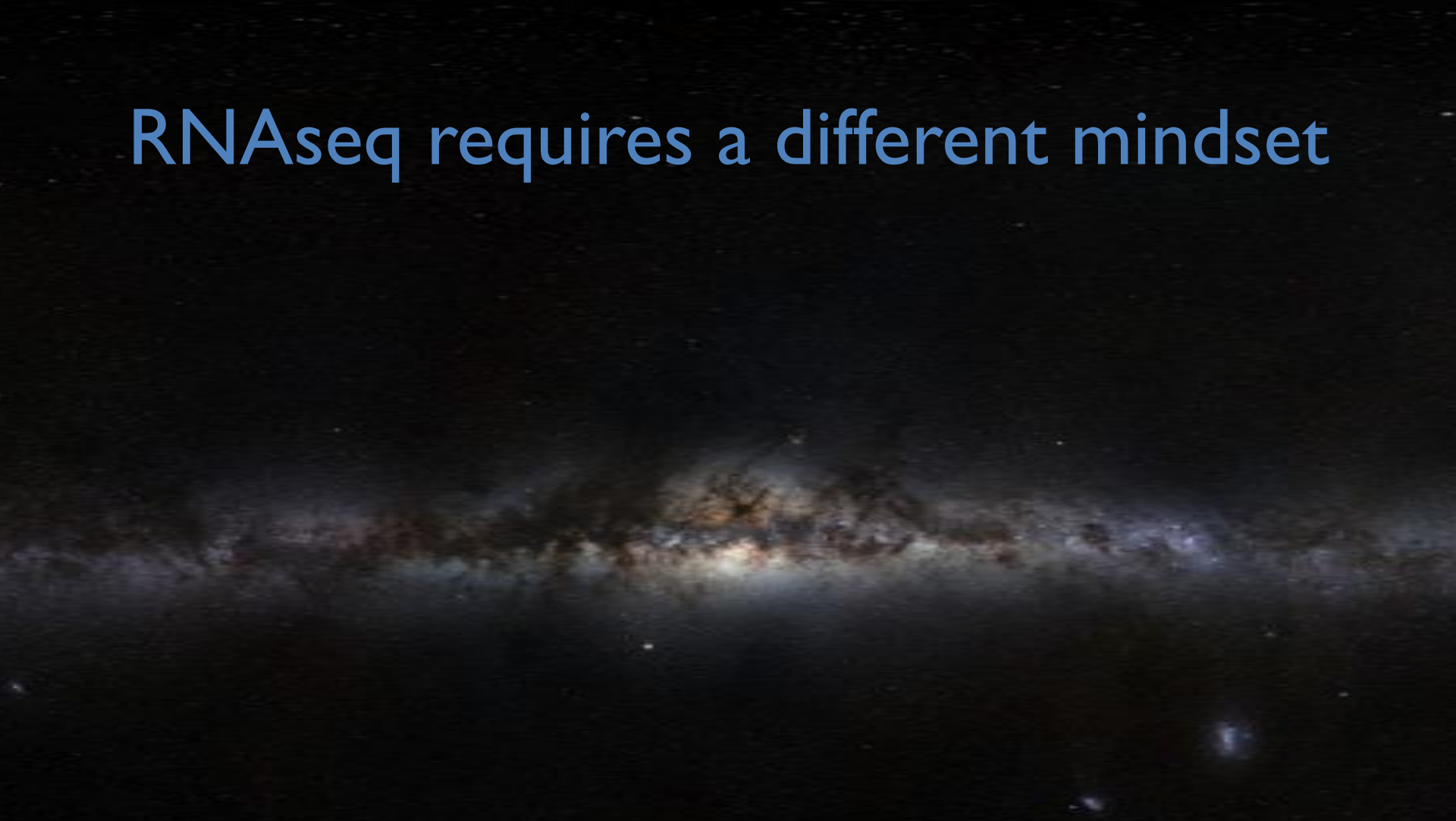


Before interpreting function

- Remember:
  - RNA decay
  - RNA editing
  - RNA splicing
  - Translation regulation
  - RNA interference
  - …



RNAseq requires a different mindset



It's like watching a picture of the milky way when you are used to watching a picture of the sun…

Step 1
Experimental design

Experimental design

Time

Money

Manpower

Tools

What do you have / want / need?

- Reference transcriptome?
- Reference genome?
- Differential expression data?
  - Response to treatment(s)?
  - Development over time?
- totalRNA or specific RNA species?

What do you have / want / need? II

- Depending on focus you may perform:
  - rRNA depletion
  - mRNA selection
  - Abundant transcript removal
  - smallRNA conservation
  - Skip library amplification
  - Strand specific library preparation

Experimental design –if time is of the essence

- Shorter time
  - Model organism
  - High quality RNA
  - Plain treatment/control setup or
  - Plain time-series with time 0
  - Dedicated person with support system

Experimental design –if time is of the essence

- Longer time
  - Non-model organism
  - Low quality RNA
  - Complex multifactorial designs
  - No single dedicated person / lack of support

Experimental design - toolkit

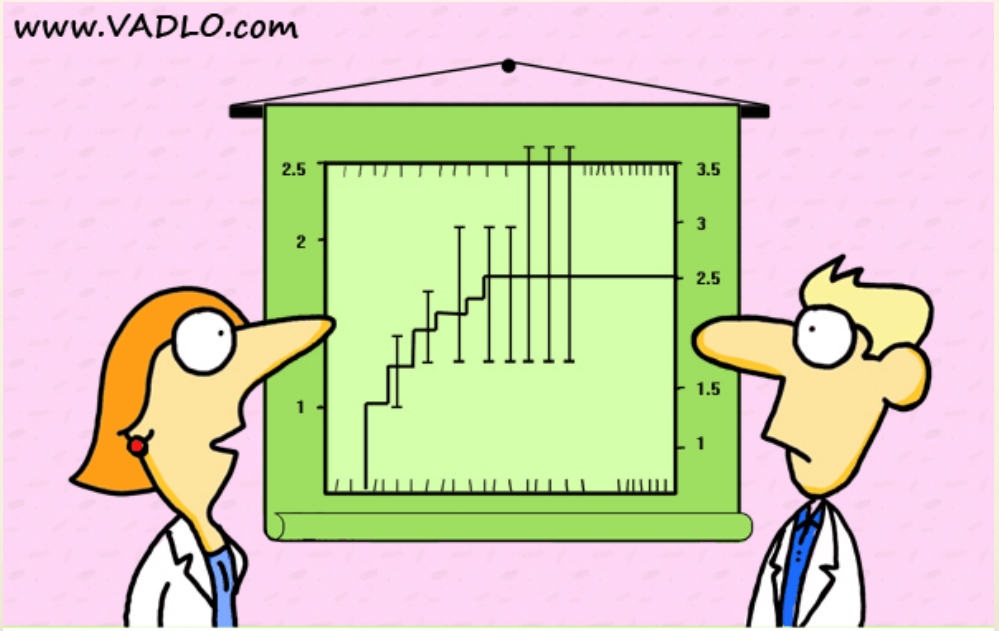
- Tools
  - Model organisms
    - Genome information
    - Several pipelines
    - Lots and lots of tools for cool vizualizations
  - Non-model organisms
    - No genome (or have to make your own)
    - Few tools designed for this

Experimental design - $

- Money
  - Biological replicates
  - Technical replicates
  - Sequencing depth
  - Sequencing technology
  - Computing resources
  - Manpower

Experimental design - statistics

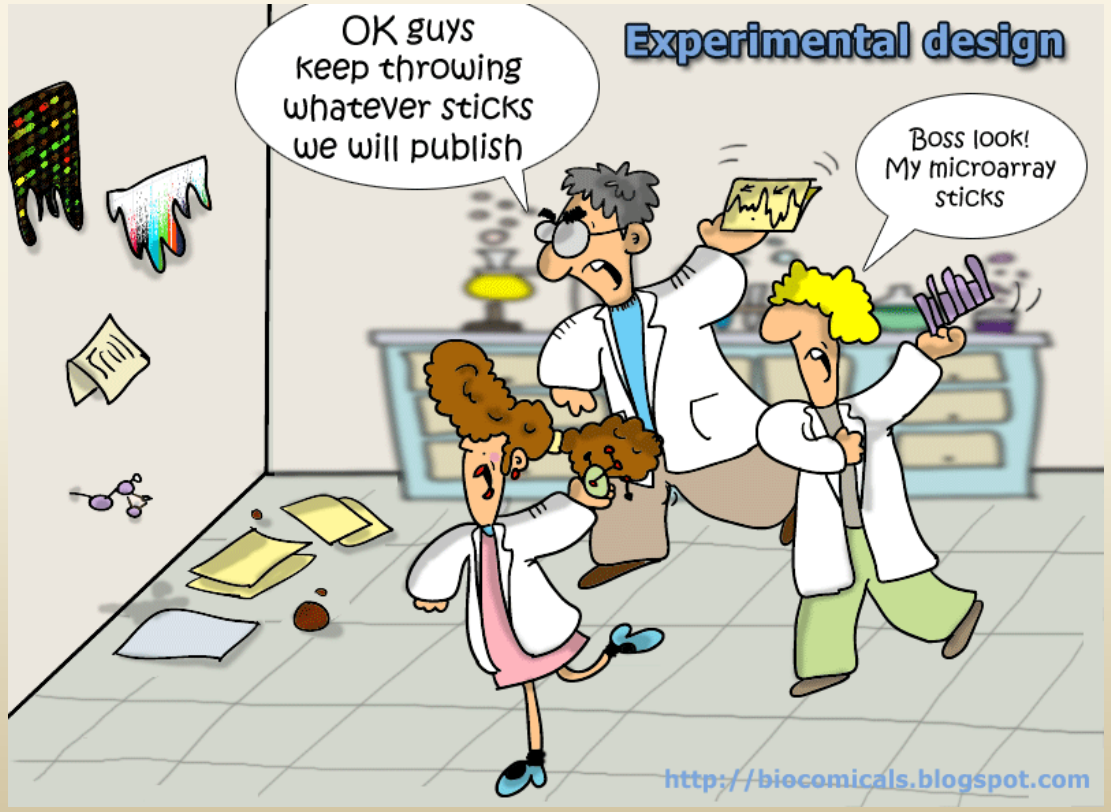
- Biological replicates are crucial
  - Aim at more than 3
  - Enables proper statistical analyses
- Technical replicates is positive
  - Eliminates sequencing bias



"Did you really have to show the error bars?"

All aspects are connected

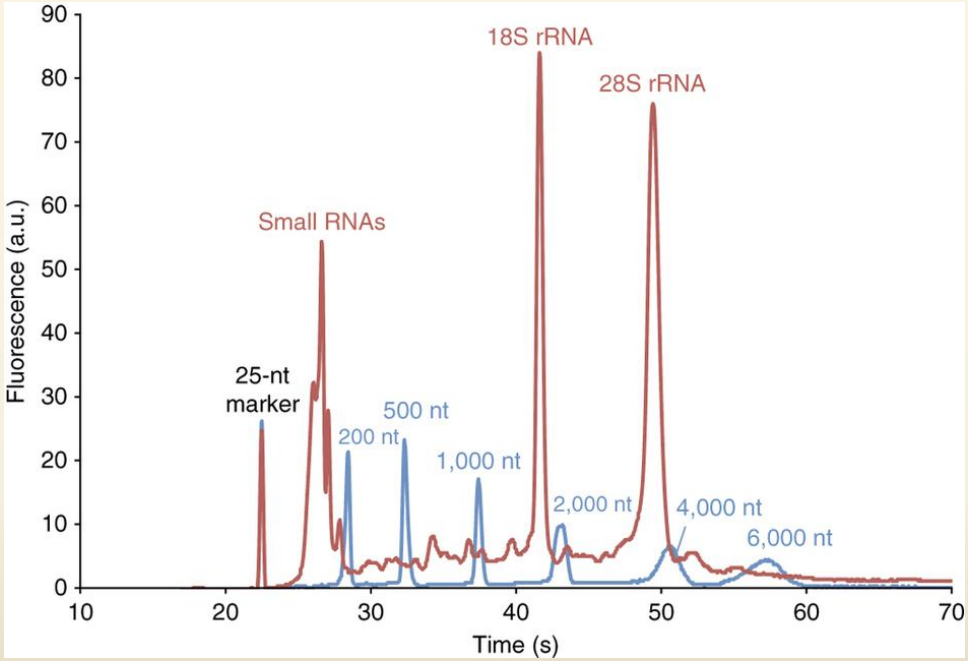
- Sample preparation must reflect experimental design!
- Otherwise this will be your outcome:



Step II
Sample preparation

RNA isolation

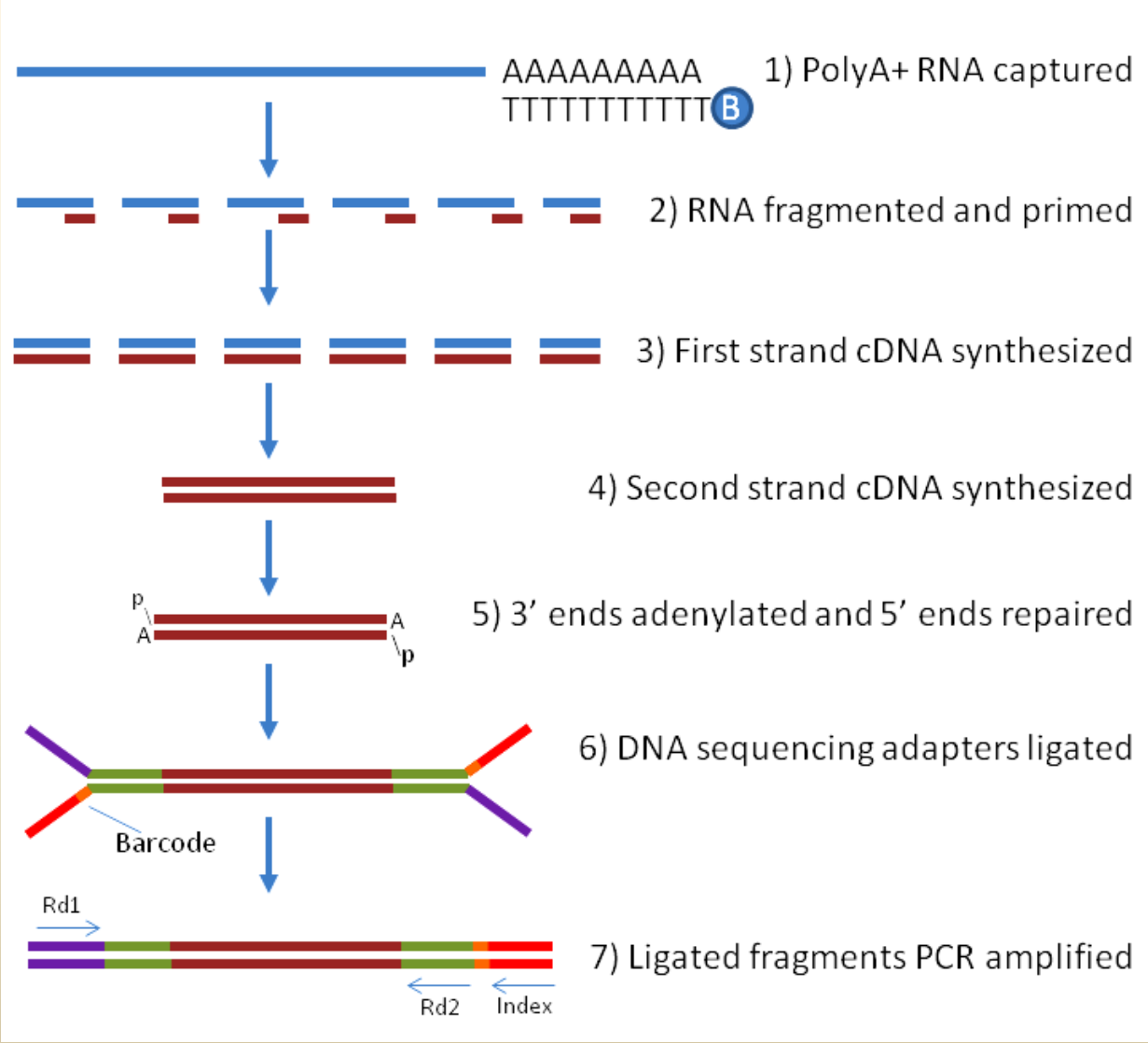
- Aim for high quality RNA with good integrity and concentration
- Column based isolation loses all small RNAs
- Chlorofom left-overs may interfere with sequencing reaction



