Introduction to computational biology and R

Adam Sorbie

Institute for Stroke and Dementia Research
Munich, Germany

Neuroimmunological methods in stroke 14.02.23

Funded by: Solorz-Żak Research Foundation

Course objectives

- Aimed at students in biological/medical sciences (any discipline).
- Develop an understanding of the basics of computational biology.
- Practical training:
  - Introduction to R
  - RNAseq analysis
  - 16S rRNA sequencing analysis

Lecture Outline

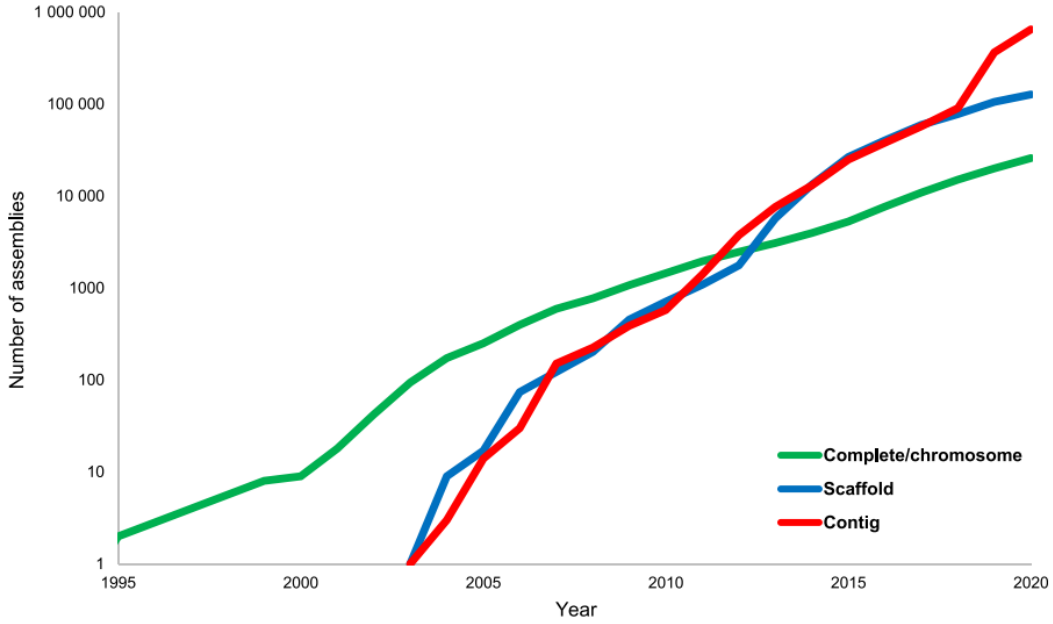
1. Introduction to Computational biology
2. Introduction to the R programming language – theory and concepts
3. Introduction to RNAseq
4. Introduction to 16S rRNA sequencing

What is computational biology and why is it important?

- Computational biology –
  - Analysis of complex, high-dimensional biological data.
  - Discovery of new biological insights.
  - Comp. bio vs bioinformatics
    - bioinformatics mostly focused on software and algorithm development

- Complex omics datasets increasingly common
  - Rare to see papers without some sort of NGS/mass spec dataset.

- Need for people who can understand and extract insights from these datasets.

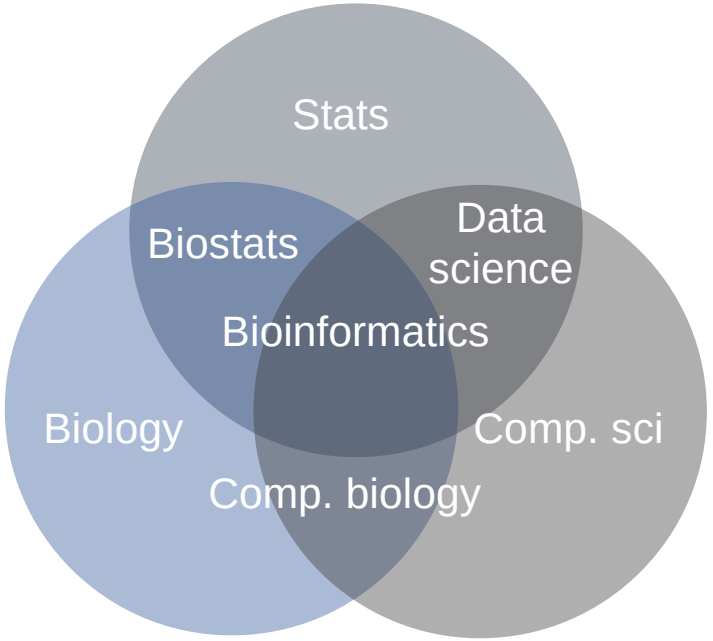


Source: https://www.genome.gov/about-genomics/fact-sheets/Sequencing-Human-Genome-cost

Koonin, Makarova and Wolf, Trends in Microbiology, 2021

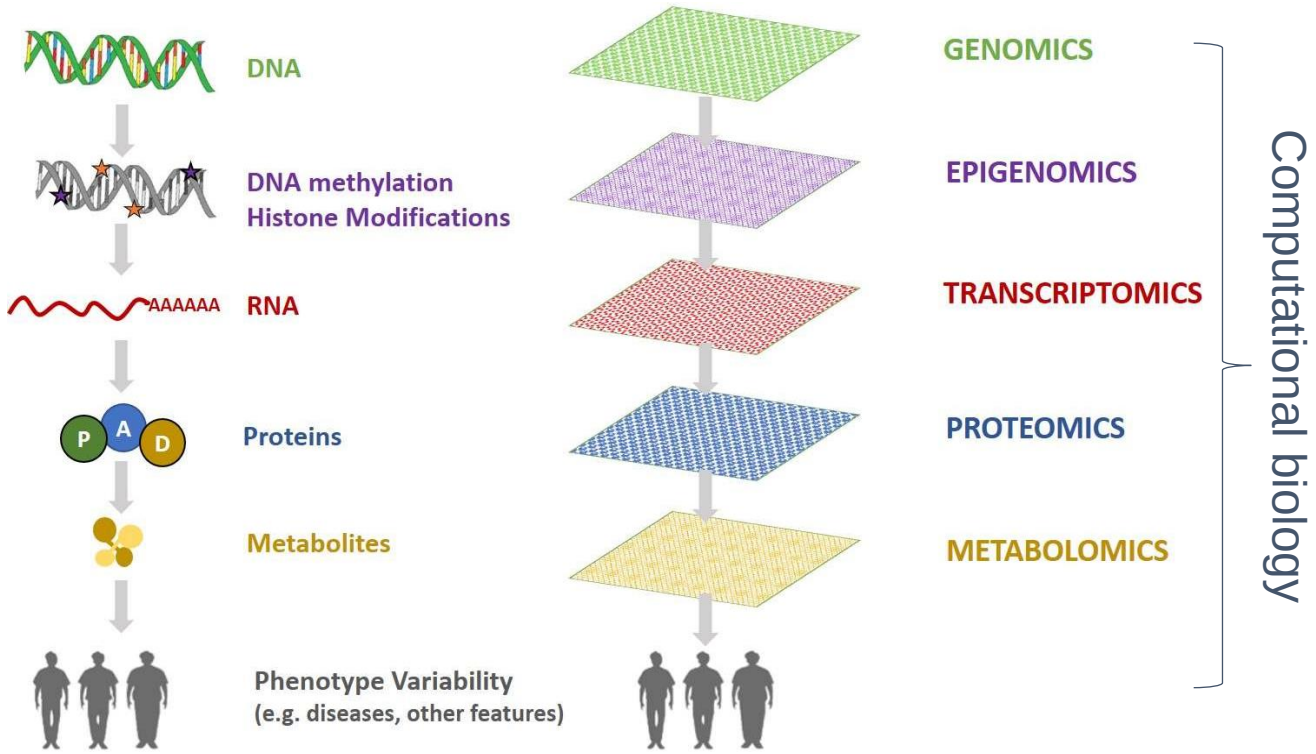
What skills are required?

- Programming/Data wrangling
  - Required for data processing, analysis and visualisation.
- High-performance computing
  - Some datasets may be too large or use too much resources for a normal laptop/desktop PC.
- Statistics
  - At least some understanding of applied statistics
- Domain knowledge
  - Understanding of the underlying biology



Omics data

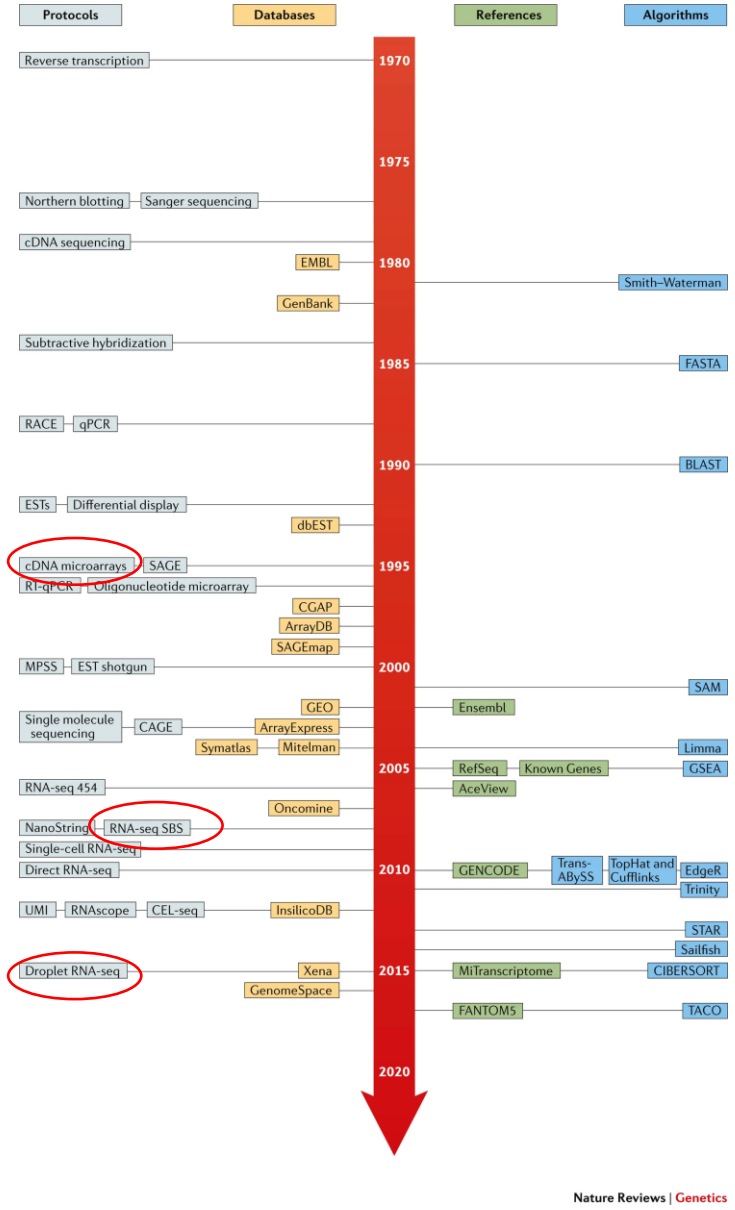
- Omics
  - High throughput
  - Measurement of all or as many as possible molecules of a given biomolecule (e.g., DNA, proteins, metabolites).
- Measurements performed using high-throughput instruments
  - E.g., Sequencers, Mass spec
  - Generally, yield large, multi-dimensional datasets.



