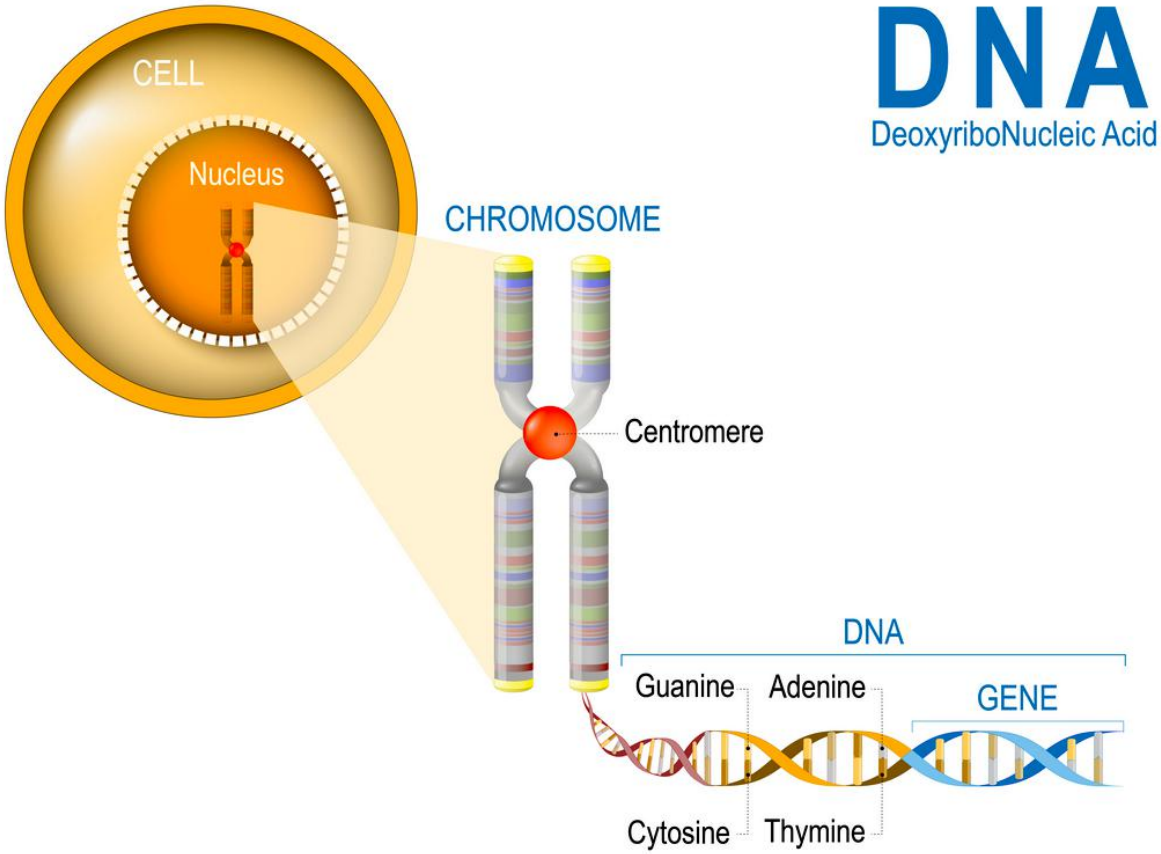
Whole Transcriptomic Analysis

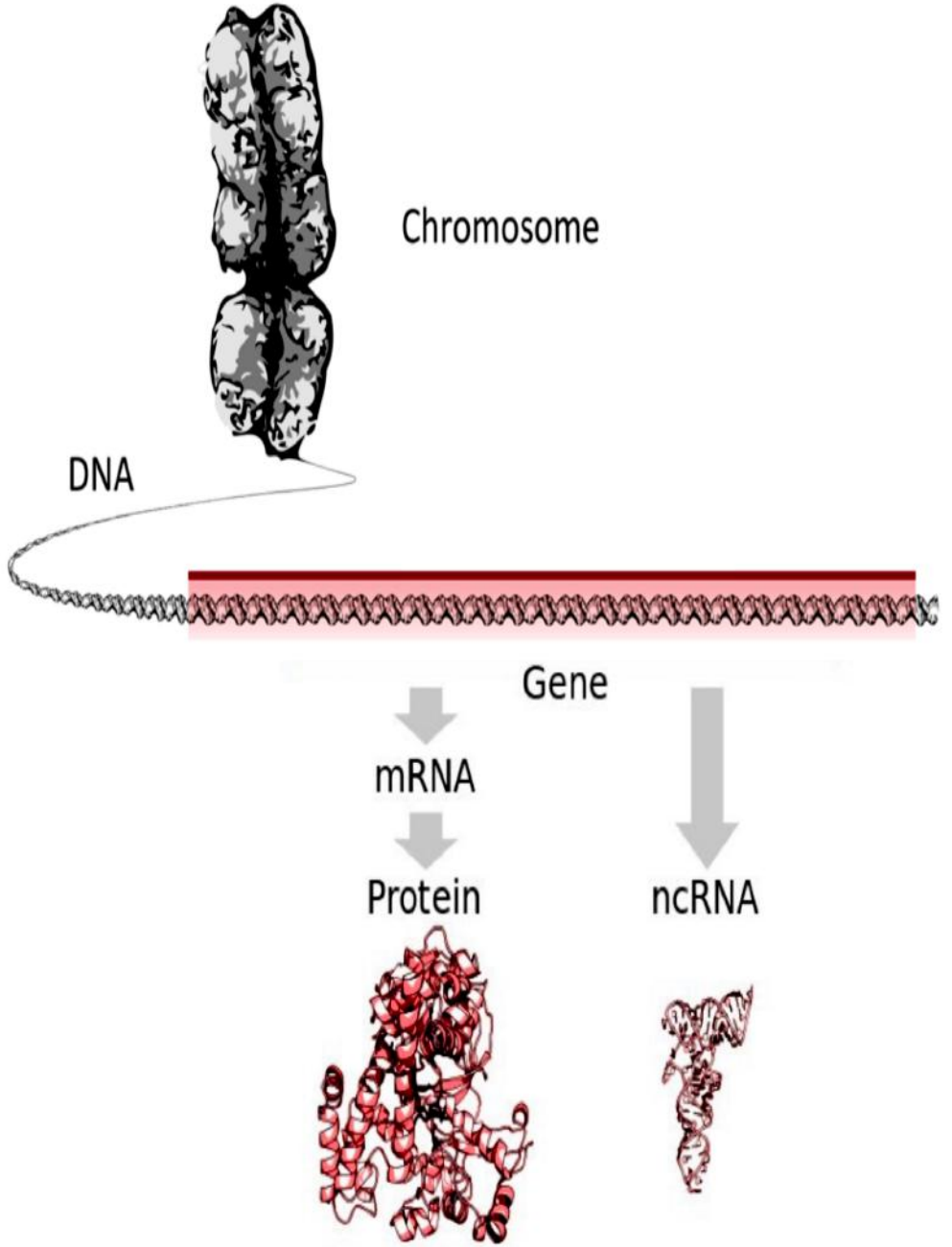
Manjula Thimma

Prof. Jesper N Tegner

DNA



DNA
DeoxyriboNucleic Acid



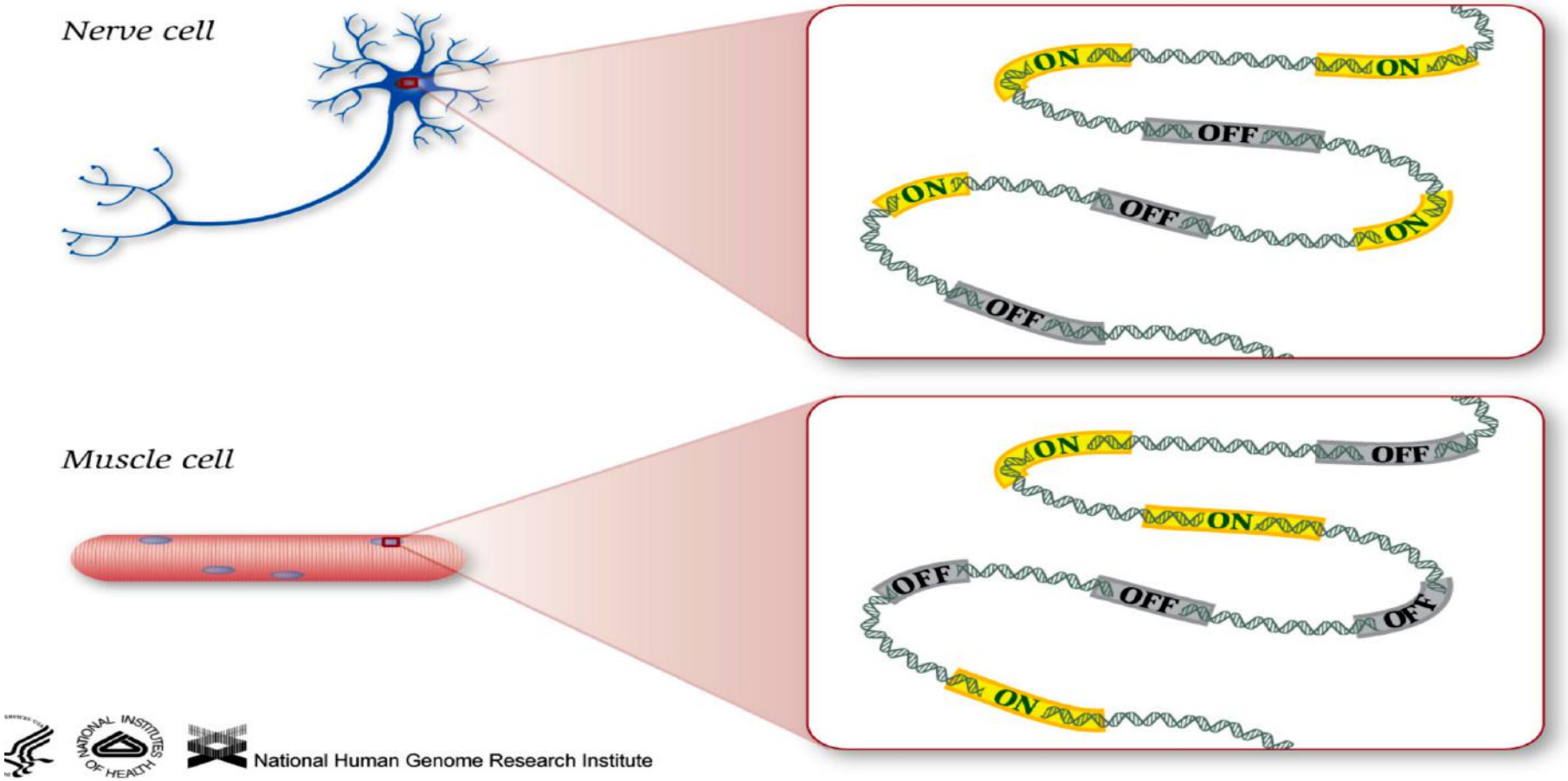
All organism contain instruction for their life in the genome; genome contains DNA ie nucleotides A. C. G, T.

Factors that determine gene expression levels

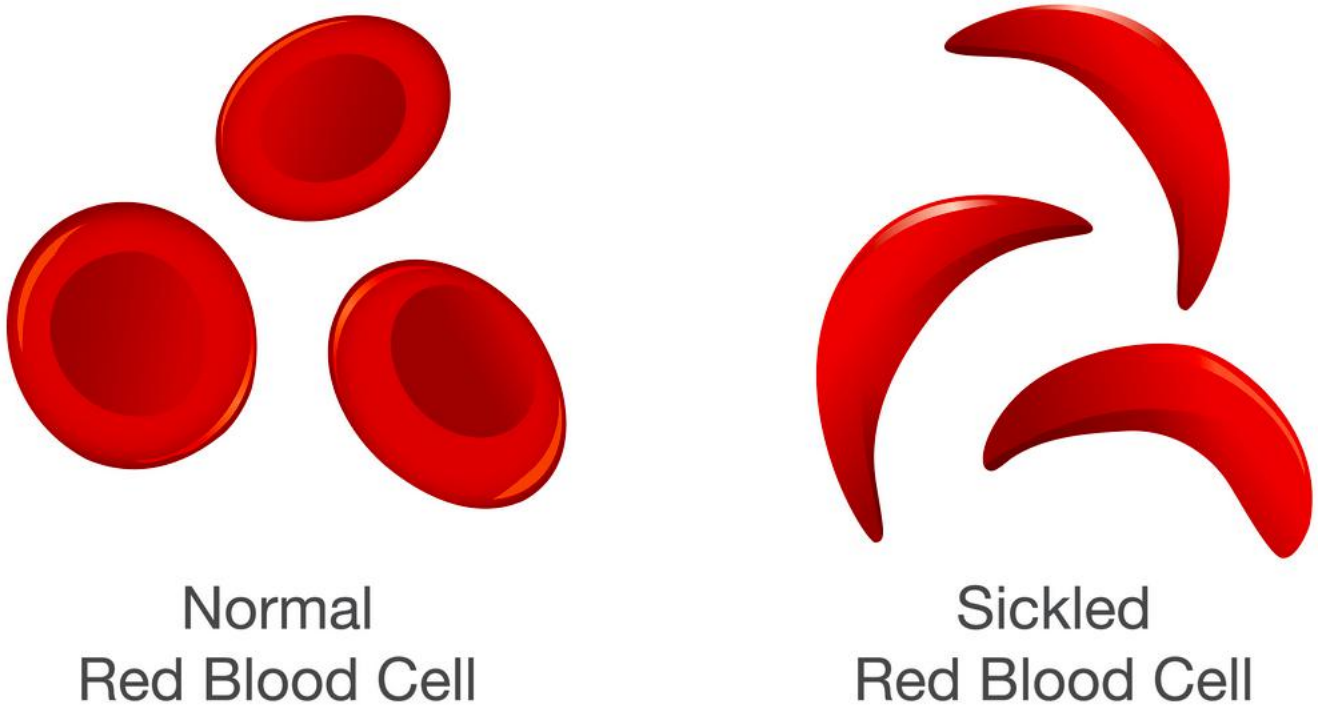
Examples:

- Cells and cell composition
- Diseases – chronic (e.g. cancer, T2D) or acute (e.g. flu)
- Infectious agents (bacteria, fungi, protozoa, virus, worms)
- Nutrient availability
- Environmental stimuli
- **Epigenetic factors**
- **Missense mutations in the genome**

Transcriptomes specific to cells

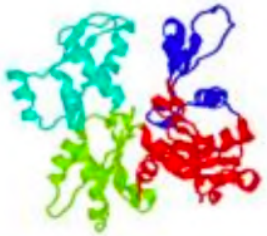


Disease specific transcriptomes

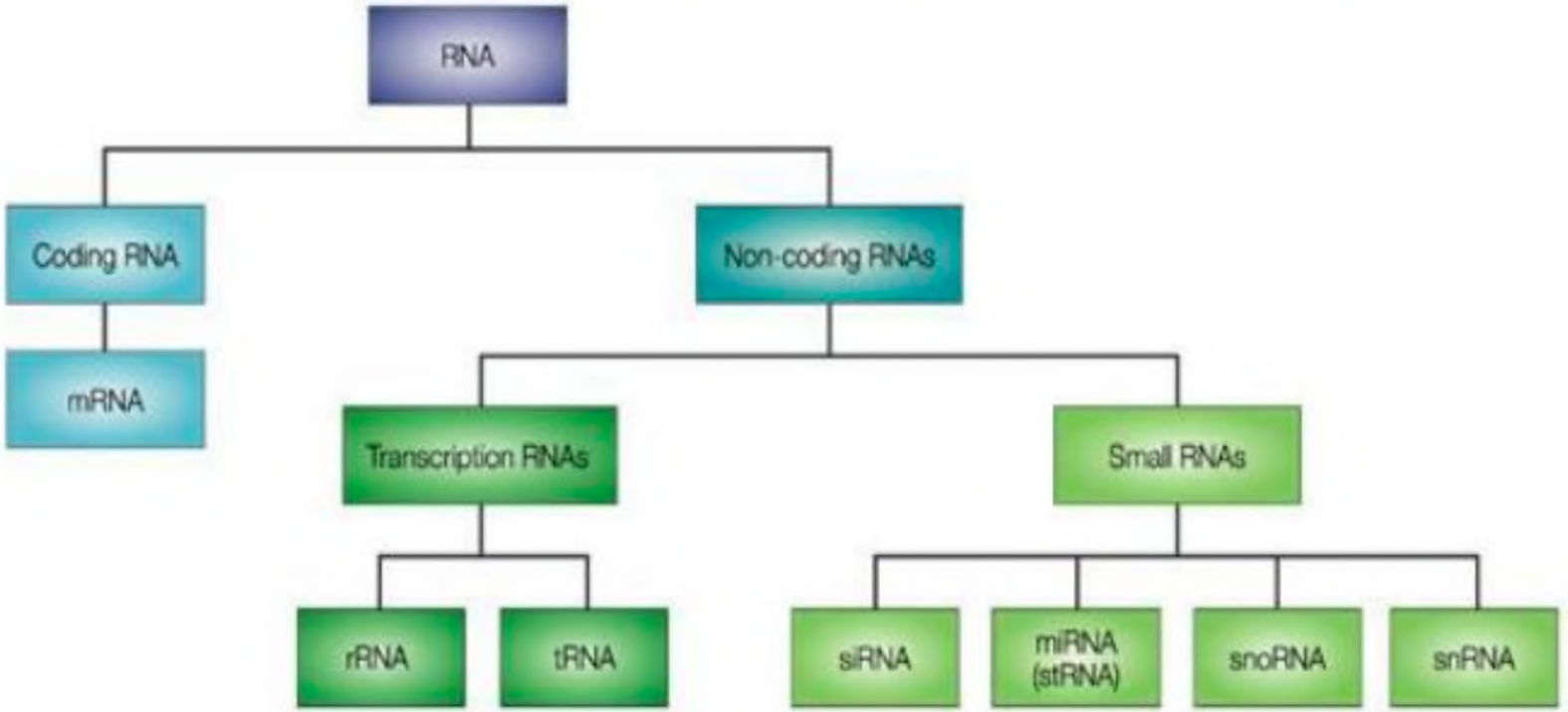


Disease can turn certain genes on or off, resulting in difference in quantity of RNA produced.

RNA-seq can help to observe whether there are significant difference in gene expression in different conditions.



RNA Types



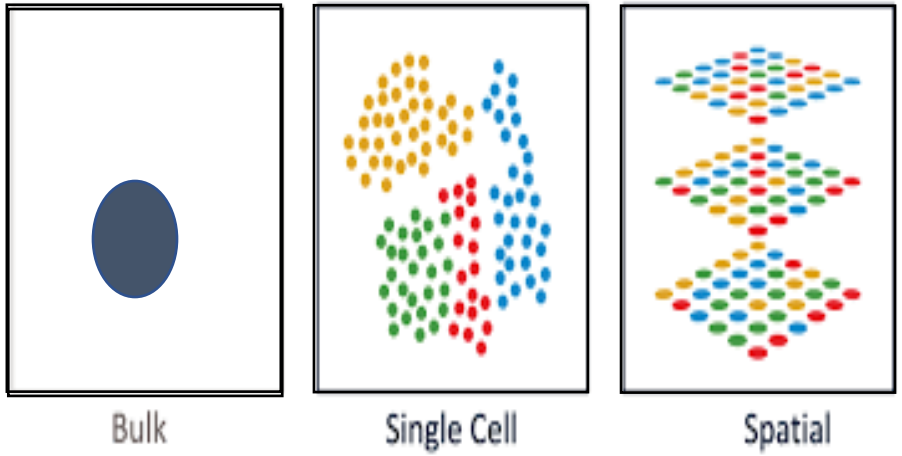
siRNA, short interfering RNA; miRNA, microRNA;
small temporal RNA stRNA; snoRNA small nucleolar RNA ;
snRNA: Small nuclear RNA.

Typical use cases for bulk RNA-seq data

- Genes/pathways that are differentially expressed
  - Between healthy and controls (diagnostic)
  - Between different molecular subtypes or stages of disease
  - Relapse vs non-relapse, metastatic vs non-metastatic, time to relapse/death (prognostic)
- Genes/pathways that are changing over time
  - Developmental stages
  - Time course experiments after acute challenge (e.g. vaccine), drug treatment or lifestyle intervention (e.g. exercise)
  - Monitoring (e.g. during pregnancy for risk of preterm)

Emerging use cases for transcriptomics

- Single cell RNA sequencing (scRNA-seq)
- Spatial transcriptomics
- Long reads using Oxford Nanopore Technology / Pacbio → direct isoform quantification
- RNA secondary structure elucidation



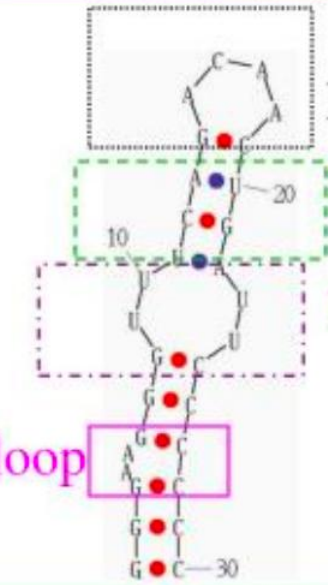
Secondary Structure Categories

Hairpin loop

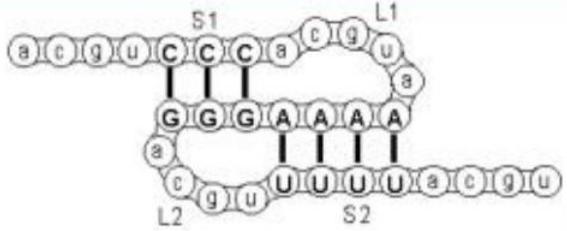
Stem

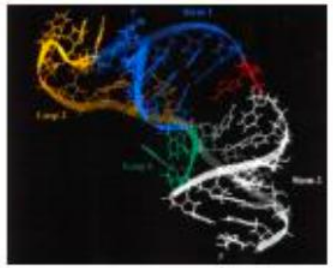
Internal loop

Bulge loop



Pseudoknots

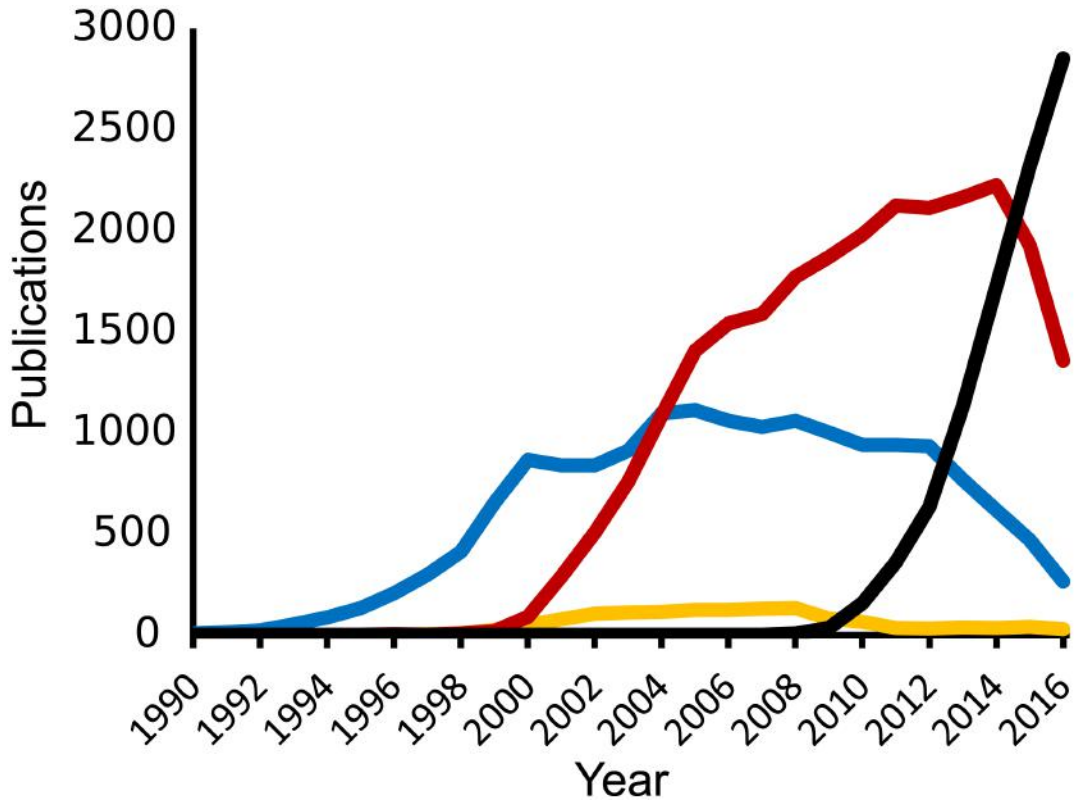




RNA-Seq questions

- What genes are differentially expressed between sample groups?
- Are there any trends in gene expression over time or across conditions.
- Which groups of genes change similarly over time or across conditions.
- What processes or pathways are important for my condition of interest?