Library preparation

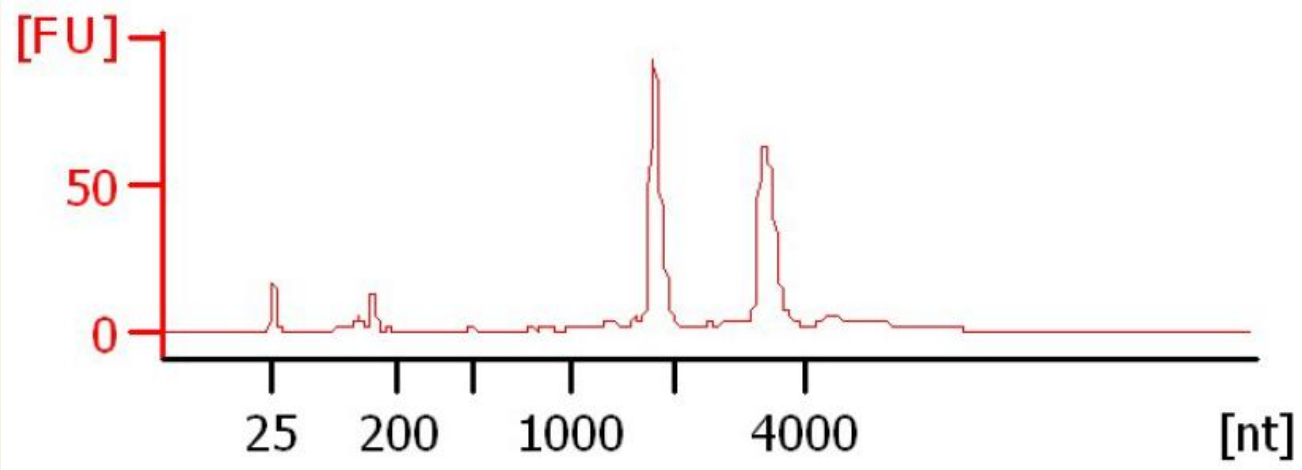
- TotalRNA works for most applications
- Physical or enzymatic shearing – fragment optimum depends on sequencing instrument
- With or without amplification – depends in RNA availability
- More than 24 samples – consider robot preparation

Library preparation – mRNA Illumina

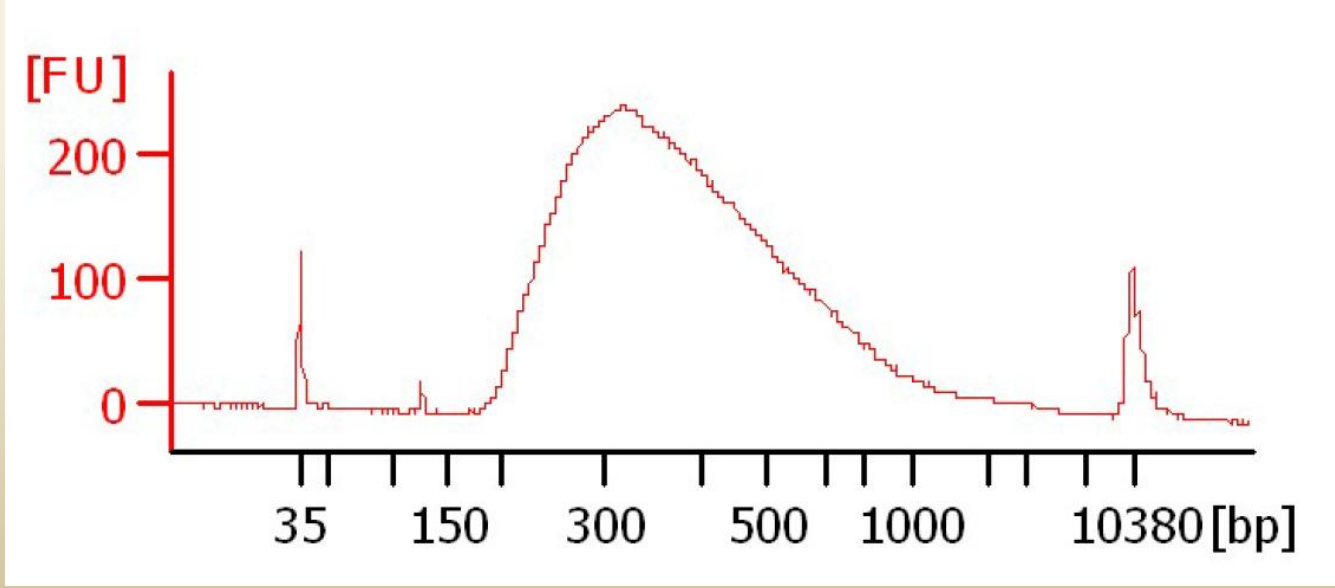


Final library – mRNA Illumina

Gone from this:
totalRNA with ribosomal peaks



To this:
mRNA selected library with ~350 bp fragment size



Step III
– RNAseq technologies

Choose your sequencing technology

Model species

Differential expression

Isoforms

Read length

Paired-end

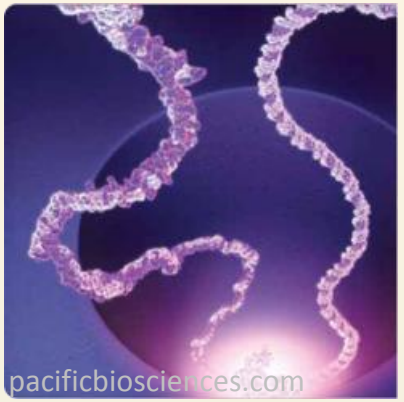
Genome resource

Money

Novel transcripts

PacBio

- Long read sequencing technology
- Sequences entire RNAs up to 10 kb
- Reconstruction of isoforms
- Detection of novel transcripts
- Expression analysis
- Great for reference transcriptomes



PacBio sequencing well

Prof. Kjetill Jakobsen showing of the NSC' PacBio

Illumina

- Short read paired-end technology
- ~150 bp x 2
- Reasonable reconstruction of isoforms
- Reasonable detection of novel transcripts
- Expression analysis
- Makes decent reference transcriptomes

www.illumina.com

Illumina HiSeq 4000

Sequencing output

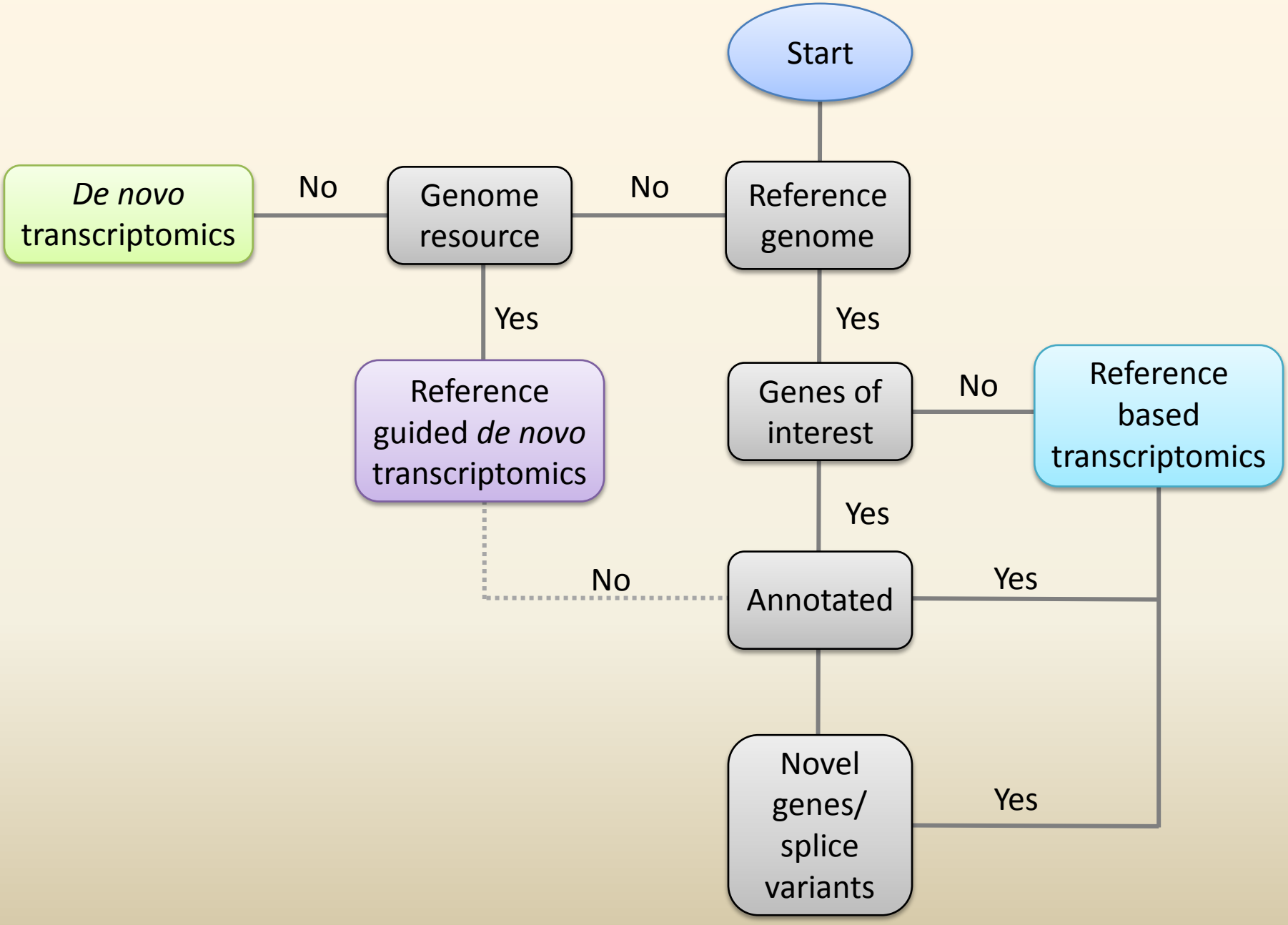
Differential expression

- Model organism
  - Illumina >= 10 mill PE / sample
  - More for rare transcripts
- Non-model organism
  - Illumina >= 20 mill PE / sample
  - More for rare transcripts

Transcriptome assembly

- Depends on species
  - Illumina 100-150 mill PE reads minimum for vertebrates
  - For comparison yeast is suffcient with 4 mill PE stranded reads
  - PacBio vertebrate example:
    - ~25 000 full-length cDNAs
    - 9 SMRT cells 1-2kb, 2-3kb and 3-6 kb

Step IV
– Available tools and resources


Start
Reference genome
No
Genome resource
No
De novo transcriptomics
Yes
Reference guided *de novo* transcriptomics
Yes
Genes of interest
No
Reference based transcriptomics
Yes
Annotated
No
Yes
Novel genes/ splice variants
Yes

Resources

- Computing power
  - One strong computer (model organism)
  - Access to a cluster
- Genome resources
  - Reference or draft genome (may be closely related species)
  - Large toolkit available
- No genome
  - Limits you significantly in terms of toolkit

Design that experiment

- Gerbils, desert rats, are small mammalian rodents related to rat/mice
- No genome resource, but your collaborator is making a draft genome
- Some populations have great resistance against disease Y
- Sick animals can be obtained
- Gene X is involved in disease Y resistance without clear function

Design a RNAseq experiment investigating the gerbil immune response focusing on a hypothesis of your choice

Design that experiment - suggestions

Step V
– The case for INF-BIO

A treatment in cod

The INFBIO case

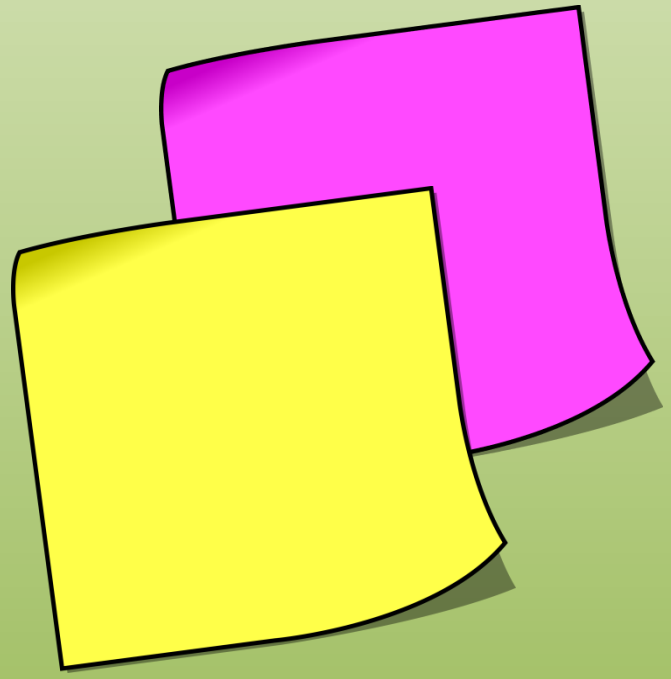
- Non-model organism – Atlantic cod
- Reference genome available but suboptimal for this case - *de novo* transcriptomics
- A treatment to investigate immune responses
- Simple treatment-control setup over time

De novo transcriptomics I

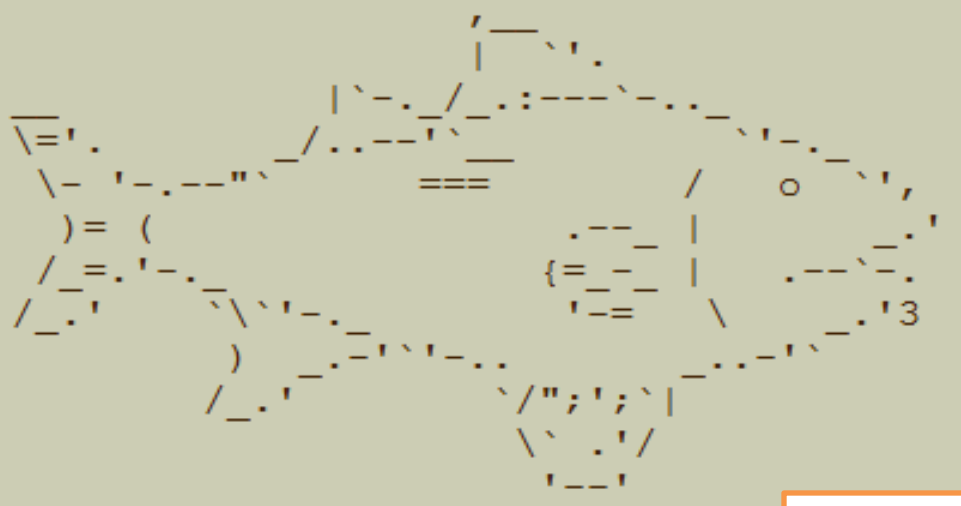
- Sequence data assessment
- Sequence trimming and reassessment

The sticky notes!

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



Where is what?