Source: https://comics.dcv.fct.unl.pt/resources/

Transcriptomics

- Relative quantification of mRNA levels (reverse transcribed into cDNA) in biological samples.
- High-throughput sequencing (HTS), RNAseq currently main method but previously microarrays were used.
- Single-cell RNAseq also now common, enabling measurement of transcriptome at single-cell level



Cieślik and Chinnaiyan 2017, Nature Reviews Genetics

Lecture Outline

1. Introduction to Computational biology
2. Introduction to the R programming language – theory and concepts
3. Introduction to RNAseq
4. Introduction to 16S rRNA sequencing

The R programming language

- Free open-source language, first released in 1993.

- Desi
- IDE

Advantages	Disadvantages
Comprehensive ecosystem	Can be difficult to learn
Many, well-maintained and well documented libraries	Choosing between base R vs tidyverse can be difficult for beginners.
R is a high-level language, which can be run in real time and does not require compilation	Relatively slow in comparison to other languages

R for computational biology

- R packages for computational biology, generally installed from two main sources: CRAN or BioConductor.
  - CRAN is mostly statistical/general purpose packages
  - BioConductor comprises packages specifically designed for analysis of biological data.
- Enables access to methods/algorithms to facilitate analysis
  - Both rep… …entation and tutorials
- Code is v… (Rmarkd…
  - Writing c…

CRAN
Mirrors
What's new?
Search
CRAN Team

About R
R Homepage
The R Journal

Software
R Sources
R Binaries
Packages
Task Views
Other

Documentation
Manuals
FAQs
Contributed

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Download R for Linux (Debian, Fedora/Redhat, Ubuntu)
- Download R for macOS
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

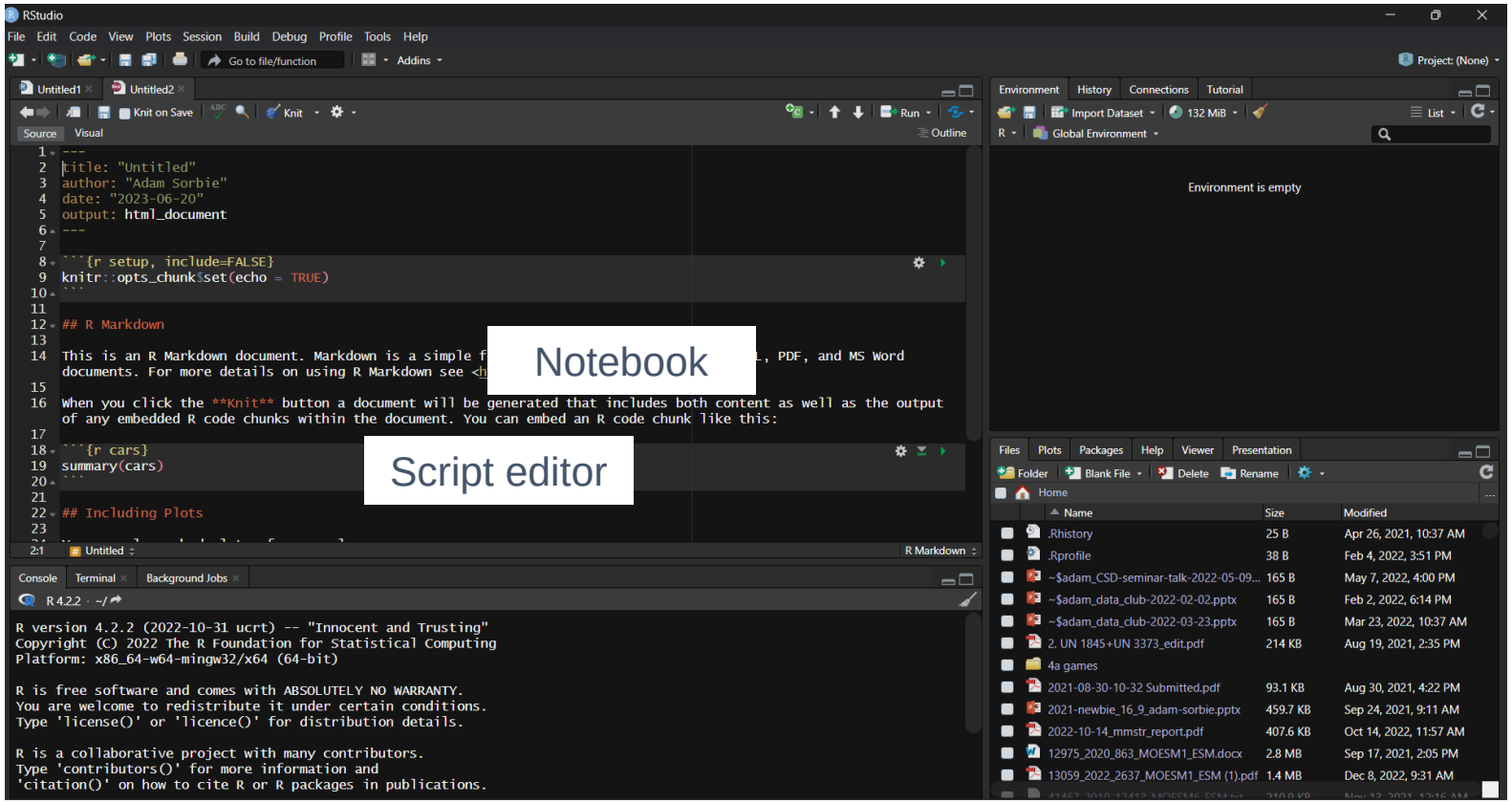
- The latest release (2023-06-16, Beagle Scouts) R-4.3.1.tar.gz, read what's new in the latest version.
- Sources of R alpha and beta releases (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are available here. Please read about new features and bug fixes before filing corresponding feature requests or bug reports.
- Source code of older versions of R is available here.
- Contributed extension packages

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

What are R and CRAN?

Working with R/Rstudio

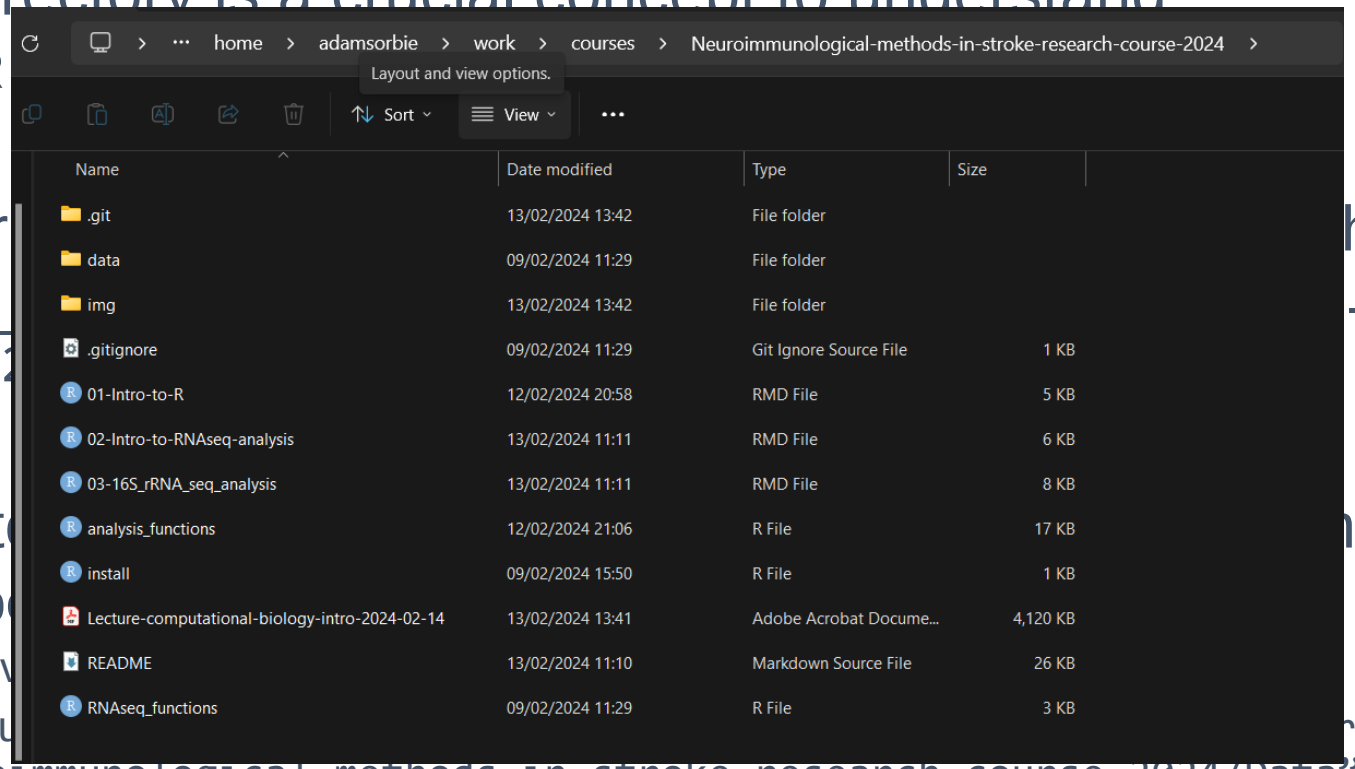


- Script editor – where you write scripts
- Console – run code interactively
- Environment – things you create/source stored here.

- In an Rmarkdown notebook the script editor is replaced with a notebook allowing you to intersperse text with code blocks.

Working with R/Rstudio – working directory

- Working directory is a crucial concept to understand
  - Where R
- Check curr h `set_wd()`
  - E.g. `set_` `-in-stroke-research-course-2`
- Important t nd absolute paths.
  - Using ab
    - Relati
    - Absolu `research-course-2024/Neuroimmunological-methods-in-stroke-research-course-2024/Data`”



Working with R/Rstudio – writing and running code

- You can write code in the console or script editor/notebook in case of Rmarkdown.
  - Always better to write a script/notebook as the code is recorded – reproducible.