Transcriptomic methods used over time



Published papers since 1990, referring to

RNA sequencing (black)

RNA microarray (red)

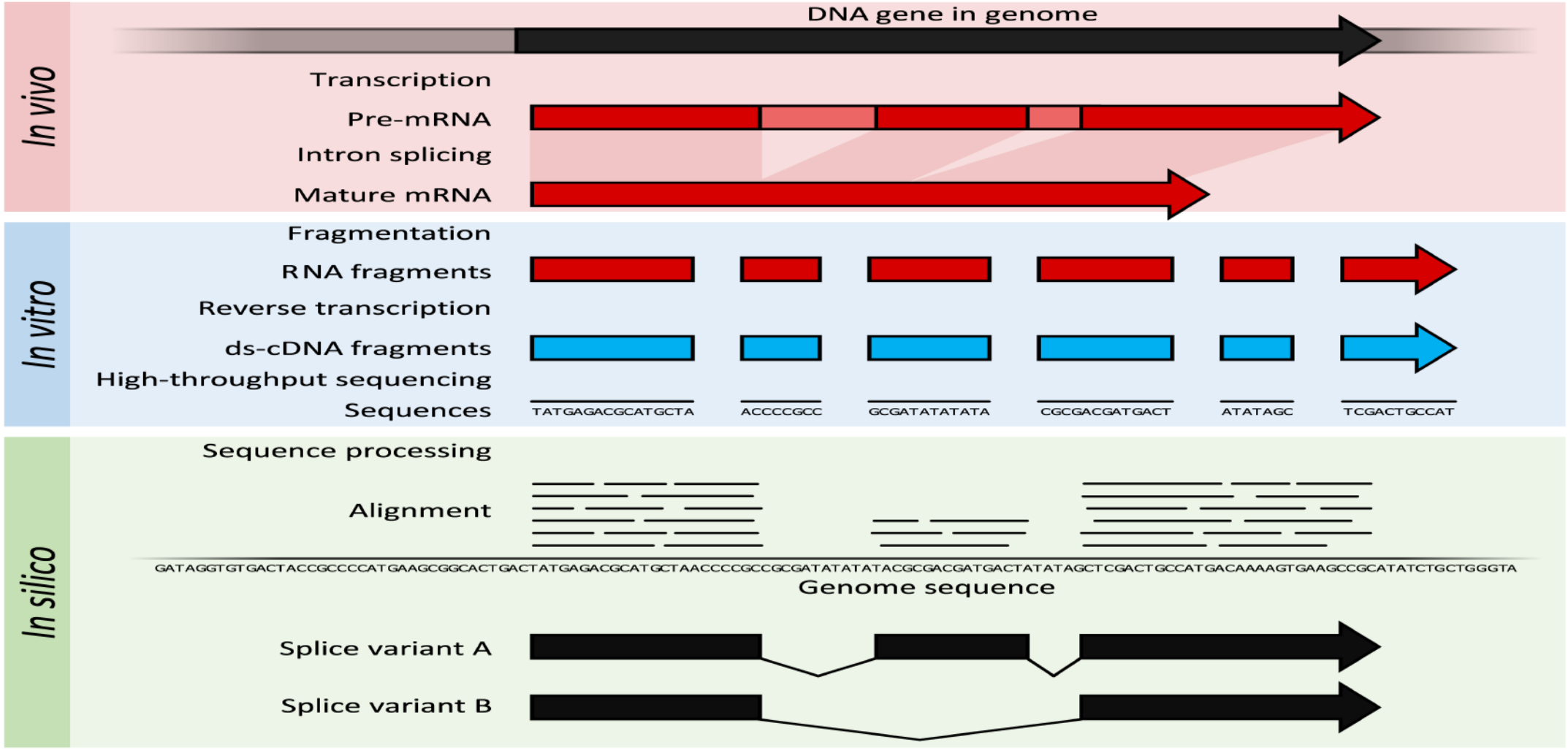
expressed sequence tag (blue)

serial/cap analysis of gene expression (yellow)

Rohan Lowie et al, **Transcriptomics technologies,** PLOS computational biology, 2017
http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005457

	Microarray	RNA-sequencing
Resolution	Difficult to measure low expressed genes.	Higher resolution but depends on sequencing depth. Note: RNA-seq with 20 – 30 million has same resolution as microarray (human samples)
Novel transcripts and splicing variants	No. Probes are designed from one reference genome. So transcripts with mutations are not represented well.	Yes, but depends on sequencing depth to some extent. Can also infer SNPs, exon junctions etc. **Cons:** If Globin mRNA or ribosomal RNA is not cleared properly, most of the reads will be wasted.
Organisms	Limited. Only model organisms like human, mouse, rats.	Any. For organisms without reference genome, we could use *de novo* assembly.
Flexibility	Restricted to the set of probes spotted onto array. Most arrays require 8 or 12 samples to be run on same chip.	Can use different library construction kits to measure mRNA, miRNA, lncRNA. Can run one samples per lane or multiplex several per lane.
Cost	Cheap but no longer in production	Varies. Library construction cost > sequencing cost

Simplified RNA-seq workflow



Rohan Lowie et al, **Transcriptomics technologies,** PLOS computational biology, 2017
http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005457

A more detailed RNA-seq workflow

Sample Preparation	Sequencing center	Primary Analysis	Secondary Analysis	Tertiary Analysis
Study design	Library construction	FASTQ	Read count table	Other analysis
Sample collection	Sequencing	FASTQC	Filter & Normalize	Validation
RNA extraction	Image analysis & Base calling	Read trimming	Sample QC	Paper / Grant / Patent
RNA QC evaluation	De-multiplexing	Read alignment	Differential expression	
Randomize & transfer	FASTQ	Quantification	Pathway enrichment	

Types of RNA-Seq

Types of RNA-Seq analysis

- Gene expression analysis
- Single cell RNA-Seq (scRNA-Seq)
- Small RNA-Seq (miRNA-Seq)
- Analysis of RNA-protein/RNA-RNA-interaction

Applications of RNA-Seq experiments

- Differential expression
- Gene fusion (arising due to translocation, deletion, chromosomal inversion)
- Alternative splicing
- Novel transcribed regions
- Allele-specific expression
- RNA editing
- Transcriptome for non-model organisms

Experimental planning is essential

RNA-Seq Workflow: RNA-Seq Experimental Design

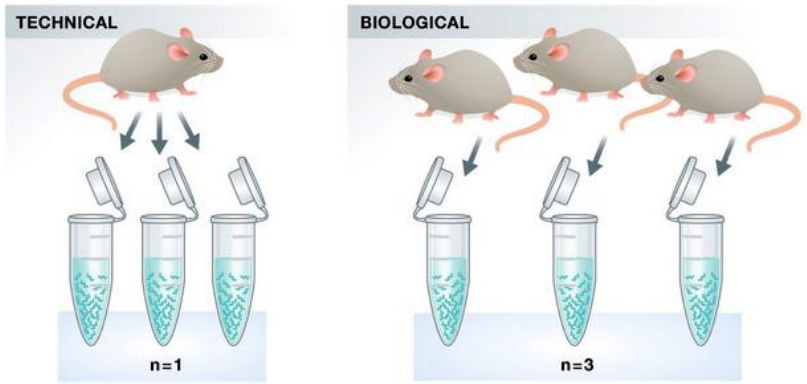


Image adapted from: Klaus B., EMBO J (2015) 34: 2727–2730

- **Technical replicates:** Generally low technical variation, so unnecessary.
- **Biological replicates:** Crucial to the success of RNA-Seq differential expression analyses. The more replicates the better, but at the very least have 3.
- **Batch effects:** Avoid as much as possible and note down all experimental variables.

- Try to avoid batch effect either by processing all samples in one batch or distribute all of your samples in different batches.
- Avoid confounding variations like using only male mice as control and female mice as treated/samples.

N=3 might be OK if you are working with cell lines but …
Generally N > 5 with animal/human work to cover the variability, outliers and sample failures.

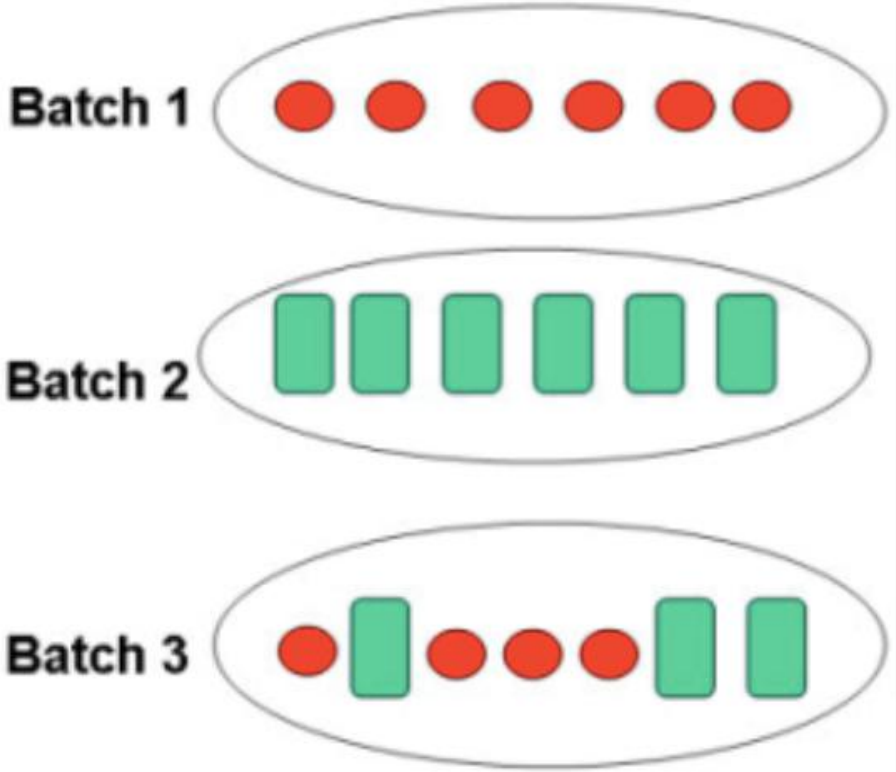
Sample Preparation

- Study design
- Sample collection
- RNA extraction
- RNA QC evaluation
- **Randomize & transfer**

Controls

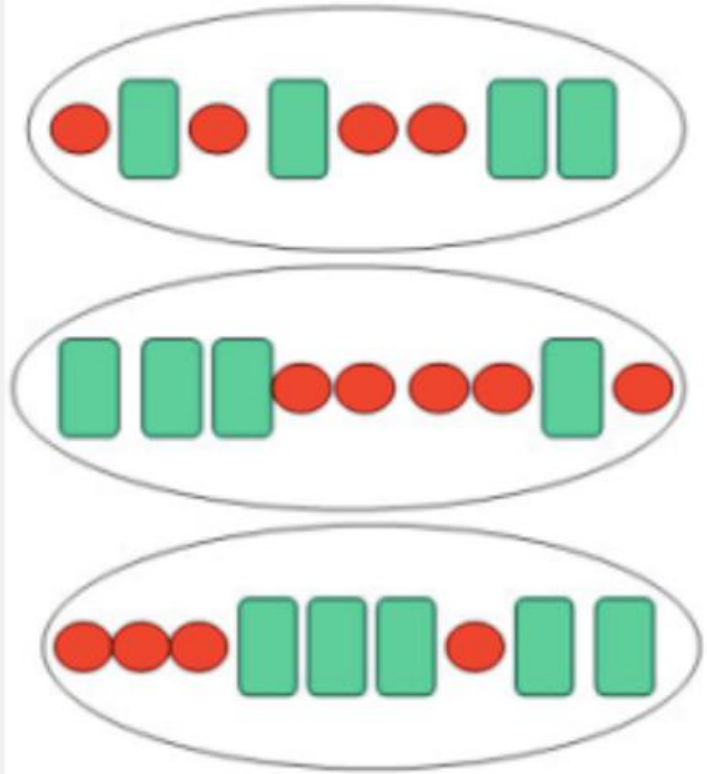
Cases

Experiment 1



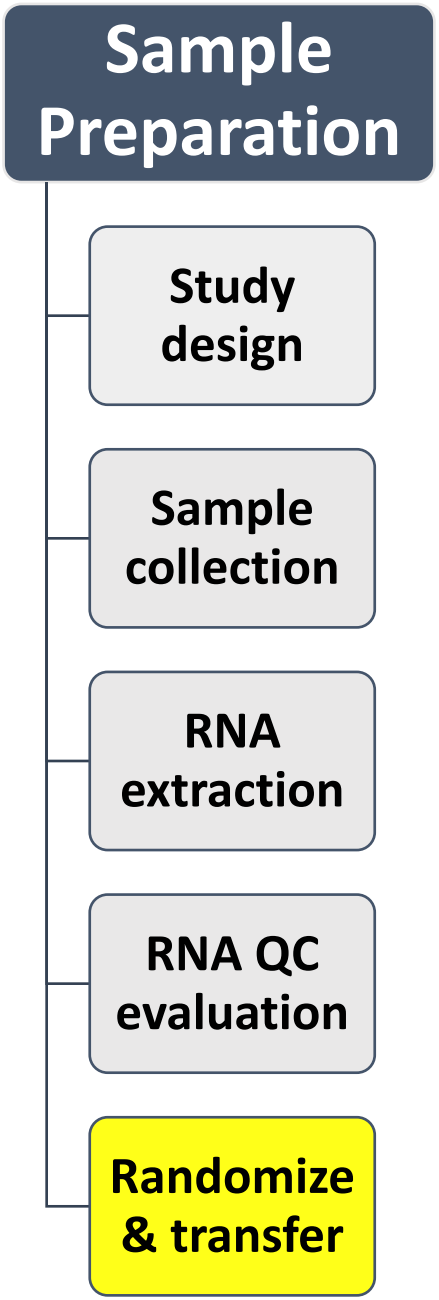
Batch effect cannot be corrected

Experiment 2



A much better design

Adopted from Walker WL et al. BMC Genomics 2008



Randomize samples with respect to

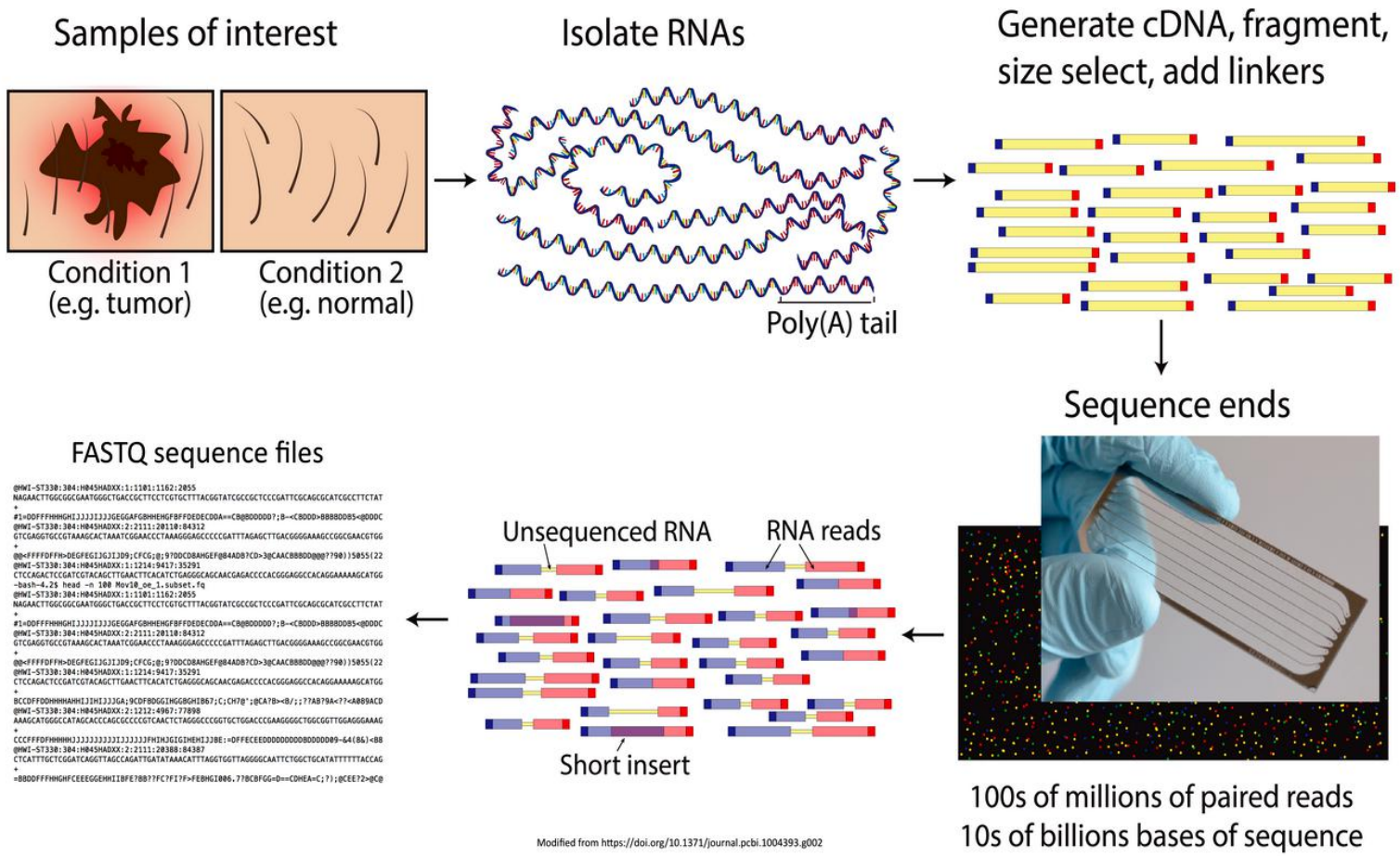
- Experimental conditions (if possible)
- RNA extraction
- Sequencing flow cells and lanes
- Multiplexing
- etc

Objective is to achieve approximately equal balance of samples within each stratum

Transfer samples to sequencing centers

- Protocol (dry ice, vials or plate, infectious agents/quarantine)
- Receipt acknowledgement

Biological sample/library preparation



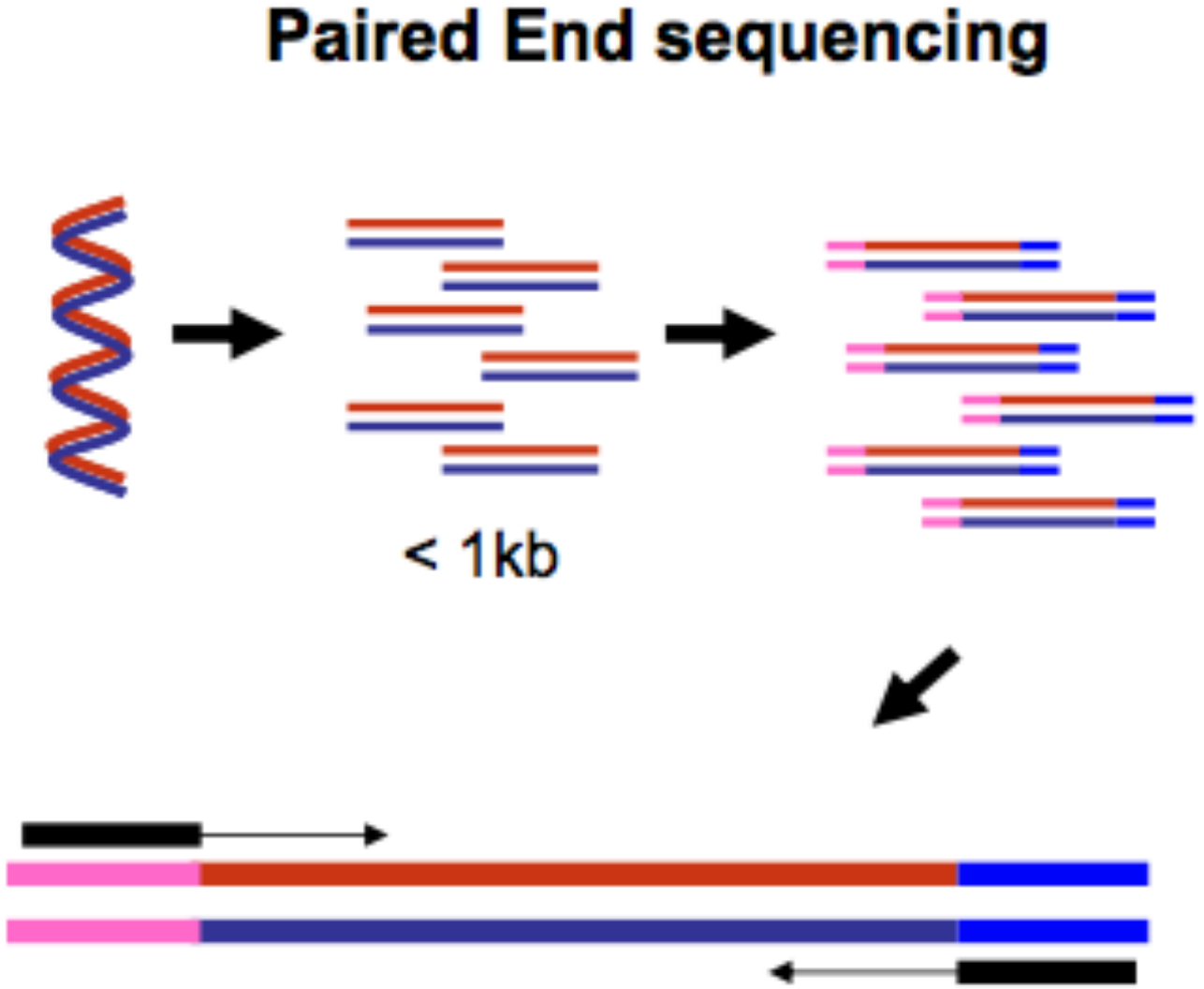
Samples are harvested. RNAs isolated, DNA contamination removed, rRNAs removed, mature RNAs are selected by their polyA tails.
RNA to cDNA, fragmented, size selected, adapters ligated to get RNA-seq library to be sequenced.
Ends of fragments sequenced are READS.