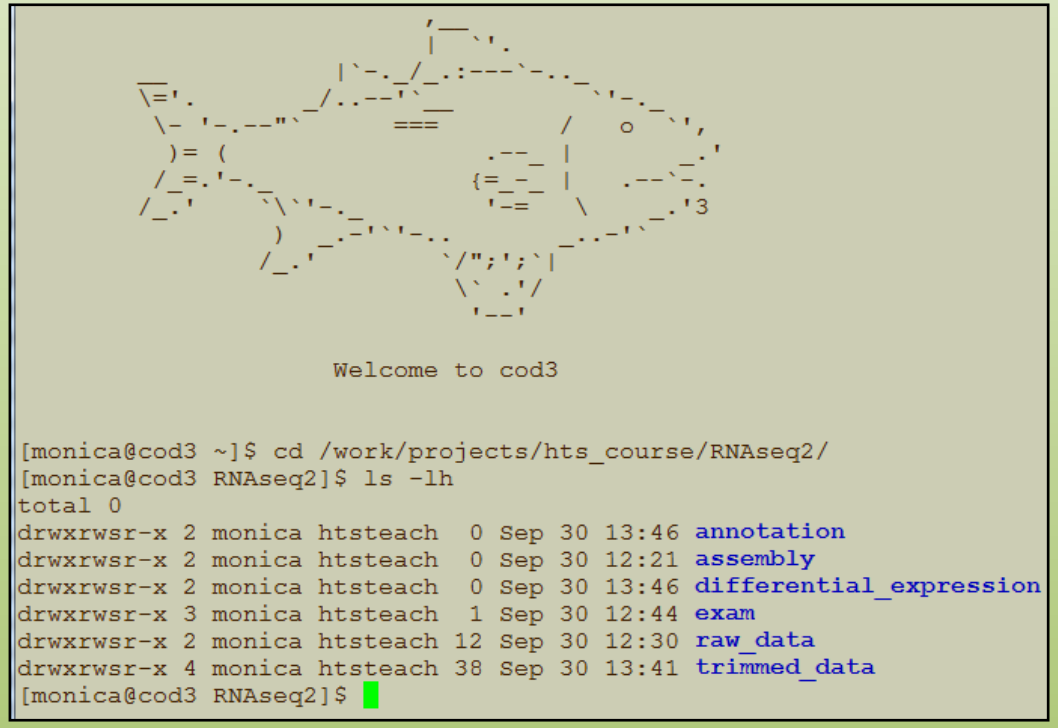


Welcome to cod3

/data/RNAseq2

```
[monica@cod3 ~]$ cd /work/projects/hts_course/RNAseq2/
[monica@cod3 RNAseq2]$ ls -lh
total 0
drwxrwsr-x 2 monica htsteach  0 Sep 30 13:46 annotation
drwxrwsr-x 2 monica htsteach  0 Sep 30 12:21 assembly
drwxrwsr-x 2 monica htsteach  0 Sep 30 13:46 differential_expression
drwxrwsr-x 3 monica htsteach  1 Sep 30 12:44 exam
drwxrwsr-x 2 monica htsteach 12 Sep 30 12:30 raw_data
drwxrwsr-x 4 monica htsteach 38 Sep 30 13:41 trimmed_data
[monica@cod3 RNAseq2]$
```

Where is what?



Your home area is:
/cluster/teaching/homedirs/<username>

Do NOT WRITE to these folders - write to your own area!

Your physical location (`pwd`) should be in your home area and you point the command to where the raw / trimmed data are

Protect long-running commands with screen, nohup or similar

Short on syntaxt

```
module load trim-galore
module load fastqc

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

Load all programs you need – in this case trim-galore also needs fastqc to run properly

\ breaks up the command to make it more readable. Use a white space in front of \

<…> fill in the true filename

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

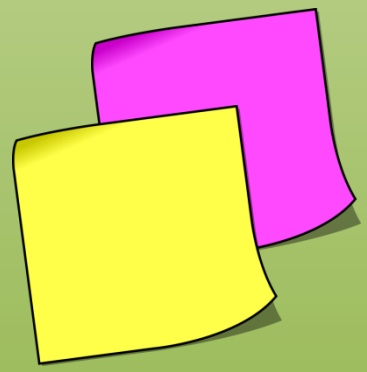
& sends the command to the background to free the command line.

Permission errors

- If you get permission denied errors your are likely trying to:
  - write to /data/RNAseq2

- Make sure you are «home» in

`/cluster/teaching/homedirs/<username>`



The case

- 2 time-points post treatment
  - 4 day
  - 21 day
- 12 samples x 2
  - 6 controls (K)
  - 6 treated (V)

```
[monica@cod3 raw_data]$ ls -lh
total 46G
-rwxr-xr-x 1 monica htsteach 1.8G Oct  5 11:25 21dayK_F4_R1.fastq.gz
-rwxr-xr-x 1 monica htsteach 1.8G Oct  5 11:26 21dayK_F4_R2.fastq.gz
-rwxr-xr-x 1 monica htsteach 2.1G Oct  5 11:27 21dayK_F5_R1.fastq.gz
-rwxr-xr-x 1 monica htsteach 2.1G Oct  5 11:28 21dayK_F5_R2.fastq.gz
-rwxr-xr-x 1 monica htsteach 2.7G Oct  5 11:30 21dayK_F6_R1.fastq.gz
-rwxr-xr-x 1 monica htsteach 2.7G Oct  5 11:31 21dayK_F6_R2.fastq.gz
-rwxr-xr-x 1 monica htsteach 1.7G Oct  5 11:32 21dayV_F1_R1.fastq.gz
-rwxr-xr-x 1 monica htsteach 1.7G Oct  5 11:33 21dayV_F1_R2.fastq.gz
-rwxr-xr-x 1 monica htsteach 1.8G Oct  5 11:34 21dayV_F2_R1.fastq.gz
-rwxr-xr-x 1 monica htsteach 1.7G Oct  5 11:35 21dayV_F2_R2.fastq.gz
-rwxr-xr-x 1 monica htsteach 1.8G Oct  5 11:36 21dayV_F3_R1.fastq.gz
-rwxr-xr-x 1 monica htsteach 1.8G Oct  5 11:37 21dayV_F3_R2.fastq.gz
-rwxr-xr-x 1 monica htsteach 1.8G Sep 30 12:22 4day_K_F4_R1.fastq.gz
-rwxr-xr-x 1 monica htsteach 1.7G Sep 30 12:23 4day_K_F4_R2.fastq.gz
-rwxr-xr-x 1 monica htsteach 1.6G Sep 30 12:23 4day_K_F5_R1.fastq.gz
-rwxr-xr-x 1 monica htsteach 1.5G Sep 30 12:24 4day_K_F5_R2.fastq.gz
-rwxr-xr-x 1 monica htsteach 1.5G Sep 30 12:25 4day_K_F6_R1.fastq.gz
-rwxr-xr-x 1 monica htsteach 1.4G Sep 30 12:25 4day_K_F6_R2.fastq.gz
-rwxr-xr-x 1 monica htsteach 2.1G Sep 30 12:26 4day_V_F1_R1.fastq.gz
-rwxr-xr-x 1 monica htsteach 2.1G Sep 30 12:27 4day_V_F1_R2.fastq.gz
-rwxr-xr-x 1 monica htsteach 2.7G Sep 30 12:28 4day_V_F2_R1.fastq.gz
-rwxr-xr-x 1 monica htsteach 2.6G Sep 30 12:29 4day_V_F2_R2.fastq.gz
-rwxr-xr-x 1 monica htsteach 1.8G Sep 30 12:30 4day_V_F3_R1.fastq.gz
-rwxr-xr-x 1 monica htsteach 1.7G Sep 30 12:30 4day_V_F3_R2.fastq.gz
drwxrwsr-x 2 monica htsteach   24 Oct  5 11:21 fastqc_reports
[monica@cod3 raw_data]$
```

Start fastqc and trim_galore

- Step 1 – evaluate data from sequence provider
- Step 2 – trim data if necessary (assume yes)
- Step 3 – evaluate trimmed data output

Start fastqc and trim_galore

- Step 1 – evaluate data from sequence provider
- Choose one sample-set in /raw_data (R1 & R2)
- Copy these to your home area
- fastqc takes ~5 min / sample

```
cp  /data/RNAseq/raw_data/<sampleR1> ~
cp  /data/RNAseq/raw_data/<sampleR2> ~
```

```
module load fastqc
```

```
fastqc <inputfile.gz>
```

Running fastqc on a single input file. Writes to current directory

Start fastqc and trim_galore

- Step 2 – trim data
- Takes about 45 minutes