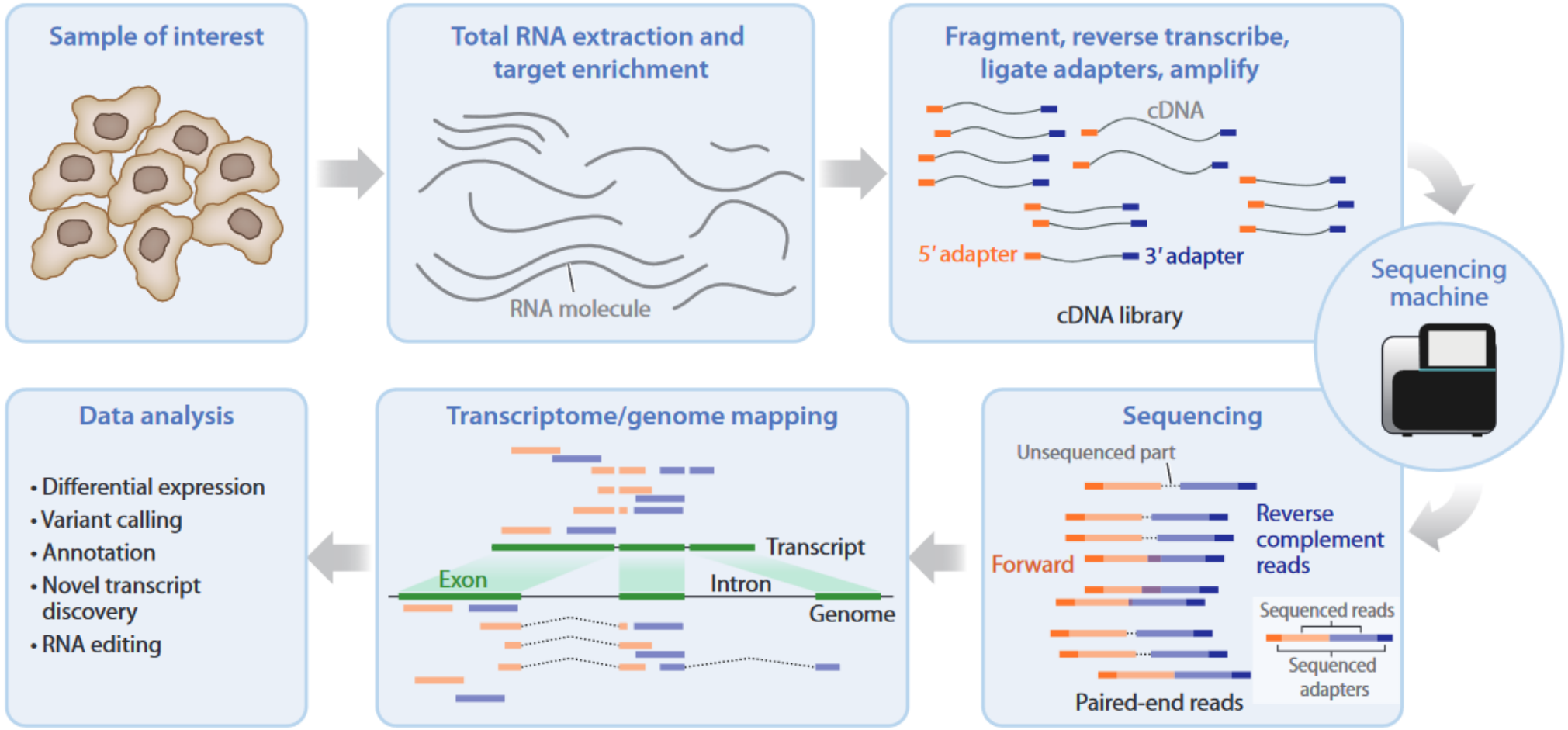
- To run a command press `ctrl + Enter` (`cmd + Return` on Macs)
  - In an Rmarkdown notebook, code can also be ran by pressing the green play button

- Rmarkdown
  - Typing outside of a code block is interpreted as markdown text
  - To insert a new code block press `ctrl + alt + I` (again replace `ctrl` with `cmd` on Macs)

Lecture Outline

1. Introduction to Computational biology
2. Introduction to the R programming language – theory and concepts
3. Introduction to RNAseq
4. Introduction to 16S rRNA sequencing

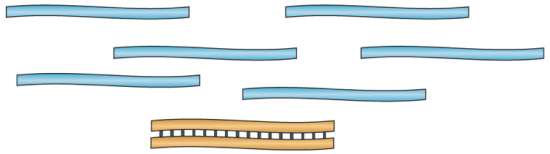
RNAseq - principle



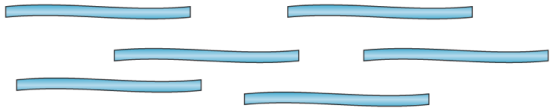
Van den Berge et al, 2019. Annual Review of Biomedical Data Science

Library preparation

① mRNA or total RNA

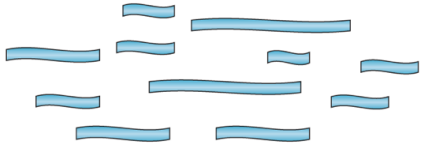


② Remove contaminant DNA



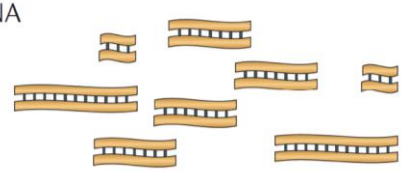
③ Fragment RNA

Remove rRNA

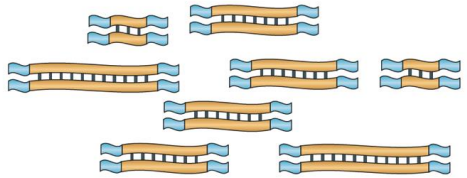


PCR amplification

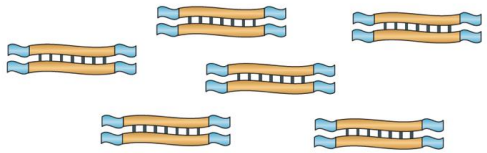
④ Reverse transcribe into cDNA



⑤ Ligate sequence adaptors



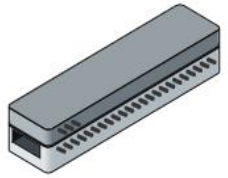
⑥ Select a range of sizes



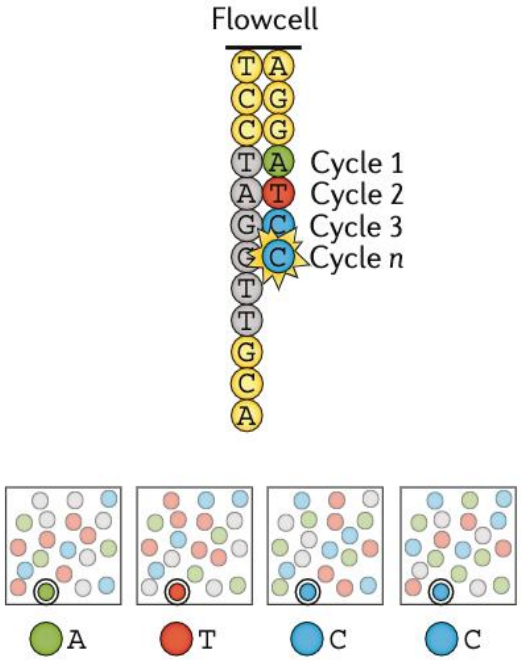
Sequencing platforms

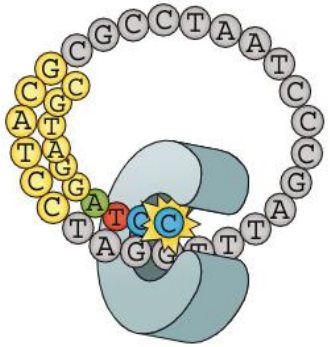
Illumina

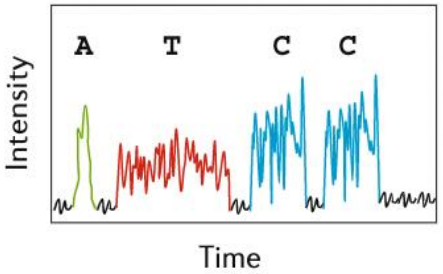
Pacific Biosciences

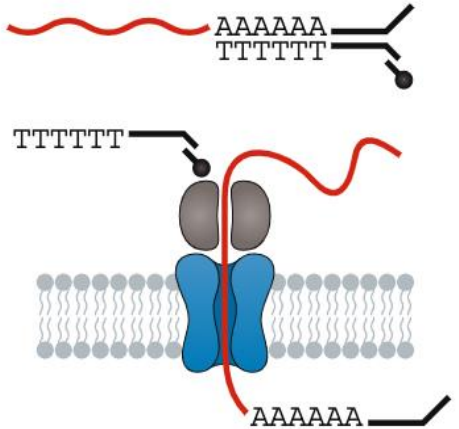


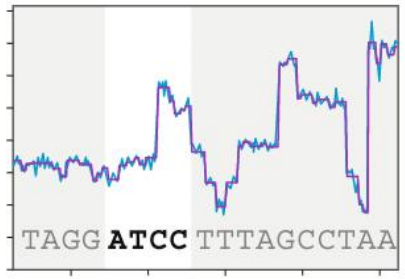
Oxford Nanopore







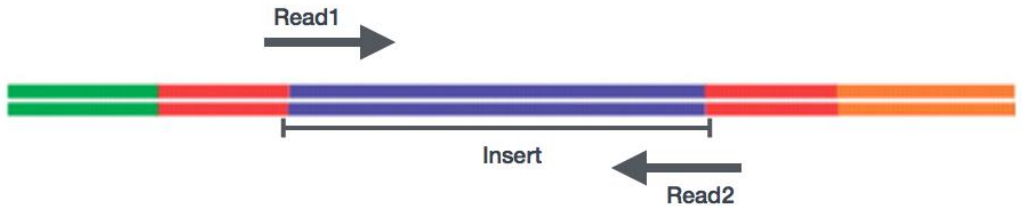




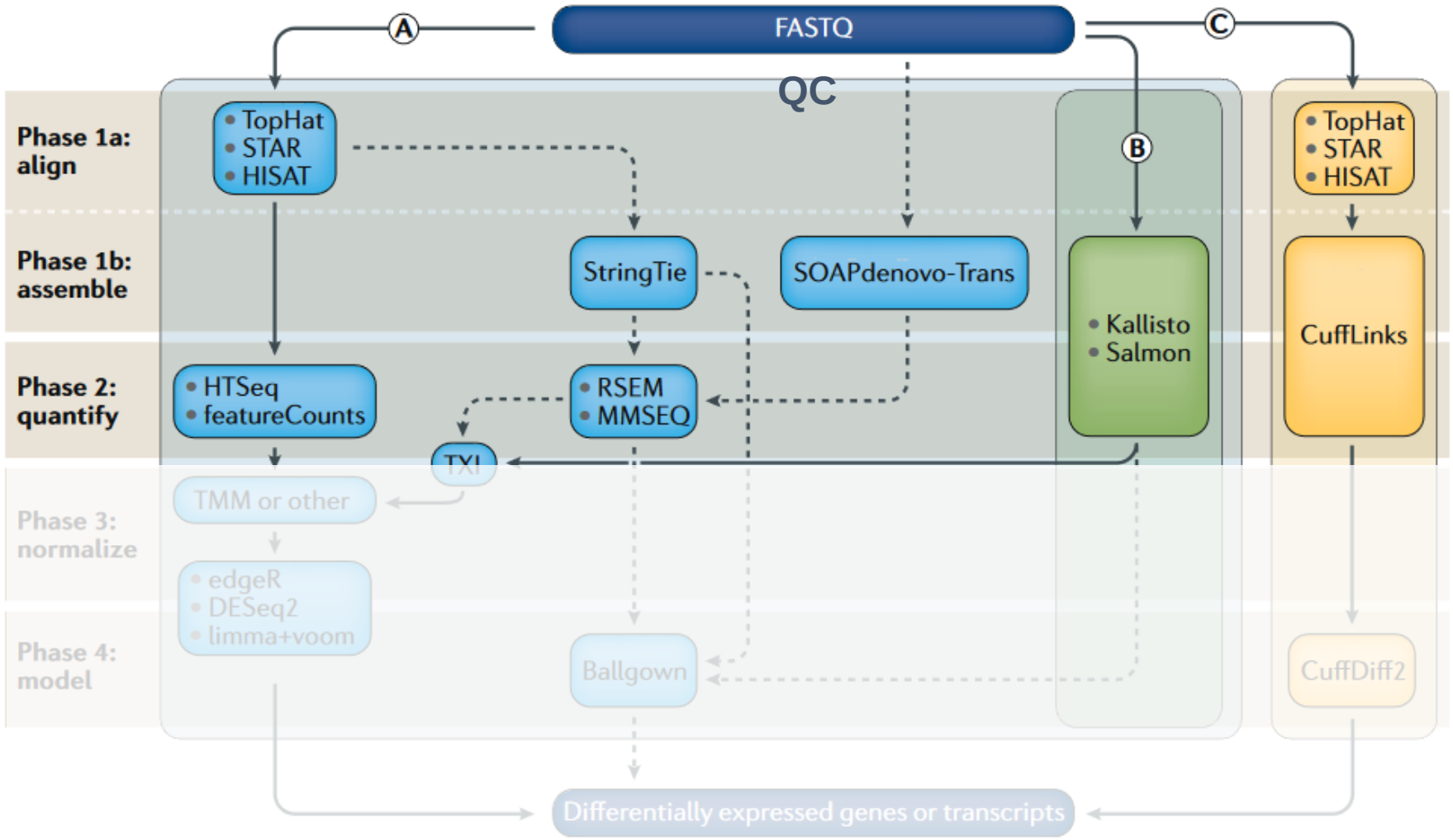
Stark, Grzelak and Hadfield, 2019 Nature Reviews Genetics