Sequencing strategies

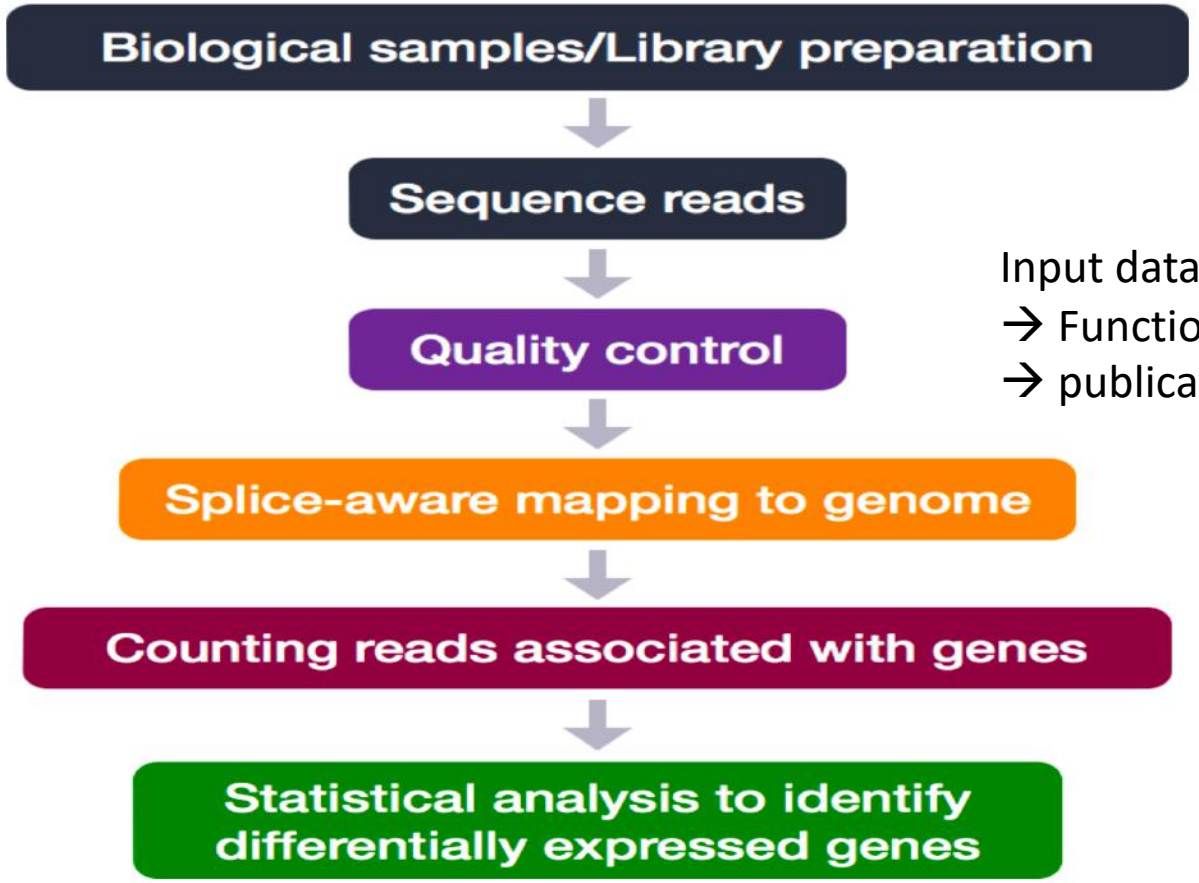
- Which library preparation protocol to use?
- How many replicates?
- What is the optimal library size (sequencing depth)?
- Paired end or single end?
- Which data analysis pipeline to use?

Sequencing center

- Library construction
- **Sequencing**
- Image analysis & Base calling
- De-multiplexing
- FASTQ

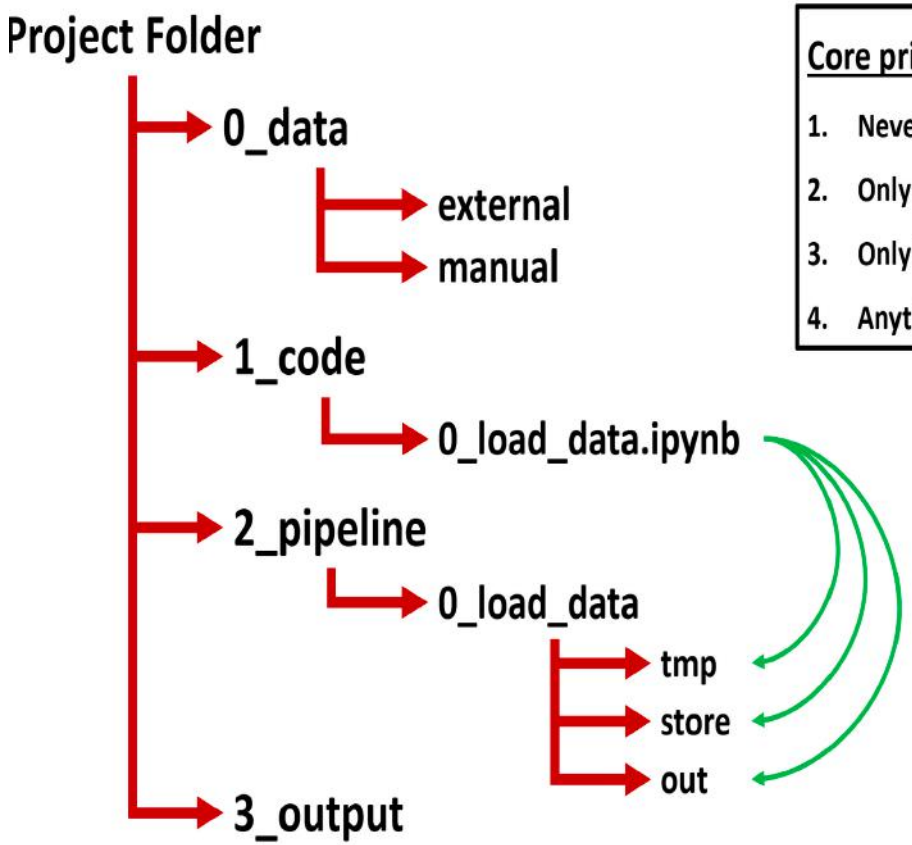


RNA-Seq Analysis Workflow



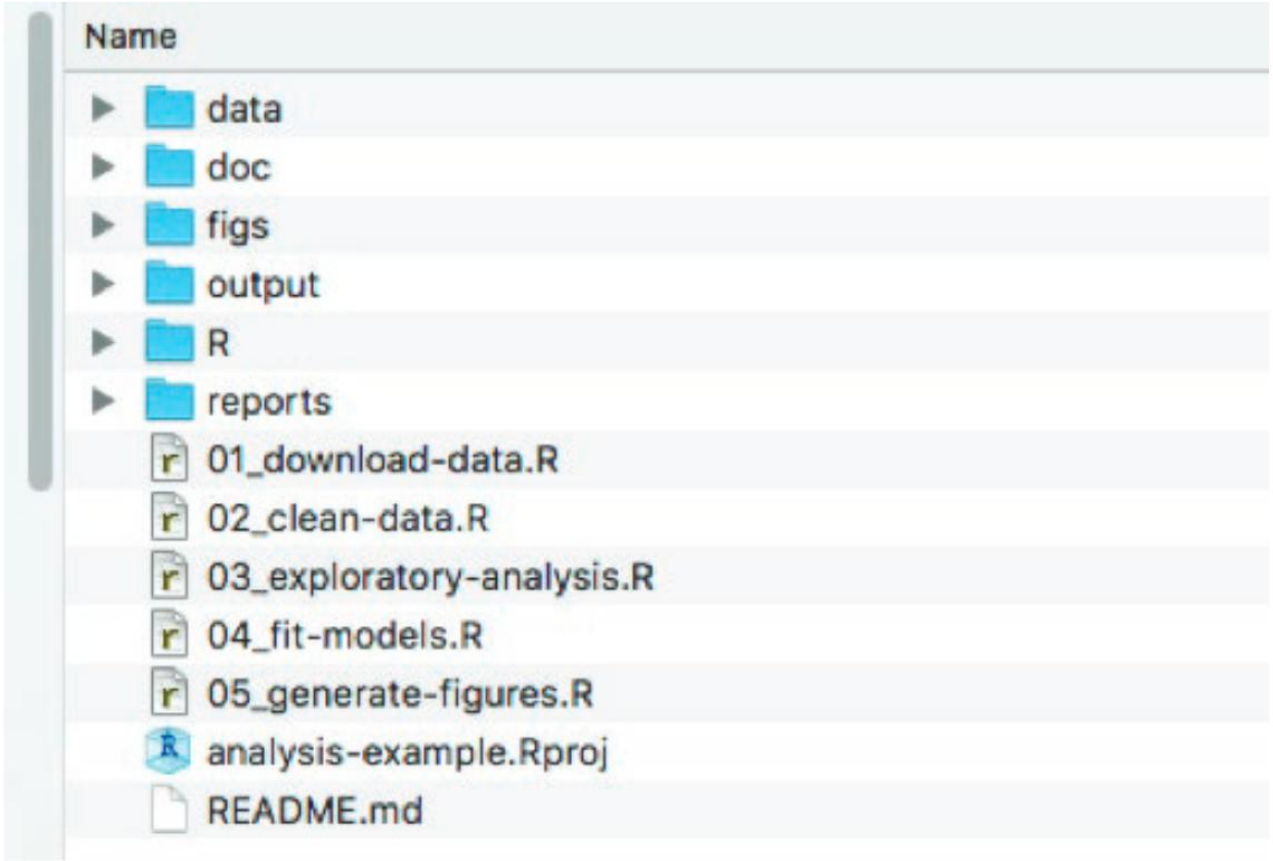
Input data → quality filtering → Mapping → Counting → DEG → Functional Analysis → Alternative splicing → data validation → publication (Bingo!!!)

Folder structure might help project organisation



Core principles:

1. **Never modify** 0_data
2. **Only save to** pipeline **folder**
3. **Only load from** 0_data **or** out
4. **Anything in** tmp **can be deleted**



Modules in this course

- Data download
- Quality assessment
- Data filtering
- Post filtering qc
- Mapping
- Mapped qc
- Count table
- DEG / Visualisation
- Functional Analysis
- Alternative splicing study

Module 1 : Input Data

Data for analysis

1. Can come from your own experiment
2. From your collaborators' lab
3. From public domain
4. Ways to get them could be either manual down or automatic download (using scripts/web interface scratching)
5. Next few slides will take you through manual download.

Our Reference for hands on and exercise

MOLECULAR ONCOLOGY 8 (2014) 1653–1666

available at www.sciencedirect.com

ScienceDirect

www.elsevier.com/locate/molonc

A nineteen gene-based risk score classifier predicts prognosis of colorectal cancer patients

Seon-Kyu Kim[a,1], Seon-Young Kim[a,1], Jeong-Hwan Kim[a], Seon Ae Roh[b,c], Dong-Hyung Cho[c,d], Yong Sung Kim[a,c,**], Jin Cheon Kim[b,c,*]

[a]Medical Genomics Research Centre, Korea Research Institute of Bioscience and Biotechnology, Daejeon, Korea
[b]Department of Surgery, University of Ulsan College of Medicine, Seoul, Korea
[c]Department of Cancer Research, Institute of Innovative Cancer Research and Asan Institute for Life Sciences, Asan Medical Centre, Seoul, Korea
[d]Graduate School of East-West Medical Science, Kyung Hee University, Gyeonggi-do, Korea

ARTICLE INFO

Article history:
Received 13 June 2014
Accepted 16 June 2014
Available online 4 July 2014

Keywords:
Colorectal cancer
Metastasis
Markers
Prognosis

ABSTRACT

Colorectal cancer (CRC) patients frequently experience disease recurrence and distant metastasis. This study aimed to identify prognostic indicators, including individual responses to chemotherapy, in CRC patients. RNA-seq data was generated using 54 samples (normal colon, primary CRC, and liver metastases) from 18 CRC patients and genes associated with CRC aggressiveness were identified. A risk score based on these genes was developed and validated in four independent CRC patient cohorts ($n = 1063$). Diverse statistical methods were applied to validate the risk scoring system, including a generalized linear model likelihood ratio test, Kaplan–Meier curves, a log-rank test, and the Cox model. *TREM1* and *CTGF* were identified as two activated regulators associated with CRC aggressiveness. A risk score based on 19 genes regulated by *TREM1* or *CTGF* activation (TCA19)

Seek for the accession

2.3. *RNA-seq data processing*

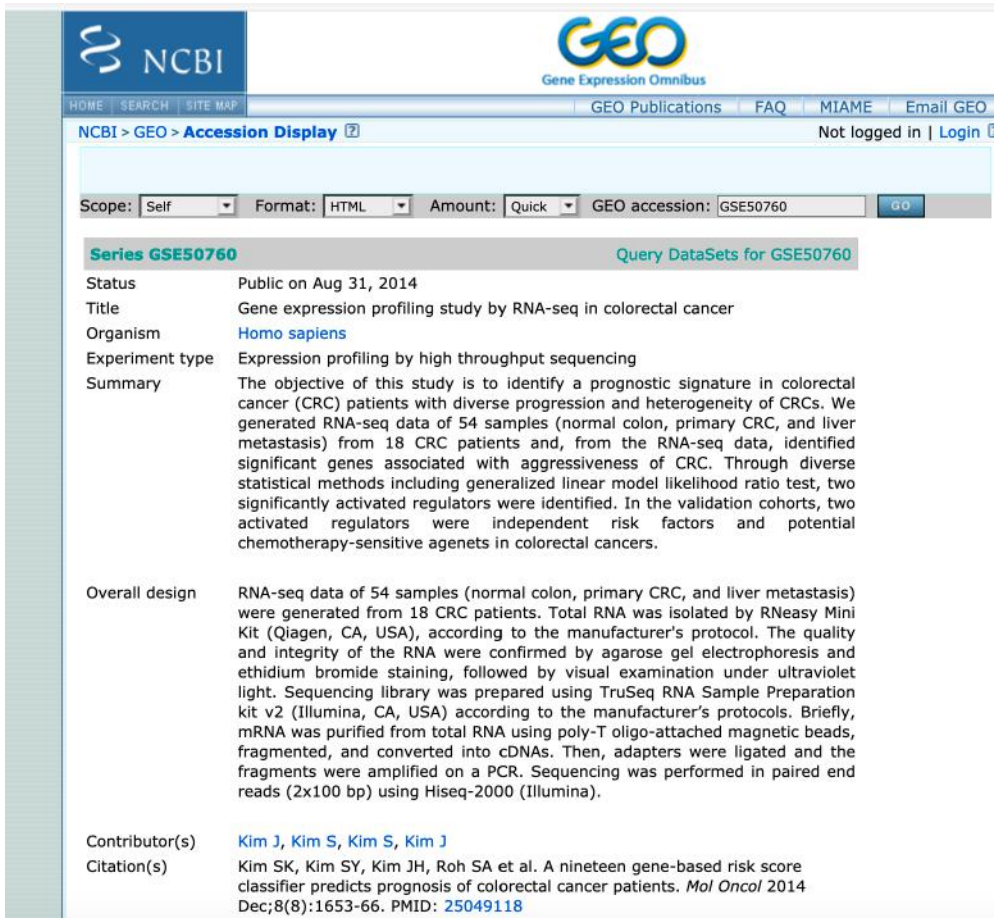
Reference genome sequence data from *Homo sapiens* were obtained from the University of California Santa Cruz Genome Browser Gateway (assembly ID: hg19). The reference genome index was built using the Bowtie2-build component of Bowtie2 (ver. 2.0) and SAMtools (ver. 0.1.18). Tophat2 was applied to tissue samples for mapping reads to the reference genome (ver. 2.0). The statistics of the mapping activity of Tophat2 are described in Supplementary Table 1. The data set generated by RNA-seq is available in the NCBI Gene Expression Omnibus public database under the data series accession number GSE50760.

Usually publications are accompanied with data submission accession details. They can be either GEO or SRP or other database.

Skim for those accession numbers and reach their source.

Here we could see the data accession is GSE85209

Data Acquisition



- Make sure have reached the right accession number.
- Is it linked to the paper you were interested by confirming with the title, authors, journal and year.
- Get the list of samples used in the experiment.

Biosamples of this project

Platforms (1) GPL11154 Illumina HiSeq 2000 (Homo sapiens)

Samples (54) GSM1228184 primary colorectal cancer AMC_2-1
More... GSM1228185 primary colorectal cancer AMC_3-1
GSM1228186 primary colorectal cancer AMC_5-1

Relations

BioProject PRJNA218851
SRA SRP029880

Download family	Format
SOFT formatted family file(s)	SOFT
MINiML formatted family file(s)	MINiML
Series Matrix File(s)	TXT

Supplementary file	Size	Download	File type/resource
GSE50760_RAW.tar	7.4 Mb	(http)(custom)	TAR (of TXT)

SRA Run Selector

Raw data are available in SRA

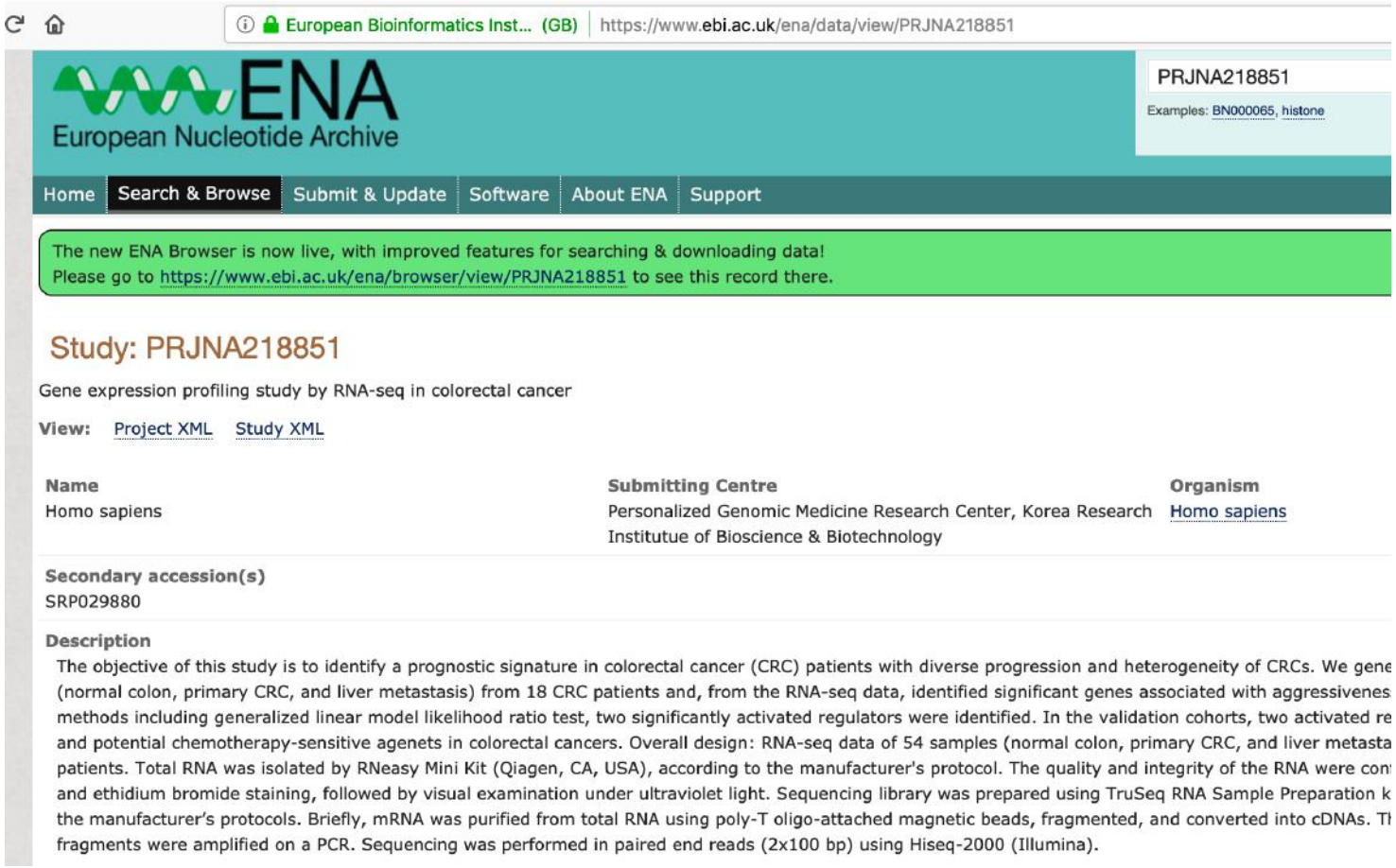
Processed data provided as supplementary file

The link for supplementary file has raw counts from each of the 14 samples.

Those who are eager to start from raw count, can use this file and their analysis.

But we wanted you to be more self dependent, hence teaching you how to start from fastq files!

Getting data from ENA



Downloading fastq files

Download: 1 - 54 of 54 results in TEXT

Select columns

Showing results 1 - 10 of 54 results

Study accession	Sample accession	Secondary sample accession	Experiment accession	Run accession	Tax ID	Scientific name	Instrument model	Library layout	FASTQ files (FTP)	FASTQ files (Galaxy)	Submitted files (FTP)	Submitted files (Galaxy)	NCBI SRA file (FTP)	NCBI SRA file (Galaxy)
PRJNA218851	SAMN02353232	SRS478664	SRX347887	SRR975551	9606	Homo sapiens	Illumina HiSeq 2000	PAIRED	File 1 File 2	File 1 File 2			File 1	File 1
PRJNA218851	SAMN02353258	SRS478665	SRX347888	SRR975552	9606	Homo sapiens	Illumina HiSeq 2000	PAIRED	File 1 File 2	File 1 File 2			File 1	File 1
PRJNA218851	SAMN02353264	SRS478667	SRX347889	SRR975553	9606	Homo sapiens	Illumina HiSeq 2000	PAIRED	File 1 File 2	File 1 File 2			File 1	File 1
PRJNA218851	SAMN02353233	SRS478666	SRX347890	SRR975554	9606	Homo sapiens	Illumina HiSeq 2000	PAIRED	File 1 File 2	File 1 File 2			File 1	File 1
PRJNA218851	SAMN02353255	SRS478668	SRX347891	SRR975555	9606	Homo sapiens	Illumina HiSeq 2000	PAIRED	File 1 File 2	File 1 File 2			File 1	File 1
PRJNA218851	SAMN02353269	SRS478669	SRX347892	SRR975556	9606	Homo sapiens	Illumina HiSeq 2000	PAIRED	File 1 File 2	File 1 File 2			File 1	File 1
PRJNA218851	SAMN02353234	SRS478670	SRX347893	SRR975557	9606	Homo sapiens	Illumina HiSeq 2000	PAIRED	File 1 File 2	File 1 File 2			File 1	File 1
PRJNA218851	SAMN02353260	SRS478671	SRX347894	SRR975558	9606	Homo	Illumina	PAIRED	File 1	File 1			File 1	File 1

Data Download Hands on

- create a folder to download data
- download the data
- Go to ENA, enter the GSE id, download fastq file
- copy the link <sample file link> to your folder.
- wget <the link> .