```
module load fastqc #if new session
module load trim-galore

trim_galore \
--fastqc --gzip \
--length 40 --paired \
<R1 file> \
<R2 file>
```

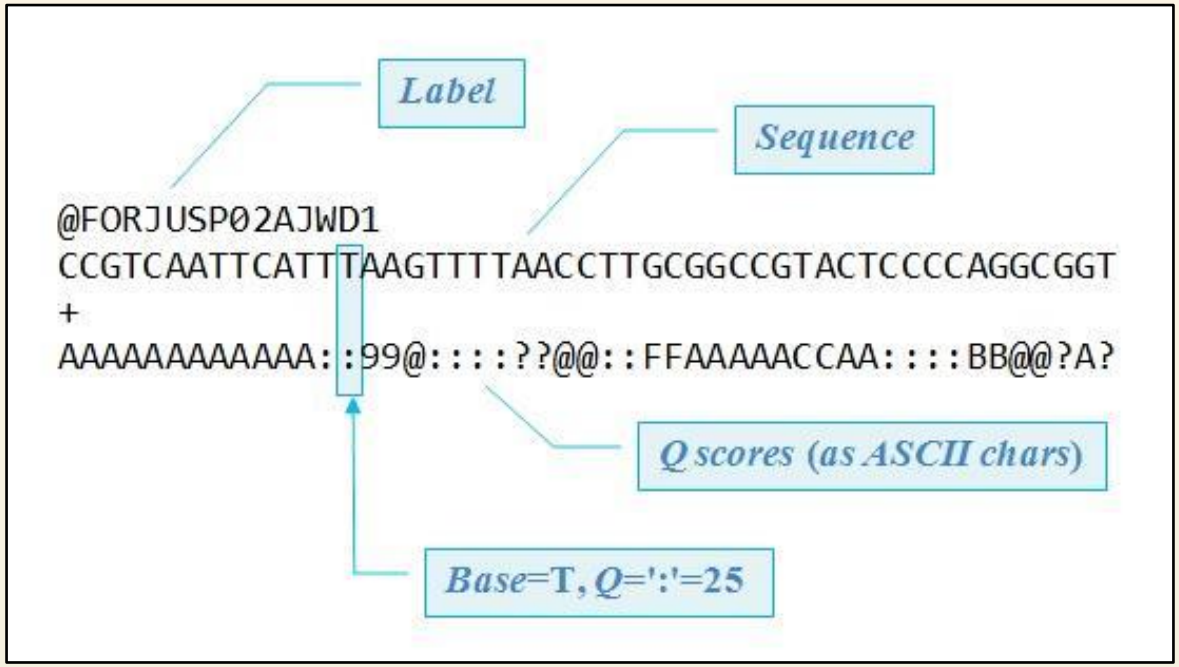
Running trim-galore on two paired compressed files, retaining sequences of 40 bp length and giving a fastqc report on output. Default quality of 20 and default stringet adapter removal

Short lecture
– Sequence evaluation / trimming and their effects on transcriptome assemblies

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

This is the fastq format

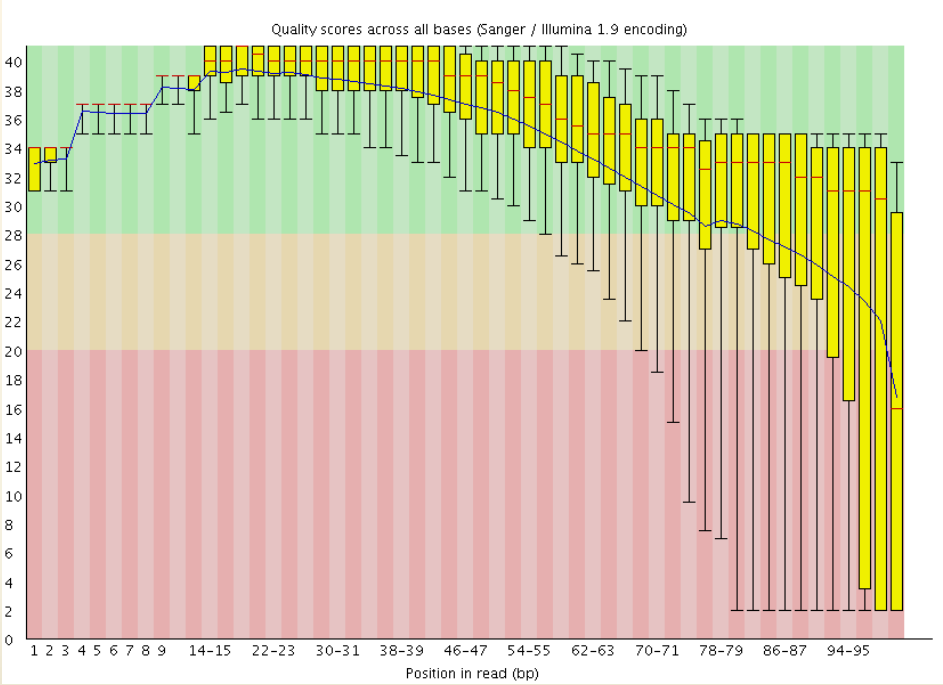


- Beware! Different sequencing technologies uses different quality encodings - ex : might correspond to different qualities
- Be sure to know which quality encoding your sequence provider use!

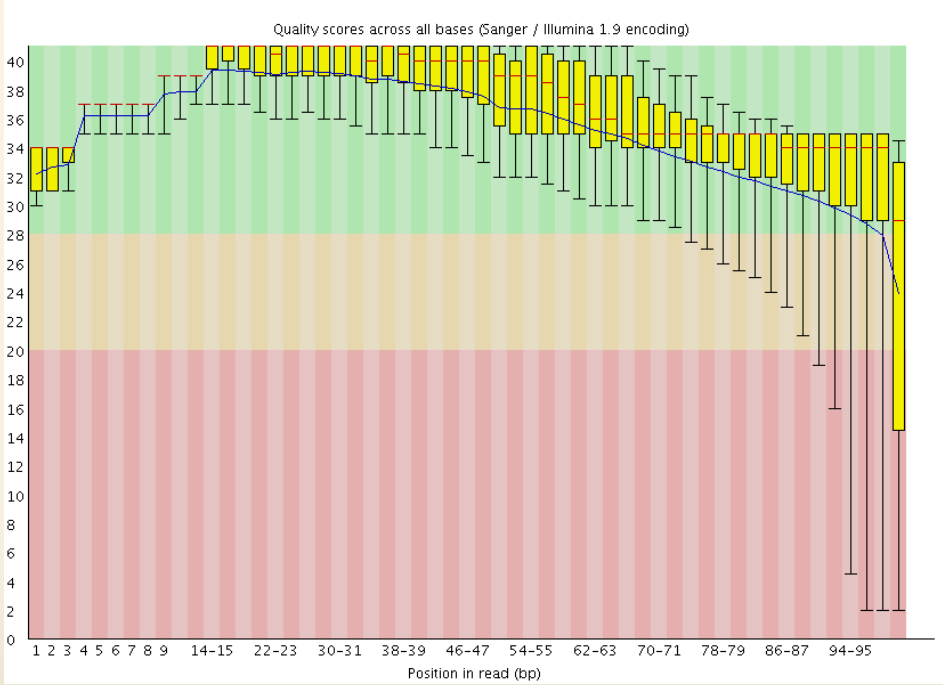
Raw sequence evaluation

- 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

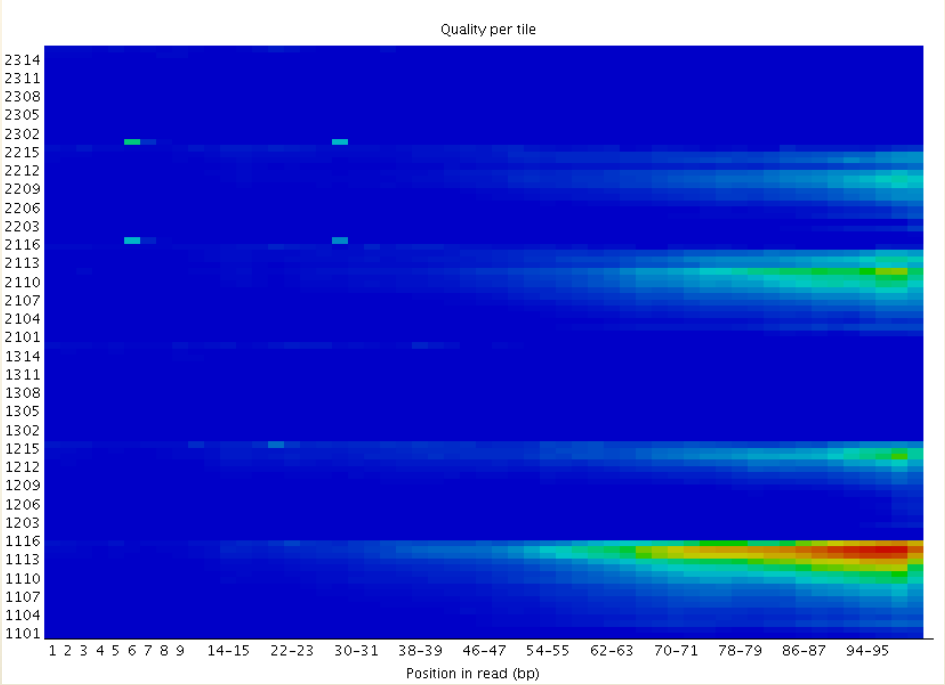


Read 1

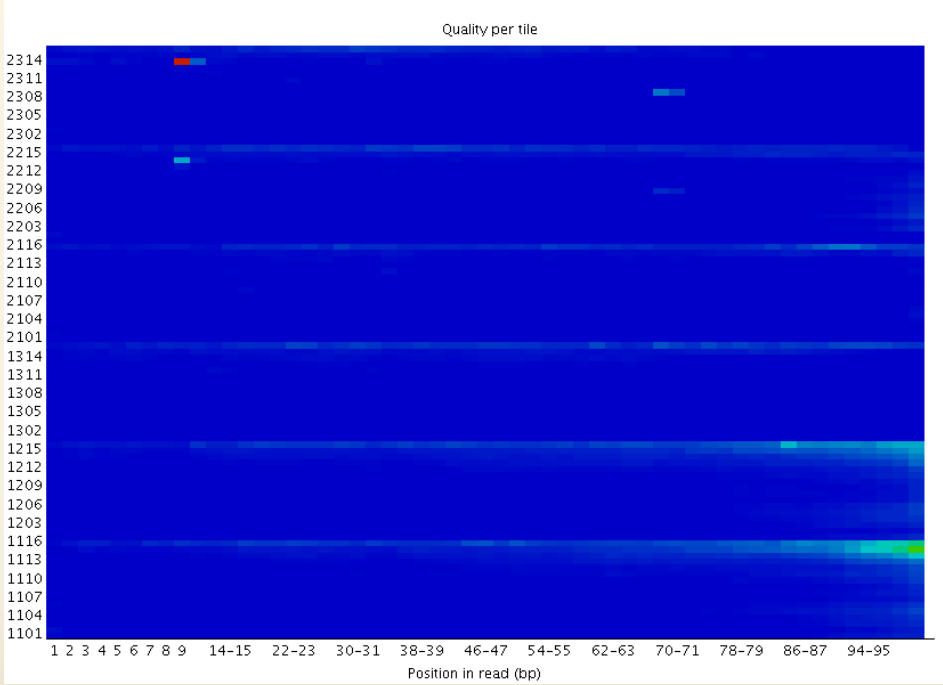


Read 2

Check the fastqc reports II

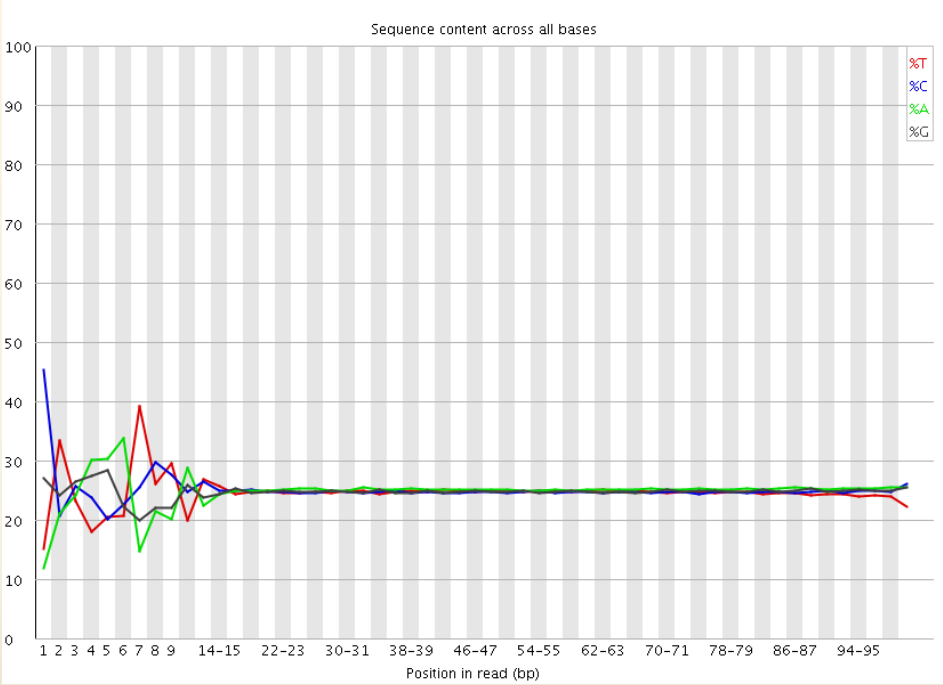


Read 1

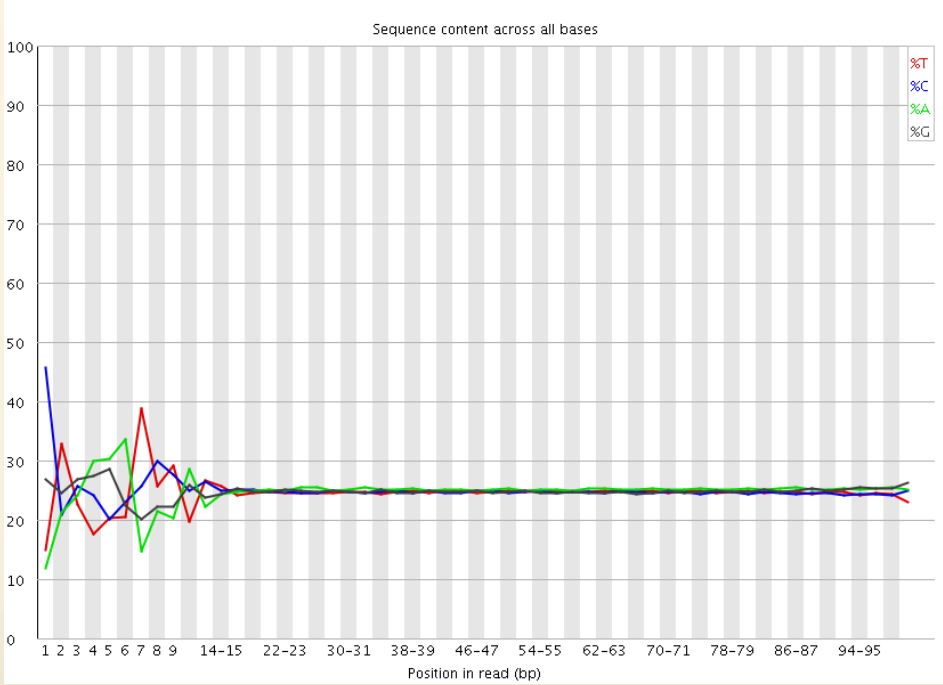


Read 2

Check the fastqc reports III

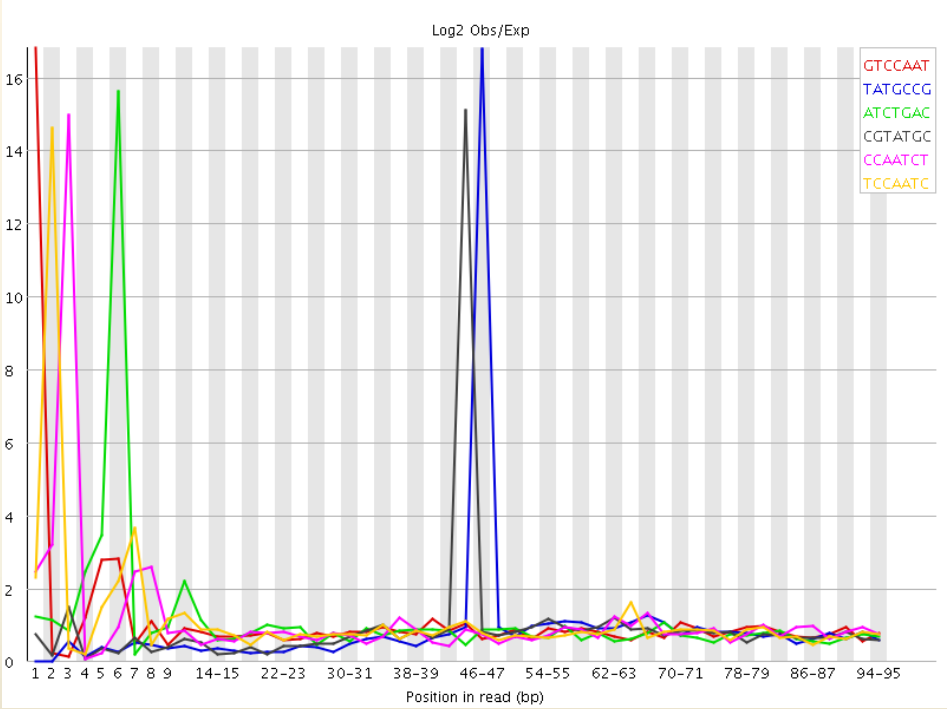


Read 1

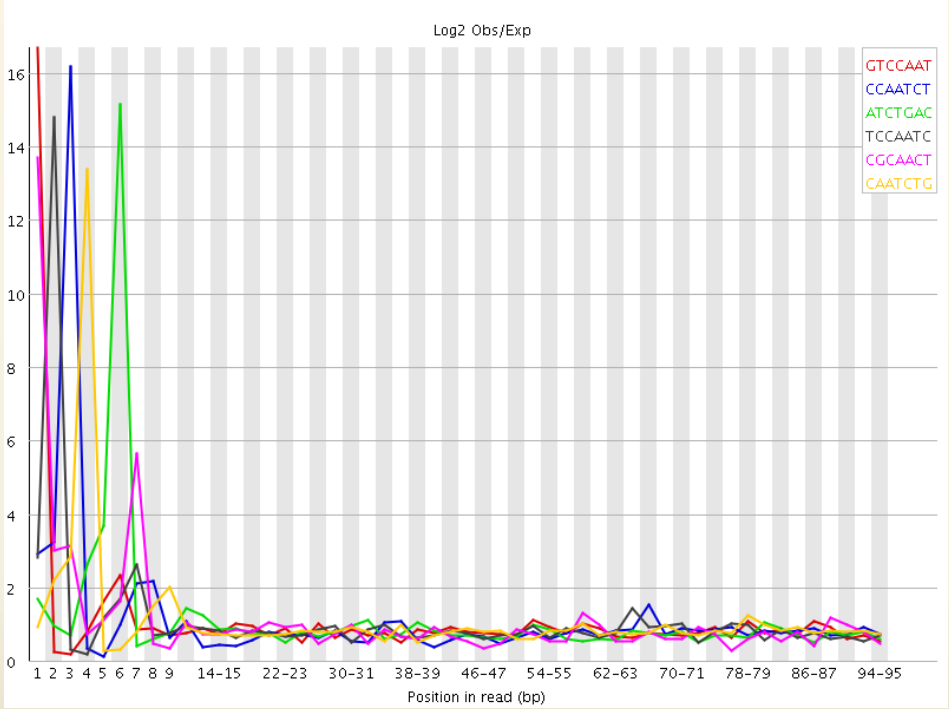


Read 2

Check the fastqc reports IV



Read 1



Read 2

Why trim your sequences?

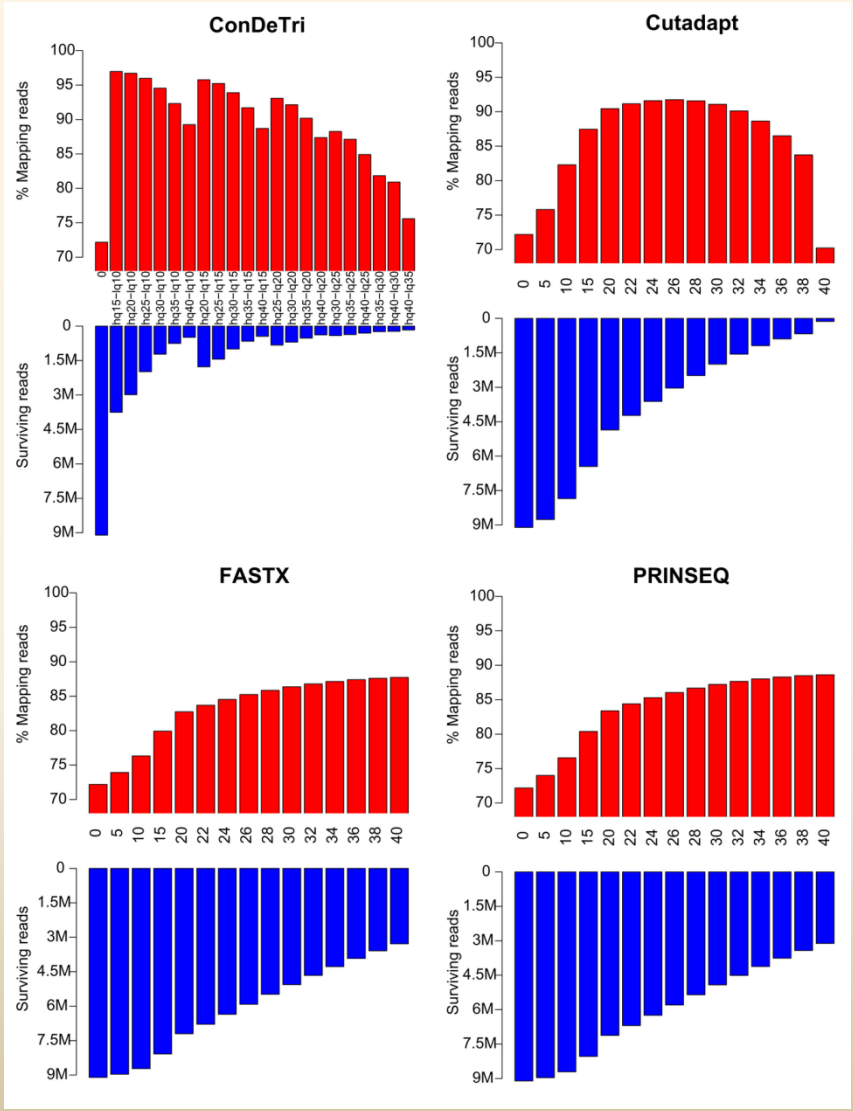
It improves read quality which further improves assembly accuracy and efficiency.

Different trims

- 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

Trimming pipelines

- There are a multitude of programs
- Differences lie in:
  - Specify adapters or use conserved part of all Illumina adapters
  - Trims on quality and/or adapters
  - Trims from single end or both ends

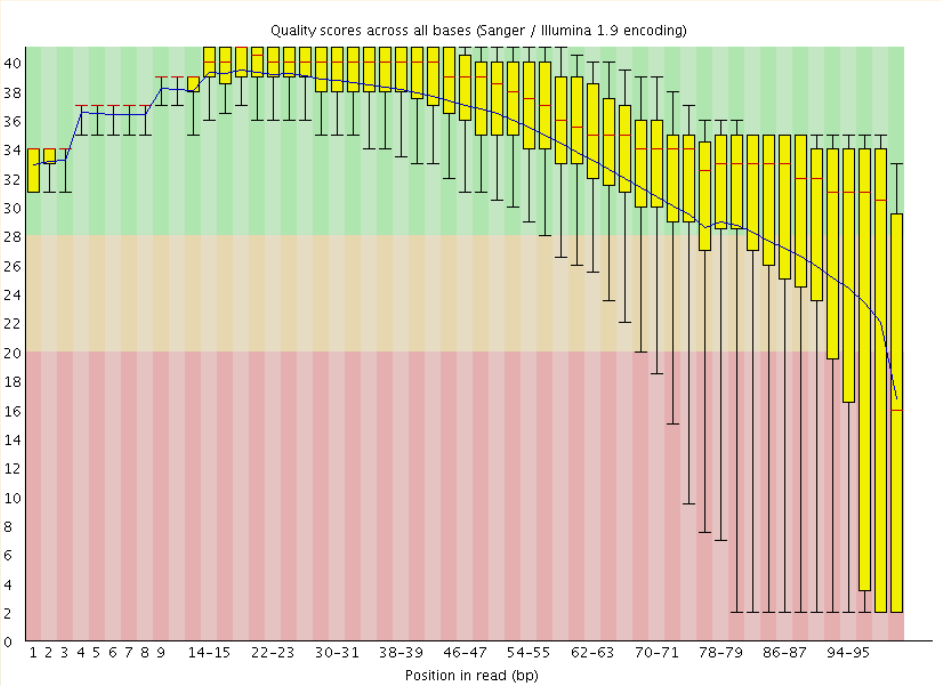


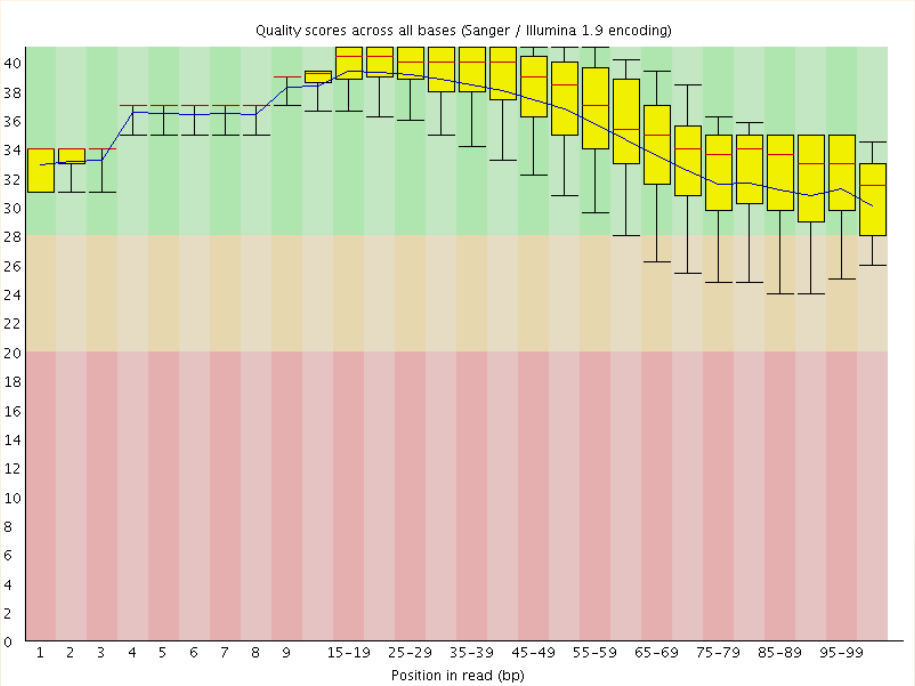
PMID: 24376861

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

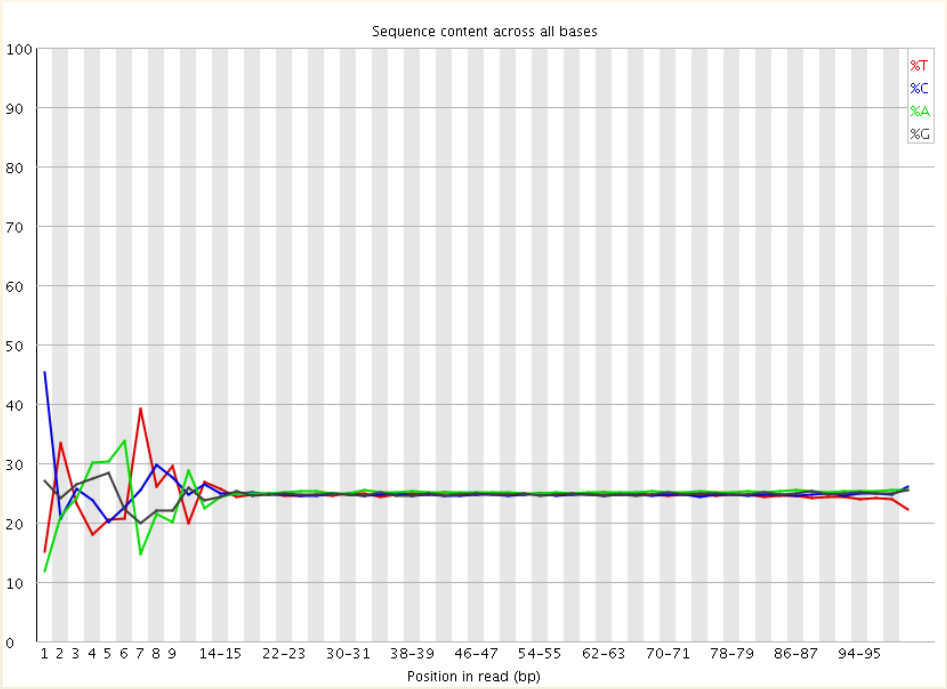
What to trim II

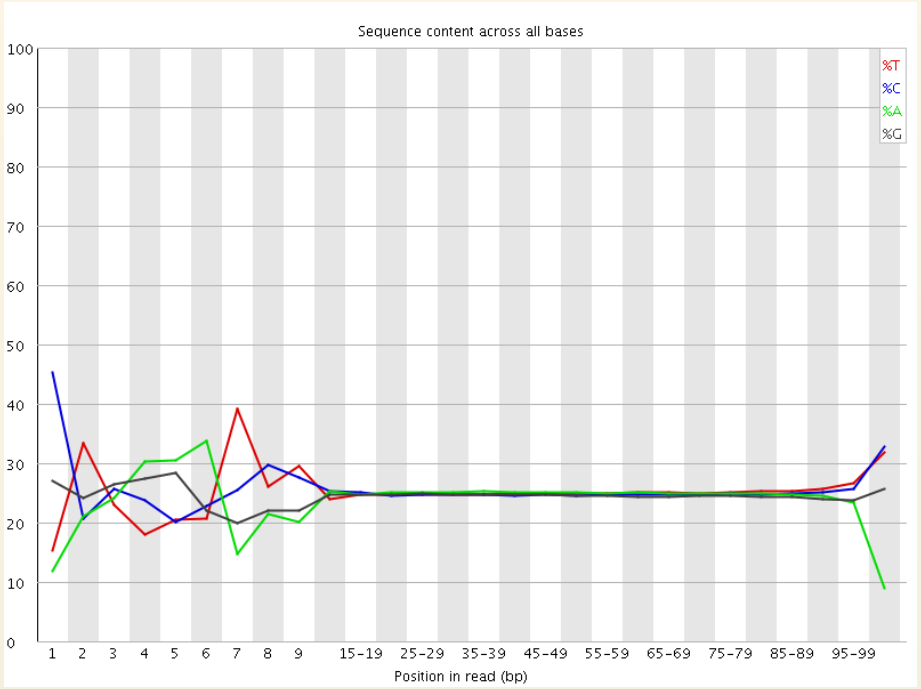




- Per base quality pre and post end quality trim on R1
- Improves assembly accuracy and efficiency

What to trim III

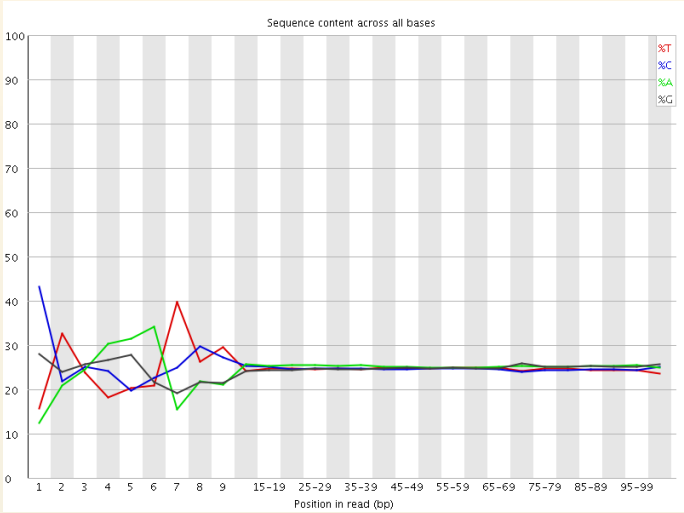


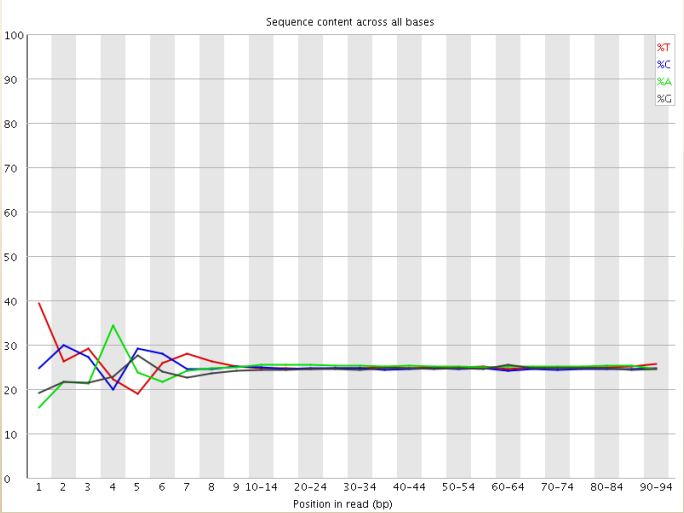


- Adapters / sequencing primers
- Still signs of bias in graphs post trim but not mentioned in the report

What to trim IV

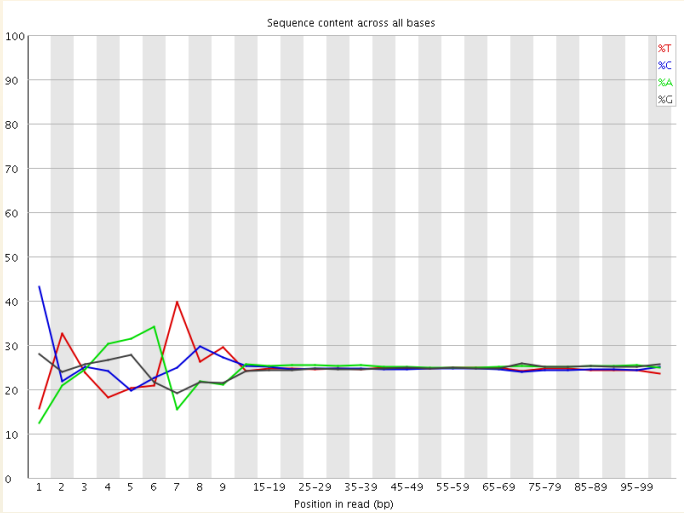
- You avoid adapter/primer transcripts in the assembly
- Mapping is quicker however many mappers can work with sequences containing adapters

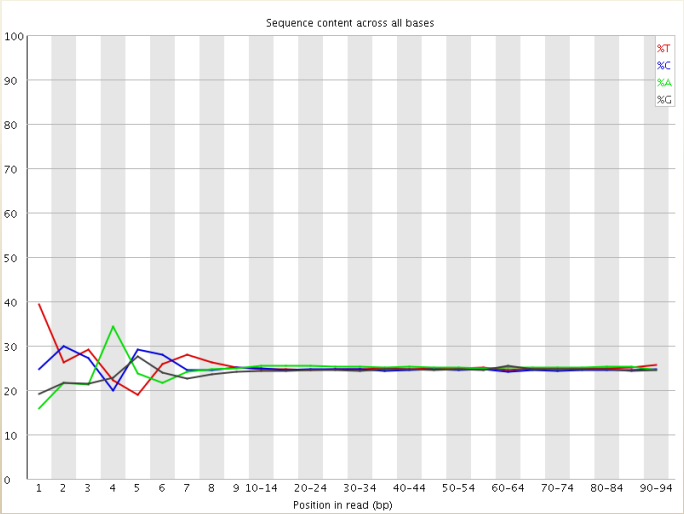