Sequencing – Key concepts

- Typical read length: 50-150bp.
- Single-end (SE) vs. paired-end (PE)
  - SE: each (amplified) fragment is only sequenced once, from one direction.
  - PE: each (amplified) fragment is sequenced from both directions:
- Sequencing depth
  - Total number of reads. Should be >20 million for standard gene-expression analysis without bells and whistles.
- Phred (Q) score: each base assigned quality score
  - Probability (P) of base call being wrong. Q = -10 log10 P. E.g., Q = 10, P = 0.1 (poor); Q = 40, P = 0.0001 (good)



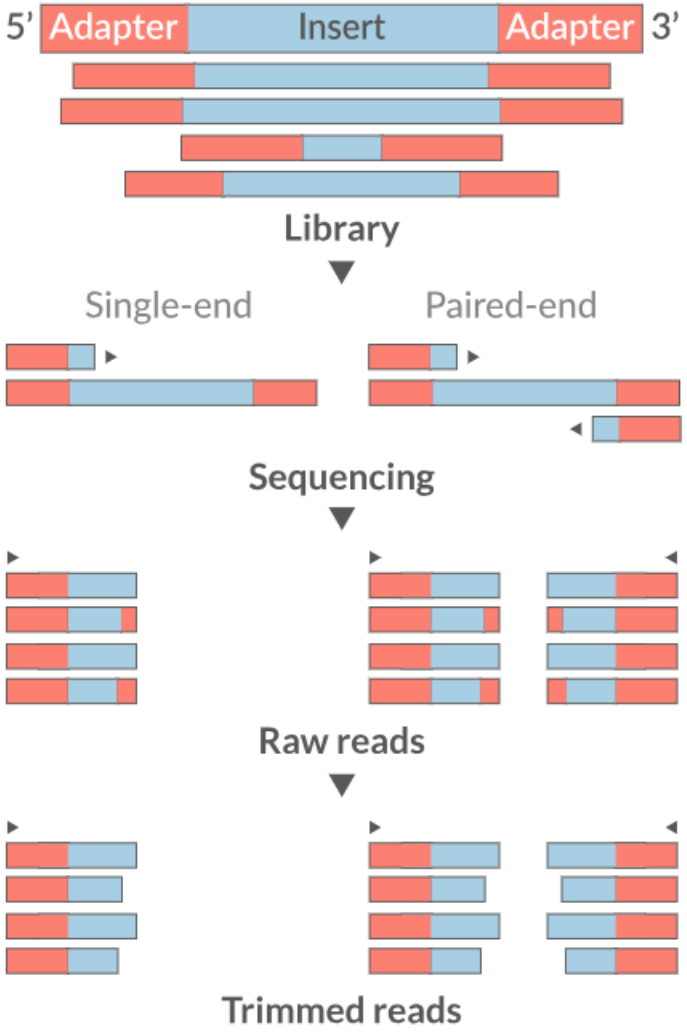
RNAseq data processing steps



Stark, Grzelak and Hadfield, 2019 Nature Reviews Genetics

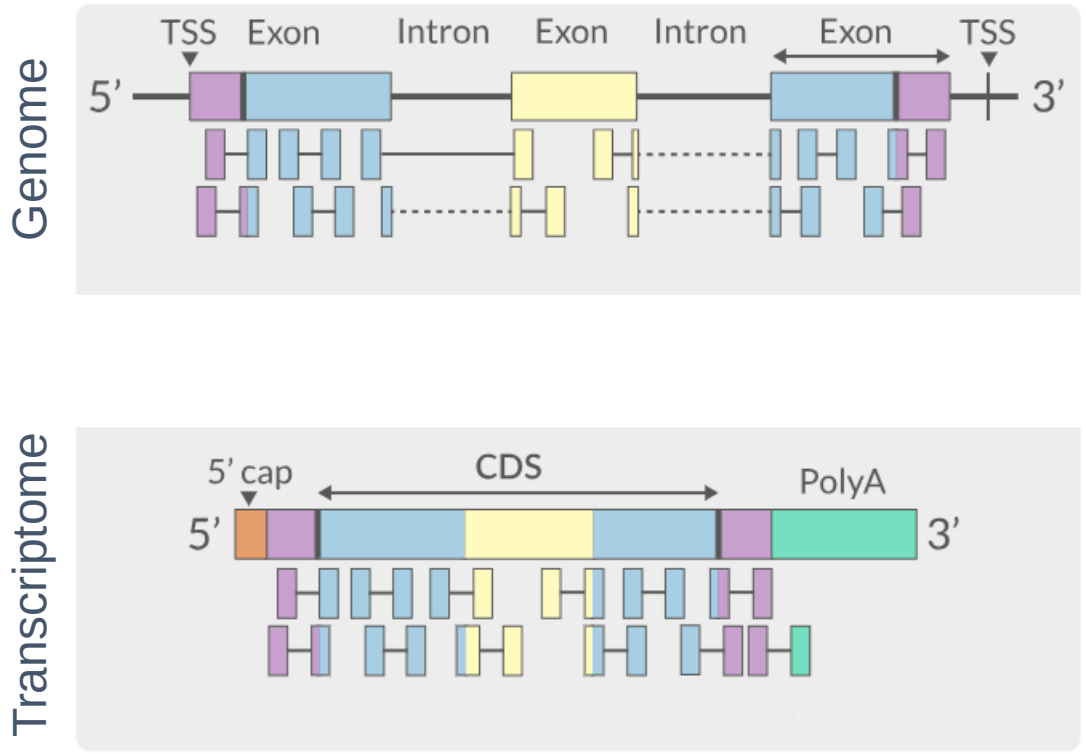
Quality control

- Examine read quality
  - FastQC, MultiQC
- Remove any adapter sequences, filter low quality reads
  - Trimmomatic, Cutadapt
- Trimming and filtering poor quality bases/reads improves mapping rate

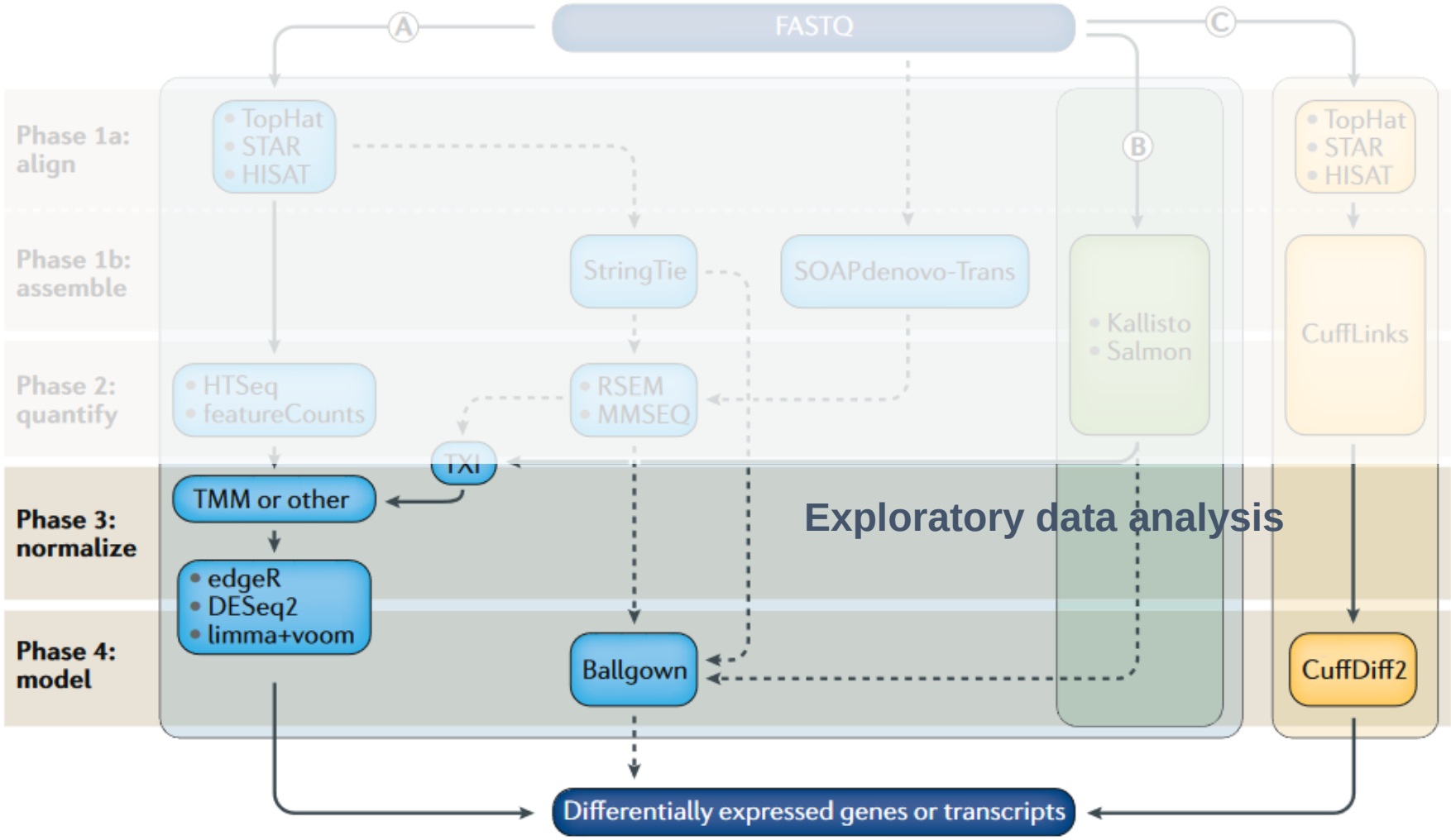


Mapping and quantification

- Aligning trimmed and filtered reads to reference sequence
  - Genome (splice-aware) or transcriptome.
- Quantifying number of hits to each gene/transcript