End of data download

- Do the same for all the files
- DATA DOWNLOAD is done!
- Next Module is about data quality

Module 2 : Quality control of the data

- We need FastQC installed in our computers/servers.
- https://www.bioinformatics.babraham.ac.uk/projects/fastqc/
- It is both command line and GUI tool.
- Download and install (NOTE, For hands one we have then on ibex)
- Open our data file using FastQC.
- Let us understand more on the quality and to clean the bad data in the following slides

Data Quality and Cleaning

Factors influencing sequencing outcomes

- Human-independent factors
    - Ability to measure fluorescence signal intensities accurately
    - Loss of efficacy of reagents over time – duration of the sequencing run
- Human Errors
    - Excessive fragmentation of the input DNA
    - Contaminants

Your RNA-Seq file looks like this

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

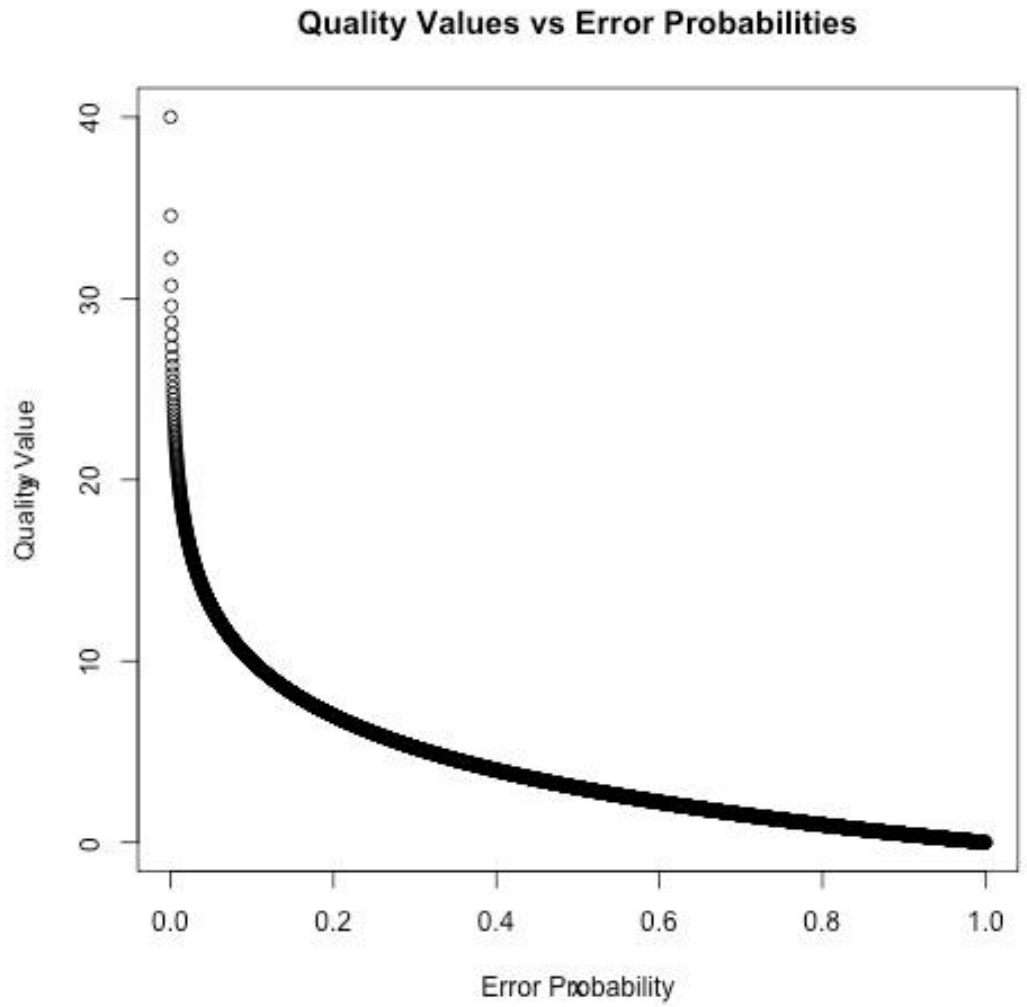
```
@HWUSI-EAS525:2:1:13336:1129#0/1
GTTGGAGCCGGCGAGCGGGACAAGGCCCTTGTCCA
+
ccacacccacccccccccc[[cccc_ccaccbbb_
@HWUSI-EAS525:2:1:14101:1126#0/1
GCCGGGACAGCGTGTTGGTTGGCGCGCGGTCCCTC
+
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
@HWUSI-EAS525:2:1:15408:1129#0/1
CGGCCTCATTCTTGGCCAGGTTCTGGTCCAGCGAG
+
cghhchhgchehhdffccgdgh]gcchhcahWcea
@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
```

Data quality and its relevance

- Significance
- How NGS data quality is quantified?
- Phred quality scores

$Q = -10 \log_{10} P$

where P is the probability that the base called is incorrect



Quality score function and its significance

- $Q = -10 \log_{10} P$, where P is the probability that the base called is incorrect

- Significance

P	% Error	% Reliability	*Q*
1/10	10	90	10
1/100	1	99	20
1/1000	0.1	99.9	30
1/10000	0.01	99.99	40

and so on…

Illumina's Sequence Quality Encoding

Symbol	ASCII Code	Q-Score
!	33	0
"	34	1
#	35	2
$	36	3
%	37	4
&	38	5
'	39	6
(	40	7
)	41	8
*	42	9
+	43	10
,	44	11
-	45	12
.	46	13
/	47	14
0	48	15
1	49	16
2	50	17
3	51	18
4	52	19
5	53	20

Symbol	ASCII Code	Q-Score
6	54	21
7	55	22
8	56	23
9	57	24
:	58	25
;	59	26
<	60	27
=	61	28
>	62	29
?	63	30
@	64	31
A	65	32
B	66	33
C	67	34
D	68	35
E	69	36
F	70	37
G	71	38
H	72	39
I	73	40

How to decipher the Sequence Quality string?

Sequence: GTGTATG

Quality string: ?A>F@JI

Nucleotide	Character from Quality string	ASCII value	Quality Value
G	?	63	63-33 = 30
T	A	65	65-33 = 32
G	>	62	62-33 = 29
T	F	70	70-33 = 37
A	@	64	64-33 = 31
T	J	74	74-33 = 41
G	I	73	73-33 = 40

Why FastQC?

- Large sequence throughput
- Need to check quality before any analysis aimed at biological inference
- Limitations of quality report from the sequencer

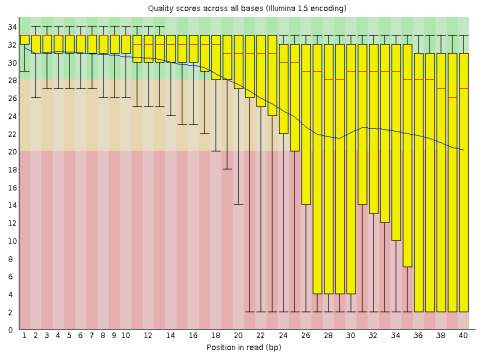
Highlights of the FastQC tool

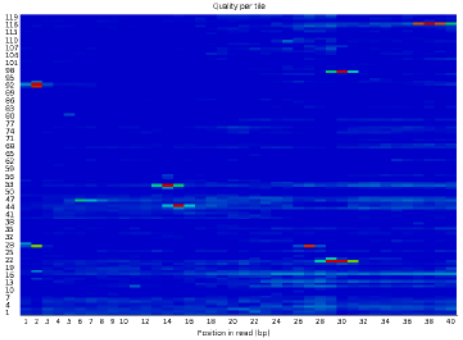
- FastQC is a program written in Java
- Runs on all standard software platforms – Linux, Mac, Windows
- Easy to install
- “Light weight” on its compute requirements
- Well-maintained and regularly updated
- Well-documented
- Widely used and discussed in the community
- Can be run in either standalone interactive / non-interactive (batch) mode

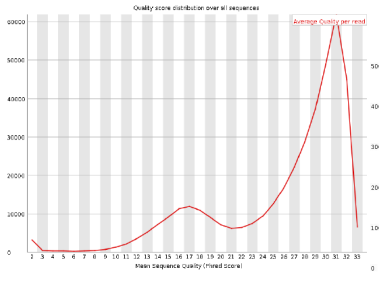
Reports generated by FastQC

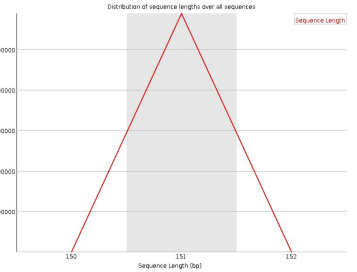
Basic Statistics

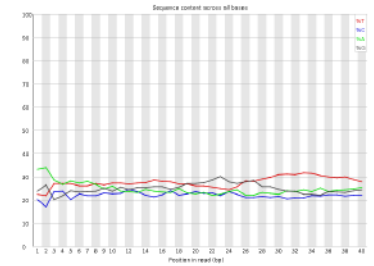
Measure	Value
Filename	M_19_0108_AM7962_AD004_L003_R1_001.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	5885062
Sequences flagged as poor quality	0
Sequence length	151
%GC	51

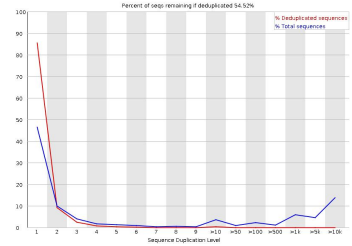


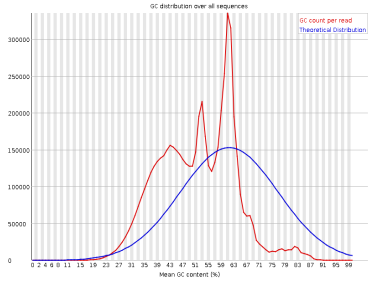












Overrepresented sequences

Sequence	Count	Percentage	Possible Source
CTCGAGTCGACGTCGCTATTCACAGCCCGATCCCACTACTGATCAGCACC	81075	1.377640541425052	No Hit
CCAGCCTCGAGTCGACGTCGCTATTCACAGCCCGATCCCACTACTGATCA	61843	1.0508470429028616	No Hit
CTCGAGTCCTCGAAGCTCGACTACCCTACGTTCTCCTACAAAATCGACCTT	57843	0.9828783452068983	No Hit
GCCTCGAGTCGACGTCGCTATTCACAGCCCGATCCCACTACTGATCGACCA	56821	0.9655123429455799	No Hit
GCCAACCTCGTCGTCCCCCCCCTCCCCCGGAGGTCACCATATTGATCCCAA	54501	0.9260904982819211	No Hit
CAGCCTCGAGTCGACGTCGCTATTCACAGCCCGATCCCACTACTGATCAG	47865	0.8133304288043184	No Hit
CCTAGCCAACCTCGTCGTCCCCCCCCTCCCCGGAGGTCACCATATTGATCC	47157	0.8012999693121329	No Hit
CCTTAGCCAACCTCGTCGTCCCCCCCTCCCCGGAGGTCACCATATTGATC	41412	0.7036799272463O6	No Hit
CTCGAGCCTCGAGTCGACGTCGCTATTCACAGCCCGATCCCACTACTGATC	39707	0.6747082698534017	No Hit
CACAAATTATCGACGTCGAGTTTCCCACATTTGGGGAAATCGCAGGGGTCA	35622	0.6052952373313993	No Hit

Explanation of FastQC modules – Basic Statistics

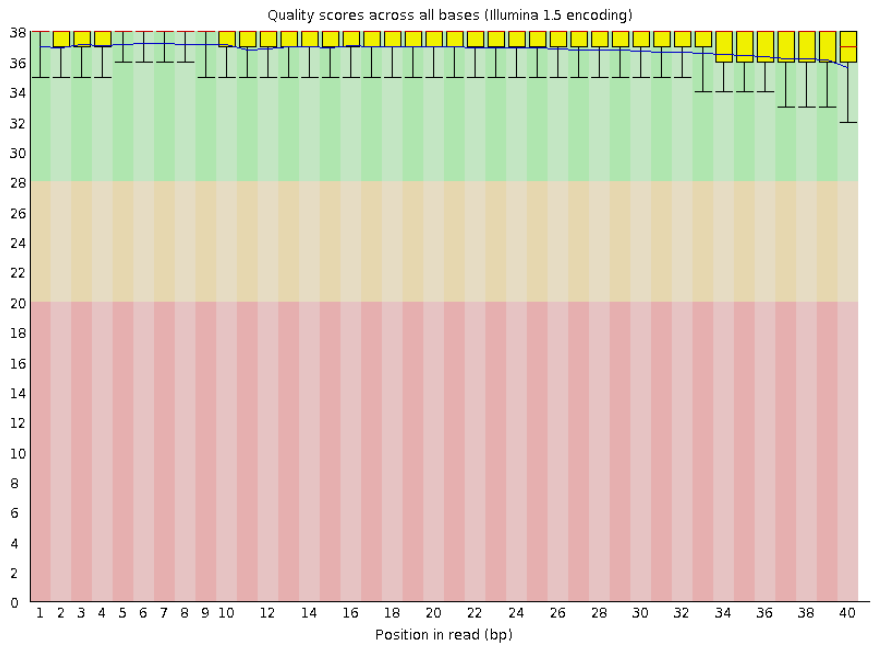
Basic Statistics

Measure	Value
Filename	M_19_0108_AM7962_AD004_L003_R1_001.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	5885062
Sequences flagged as poor quality	0
Sequence length	151
%GC	51

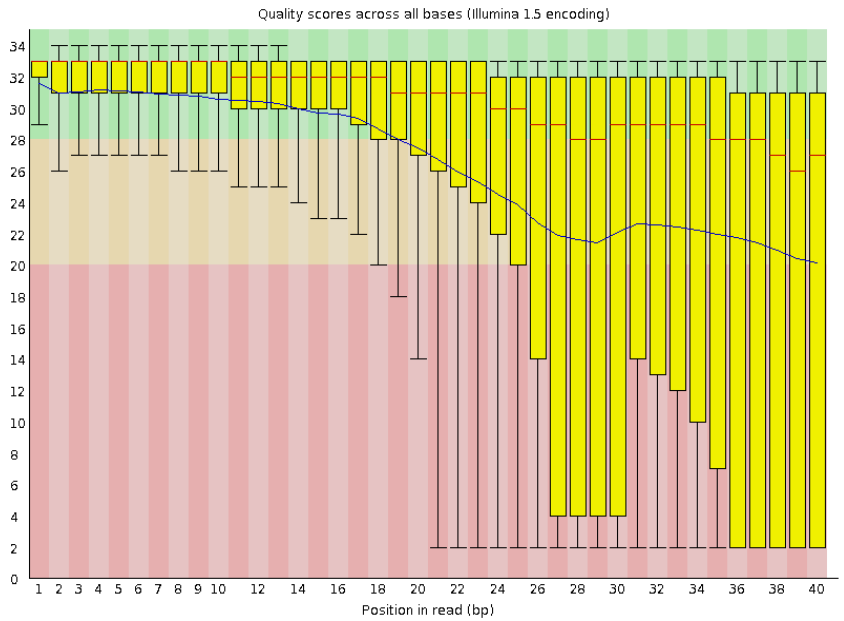
Explanation of FastQC modules – Per Base Sequence Quality

Per base sequence quality

Good data



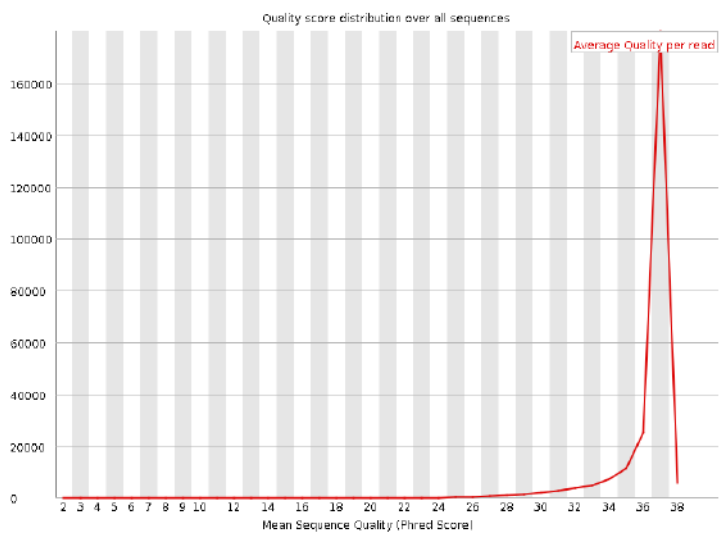
Bad data



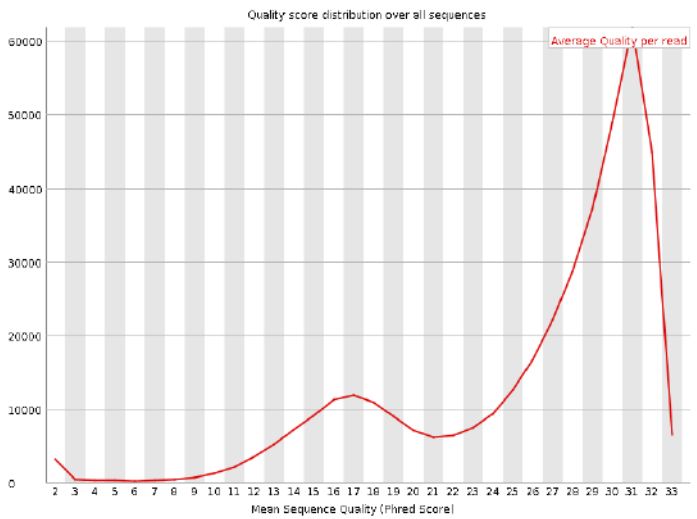
Explanation of FastQC modules – Per Sequence Quality Scores

Per sequence quality scores

Good data



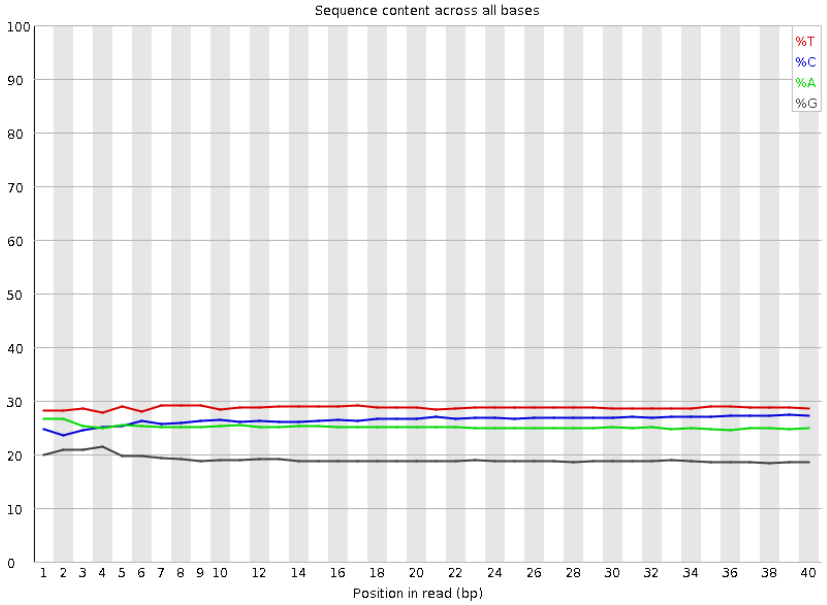
May need to be understood



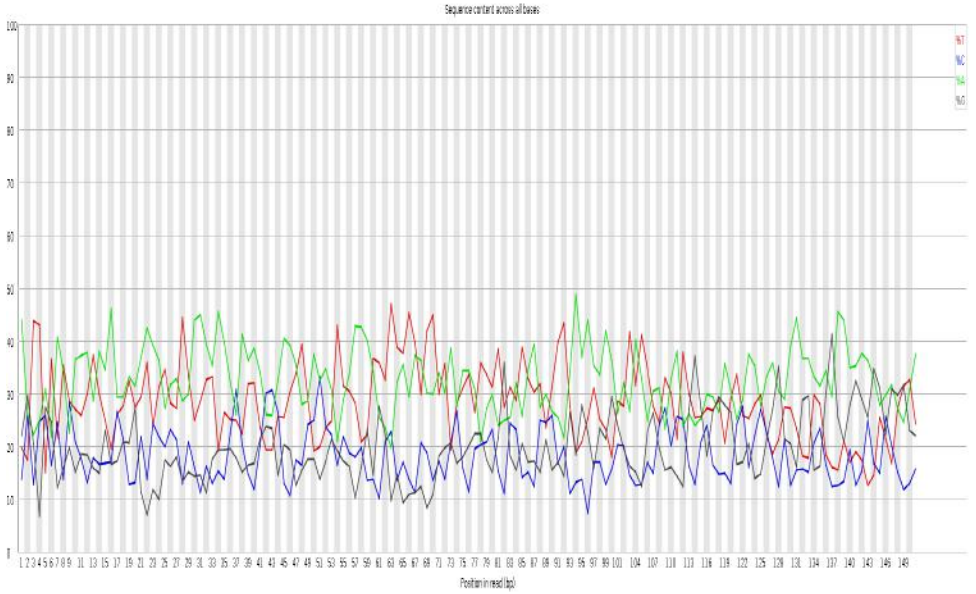
Explanation of FastQC modules – Per Base Sequence Content

Per base sequence content

Good data



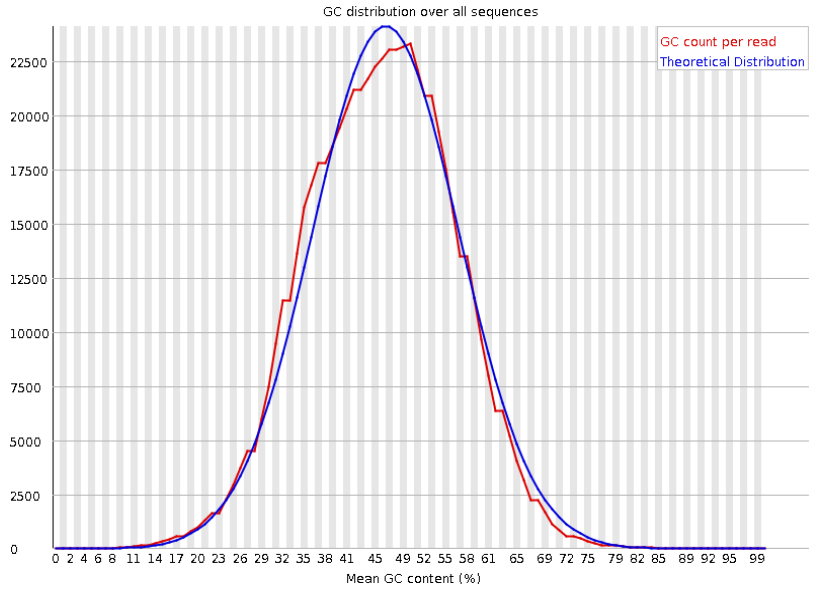
Bad data



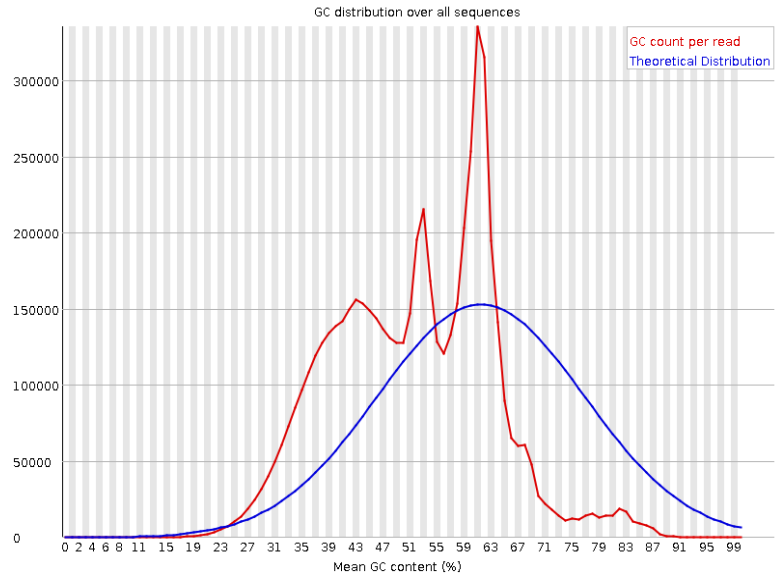
Explanation of FastQC modules – Per Sequence GC Content