What to trim V

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

Low complexity / “identical” reads?

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

Effects of read trimming

Assembly

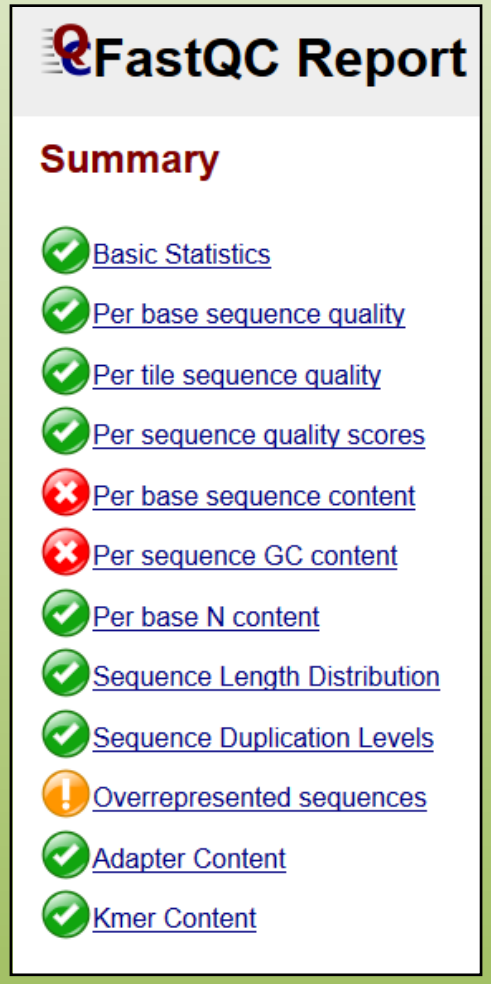
- Trimming-transcriptome completeness trade-off
- Trade-off between computation time and lower precision
- Trimming w/ sequence correction will lead to loss of rare transcripts

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

Evaluate the samples

- Move the folders made by fastqc to your computer (rsync, winscp…)
- Open the fastqc_report.html files
- Compare pre and post trim outputs



Evaluate the samples 2

- 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 and are they removed in the post trim sample?
- How much data did you loose during trimming?

Short lecture
– How to make a transcriptome assembly

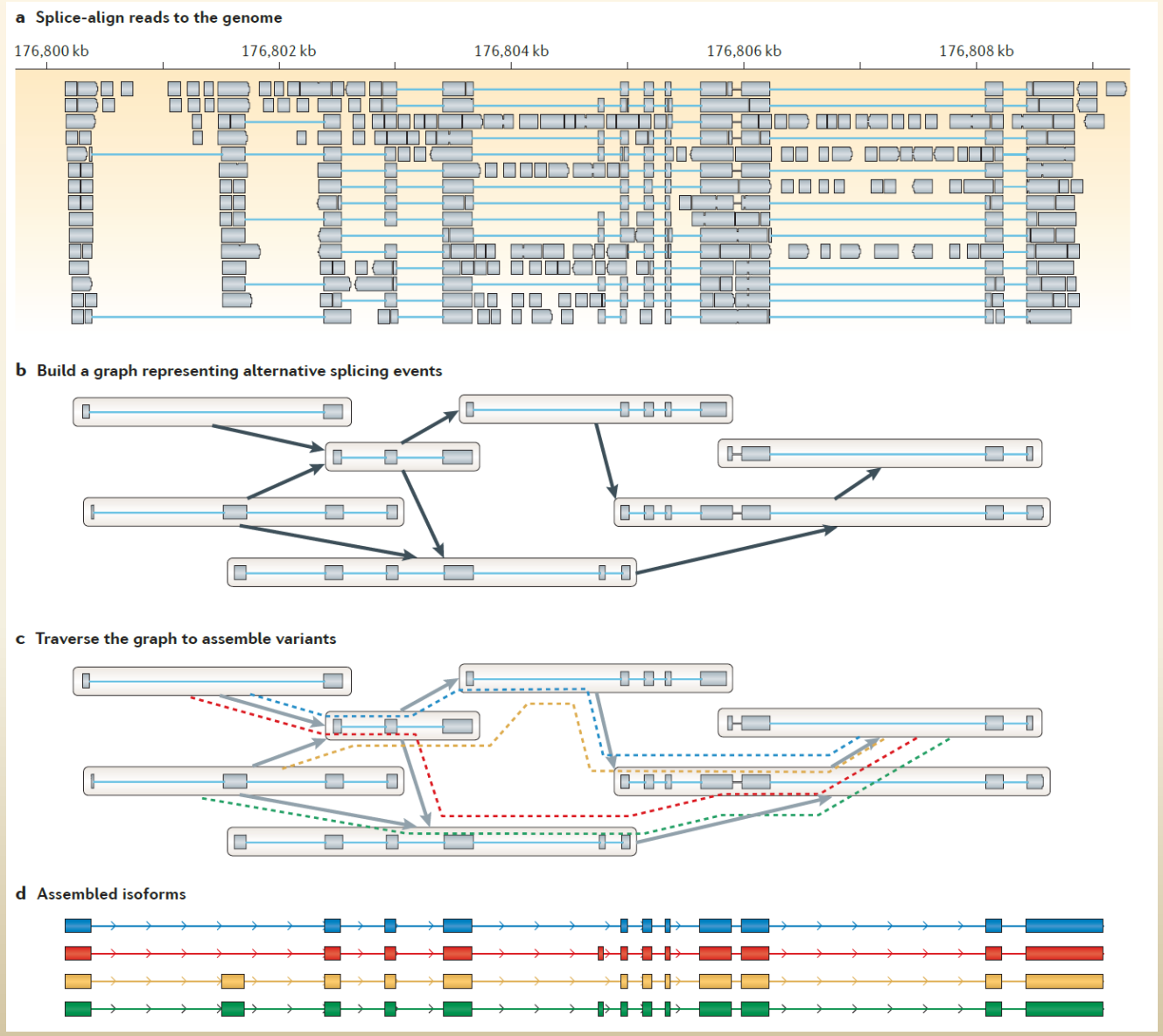
What to consider I

- What do I want?
- What will I use it for?
- Which resources are available for your species (very closely related species)?
- What kind of data do I have?
- 2n or xn ploidy?

The strategies I

- Reference based (*ab initio*)
  - Maps RNAseq reads back towards reference genome and builds transcripts
  - Needs a certain amount of splice-junction covering reads
- *De novo* (with/without genome guiding)
  - Assembly of RNAseq reads only
  - Guided: reads are clustered according to chromosome / scaffold prior to assembly
- Mixed approach
  - Merging several assemblies to one

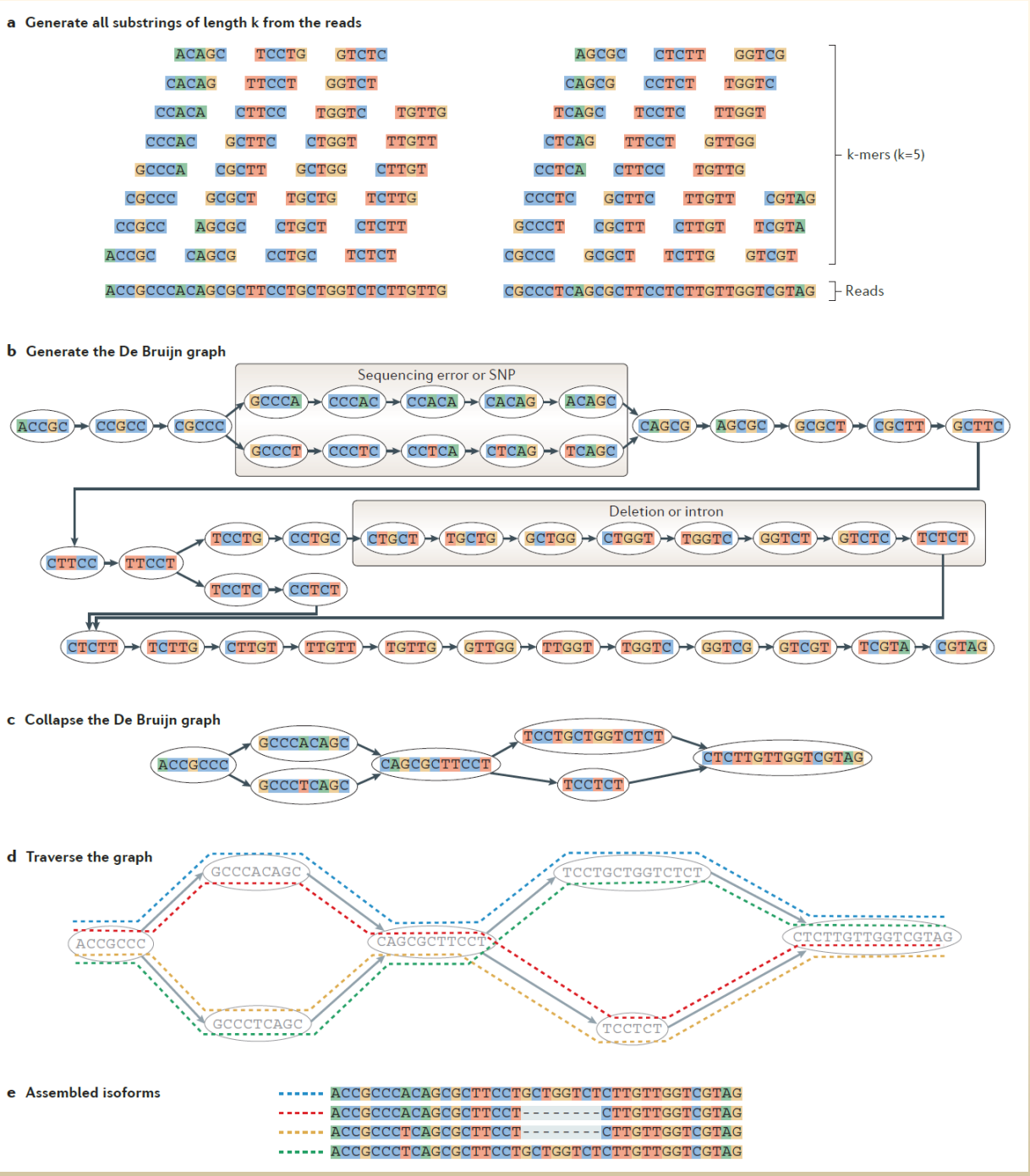
Reference based



Reference based II

- Benefits
  - Time efficient / single computer job
  - Requires less coverage of samples
  - Artefacts / contaminations does not align to the reference
  - Low abundance / novel isoforms are resolved
- Complications
  - Depends on quality of reference
  - Gene dense organisms
  - Higher eukaryotes with complex splice variants – especially *trans*-splicing
  - Software settings may discard splice variants / transcripts
  - Different treatment of multi-mapping reads

De novo


a Generate all substrings of length k from the reads
ACAGC TCCTG GTCTC
CACAG TTCCT GGTCT
CCACA CTTCC TGGTC TGTTG
CCCAC GCTTC CTGGT TTGTT
GCCCA CGCTT GCTGG CTTGT
CGCCC GCGCT TGCTG TCTTG
CCGCC AGCGC CTGCT CTCTT
ACCGC CAGCG CCTGC TCTCT
AGCGC CTCTT GGTCG
CAGCG CCTCT TGGTC
TCAGC TCCTC TTGGT
CTCAG TTCCT GTTGG
CCTCA CTTCC TGTTG
CCCTC GCTTC TTGTT CGTAG
GCCCT CGCTT CTTGT TCGTA
CGCCC GCGCT TCTTG GTCGT
k-mers (k=5)
ACCGCCCACAGCGCTTCCTGCTGGTCTCTTGTTG
CGCCCTCAGCGCTTCCTCTTGTTGGTCGTAG
Reads