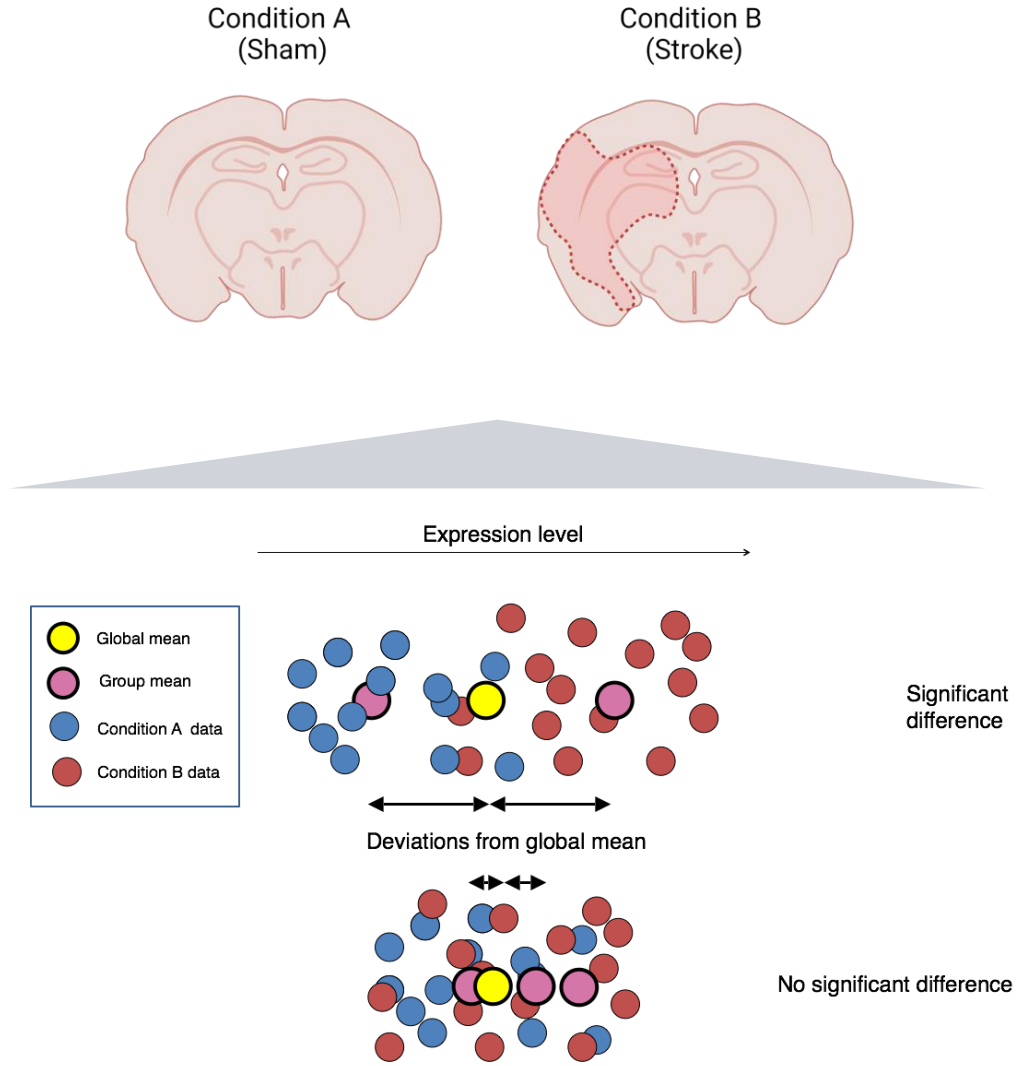
RNAseq data analysis steps



Stark, Grzelak and Hadfield, 2019 Nature Reviews Genetics

Differential expression

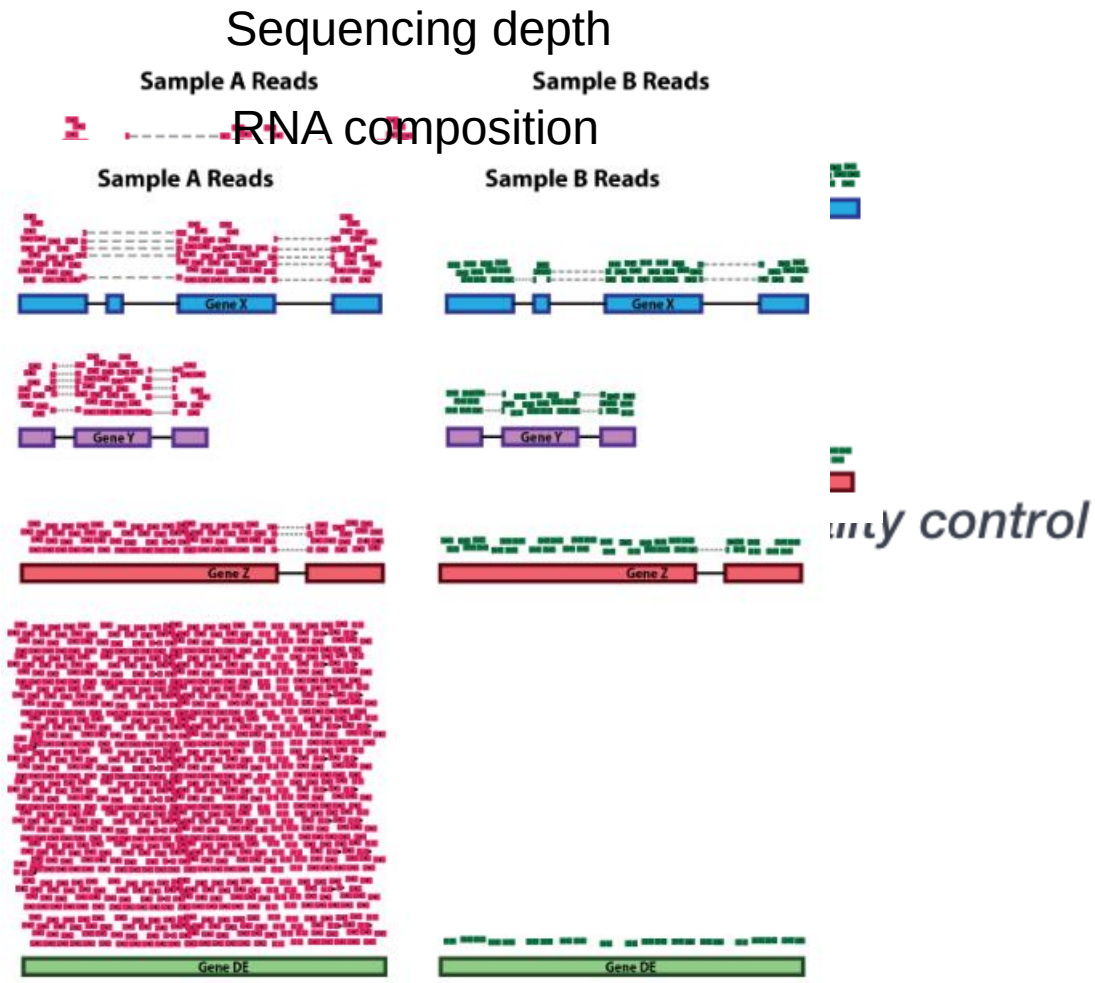
- Which genes/transcripts are different between conditions?
- Common tools include DESeq2, edgeR and limma/voom.



Differential expression - Normalization

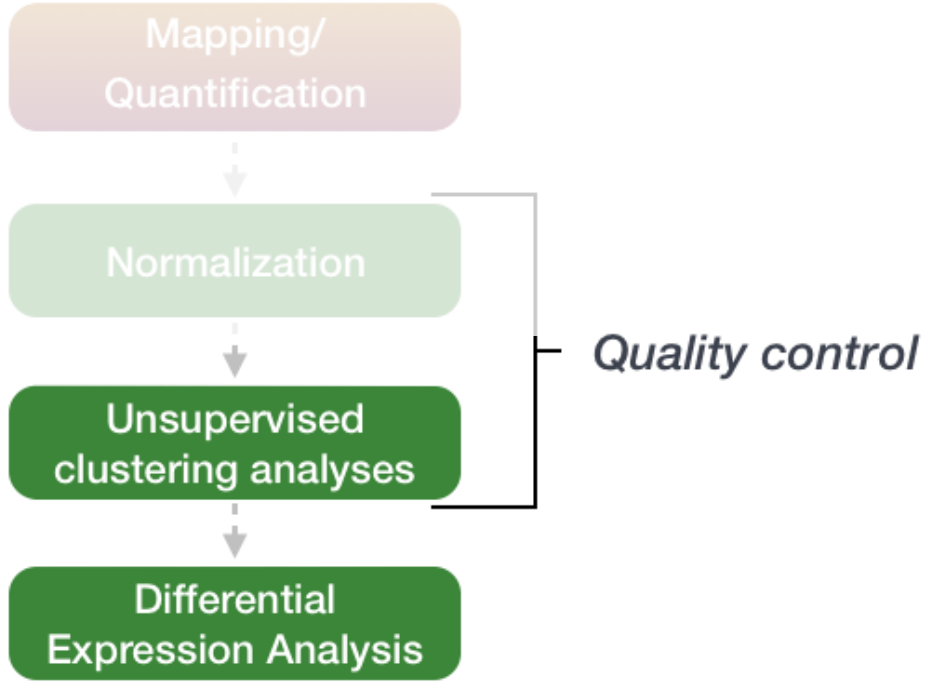
- Post-mapping –
  - Count matrix representing number of reads originating from each gene/transcript.

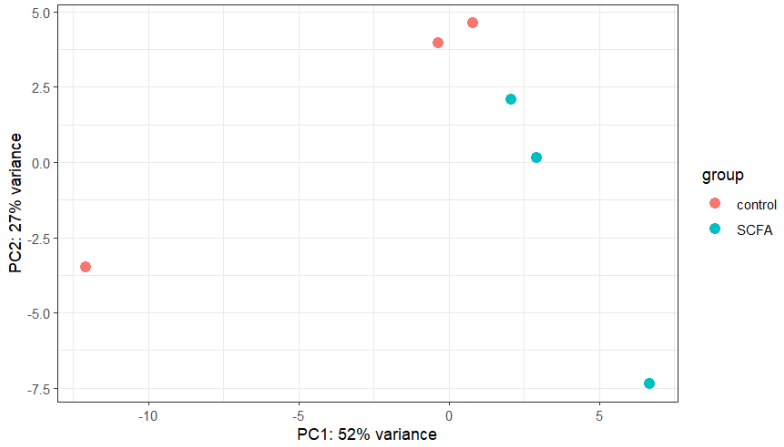
- Raw counts not comparable between samples
  - Sequencing depth and RNA composition differ.