Per sequence GC content

Good data



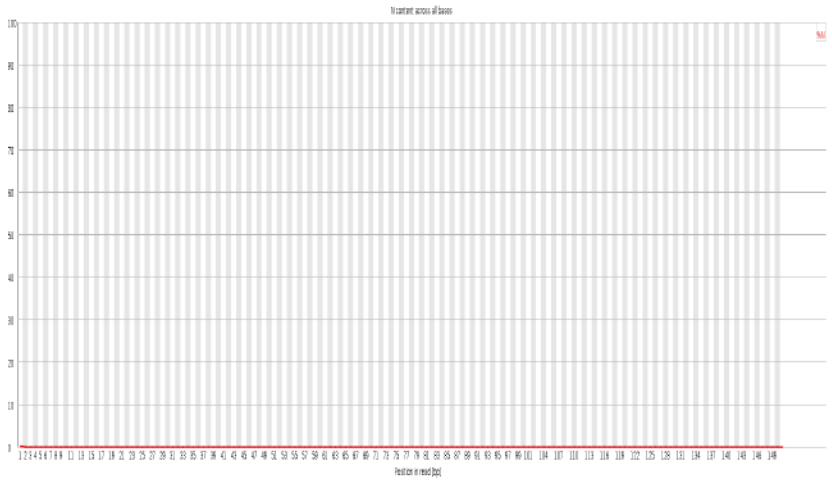
Needs investigation



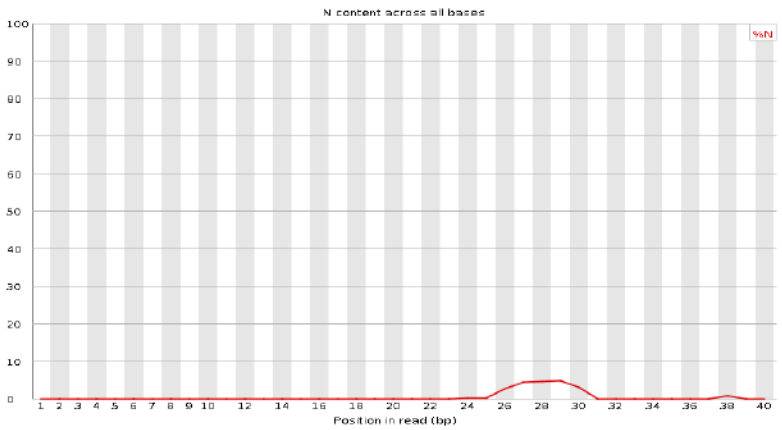
Explanation of FastQC modules – Per Base N Content

Per base N content

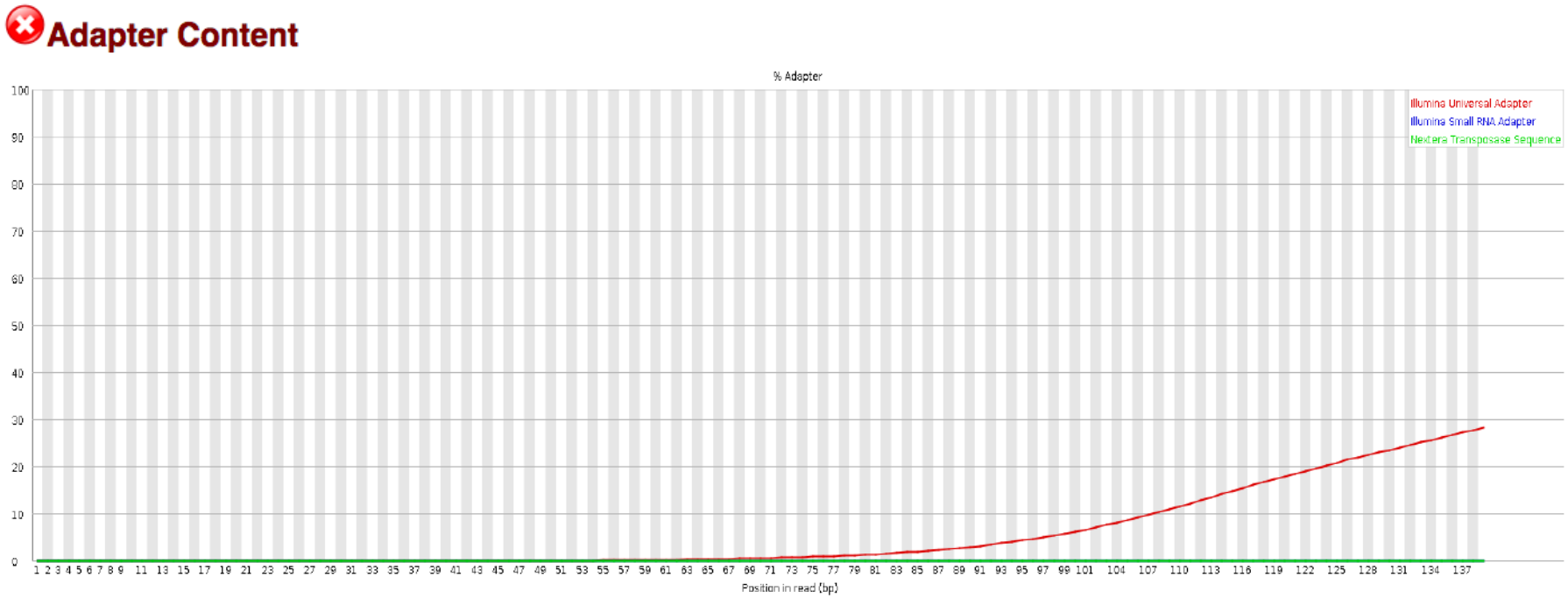
Good data



Few reads have N bases

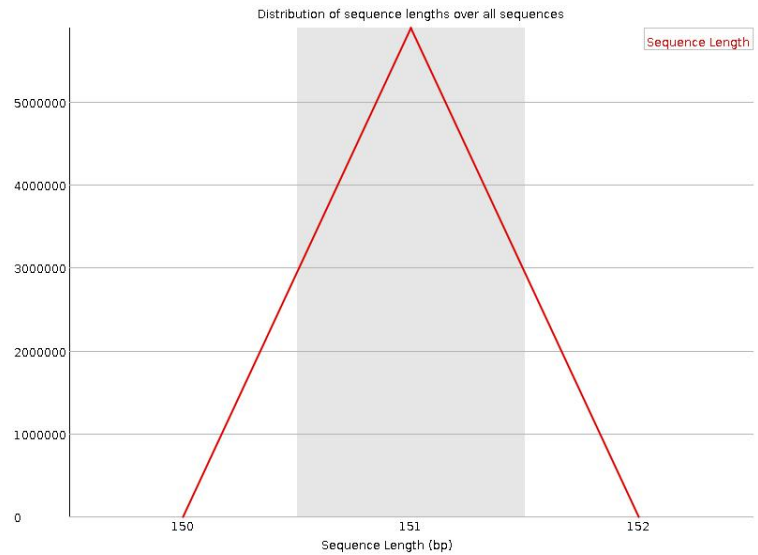


Explanation of FastQC modules – Adapter Content



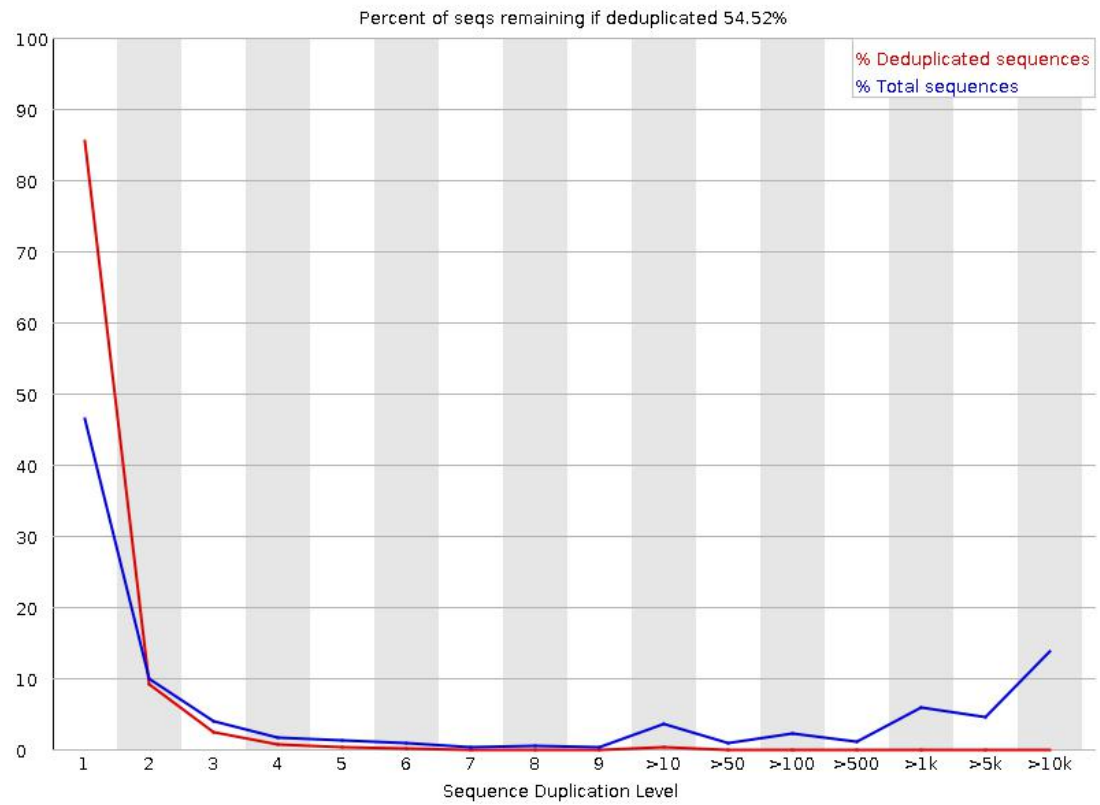
Explanation of FastQC modules – Sequence Length Distribution

Sequence Length Distribution



Explanation of FastQC modules – Sequence Duplication Levels

Sequence Duplication Levels

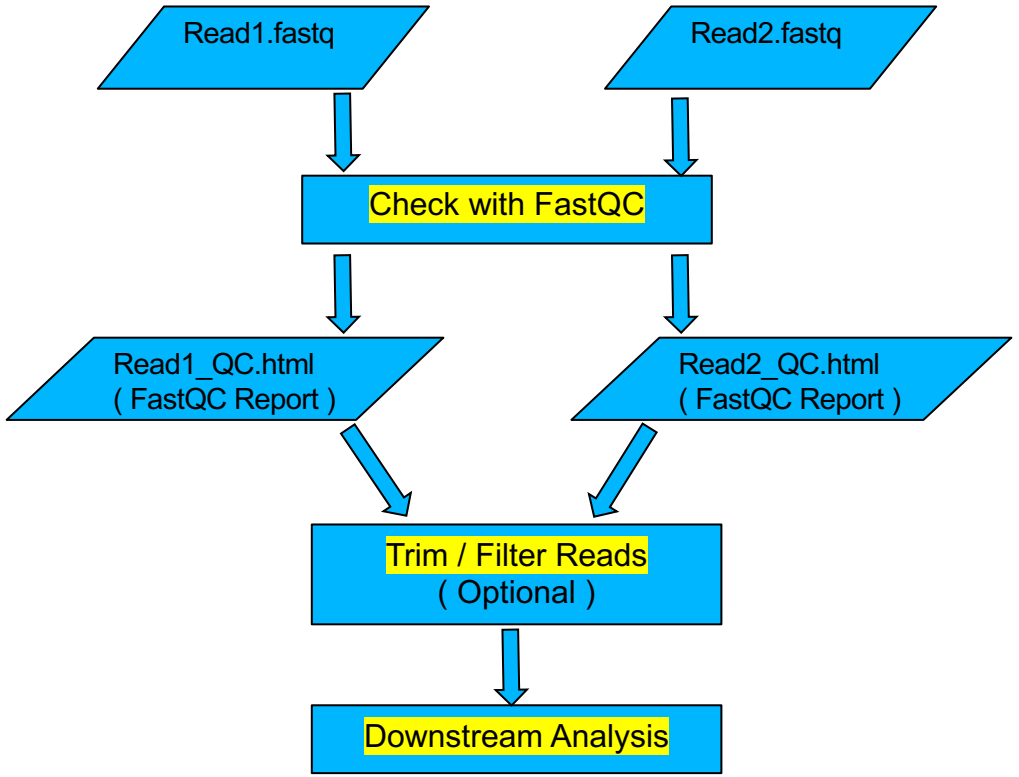


Explanation of FastQC modules – Overrepresented Sequences

Overrepresented sequences

Sequence	Count	Percentage	Possible Source
CTGGAGTGCAGTGGCTATTCACAGGCGCGATCCCACTACTGATCAGCACG	81075	1.377640541425052	No Hit
CCAGGCTGGAGTGCAGTGGCTATTCACAGGCGCGATCCCACTACTGATCA	61843	1.0508470429028616	No Hit
CTGGAGTCTTGGAAGCTTGACTACCCTACGTTCTCCTACAAATGGACCTT	57843	0.9828783452068983	No Hit
GGCTGGAGTGCAGTGGCTATTCACAGGCGCGATCCCACTACTGATCAGCA	56821	0.9655123429455799	No Hit
GGCAACCTGGTGGTCCCCCGCTCCCGGGAGGTCACCATATTGATGCCGAA	54501	0.9260904982819211	No Hit
CAGGCTGGAGTGCAGTGGCTATTCACAGGCGCGATCCCACTACTGATCAG	47865	0.8133304288043184	No Hit
CTTAGGCAACCTGGTGGTCCCCCGCTCCCGGGAGGTCACCATATTGATGC	47157	0.8012999693121329	No Hit
CCTTAGGCAACCTGGTGGTCCCCCGCTCCCGGGAGGTCACCATATTGATG	41412	0.703679927246306	No Hit
CTCAGGCTGGAGTGCAGTGGCTATTCACAGGCGCGATCCCACTACTGATC	39707	0.6747082698534017	No Hit
CACAAATTATGCAGTCGAGTTTCCCACATTTGGGGAAATCGCAGGGGTCA	35622	0.6052952373313993	No Hit

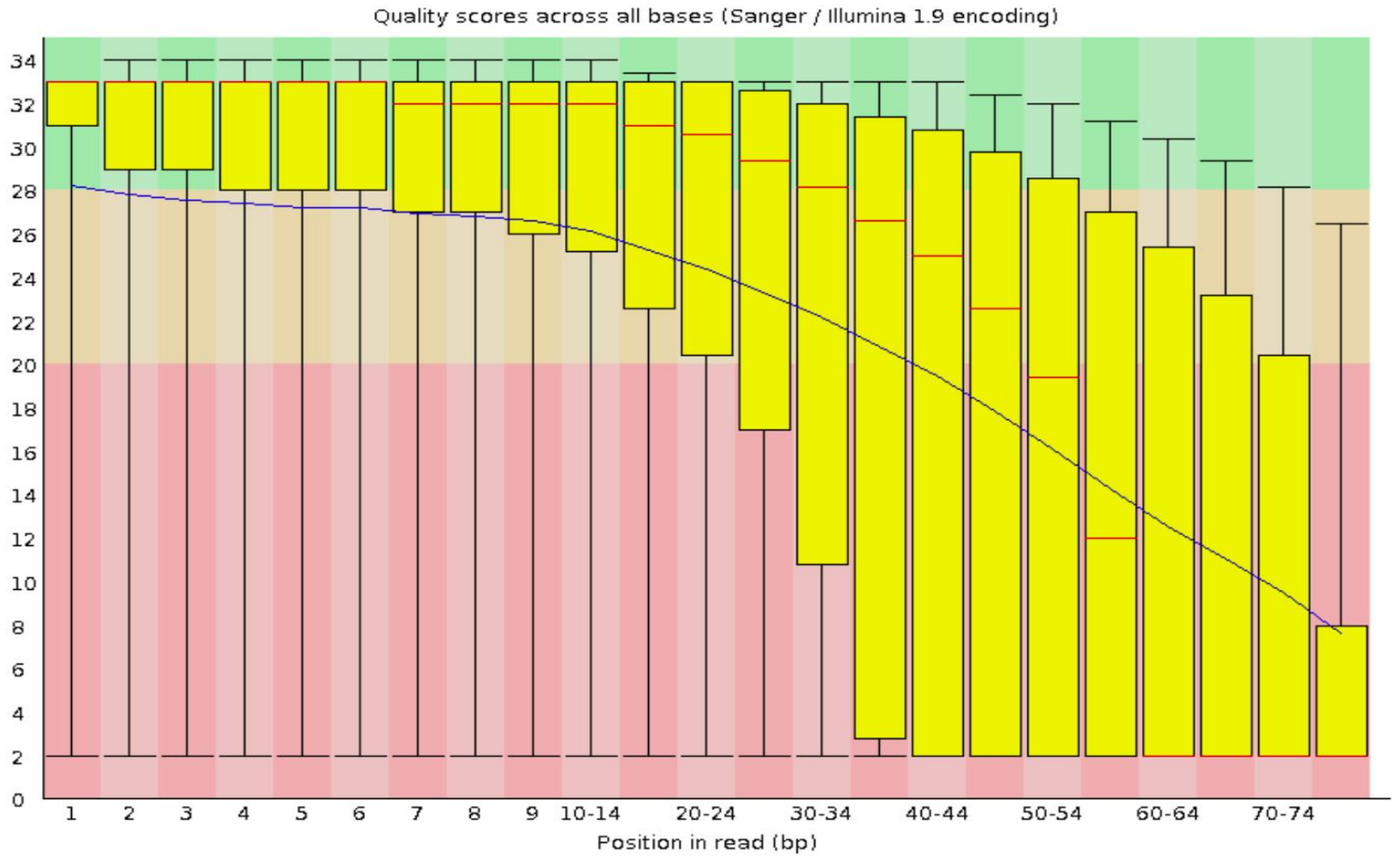
Representative QC workflow



Quality Control of RNA-Seq Reads

Step 1. Quality Control (QC) using FASTQC Software

1. Sequencing quality score

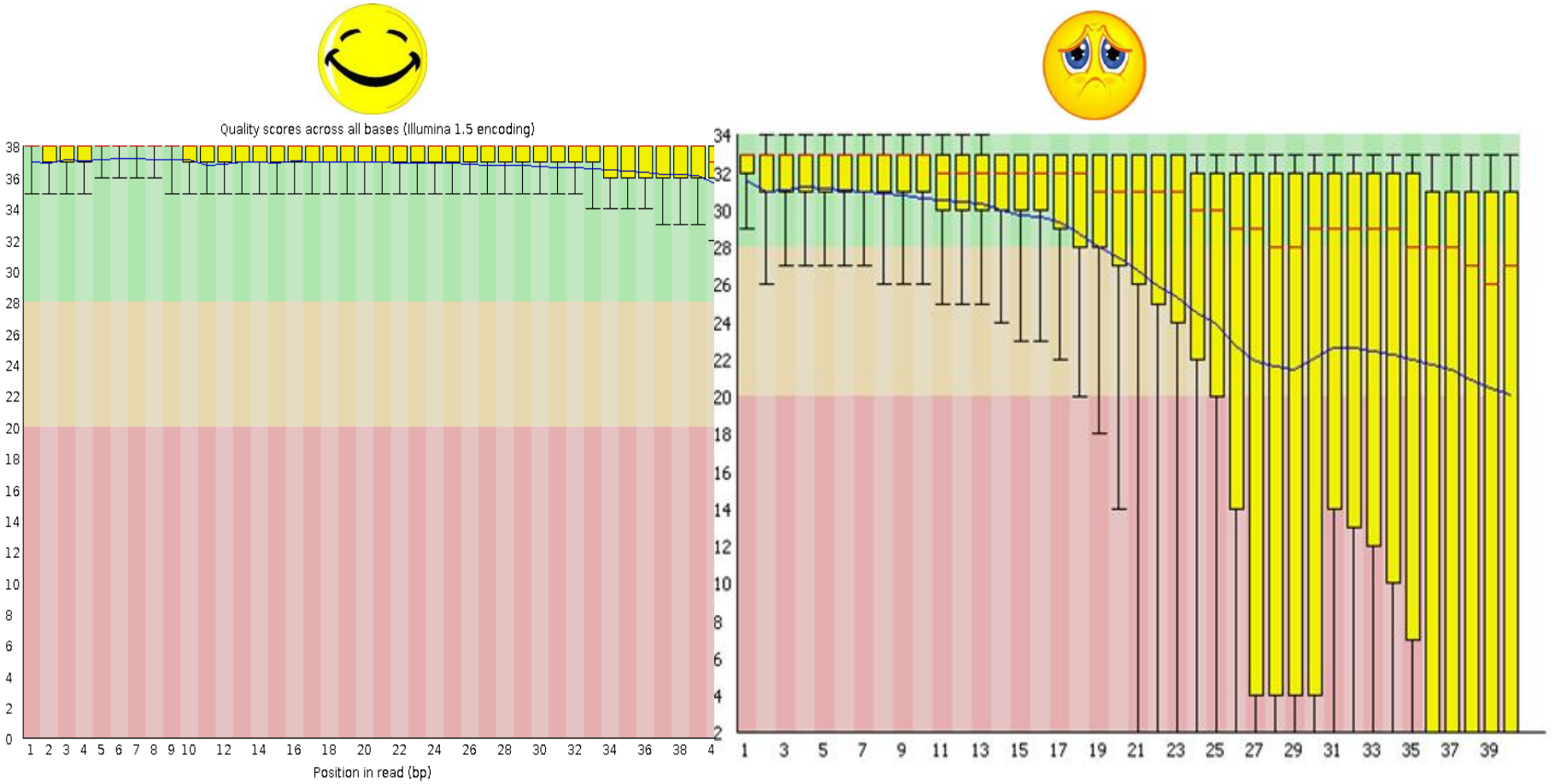


Look for any erroneous sequences, bacterial contamination, remove adapters and ribosomal RNAs