b Generate the De Bruijn graph
Sequencing error or SNP
Deletion or intron
c Collapse the De Bruijn graph
d Traverse the graph
e Assembled isoforms
ACCGCCCACAGCGCTTCCTGCTGGTCTCTTGTTGGTCGTAG
ACCGCCCACAGCGCTTCCT--------CTTGTTGGTCGTAG
ACCGCCCTCAGCGCTTCCT--------CTTGTTGGTCGTAG
ACCGCCCTCAGCGCTTCCTGCTGGTCTCTTGTTGGTCGTAG

De novo II

- Benefits
  - No reference needed
  - Detects all transcripts (coverage dependent)
  - No knowledge/prediction of splice sites needed
  - Complex splice patterns can be resolved
- Complications
  - Requires lots of computing power
  - Requires more coverage to resolve transcripts
  - Sensitive to read errors and artefacts / contaminations
  - Paralog (“gene copies”) resolution is an issue

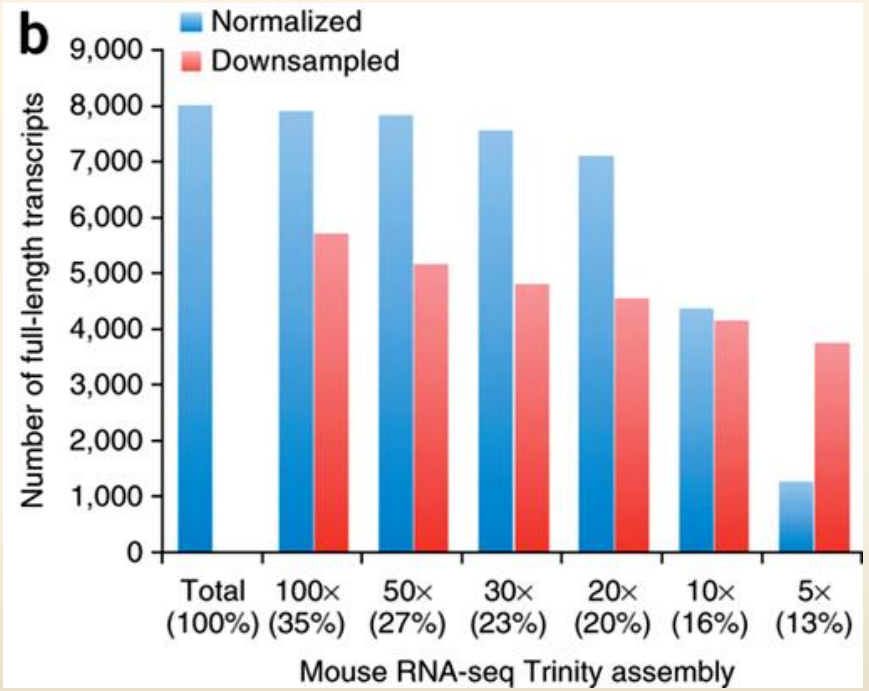
Mixed approach

- *De novo* and *ab initio* assembly concatenation
- Multiple kmer strategy

- Who benefits from a mixed approach?
  - Gene dense eukaryotes
  - Polyploid species
  - When the aim is to make a really good reference transcriptome

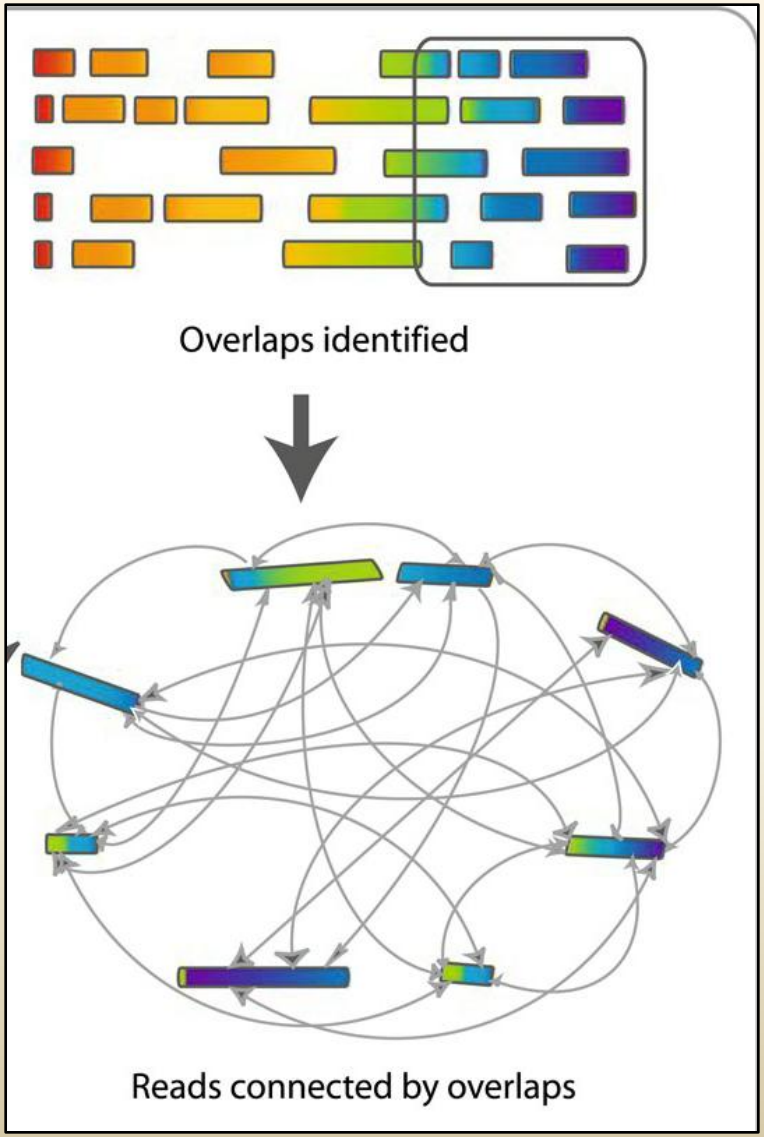
How to make it I

- Use all available data
- Consider normalization to shorten computation time and increase chances of resolving less abundant transcripts



How to make it II

- Consider the assembly algorithm
  - Large datasets with short reads benefits from using De bruijn graph based assembly programs (more than a hundred million read pairs)
  - Small datasets with short reads benefits from using Overlap-Layout-Consensus (OLC) based assembly programs



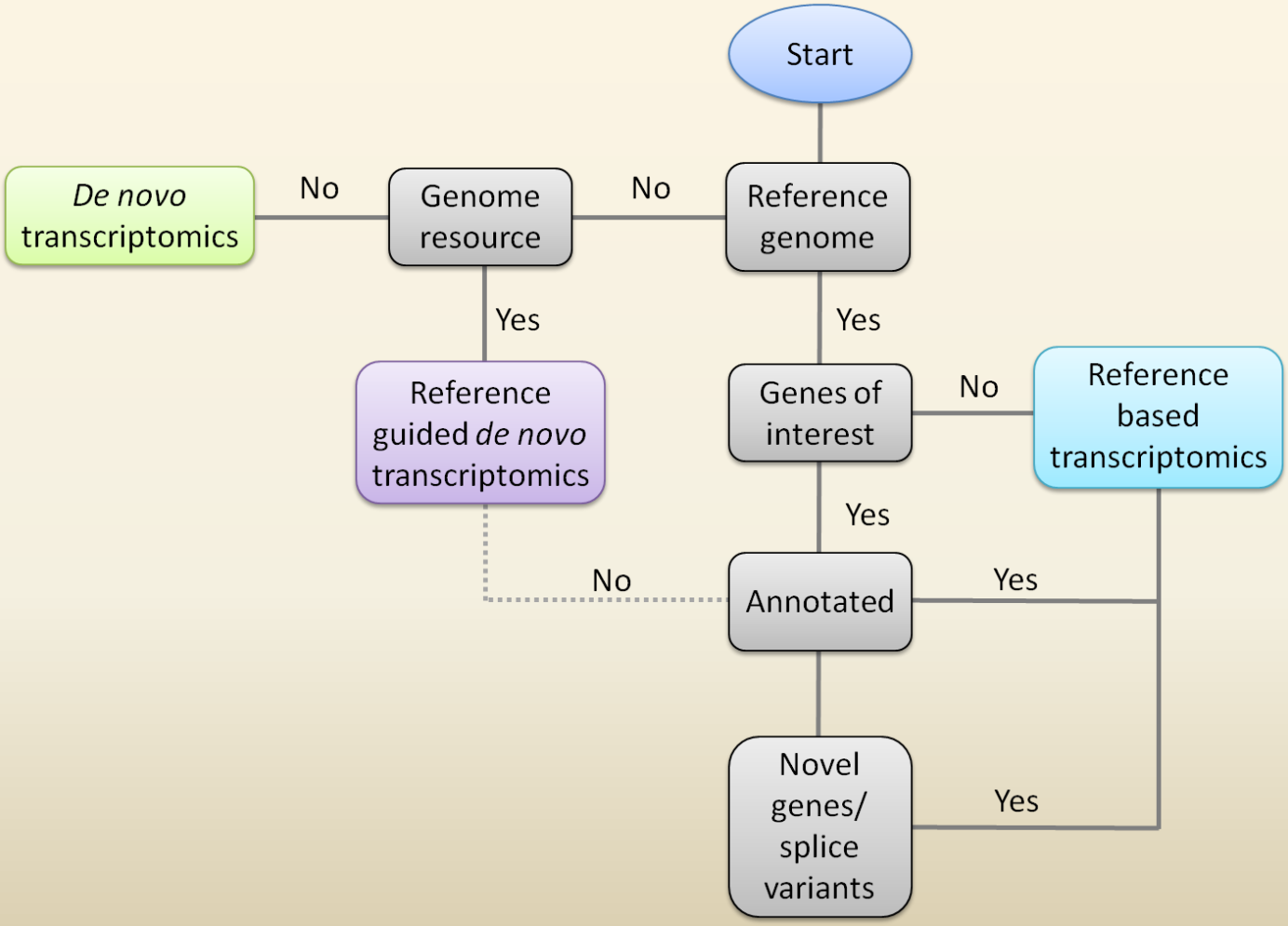
How to make III

- Do you have a an organism known to be gene dense with overlapping UTRs?
  - Select a program with options like jaccard clip to improve algorithm
  - The cost is more computation time so do not use it unless necessary

What to consider – INFBIO case

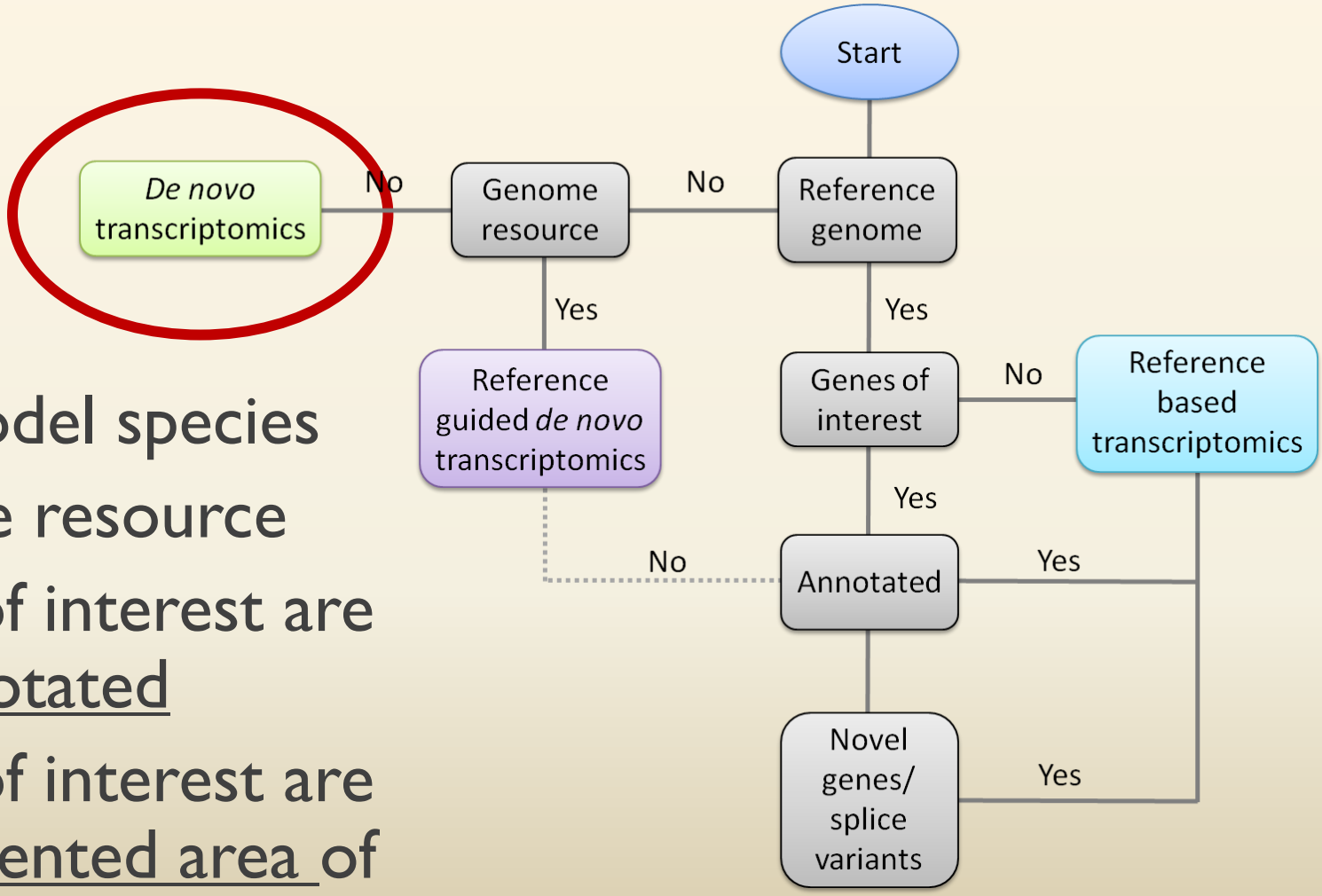
- What do I want? – transcriptome
- What will I use it for? – differential expression
- Which resources are available for your species (very closely related species)? – genome
- What kind of data do I have? – Illumina PE
- 2n or ploidy? – 2n

Choosing our strategy



Choosing our strategy

- Non-model species
- Genome resource
- Genes of interest are <u>not annotated</u>
- Genes of interest are <u>in fragmented area</u> of genome



Trinity assembler

- Trinity is the best single parameter *de novo* RNA assembly pipeline available
- Good on splice variants, full length transcripts and resolution of lowly expressed transcripts
- Contains tools to help with visualizations

Starting a Trinity *de novo* assembly I