Differential expression - Unsupervised clustering

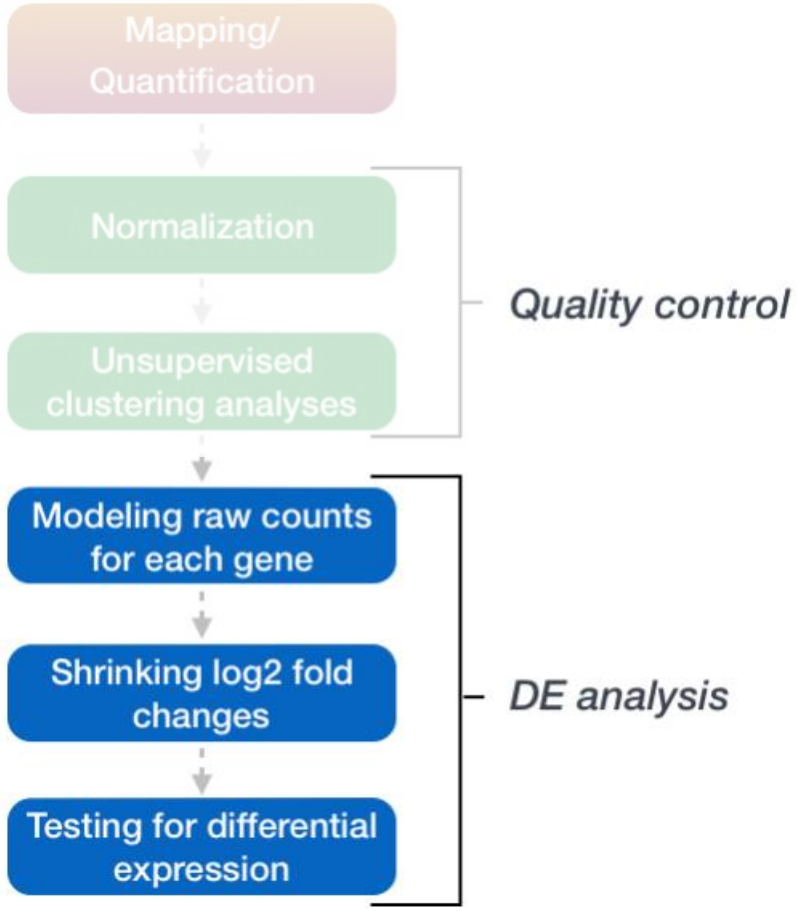
- Important to understand how similar/different samples are.
- Also, useful to examine data for outliers/confounding variables
- Principal component analysis (PCA) is a useful tool for this





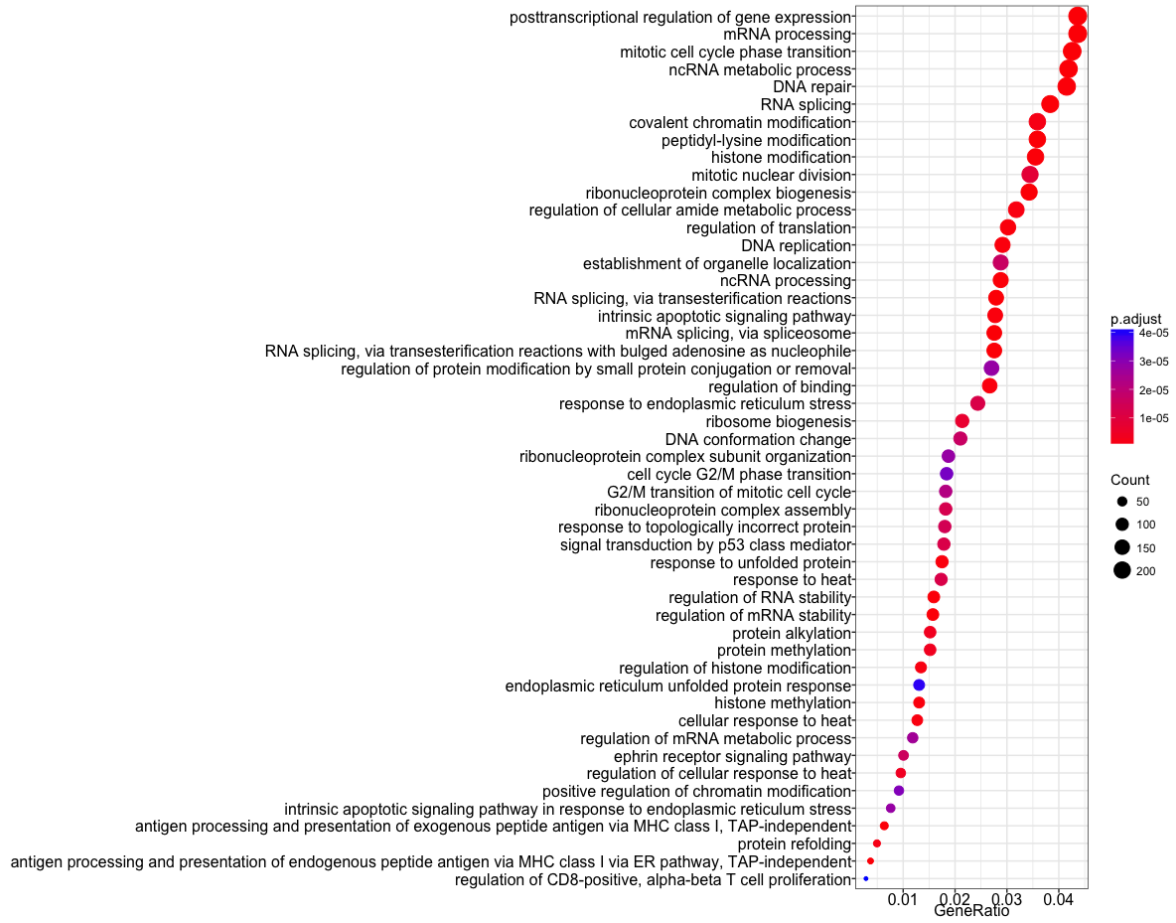
Differential expression – Identifying DEGs

- Using DESeq2 as an example, identification of DEGs can be split into 3 steps:
  - Apply statistical model (in this case a negative binomial model) to the raw counts for each gene.
  - Estimate Log2FC and shrink imprecise estimates
  - Identify differentially expressed genes using hypothesis testing (in this case a Wald test, with the null hypothesis that there is no difference in expression between groups.



Gene ontology/Enrichment analysis

- After identifying DEGs assigning pathways or functions to groups of genes can help make sense of results
- Three main types: Over-representation analysis, functional-class scoring and pathway topology.
- Common tools include R-based tools such as clusterProfiler and enrichR or online tools, like GSEA or DAVID.



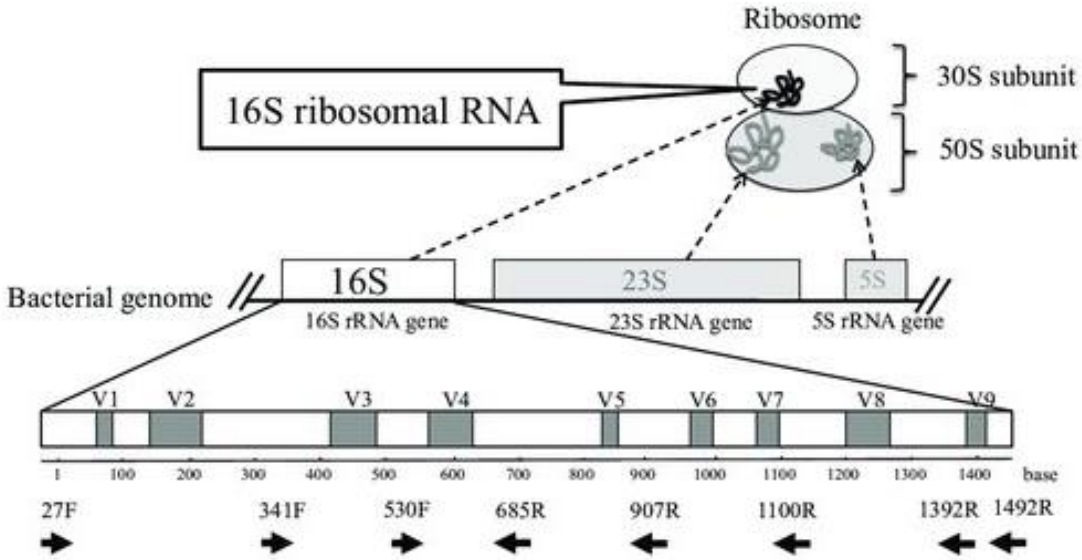
Source: https://hbctraining.github.io/Training-modules/DGE-functional-analysis/lessons/02_functional_analysis.html

Lecture Outline

1. Introduction to Computational biology
2. Introduction to the R programming language – theory and concepts
3. Introduction to RNAseq
4. Introduction to 16S rRNA sequencing

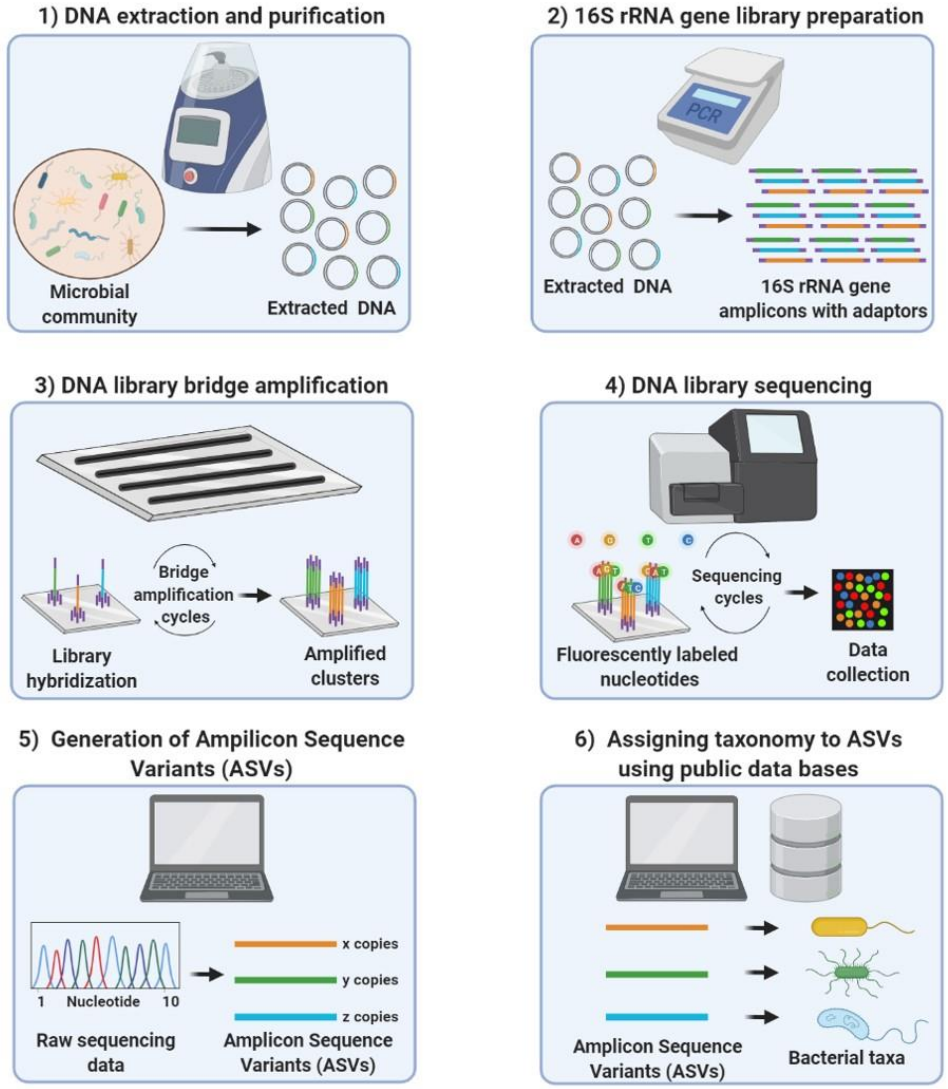
16S rRNA gene

- Small ribosomal subunit
- Composed of conserved and hypervariables (V) regions
  - Can be exploited be identify bacteria (and in some cases archaea)
- Relatively inexpensive
- Short amplicon sequences ~150-500bp matched against database of full-length sequences to identify bacteria
  - Usually limited to family/genus level classification