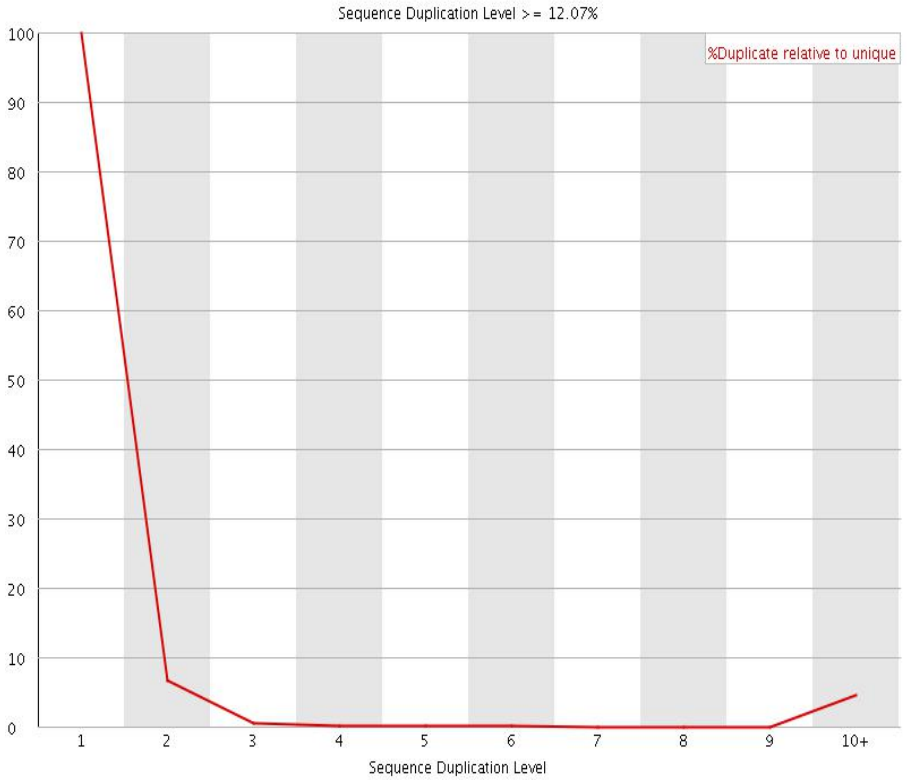
Overrepresented sequences

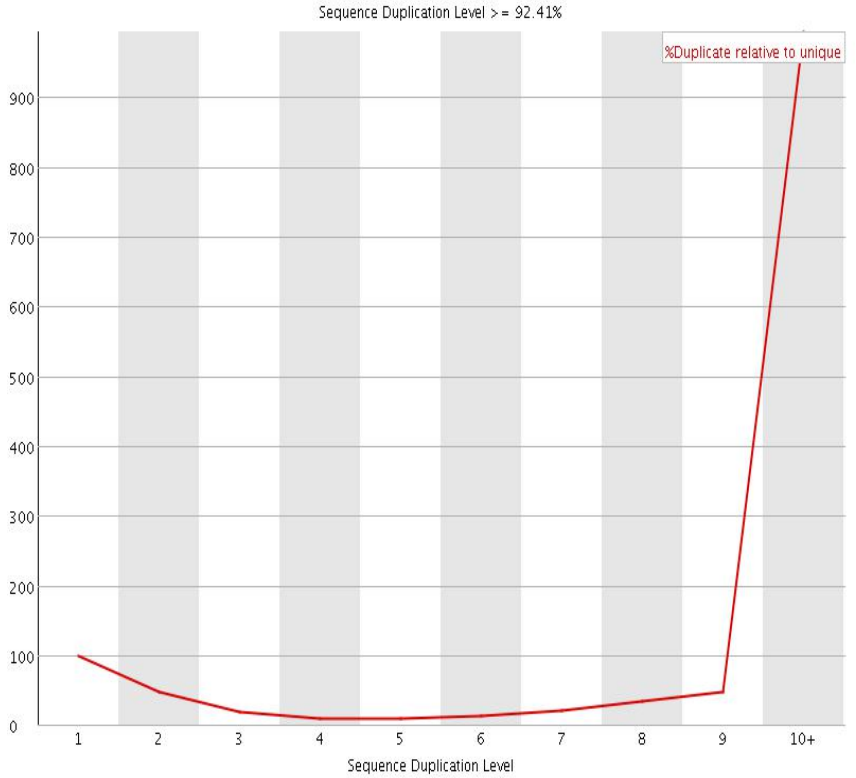
Sequence	Count	Percentage	Possible Source
CTGCTATGGCCACCAGACTCTCAGGCTCCATGCAGTGGCCAGCCTCATCG	2554	0.8349133703824779	No Hit
CAGCGGTCTAGTTTGAAGAACCTGACCCGAGTCTTGGTGACGAAGGCCAG	2463	0.8051650866296176	No Hit
GTTTGAAGAACCTGACCCGAGTCTTGGTGACGAAGGCCAGATTTGCGATC	1920	0.6276560967636483	No Hit
CCACAGGGTCCCAGGTCATGGGTACCGAGTCCAGGTCATAGTGCCGGATG	1219	0.39849624060150374	No Hit
GAAGAACCTGACCCGAGTCTTGGTGACGAAGGCCAGATTTGCGATCTTCA	1186	0.3877084014383786	No Hit
GGCAGGTGGACCCGGAGCCGCTGACAGAGGAGGTCAGCCCCTGAGTTGGA	1111	0.3631905851585486	No Hit
CACAGGGTCCCAGGTCATGGGTACCGAGTCCAGGTCATAGTGCCGGATGT	1079	0.35272965021248776	No Hit

Per base sequence quality



Duplication level



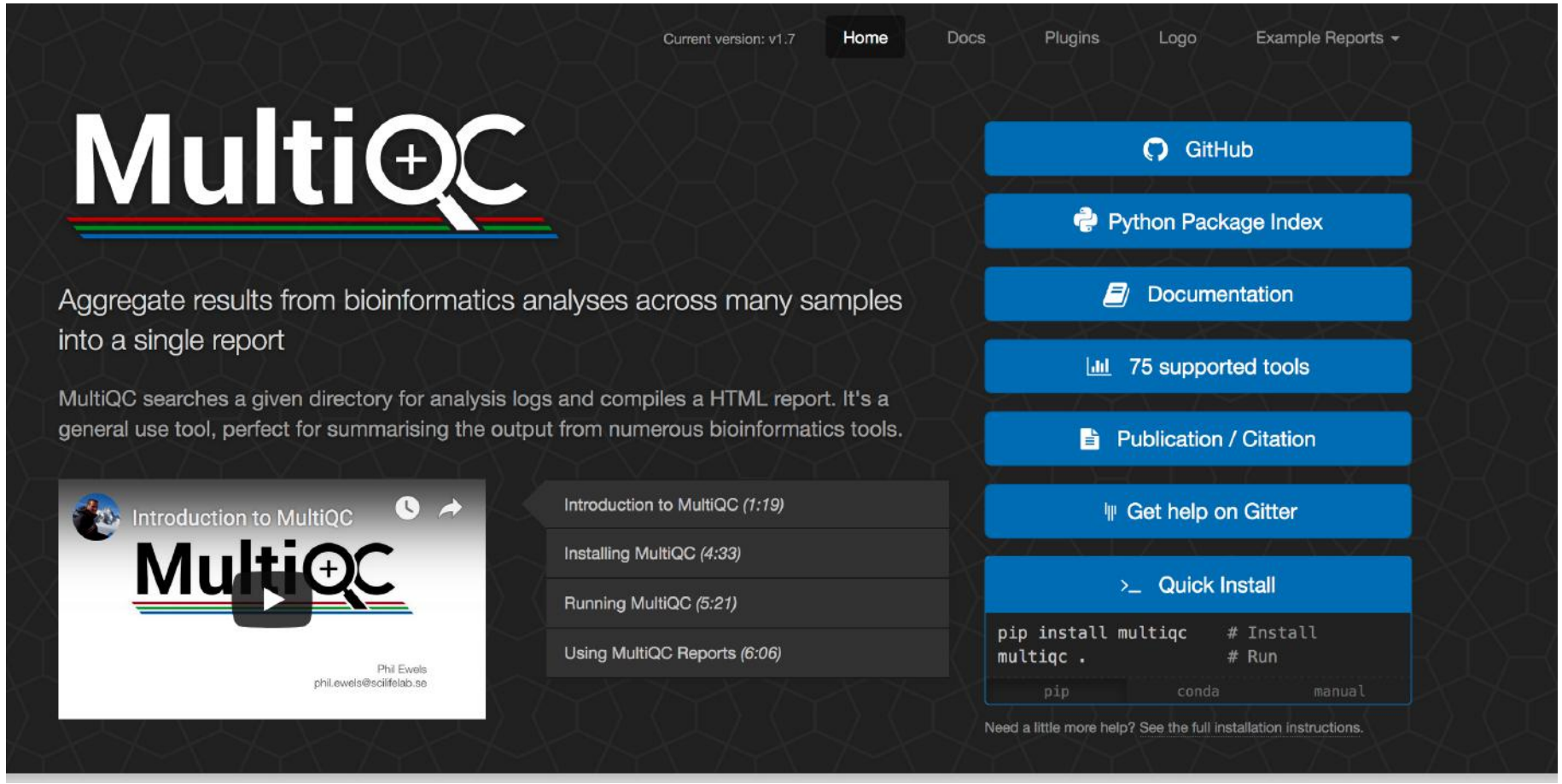


Summary of take-home messages (A short re-cap)

- We saw about the nature of NGS data – massively parallel sequencing and its challenges in terms of quality inference.
- A peek in to some concepts related to quality encoding.
- An example of how we could obtain the quality of a base from its corresponding quality string character.
- Introduced the FastQC tool.
- Visited the various modules in FastQC.
- **Importantly!** Always, evaluate and understand the quality of your sequenced data BEFORE ANY analysis.

Hands on QC using fastqc (both single fastq and multiple fastq files

MultiQC



To look at the summary qc report for all your samples.

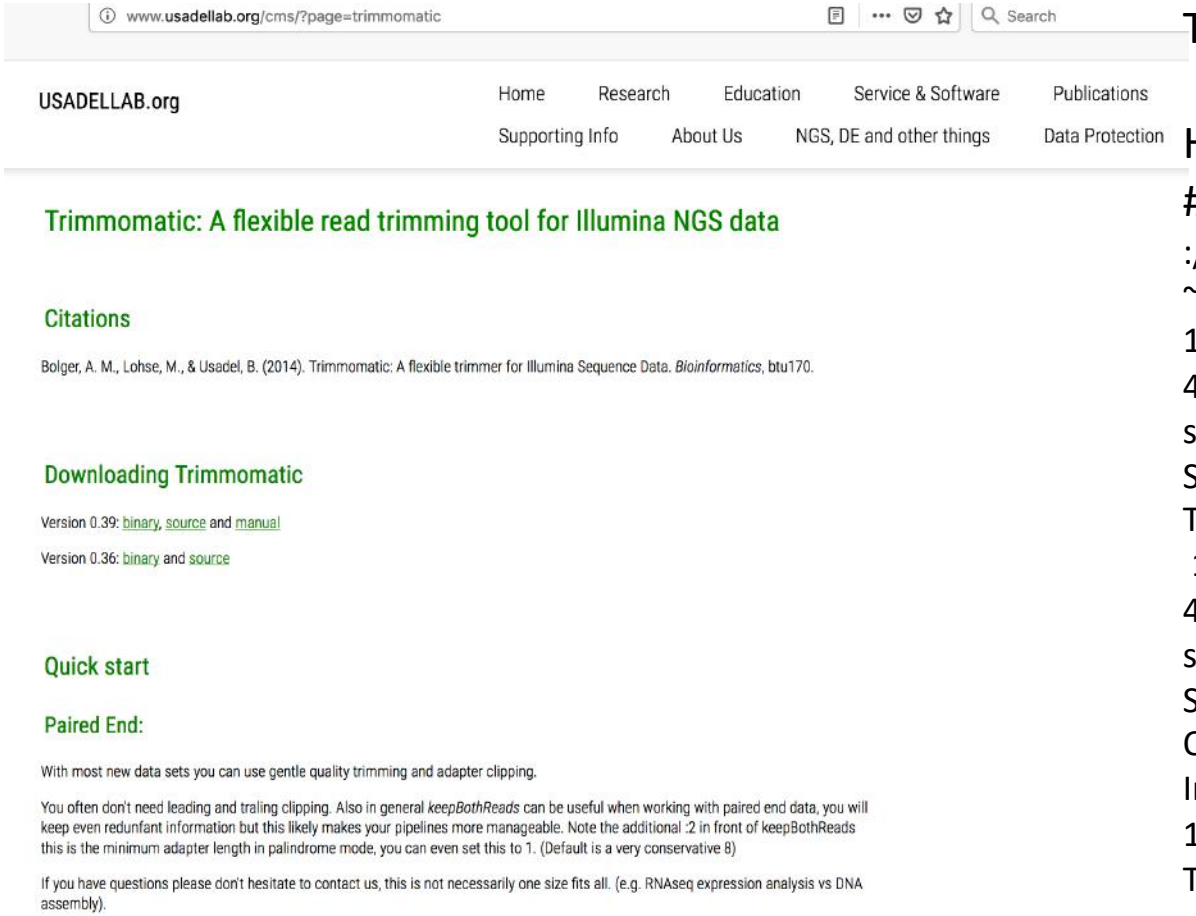
Command to run is
> Multiqc <your qc folder with fastqc output>

End of Module 2 QC

Module 3 : Data Trimming

- After scanning through the quality of your raw data, it is time to tailor your data cleaning strategy
- Remove low quality bases
- Trim off adapter and primer sequences
- make sure minimal read length to be carried for further down stream analysis.

Trimmomatic



This is java based tool, to trim your reads.

How to run:

####Trimming

:/data1/BESE_COURSE2019$ time java -jar ~/bioinformatics_tools/Trimmomatic-0.38/trimmomatic-0.38.jar SE 1_SMOC_RAWDATA/WT_NORMAL1_509.fastq.gz 4_TRIMMING/WT_NORMAL1_509_Trimmed.fastq.gz -threads 12 -summary WT_NORMAL1_Trimm_Summary.txt LEADING:3 TRAILING:3 SLIDINGWINDOW:4:30 MINLEN:75
TrimmomaticSE: Started with arguments:
1_SMOC_RAWDATA/WT_NORMAL1_509.fastq.gz 4_TRIMMING/WT_NORMAL1_509_Trimmed.fastq.gz -threads 12 -summary WT_NORMAL1_Trimm_Summary.txt LEADING:3 TRAILING:3 SLIDINGWINDOW:4:30 MINLEN:75
Quality encoding detected as phred33
Input Reads: 36746448 Surviving: 22973692 (62.52%) Dropped: 13772756 (37.48%)
TrimmomaticSE: Completed successfully
real 5m12.718s
user 8m43.928s
sys 0m36.734s

Hands on Trimmomatic and check quality after trimming

Module 4 : Mapping/aligning

- The Clean reads are next mapped to the reference genome of interest to get the quantification of genes/transcripts.

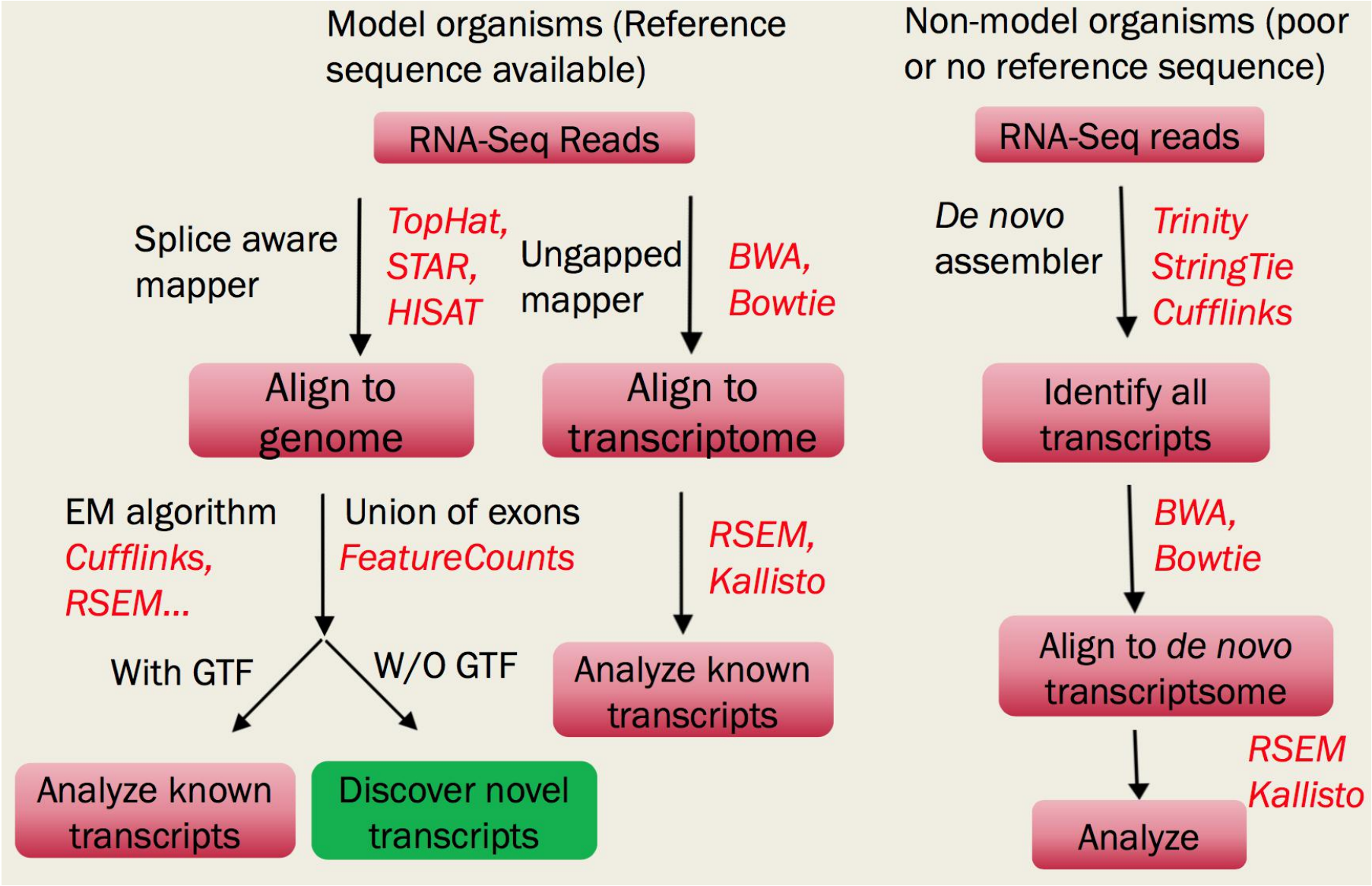
- We are going to use Tophat for this task.

- TopHat can identify splicing junctions without relying on database of known splice junctions.

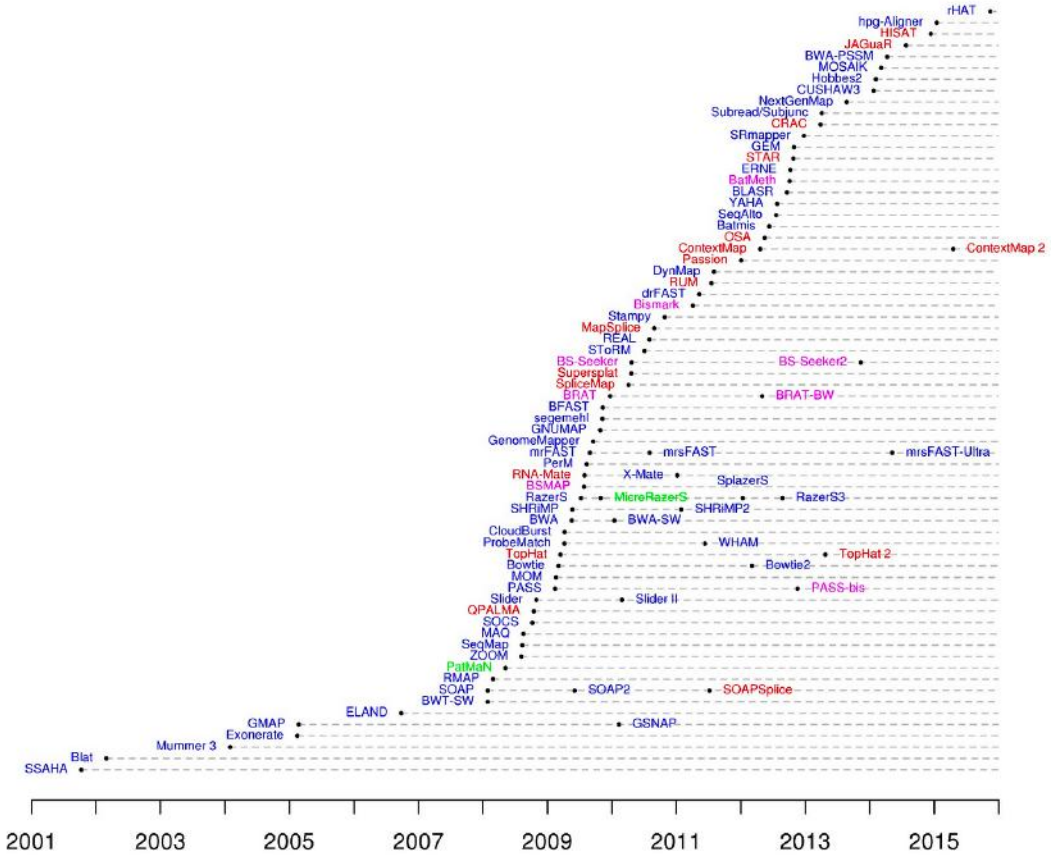
Which aligner and counting method?

Primary Analysis

- FASTQ
- FASTQC
- Read trimming
- Read alignment
- Quantification



Many Aligners available out there



How to compare read alignment software?

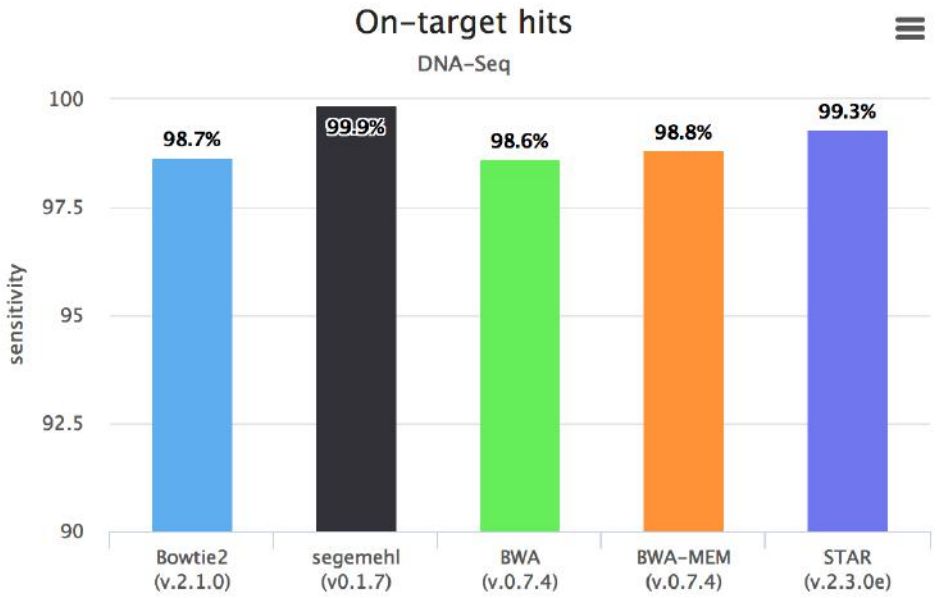
Every time a new read alignment software is developed and officially presented in a peer reviewed journal, the authors are asked to provide a comparison to existing tools. This is typically done in a benchmark where certain aspects of a software tool are assessed (ideally) in a scientifically sound manner. You can then compare these benchmarks and use them to decide on the optimal tool for your case. However, this procedure has its limitations: only a small set of the many aspects - typically things like mapping rate, sensitivity, speed - can be assessed in a short paper. And only certain program versions, parameter settings and data can be assessed.