```
module load samtools/samtools-1.1
module load trinityrnaseq
module load perlmodules/5.10_2
module load gcc/5.2.0
ulimit -s unlimited
```

Trinity needs several programs to work properly.

ulimit changes how much resources an individual can use on the node.

Trinity will run for a few days

Starting a Trinity *de novo* assembly II

```
Trinity \
--seqType fq \
--left \
<R1_val_1.fq.gz> \
--right \
<R2_val_2.fq.gz> \
--max_memory 20G \
--CPU 4 \
--bflyCPU 2 \
1> trinity_denovo.out 2>trinity_denovo.err &
```

Run Trinity in ~
Allocate 20 Gb RAM
and 4 CPUs
Restrict Butterfly (java)
with 2 CPUs
Print standard out and
err to files

```
module load samtools/samtools-1.1
module load trinityrnaseq/trinityrnaseq-2.0.6
module load perlmodules/5.10_2
module load gcc/5.2.0
ulimit -s unlimited

Trinity \
--seqType fq \
--left \
<R1_trimmed> \
--right \
<R2_trimmed> \
--max_memory 20G \
--CPU 4 \
--bflyCPU 2 \
1> trinity_denovo.out 2>trinity_denovo.err
```

Choose a node to work on.

Protect command using screen

```
screen bash -l
```

Load all modules

Run Trinity

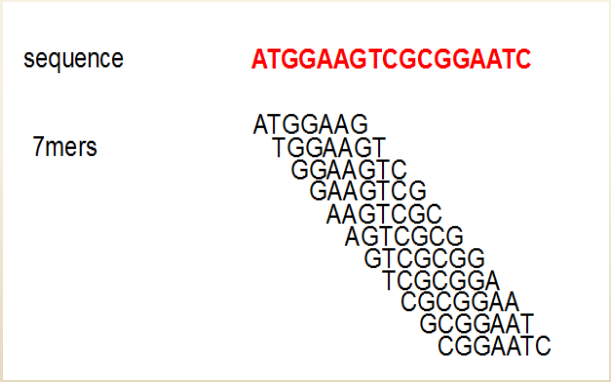
16 students on cod4

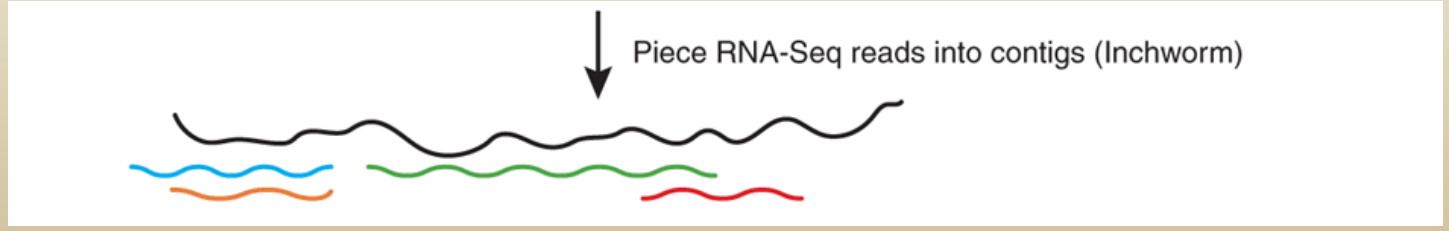
6 students on cod1

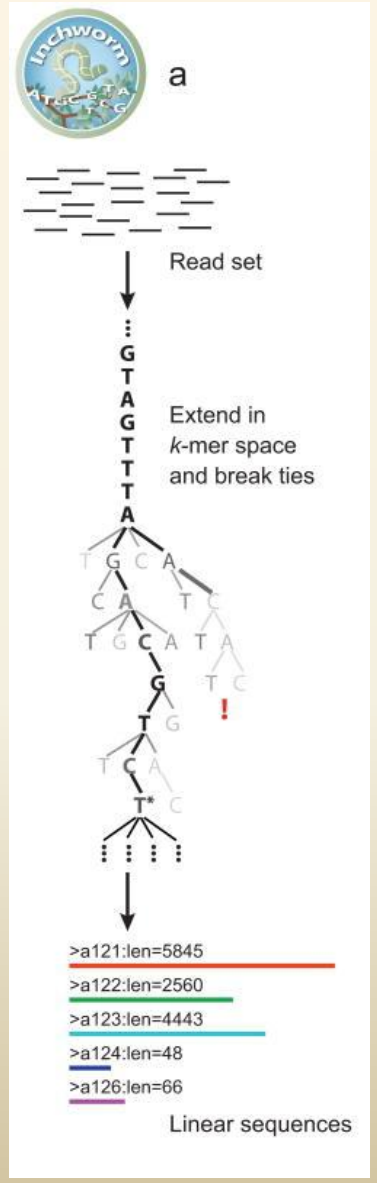
8 students on cod 3

Trinity pipeline - Inchworm

- It employs a greedy kmer based approach to reconstruct the best representative for a transcriptionally active region (often full-length dominant isoform).

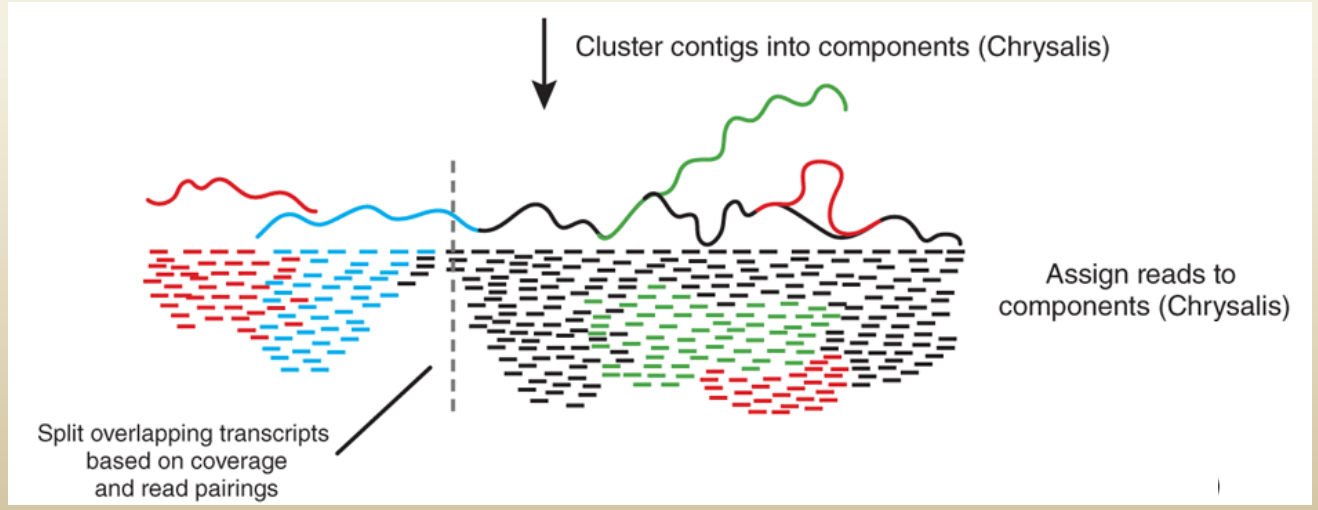


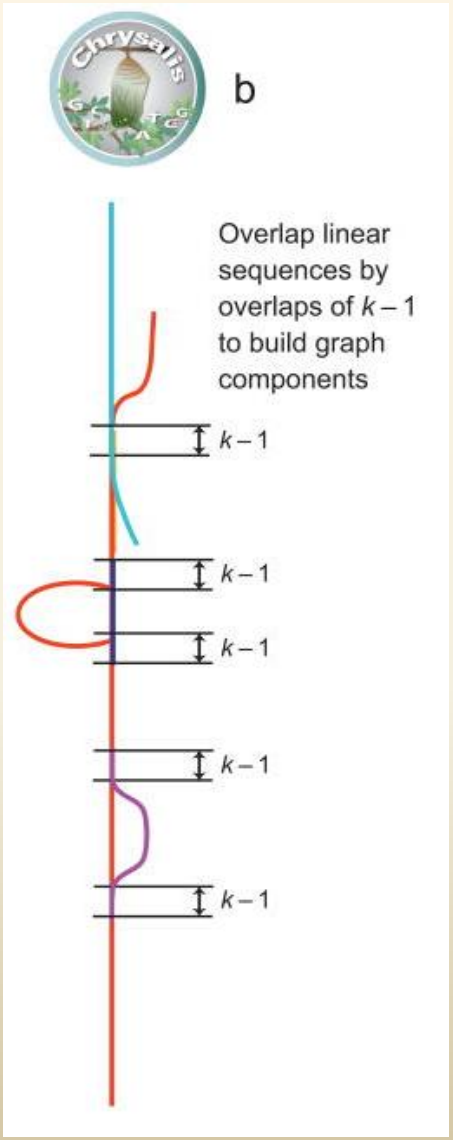




Trinity pipeline

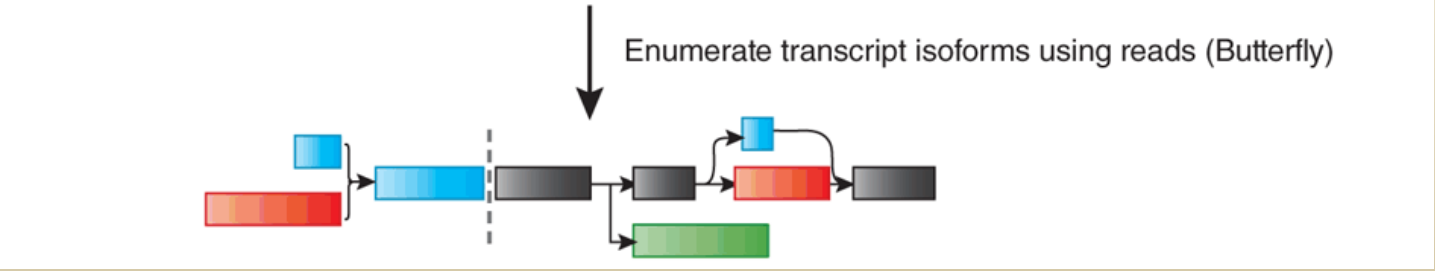
- Chrysalis clusters Inchworm related contigs into components (alternatively spliced variants)
- Then a De Bruijn graph is made for each component

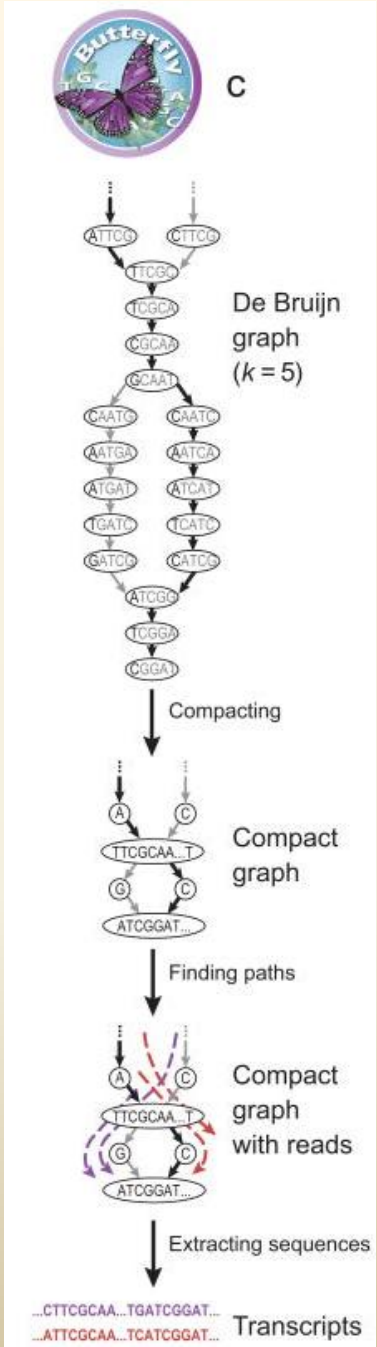




Trinity pipeline

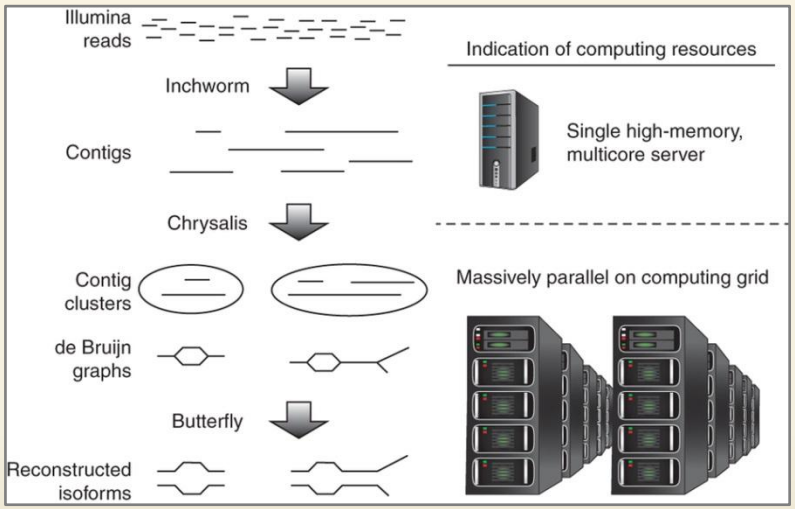
- Butterfly analyzes the paths taken by reads and read pairings in the graphs and reports all plausible transcripts including splice variants and transcripts derived from paralogs (duplicated genes)



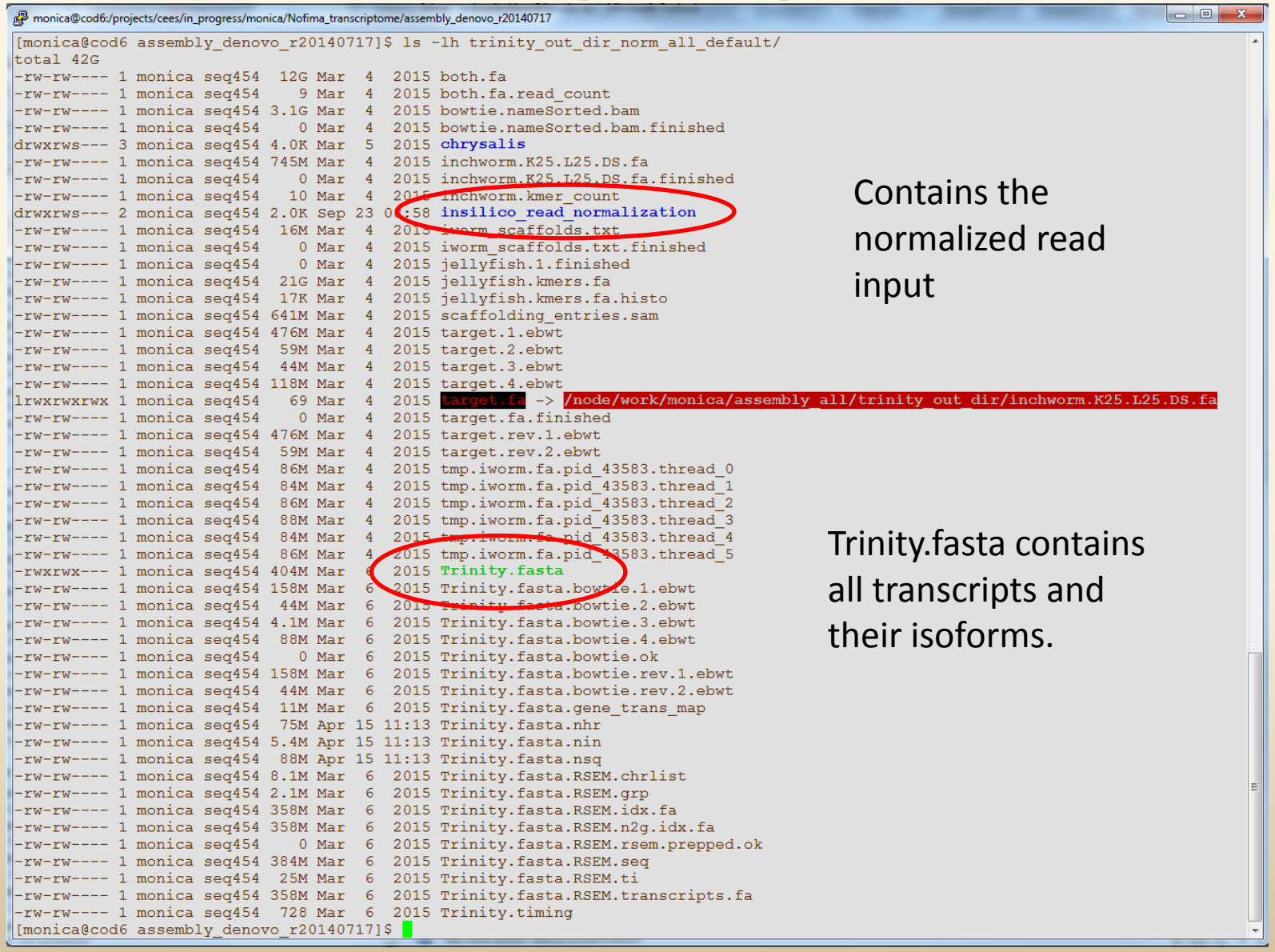


Trinity computation requirements

- Assembly algorithms require large amounts of memory
- 2/3rds of Trinity is parallelized to save computation time
- Estimate at least 1 week of trial/error/final computation
- Remember to calculate memory/time requirements before starting!
  - 1Gb RAM / million reads
  - ½ - 1 hour / million reads



Trinity output



Contains the normalized read input

Trinity.fasta contains all transcripts and their isoforms.

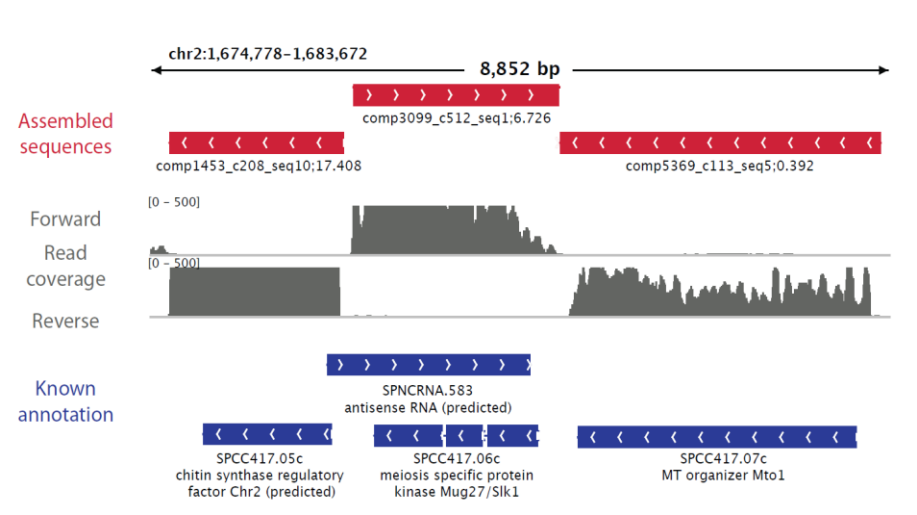
What to expect

- Significantly more transcripts than predicted in the same or closely related species!
- Low coverage over splice junctions, sequencing errors and heterozygosity restricts full-length transcript reconstruction