Fukuda *et al*., J UOEH, 2016

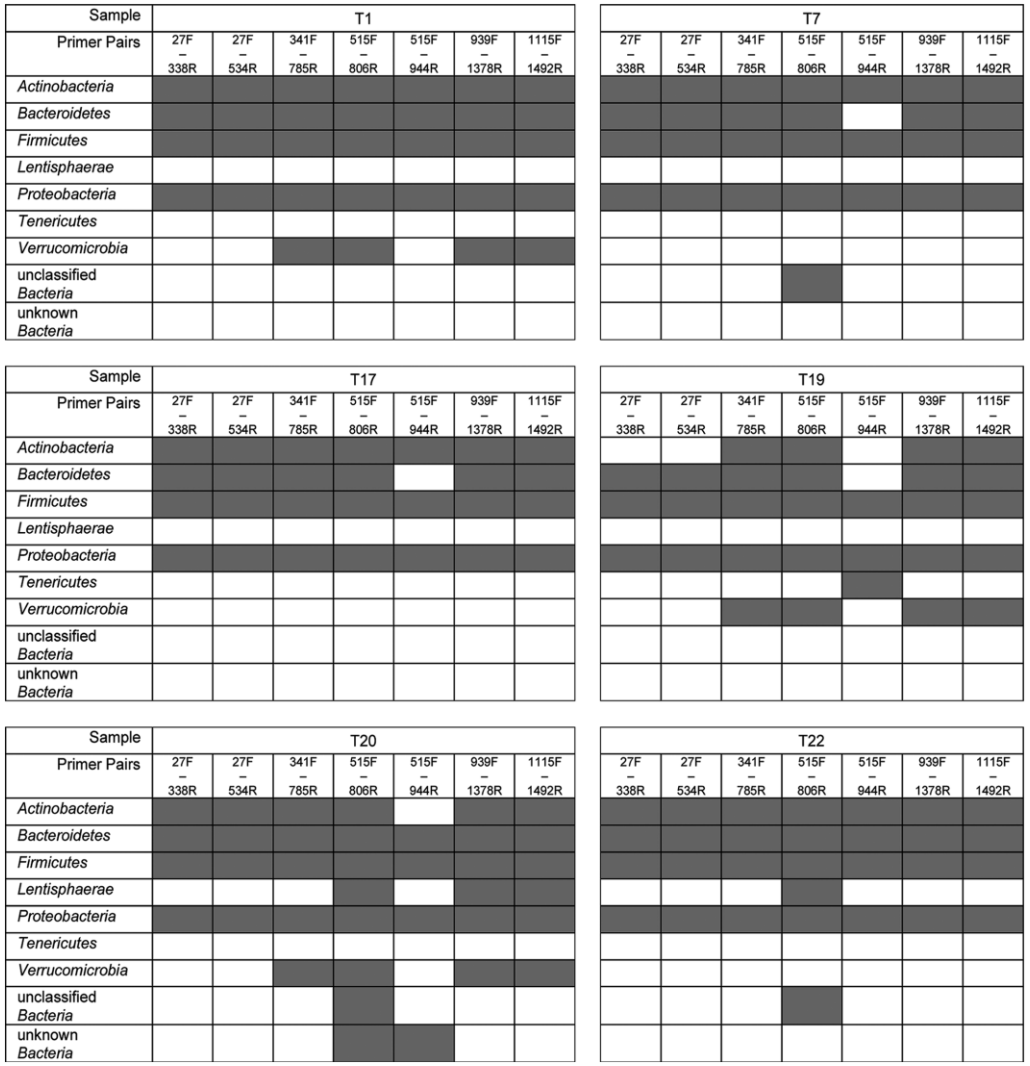
16S rRNA Sequencing – Principle



Source: Helmholtz UFZ - https://www.ufz.de/index.php?en=48523

Library preparation and sequencing

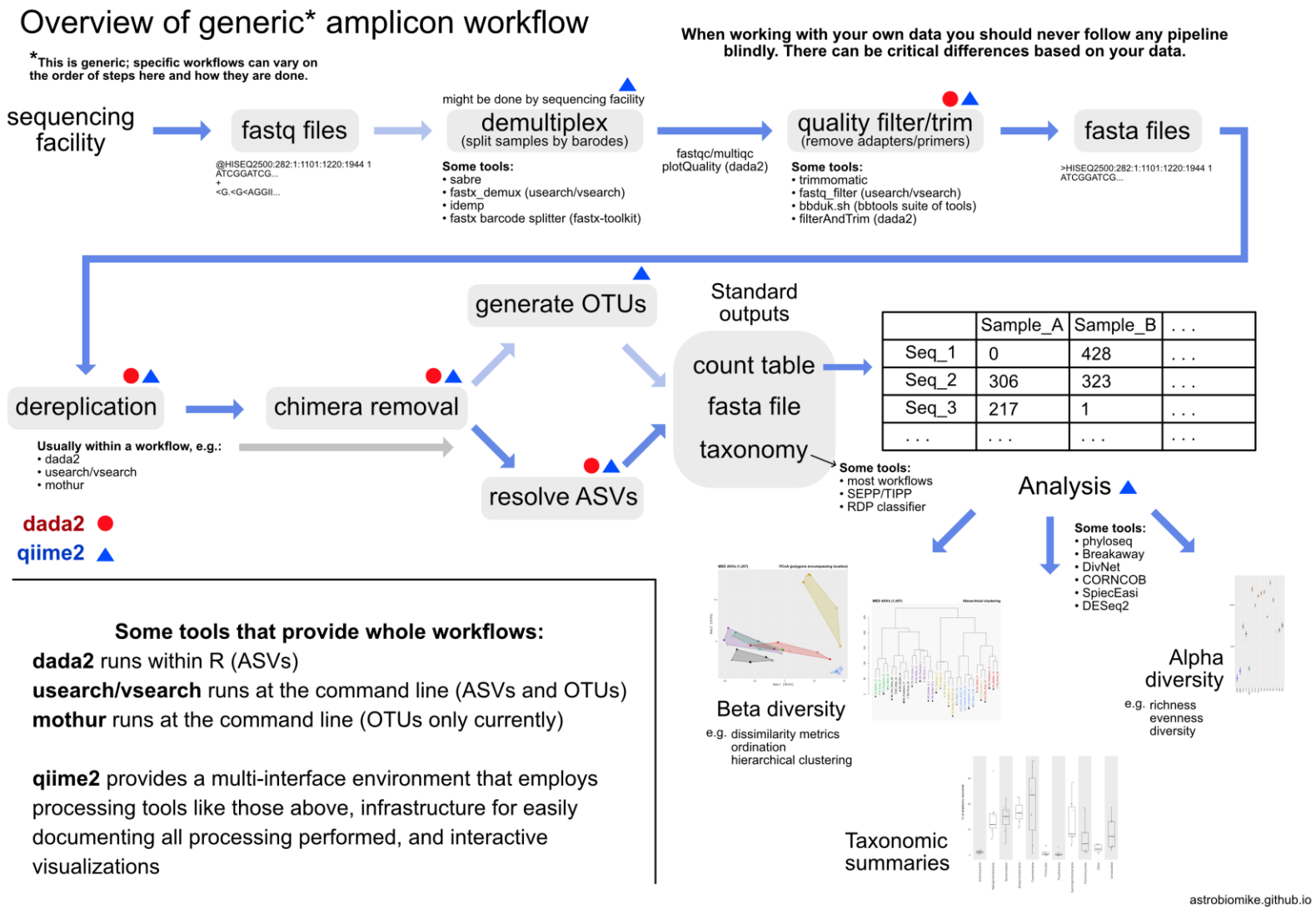
- Primer/V region choice has a massive impact on resulting microbial community
- Number of cycles used during PCR also impact composition/presence of artifacts
- Illumina HiSeq and MiSeq most commonly used for 16S studies



Abellan-Schneyder et al 2021, mSphere

Data processing steps

- Like RNAseq raw data is delivered as FASTQ files.
- QC
- Denoising
- Taxonomic assignment/Phylogenetic tree generation

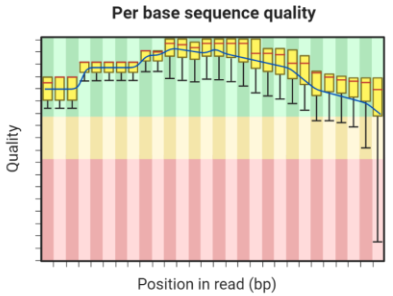


Source: https://astrobiomike.github.io/amplicon/

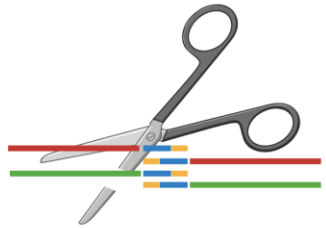
QC

- Adapter/primer trimming
- Truncate and filter low quality reads
- Non-biological sequences/low quality bases interfere with taxonomic assignment.