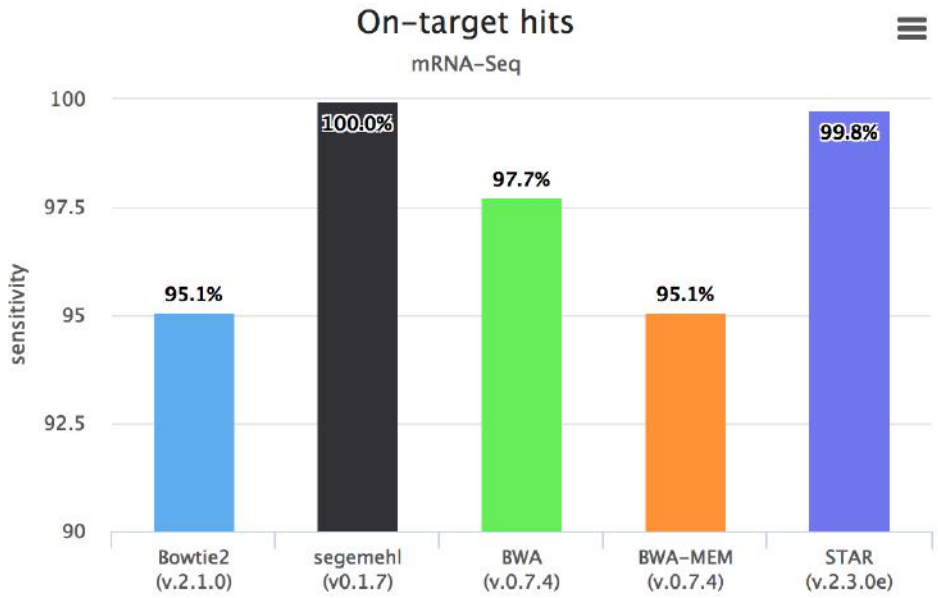
Comparing performance of aligners

Here, we provide detailed performance comparisons of NGS read aligners. In light of heated debates, we would like to stress that benchmarks only measure specific aspects and may not be used to claim any universal superiority or inferiority of a particular tool.

In order to compare different short read aligners, we use a published real-life paired-end DNA/RNA-Seq dataset. All optimal alignments (also multiple mapping loci) of 100,000 read pairs of each sample were obtained by RazerS 3 (full sensitivity mapping tool). In the benchmark shown below, we measured the performance in finding all optimal hits of different NGS mappers with default parameters. True positives are reads with up to 10 multiple mapping loci, allowing up to 10 errors (mismatches and indels).

Note that we explicitely want to find all multiple mapping loci in this benchmark and not only unique mapping loci or just one random hit of several. We believe that reads mapping multiple times should not be discarded since gene duplications and repeat regions are known to be biologically relevant.





Reference: https://www.ecseq.com/support/benchmark

How to decide on the best alignment software?

Assume you have a benchmark of your favorite alignment tools, what aspects should you look for? In general, you should try to answer the following questions:

1. What kind of sequences/experiment do you have? Do you have fragmented DNA inserts or spliced sequences from total RNA-seq? Is there a special protocol, DNA treatment or enrichment involved? What species is it and how is the quality of its genome assembly?
2. What sequencing platform do you have? Do you deal with Illumina,Ion Torrent reads or PacBio? Each machine has its characteristic read length and error types and not all mapping tools can handle them.
3. What kind of further analysis do you plan to carry out with the alignments? Do subsequent tools maybe depend on the reported alignment types and formats.
4. What kind of infrastructure do you have? Can you run your computations on a high performance computing cluster or does it need to run on your desktop computer?

Note that besides this “hard” benchmark there are also other factors to consider: is the output or input format of the program usable for you? Does the software have special features relevant for you? Is the program easy to use? Are there special license requirements or fees associated with it?

And the answer is...

...there is no best read aligner. It really depends on your goals and the specific case. What is the application? What sequencing technology has been used? What is the species? What are the computational constraints, etc.? You need to take into account the answer to those questions and then decide on the best read mapper according to the performance in the aspects important to you as well as in the software's features.

Alignment to reference genome

Alignment to reference genome/transcriptome

- **Goal is to find out where a read originated from**
  - Challenge: variants, sequencing errors, repetitive sequence
- **Mapping to**
  - transcriptome allows you to count hits to known transcripts
  - genome allows you to find new genes and transcripts
- **Many organisms have introns, so RNA-seq reads map to genome non-contiguously → spliced alignments needed**
  - Difficult because sequence signals at splice sites are limited and introns can be thousands of bases long

Old Tuxedo tools

NATURE PROTOCOLS | PROTOCOL

Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks

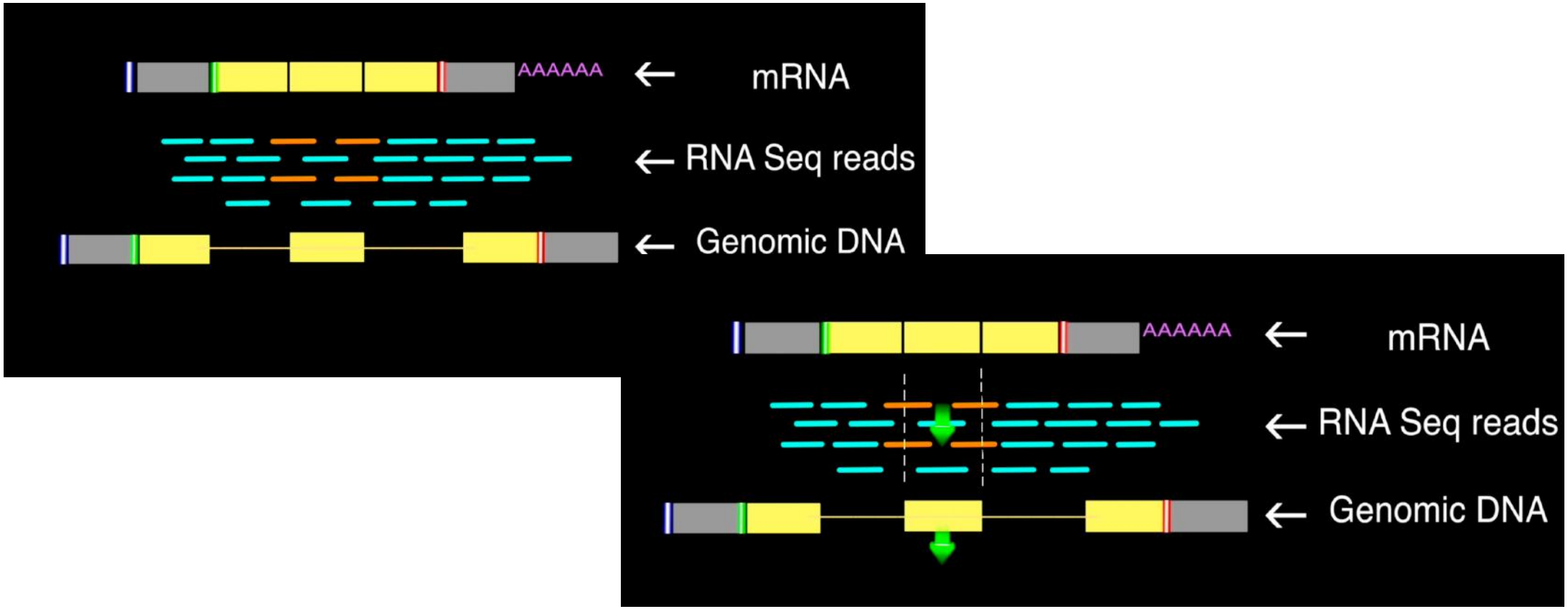
Cole Trapnell, Adam Roberts, Loyal Goff, Geo Pertea, Daehwan Kim, David R Kelley, Harold Pimentel, Steven L Salzberg, John L Rinn & Lior Pachter

Affiliations | Contributions | Corresponding author

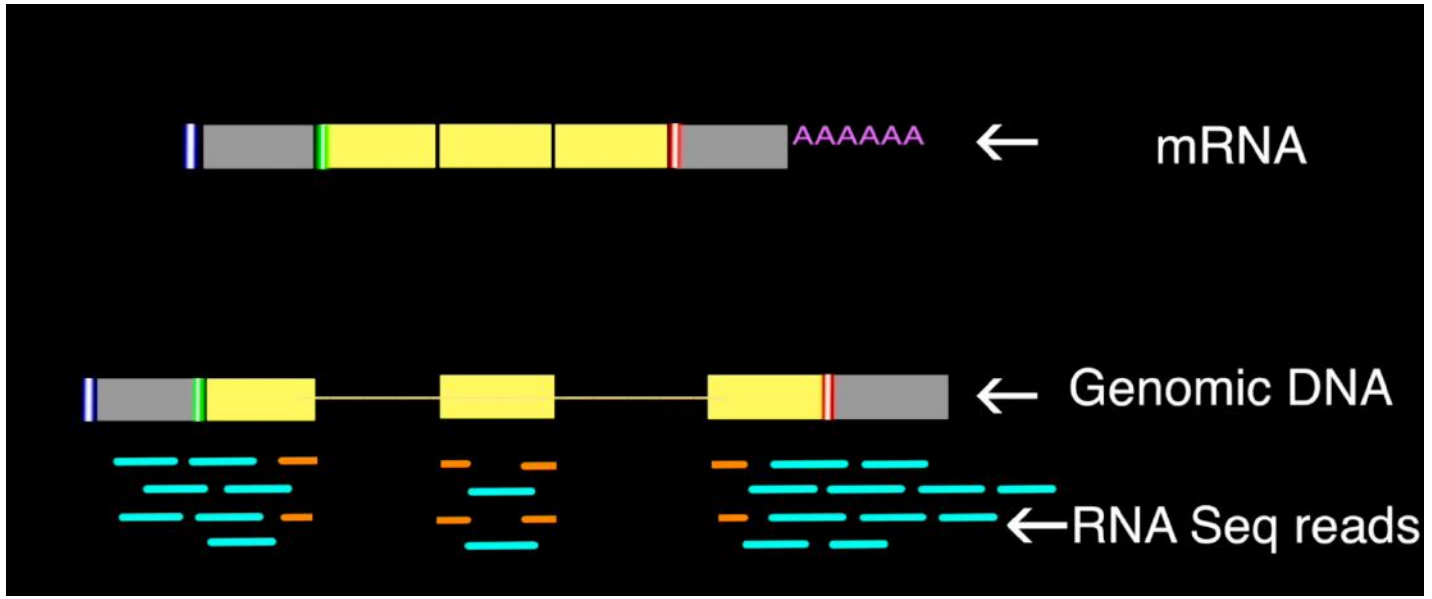
Nature Protocols **7**, 562–578 (2012) | doi:10.1038/nprot.2012.016
Published online 01 March 2012

Abstract

Recent advances in high-throughput cDNA sequencing (RNA-seq) can reveal new genes and splice variants and quantify expression genome-wide in a single assay. The volume and complexity of data from RNA-seq experiments necessitate scalable, fast and mathematically principled analysis software. TopHat and Cufflinks are free, open-source software tools for gene discovery and comprehensive expression analysis of high-throughput mRNA sequencing (RNA-seq) data. Together, they allow biologists to identify new genes and new splice variants of known ones, as


AAAAAA
mRNA
RNA Seq reads
Genomic DNA
AAAAAA
mRNA
RNA Seq reads
Genomic DNA

Reads containing splicing junctions



TopHat builds the database of splice junctions from the alignment output.

By this method, TopHat can identify novel splice junctions as well.

Splice-aware aligners

- **TopHat, HISAT (use Bowtie aligner internally)**
- **STAR**
- **GSNAP**
- **RUM**
- **MapSplice**

...

Systematic evaluation of spliced alignment programs for RNA-seq data

Pär G Engström[1,13], Tamara Steijger[1], Botond Sipos[1], Gregory R Grant[2,3], André Kahles[4,5], The RGASP Consortium[6], Gunnar Rätsch[4,5], Nick Goldman[1], Tim J Hubbard[7], Jennifer Harrow[7], Roderic Guigó[8,9] & Paul Bertone[1,10–12]

Nature methods 2013 (10:1185)

Mapping quality

- **Confidence in read's point of origin**
- **Depends on many things, including**
  - uniqueness of the aligned region in the genome
  - length of alignment
  - number of mismatches and gaps

- **Expressed in Phred scores, like base qualities**
  - $Q = -10 * \log_{10}$ (probability that mapping location is wrong)

- **TopHat mapping qualities**
  - 50 = unique mapping
  - 3 = maps to 2 locations
  - 1 = maps to 3-4 locations
  - 0 = maps to 5 or more locations

Mapping with Tophat
(more parameters than you care to care about)

- What do you absolutely need to specify for each run?

  - The genome index file (just the prefix)
  - The reads files, one for each paired end
  - The transcript files (gtf format)
  - The insert size (-r). This is the size of the fragment minus reads length
  - The library type (stranded or not, illumina-like or not)

Mapping with Tophat

… more parameters than you'd care to care about.

You need to give Tophat a reference genome for mapping your reads.

We actually give it an "index file" which is a compressed map of the genome.
There are 6 genome index files.
If you built these with bowtie1-build, they look like this

Just specify the prefix, e.g.: hg19

The genome index file (just the prefix)
The reads files, one for each paired end
The transcript files (gtf format)
The insert size (-r). This is the size of the fragment minus reads length
The library type (stranded or not, illumina-like or not)

Mapping with Tophat

… more parameters than you'd care to care about.

Next, we give Tophat your actual sequenced reads.

These are FastQ files, which contain both sequence information as well as quality scores for each read.
For paired end reads, we tell Tophat about both of them, (pair the reads) so it can try to map them together.

Example: First WT replicate (rep1); Read 1 paired with Read 2

wt-rep1_R1.fq.gz
wt-rep1_R2.fq.gz

The genome index file (just the prefix)
The reads files, one for each paired end
The transcript files (gtf format)
The insert size (-r). This is the size of the fragment minus reads length
The library type (stranded or not, illumina-like or not)

Mapping with Tophat

… more parameters than you'd care to care about.

Tophat works much better if you give it the exon/intron structure of known genes, i.e., a gtf file.

You can still choose to find novel transcripts later, but providing this file makes it easier to find reads that span known introns, which makes the search faster.

Example: the human file is hg19_refseq_genes.gtf

****Be careful – tophat is very picky about the format of this file. Either download one from their website, or get someone to give you one that plays nicely with tophat.**

The genome index file (just the prefix)
The reads files, one for each paired end
The transcript files (gtf format)
The insert size (-r). This is the size of the fragment minus reads length
The library type (stranded or not, illumina-like or not)

Mapping with Tophat

… more parameters than you'd care to care about.

Mate Inner Distance

Let's say your RNAseq library prep kit preferentially fragments the RNA into 500 bp fragments. Then, you did a PE100bp run.

Your "insert size" is 500 – 100 – 100 = **300**.

You did a PE300 run? Okay, your insert size is 0.

Why does Tophat care? Tophat judges how well the two ends of a paired-end segment mapped by whether the inferred distance between them is consistent with the expected insert size. This is part of "mapping quality."

The genome index file (just the prefix)

The reads files, one for each paired end

The transcript files (gtf format)

The mate inner distance (-r). This "insert size" is the size of the fragment minus reads length

The library type (stranded or not, illumina-like or not)

Mapping with Tophat

… more parameters than you'd care to care about.

In general, an Illumina Tru-seq Stranded RNA-seq library prep kit should use **fr-firststrand.** That means the library is stranded and the complementary strand is the first one sequenced. In other words, all your reads are the reverse-complement of the transcriptome!

The other options:

fr-unstranded	common for non-stranded libraries
fr-secondstrand	used for ABI Solid libraries – not common

The genome index file (just the prefix)
The reads files, one for each paired end
The transcript files (gtf format)
The insert size (-r). This is the size of the fragment minus reads length
The library type (stranded or not, illumina-like or not)

Mapping with Tophat

What other options might you care to change?

Preset default options

-N 2 number of mismatches per read (default is 2)

-g 2 maximum number of alignments per read to report (default is 2)

--suppress-hits set this if you want to suppress reads that map more than max (2)

--no-mixed don't report an alignment if you can't map both ends of the fragment

--no-novel-juncs don't look for novel splice junctions – just use the ones in the GTF file
*(if you will be skipping CuffLinks, you must run TopHat Advanced and choose this option to **not** look for novel junction)*

Mapping with Tophat

What should your output look like?

A typical tophat output directory should have these files:
(The output files will be in your Project folder in the Data Store)

accepted_hits.bam deletions.bed junctions.bed **logs**
insertions.bed left_kept_reads.info right_kept_reads.info

The bam file is your main output – the aligned reads.
It's in binary sam format.
Here is an example of what one line of a bam file looks like:

read name | map flags | map position chr/start | map quality | other PE read is mapped to same chr (=) at pos 709587, which is 699060 bp away

```
MENDEL_0001_FC61FR7AAXX7:69:18748:7104#0        81          chr1        10563       255         36M         =
            709587      699060      CGCAGCTCCGCCCTCGCGGTGCTCTCCGGGTCTGTG
            CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC                NM:i:0      NH:i:1
```

read sequence | read qualities | tophat flags

NM:i:0 = 0 mismatches
NH:i:1 = aligns uniquely

Genome Annotation and output bam file

```
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";
chr12   unknown exon    96077274        96077487        .       -       .
gene_id "NTN4"; gene_name "NTN4"; p_id "P12149"; transcript_id
"NM_021229"; tss_id "TSS6395";
chr12   unknown CDS     96104219        96104407        .       -       2
gene_id "NTN4"; gene_name "NTN4"; p_id "P12149"; transcript_id
"NM_021229"; tss_id "TSS6395";
chr12   unknown exon    96104219        96104407        .       -       .
gene_id "NTN4"; gene_name "NTN4"; p_id "P12149"; transcript_id
"NM_021229"; tss_id "TSS6395";
```

```
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
CCCFFFFFFGFFHHJGIHHJJJFGGJJGIIIIIIGJJJJJJJJJIJJJJJE    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
```

Hands on For Mapping

Output looks like this:

[2019-07-12 16:14:43] Beginning TopHat run (v2.1.1)

[2019-07-12 16:14:43] Checking for Bowtie

Bowtie version: 2.2.6.0

[2019-07-12 16:14:43] Checking for Bowtie index files (genome)..

[2019-07-12 16:14:43] Checking for reference FASTA file

[2019-07-12 16:14:43] Generating SAM header for /home/thimmamp/References/mm10/Mus_musculus/UCSC/mm10/Sequence/Bowtie2Index/genome

[2019-07-12 16:15:11] Preparing reads

left reads: min. length=75, max. length=75, 34183396 kept reads (5258 discarded)

[2019-07-12 16:20:16] Mapping left_kept_reads to genome genome with Bowtie2

[2019-07-12 16:32:56] Mapping left_kept_reads_seg1 to genome genome with Bowtie2 (1/3)

[2019-07-12 16:34:08] Mapping left_kept_reads_seg2 to genome genome with Bowtie2 (2/3)

[2019-07-12 16:35:21] Mapping left_kept_reads_seg3 to genome genome with Bowtie2 (3/3)

[2019-07-12 16:36:31] Searching for junctions via segment mapping

[2019-07-12 16:38:46] Retrieving sequences for splices

[2019-07-12 16:39:56] Indexing splices

Building a SMALL index

[2019-07-12 16:40:07] Mapping left_kept_reads_seg1 to genome segment_juncs with Bowtie2 (1/3)

[2019-07-12 16:40:43] Mapping left_kept_reads_seg2 to genome segment_juncs with Bowtie2 (2/3)

[2019-07-12 16:41:17] Mapping left_kept_reads_seg3 to genome segment_juncs with Bowtie2 (3/3)

[2019-07-12 16:41:47] Joining segment hits

[2019-07-12 16:43:47] Reporting output tracks

Mapping QC

Information we need to check

- Percentage of reads properly mapped or uniquely mapped
- Among the mapped reads, the percentage of reads in exon, intron, and intergenic regions.
- 5' or 3' bias
- The percentage of expressed genes

Genome analysis **2012, Bioinformatics** Advance Access publication April 25, 2012

RNA-SeQC: RNA-seq metrics for quality control and process optimization

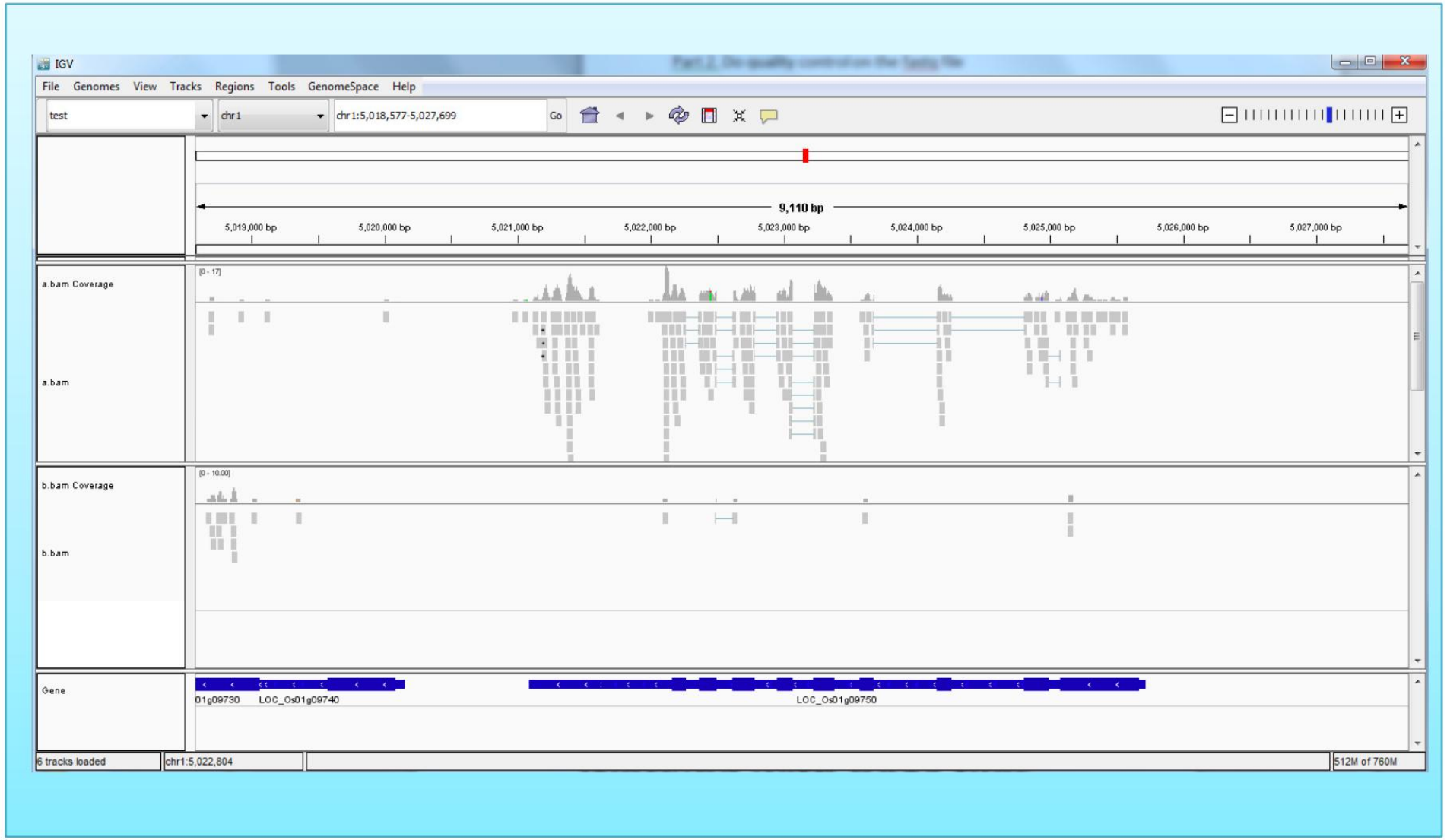
David S. DeLuca*, Joshua Z. Levin, Andrey Sivachenko, Timothy Fennell,
Marc-Danie Nazaire, Chris Williams, Michael Reich, Wendy Winckler and Gad Getz*
The Broad Institute of MIT and Harvard, Cambridge, MA, USA