```
################################
## Counts of transcripts, etc.
################################
Total trinity 'genes':  320520
Total trinity transcripts:      468626
Percent GC: 47.31

########################################
Stats based on ALL transcript contigs:
########################################

        Contig N10: 3657
        Contig N20: 2645
        Contig N30: 2042
        Contig N40: 1597
        Contig N50: 1235

        Median contig length: 459
        Average contig: 784.28
        Total assembled bases: 367534825


#####################################################
## Stats based on ONLY LONGEST ISOFORM per 'GENE':
#####################################################

        Contig N10: 3360
        Contig N20: 2278
        Contig N30: 1635
        Contig N40: 1193
        Contig N50: 880

        Median contig length: 382
        Average contig: 634.85
        Total assembled bases: 203483069
```

How to make it comparable

- Trinity comes with:
  - Full length estimation (BLAST based)
  - Abundance estimation (simple expression analysis)
- Consider mapping transcripts towards a reference genome if available



What to evaluate

- Assembly with 320 520 genes in a marine fish with estimated gene number ~22 000
- Full length analysis
  - ORFs more than 80 % full length: ~8 800
- Abundance estimation
  - 30 779 genes expressed 2 fold or more
  - Also check that the majority of (un-normalized) reads map back towards the assembly

Filtering away contigs?

- There is likely a small percentage of assembly artefacts increasing the number of contigs
- You may use full length analysis and abundance estimation to filter your assembly
  - Filtering on contig size: what about small transcripts like interleukins?
  - Filtering on expression: what about low constitutive/house keeping genes
  - Filtering on full length: some splice variants might lack exons on purpose

The only benefit of filtering before analyses is saving a few hours of computing time!

The Trinity.fasta

```
>c115_g5_i1 len=247 path=[31015:0-148 23018:149-246]
```

- `c115` refers to the read cluster
- `g5` the "gene"
- `i1` isoform number 1
- `len` is the length of the isoform
- `path` refers to how Butterfly traversed through the nodes in the De bruijn graph.

How to work with Trinity.fasta

- A large text file not suitable for text editors
- Use UNIX commands to handle it / extract information
- There are also various fasta tools installed on the cluster

```
>c10_g1_i1 len=265 path=[446:0-66 15:67-264]
TGGCGCTACACCCGGACAGGAAGTATTGTCGGCCCAGGATGGTGTTACCTGAGGCGCTCT
TCCCACATCCCGTGTTTCCTATCAGGATCAGACTCAGCTCTTCTTCTGGATCATTCCCTC
CCATGTCTGTTGACCACGGCCCCTTGGCTCCCAGGTTAACCTTTGGTCTGCCTTGGCTTT
GGCGGTAATGGCGAAGTGGCCATGTTTTCTAGCCTTTAACTTATCTGTTGGCTCTGTAGG
TGAACTGCATGGGGTCCAGGAGAGG
>c11_g1_i1 len=217 path=[195:0-216]
CGGCAACGGCGGCGGCAGCGGCGTTTGGGATGGTCTGGTTGCTGGTTCAAGGCCTTGCCG
CCTCTGGGCAGAGCAGTGCATGCTGGGAGCGGACAGGTCCAGGGGTGGCGTGGAACCACA
GCTGTCCAGGTCCAGACCCCTACGTTATGTTCCTCCACGGCGAGGCGCTCCACCGGATGG
ATCTGGACGGAGGGAACCACCGCCGGGTGGTGTCGGG
>c12_g1_i1 len=203 path=[181:0-202]
AGGAAAGAACGAAAGAAAGAGAGAAAGAGAAAAAGGGAGAGAGAGATTAATTTCTGTGTT
AGCATGACCCAAAGAGAAAGTGTATTGATGAACAAATTGAGGAGGATTGGTGTGTGAATT
GGCAACATCCAAACCAAGATACAAAGGCTGATACCAAACCATTGGCAAACAATTATAGGG
AATGGACTTGAGCTTGCTGAGAG
>c21_g1_i1 len=213 path=[1:0-212]
GTCCTCCTCAAACTGGTAACCCTGGAACAGCCAAGTAAAGGCAGCGTGACAATCTCCTCT
CCAAGGCAACAAGTGTTCTTGACACATTGTCGCTTGTCATACCATGTGTTGGATGTCACA
CTGGGATGAAAGTATAAAGGTGAAAACTACATTTCCCAAAACAGTGGTTCTAGGAGACAT
TTAAAACAATACACAGTGAGTGGTACTGACGGG
>c23_g1_i1 len=222 path=[200:0-221]
AGAGGCTGCCATGTTTCCTGCCATGTTCCTACAGCAGGCCAGGACAGACAAACAACTCTG
GAGGGGACGCTTTTATTCTTTAATCATAACCACTGCCGTGCTTTCAAGCGATGACCTTGA
GTTTTACAGACTTAATAACTAAACAGAACTCATGAATCTGTCGAGGAGCTATCACTTGAC
ATTCGGCTACTCAATGCTGCAACAGACGATGACATCATTTAG
>c24_g1_i1 len=212 path=[190:0-211]
GTTTTATATAGTTGAACCATTTCTATTTGTGCTGCATGTTGTGGTTATGTTGCTTGTTAT
GTTTATTCTATTCTTTCTGGCTTGAGCTGCGCAACTGTAAAAAAATATTTCCTCCTAGAT
CCATAAAGTAAGTTCAGAAAGATAACATGCAGGCAACTCATGACAAGCGGATTATATCCA
CGTTTATCTCATGTGGTTTGAACAAGGCCGCT
>c29_g1_i1 len=274 path=[1:0-273]
GATCCCTGCTTACCGGGCTCACCCTGACGGGGGTCAAGACAGAAAGAGGTTGGATCACTG
AACGGAAGGGGAGCAGGACAAGGGCCCGACGGGAAAGACTAAACAGTGTAAACATAACAT
GTATGTTGTGAATGTATCTGTGCCTGAATCATCACTTAATACAAAGGTTTCATTTTCAGC
AGCTTATGCACACGTTAAGCCGATAGCTAGCTGACGAGGCGCCGCGCGACGCCCCGAGAG
GGAAGCTACTCACTCCGGGGCCGGGAATGCCGGG
```

Copy Trinity.fasta

- Copy

`/data/RNAseq2/assembly/Trinity_complete.fasta`

- to your home directory

```
cp /data/RNAseq2/assembly/Trinity_complete.fasta ~
```

Extract info from Trinity.fasta I

- Find the gene(s) with the most isoforms in Trinity_complete.fasta (`/data/RNAseq2/assembly/`)

```
>c115_g5_i1 len=247 path=[31015:0-148 23018:149-246]
```

  - Hint: `awk`
  - Hint 2: specify delimiter

Extract info from Trinity.fasta III

- How many transcripts (incl. isoforms) are there in Trinity_complete.fasta?
  - Hint `grep`

Extract info from Trinity.fasta V

- Trinity comes with scripts to extract various stats

```
ls /cluster/software/VERSIONS/trinityrnaseq/\
trinityrnaseq-2.0.6/util/
```

- TrinityStats.pl provides assembly stats

```
/cluster/software/VERSIONS/trinityrnaseq/\
trinityrnaseq-2.0.6/util/TrinityStats.pl \
Trinity_complete.fasta > \
trinity_stats.txt
```

Extract info from Trinity.fasta VI

- Copy the remaining Trinity.fasta files in /data/RNAseq2/assembly to your home area
- Compare stats for the four assemblies using TrinityStats.pl

- Which assembly is the best?