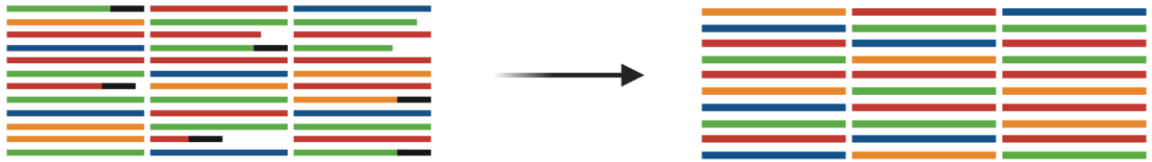
Quality control



Adapter trimming and filtering

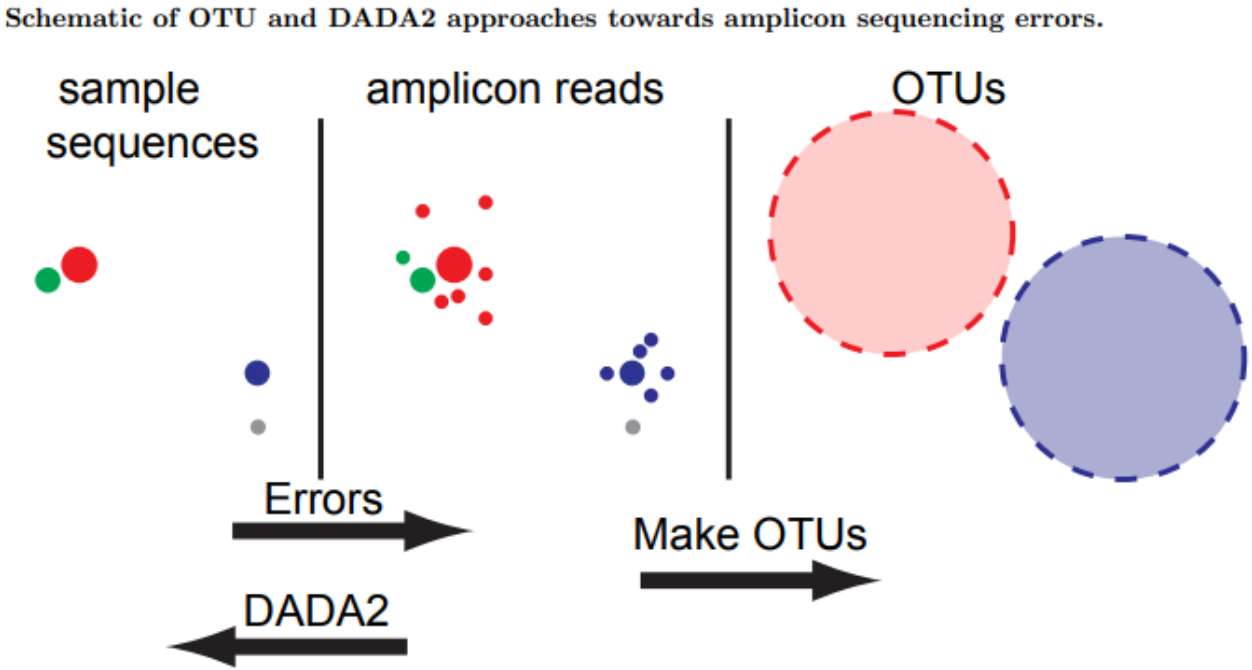


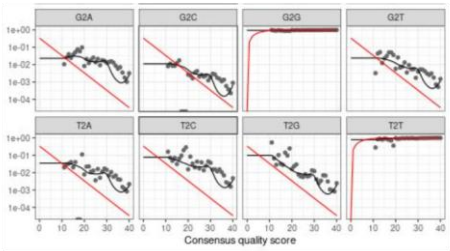
Truncate and filter poor quality reads



Denoising reads

- Improves upon previous method of OTU clustering.
  - Sequences binned at 97% sequence identity.
- Broadly speaking, most methods generate statistical models of error profiles, to identify and remove errors.
- Generates real biological sequences, at single nucleotide resolution.



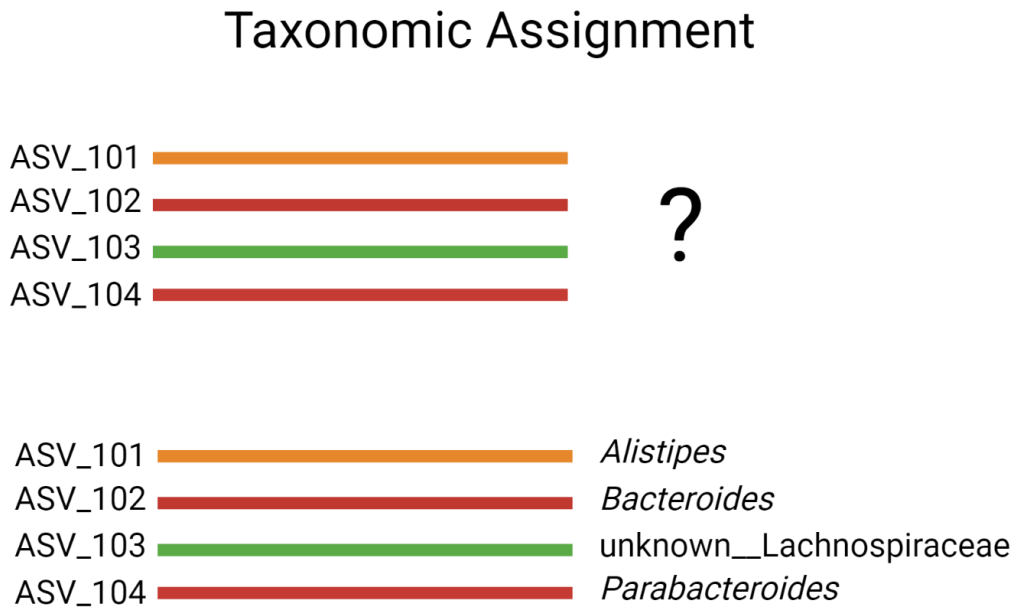


TTAACTGACGCTGAGGC...

TAAACTAACGCCAAGGC...

Taxonomic classification

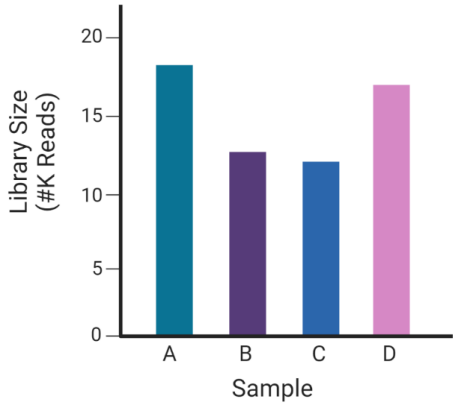
- Post-denoising, ASVs assigned taxonomic classification to identify source of sequence
- Widely used methods use ML models trained on curated full-length sequence database to classify unknowns.
  - Databases – SILVA, RDP, Greengenes2
  - Methods – DECIPHER, RDP-classifier, qiime2 feature-classifier
- Resolution depends on length and database used
  - Assignments differ between databases
  - Some environments better represented than others



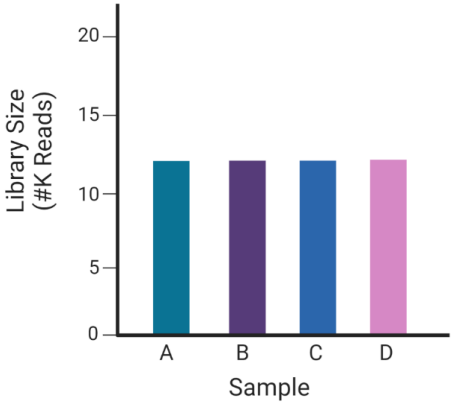
Normalisation – data transformations

- Biological and technical variation lead to different library sizes between samples.
  - Must be controlled for to limit erroneous conclusions
- Most commonly used methods are rarefaction or relative abundance.
  - Rarefaction discards data and not always recommended.
- Other methods include minimum sum scaling (MSS).
  - Scaling of counts to a common factor e.g. lowest sum
- Many publications (though not all) now maintain that microbiome data is compositional
  - Relative information, summing to a constant e.g., 1 or 100
  - This has implications for how we normalise and analyse our data.
  - Log-ratio transforms (commonly CLR) used for normalisation

Before normalisation

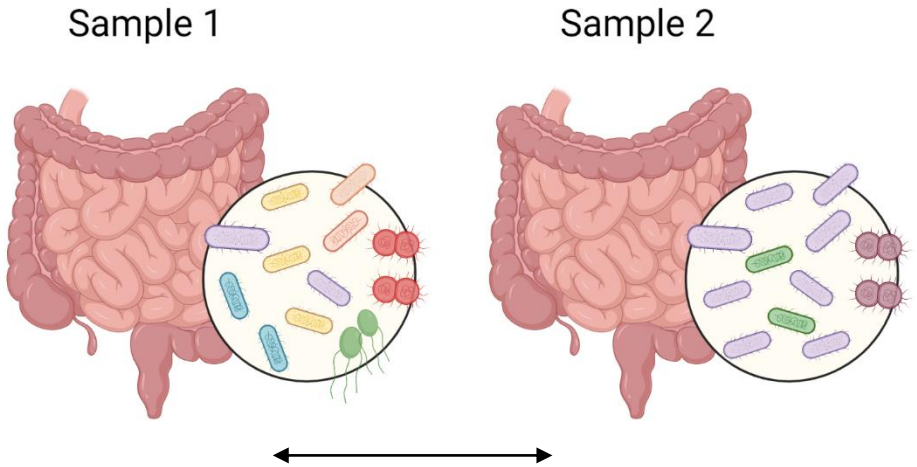


After normalisation



Diversity metrics

- Community level
- Alpha diversity – within sample, how many different species are present?
- Beta-diversity – between samples, how does the composition of species differ among samples?



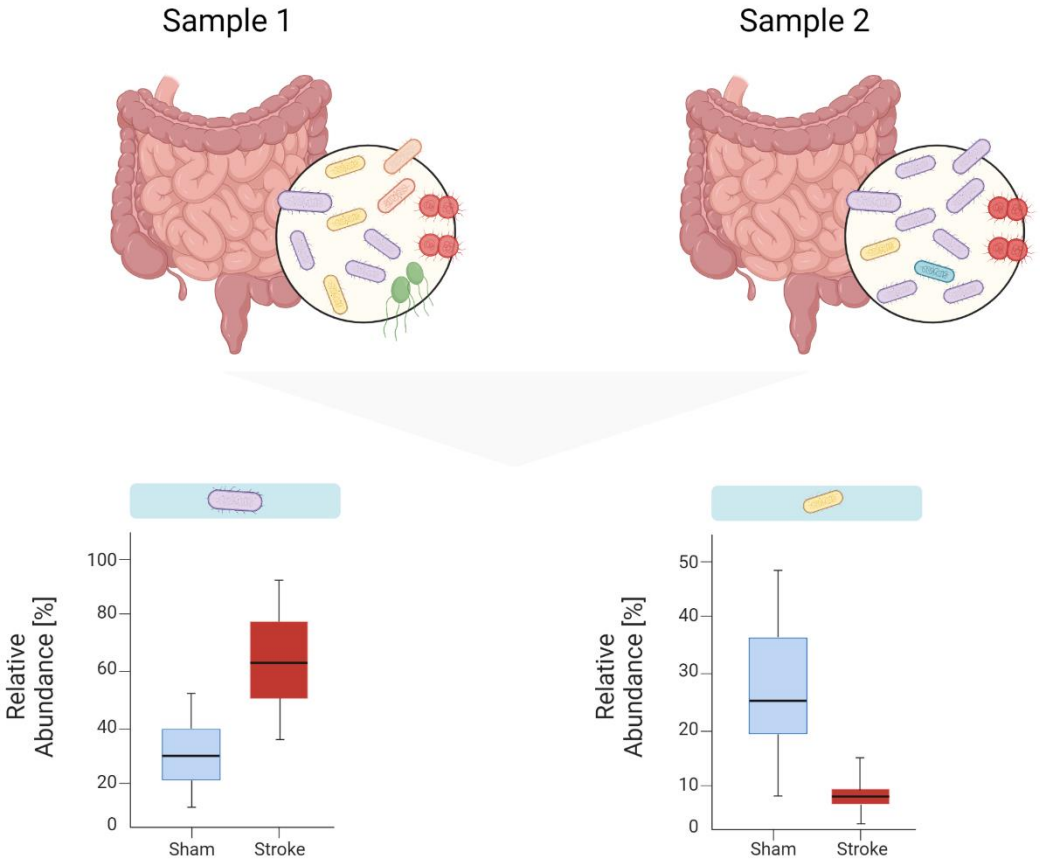
Alpha Diversity 6 3

Beta Diversity 7

Differential Abundance

- Which individual species differ between samples?

- In this example, the purple species is enriched in stroke and yellow species is depleted.



Any questions?

Institute for Stroke and
Dementia Research (ISD)