- Read Metrics
  - Total, unique, duplicate reads
  - Alternative alignment reads
  - Read Length
  - Fragment Length mean and standard deviation
  - Read pairs: number aligned, unpaired reads, base mismatch rate for each pair mate, chimeric pairs
  - Vendor Failed Reads
  - Mapped reads and mapped unique reads
  - rRNA reads
  - Transcript-annotated reads (intragenic, intergenic, exonic, intronic)
  - Expression profiling efficiency (ratio of exon-derived reads to total reads sequenced)
  - Strand specificity
- Coverage
  - Mean coverage (reads per base)
  - Mean coefficient of variation
  - 5'/3' bias
  - Coverage gaps: count, length
  - Coverage Plots
- Downsampling
- GC Bias
- Correlation:
  - Between sample(s) and a reference expression profile
  - When run with multiple samples, the correlation between every sample pair is reported

https://confluence.broadinstitute.org/display/CGATools/RNA-SeQC

Visualizing BAM files with IGV

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



Quality of mapping

Mapping quality

- **Confidence in read's point of origin**
- **Depends on many things, including**
  - uniqueness of the aligned region in the genome
  - length of alignment
  - number of mismatches and gaps

- **Expressed in Phred scores, like base qualities**
  - $Q = -10 * \log_{10}$ (probability that mapping location is wrong)

- **TopHat mapping qualities**
  - 50 = unique mapping
  - 3 = maps to 2 locations
  - 1 = maps to 3-4 locations
  - 0 = maps to 5 or more locations

How to look at the quality of mapping

```
(base) thimmamp@kw60284:/data1/BESE_COURSE2019$ samtools view 7_MAPPING/WT_NORMAL1_509_tophathout/accepted_hits.bam | grep "
NH:i:1" | wc -l
28843809
(base) thimmamp@kw60284:/data1/BESE_COURSE2019$ samtools view 7_MAPPING/WT_NORMAL1_509_tophathout/accepted_hits.bam | grep "
NH:i:1" | head
SRR4000509.26291084     0       chr1    3031449 0       75M     *       0       0       GCCAGAGTGCAAGAAGCAAGAGAGCAAGAAGCAAGA
GAGAGAGAAAACGAAACCCCGTCCCTATTAAGGAGAATT AAAAAEEAEEEEEEEEEEEEEAEEEEEEEEEEEEEEEEEEEEEEEEEAA/EEEEEEEEEEEEEEEEEEE/E     AS:i
:0      XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:75 YT:Z:UU NH:i:17 CC:Z:=  CP:i:181263616  HI:i:0
SRR4000509.6408277      272     chr1    3053490 0       75M     *       0       0       GTGCTCGCCTCCTAGCCCTTACTGAAGATTTTTTAG
AAAGCTCCCAATCGCCATCTCACAAAAGCAAAACTAAGC E/EAAEAEEEEEEEEAEEEEAEEEEEEEEEEEAAEEEEEEEEEEEEAEEEEEEEEEEAEEEEEEEEEEEEEAAAA6     AS:i
:0      XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:75 YT:Z:UU NH:i:13 CC:Z:chr15      CP:i:9225541    HI:i:0
SRR4000509.19622941     272     chr1    3053490 0       75M     *       0       0       GTGCTCGCCTCCTAGCCCTTACTGAAGATTTTTTAG
AAAGCTCCCAATCGCCATCTCACAAAAGCAAAACTAAGC A/EAAEEEAEEEEEEEEEEEEEE<EAEEEEEEEEEEEEEEEEEEEEEEEEEEEE6EEEEEEEEEAEAAAAA     AS:i
:0      XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:75 YT:Z:UU NH:i:13 CC:Z:chr15      CP:i:9225541    HI:i:0
SRR4000509.16564974     272     chr1    3054452 0       75M     *       0       0       CTCCCACTATTCGGCCATTTCCCTTATATGAAAGAG
GAGTGAGGACCTTCCTCCCTGGGATCCTCAGAAGTCTAC EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEAEEEEEAAAAA     AS:i
:-5     XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:39C35      YT:Z:UU NH:i:16 CC:Z:=  CP:i:16833476   HI:i:0
SRR4000509.26549395     272     chr1    3054480 0       75M     *       0       0       TGAAAGAGGAGTGAGGACCTTCCTCCCTGGGATCCT
CAGAAGTCTACTGGGCACAAAAAATAACTGATTCATGGC E<EEAEEEEEE/EEEEE<EEEEEEEEEEEEEEEEAEEEEEEEEEEEEEEEEEEEEEEE6EEEEEEEEEAAAAA     AS:i
:-8     XN:i:0  XM:i:2  XO:i:0  XG:i:0  NM:i:2  MD:Z:11C43G19   YT:Z:UU NH:i:15 CC:Z:=  CP:i:16833448   HI:i:0
SRR4000509.6304667      272     chr1    3056106 0       75M     *       0       0       ATAGCTATAAAAAGTAAAAGTCTTTTATTGATCTAA
ATTTATAAGATTCGTGAAGCCACTTCATTTGGTTTTTGA EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEAAAAA     AS:i
:-5     XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:25A49      YT:Z:UU NH:i:18 CC:Z:chr12      CP:i:23701234   HI:i:0
SRR4000509.34245684     16      chr1    3057761 50      75M     *       0       0       ACCATGGGCTGCAGCAAGGAGTACCAACCTGGTGCC
AACTTAAAATAAAAACAACTAAAAAAGCTGACTCCTGAG EEEEEEEEEEEEEEEEEEEEEEEEAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEAAAAA     AS:i
:-5     XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:70G4       YT:Z:UU NH:i:1
SRR4000509.28556553     272     chr1    3058013 0       75M     *       0       0       TTCTTGTCAATTAACTAATTGTGTAGATCCTAATTT
AGATAAGGAATCTACTGTTATGTTTTTGTAAGAGACCTG EEEEEEEEEEAE<EEAEE/EEEAEEEEEAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEAAAAA     AS:i
:0      XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:75 YT:Z:UU NH:i:11 CC:Z:chr3       CP:i:24237730   HI:i:0
SRR4000509.27411499     256     chr1    3066281 0       75M     *       0       0       CCATTTGTCAATTCTCGATCTTACAGCACAAGCCAT
TACTGTTCTGTTCAGGAATTTTTCCCCTGTGCCCATATC AAAAAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE/EEEEEEEEEEEEEEEEEEEEEEEEEEE<EEEEEE     AS:i
:0      XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:75 YT:Z:UU NH:i:13 CC:Z:chr10      CP:i:65670410   HI:i:0
SRR4000509.22830477     256     chr1    3087813 0       75M     *       0       0       GTCCTCTCTTTCCAGTGATGGCCGACTAGGCCATCT
TTTGATACATATGCAGCTAGAGTCAAGAGCTCCGGGGTA AAAAAEEEEAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE     AS:i
:0      XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:75 YT:Z:UU NH:i:17 CC:Z:=  CP:i:94587835   HI:i:0
```

Look at the align_summary.txt file under each sample

End of Module 4 : Mapping

Next Module 5: Quantification

Quantification of Reads Mapped to Genome

TBC in next session…

Counting rules

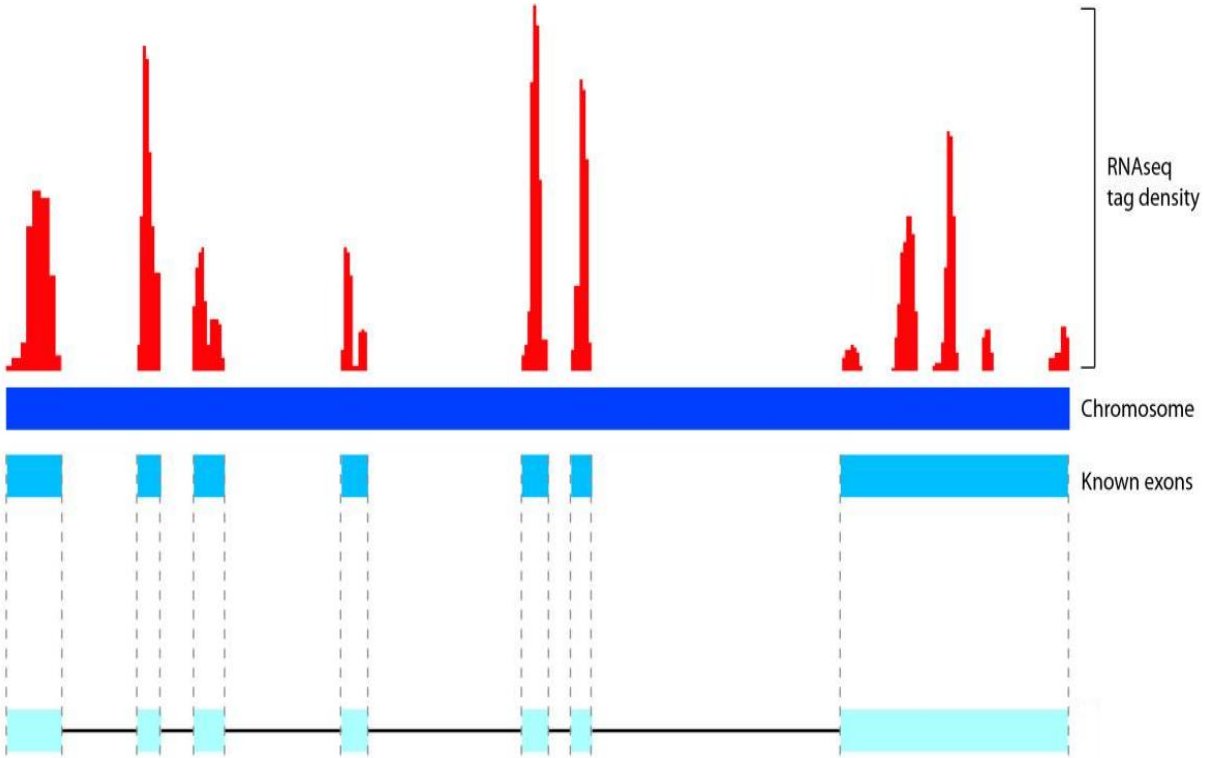
- Count reads, not base-pairs
- Count each read at most once.
- Discard a read if
  - it cannot be uniquely mapped
  - its alignment overlaps with several genes
  - the alignment quality score is bad
  - (for paired-end reads) the mates do not map to the same gene

Expression quantification

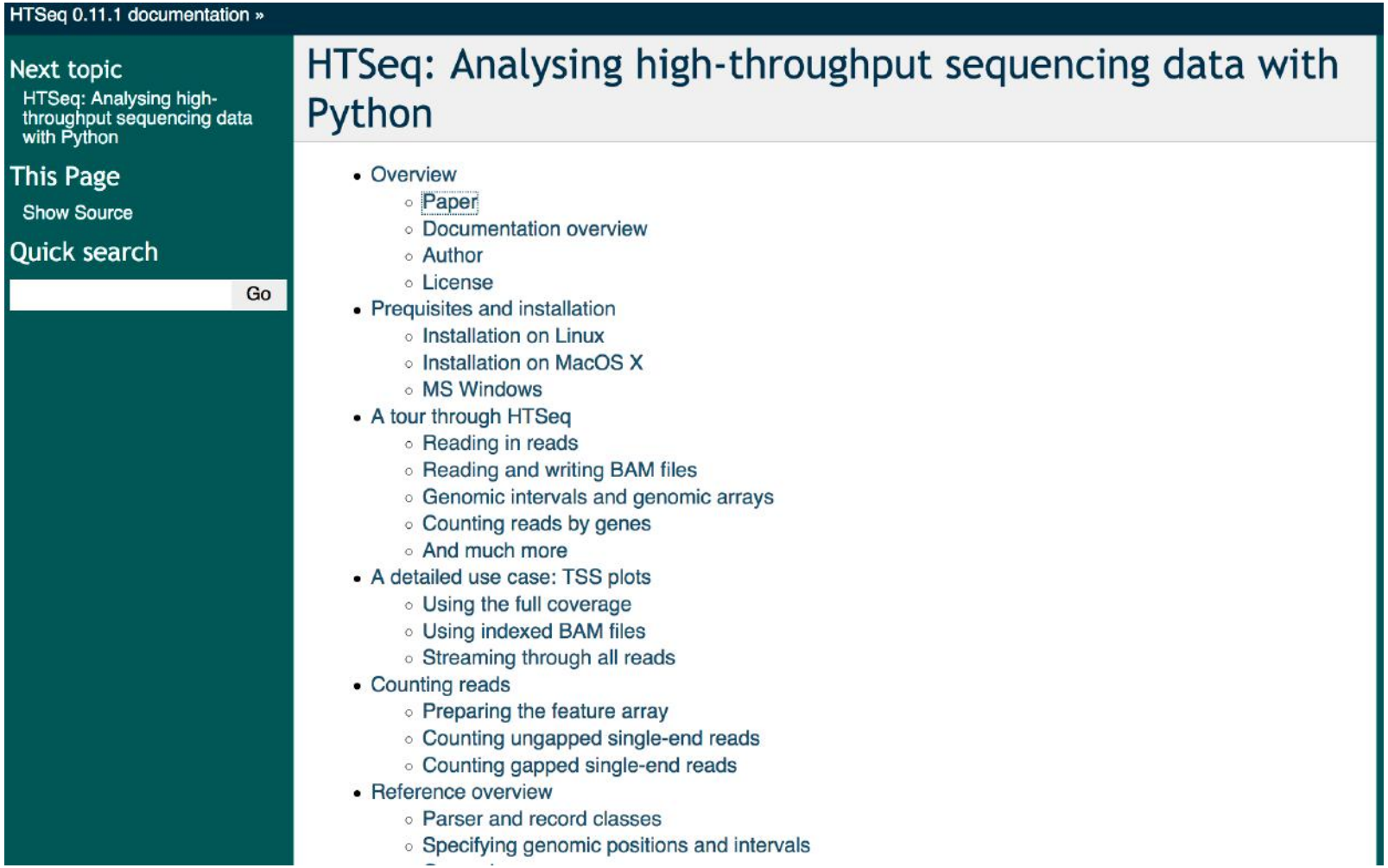
FPKM /RPKM

- Count data
  - Summarized mapped reads to CDS, gene or exon level

$$FPKM = \frac{Counts\ of\ mapped\ fragments}{Total\ mapped\ fragments\ (million) \times Exon\ length\ of\ transcript\ (KB)}$$

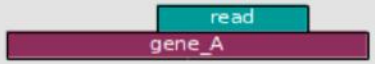
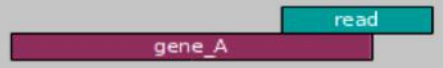
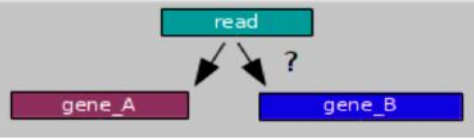


Module 5: Quantification using HTSEQ-Count



How to assign reads to features

The following figure illustrates the effect of these three modes and the `--nonunique` option:

	union	intersection_strict	intersection_nonempty
	gene_A	gene_A	gene_A
	gene_A	no_feature	gene_A
	gene_A	no_feature	gene_A
	gene_A	gene_A	gene_A
	gene_A	gene_A	gene_A
	ambiguous (both genes with --nonunique all)	gene_A	gene_A
	ambiguous (both genes with --nonunique all)		
	alignment_not_unique (both genes with --nonunique all)		

Install HTSeq python module

Prequisites and installation

HTSeq is available from the Python Package Index (PyPI):

To use HTSeq, you need Python 2.7 or 3.4 or above (3.0-3.3 are not supported), together with:

- NumPy, a commonly used Python package for numerical calculations
- Pysam, a Python interface to samtools.
- To make plots you will need matplotlib, a plotting library.

At the moment, HTSeq supports Linux and OSX but not Windows operating systems, because one of the key dependencies, Pysam, lacks automatic support and none of the HTSeq authors have access to such a machine. However, it *might* work with some work, if you need support for this open an issue on our Github page.

HTSeq follows install conventions of many Python packages. In the best case, it should install from PyPI like this:

```
pip install HTSeq
```

If this does not work, please open an issue on Github and also try the instructions below.

Installation on Linux

You can choose to install HTSeq via your distribution packages or via *pip*. The former is generally recommended but might be updated less often than the *pip* version.

Distribution package manager

- Ubuntu (e.g. for Python 2.7):

```
sudo apt-get install build-essential python2.7-dev python-numpy python-matplotlib python-pysam py
```

- Arch (e.g. using `aura`, you can grab the AUR packages otherwise):

```
sudo pacman -S python python-numpy python-matplotlib
sudo aura -A python-pysam python-htseq
```

Counting with htseq-count

Counting reads in features with `htseq-count`

Given a file with aligned sequencing reads and a list of genomic features, a common task is to count how many reads map to each feature.

A feature is here an interval (i.e., a range of positions) on a chromosome or a union of such intervals.

In the case of RNA-Seq, the features are typically genes, where each gene is considered here as the union of all its exons. One may also consider each exon as a feature, e.g., in order to check for alternative splicing. For comparative ChIP-Seq, the features might be binding region from a pre-determined list.

Special care must be taken to decide how to deal with reads that align to or overlap with more than one feature. The `htseq-count` script allows to choose between three modes. Of course, if none of these fits your needs, you can write your own script with HTSeq. See the chapter A tour through HTSeq for a step-by-step guide on how to do so. See also the FAQ at the end, if the following explanation seems too technical.

The three overlap resolution modes of `htseq-count` work as follows. For each position *i* in the read, a set *S(i)* is defined as the set of all features overlapping position *i*. Then, consider the set *S*, which is (with *i* running through all position within the read or a read pair)

- the union of all the sets *S(i)* for mode `union`. This mode is recommended for most use cases.
- the intersection of all the sets *S(i)* for mode `intersection-strict`.
- the intersection of all non-empty sets *S(i)* for mode `intersection-nonempty`.

If *S* contains precisely one feature, the read (or read pair) is counted for this feature. If *S* is empty, the read (or read pair) is counted as `no_feature`. If *S* contains more than one feature, `htseq-count` behaves differently based on the `--nonunique` option:

- `--nonunique none` (default): the read (or read pair) is counted as `ambiguous` and not counted for any features. Also, if the read (or read pair) aligns to more than one location in the reference, it is scored as `alignment_not_unique`.
- `--nonunique all`: the read (or read pair) is counted as `ambiguous` and is also counted in all features to which it was assigned. Also, if the read (or read pair) aligns to more than one location in the reference, it is scored as `alignment_not_unique` and also separately for each location.

Notice that when using `--nonunique all` the sum of all counts will not be equal to the number of reads (or read pairs), because those with multiple alignments or overlaps get scored multiple times.

The following figure illustrates the effect of these three modes and the `--nonunique` option:

Hands on counting mapped reads per gene

Run HTSeq on our bam file

Hands on

htseq-count -i gene_id --mode=union --nonunique=none --format=bam <bam file> <reference_genome_gtf_file> > samplename_count_table.txt

Output:

- head 8_COUNTING/WT_NORMAL2_510_count_table.txt
- Gene WT_NORMAL2_510
- 0610005C13Rik 3
- 0610007P14Rik 25
- 0610009B22Rik 0
- 0610009L18Rik 2
- 0610009O20Rik 12
- 0610010B08Rik 0
- 0610010F05Rik 0
- 0610010K14Rik 5
- 0610011F06Rik 2

Merge the counts from all samples into one file

- You need to know how to write python or R script to merge all the count files from individual samples into one single master count file!

```
smoc2_norm1 <- read.table(file = "SMOC2_NORMAL1_502_count_table.txt", header=TRUE,sep = "\t")
smoc2_norm3 <- read.table(file = "SMOC2_NORMAL3_503_count_table.txt", header=TRUE, sep = "\t")
smoc2_norm4 <- read.table(file = "SMOC2_NORMAL4_504_count_table.txt", header=TRUE,sep = "\t")
smoc2_uuo1 <- read.table(file = "SMOC2_UUO1_505_count_table.txt", header=TRUE,sep = "\t")
smoc2_uuo2 <- read.table(file = "SMOC2_UUO2_506_count_table.txt", header=TRUE,sep = "\t")
smoc2_uuo3 <- read.table(file = "SMOC2_UUO3_507_count_table.txt", header=TRUE,sep = "\t")
smoc2_uuo4 <- read.table(file = "SMOC2_UUO4_508_count_table.txt", header=TRUE,sep = "\t")
wt_norm1 <- read.table(file = "WT_NORMAL1_509_count_table.txt", header=TRUE,sep = "\t")
wt_norm2 <- read.table(file = "WT_NORMAL2_510_count_table.txt", header=TRUE,sep = "\t")
wt_norm3 <- read.table(file = "WT_NORMAL3_511_count_table.txt", header=TRUE,sep = "\t")
wt_uuo1 <- read.table(file = "WT_UUO1_512_count_table.txt", header=TRUE,sep = "\t")
wt_uuo2 <- read.table(file = "WT_UUO2_513_count_table.txt", header=TRUE,sep = "\t")
wt_uuo3 <- read.table(file = "WT_UUO3_514_count_table.txt", header=TRUE,sep = "\t")
wt_uuo4 <- read.table(file = "WT_UUO4_515_count_table.txt", header=TRUE,sep = "\t")
mergeCol = c("Gene")
 #<- merge(smoc2_norm1, smoc2_norm3, by=mergeCol, all = TRUE)
final <- Reduce(function(x, y) merge(x, y, by=mergeCol, all=TRUE), list(smoc2_norm1, smoc2_norm3,
                                                                        smoc2_norm4, smoc2_uuo1, smoc2_uuo2, smoc2_uuo3, sm
oc2_uuo4,

                                                                        wt_norm1, wt_norm2, wt_norm3,
                                                                        wt_uuo1, wt_uuo2, wt_uuo3, wt_uuo4))


mydf <- final[, -1] # all sample columns except gene name
rownames(mydf) <- final[,1] # now get gene column
head(mydf)
dataForDE <- mydf[rowSums(mydf)>0, ] # filter out genes with 0 for all samples
write.table(dataForDE, file = "Samples_Merged.txt", sep = "\t")
```

End of Quantification

Next Module 6: DE Analysis


THANK YOU!
QUESTIONS?