Introduction to computational biology and R

Dr Adam Sorbie

Institute for Stroke and Dementia Research
Munich, Germany

Neuroimmunological methods in stroke 25.07.24

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

LMU KLINIKUM

Institute for Stroke and Dementia Research (ISD)

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 (virtual)
  - Shotgun metagenomic analysis
  - RNAseq analysis

Lecture Outline

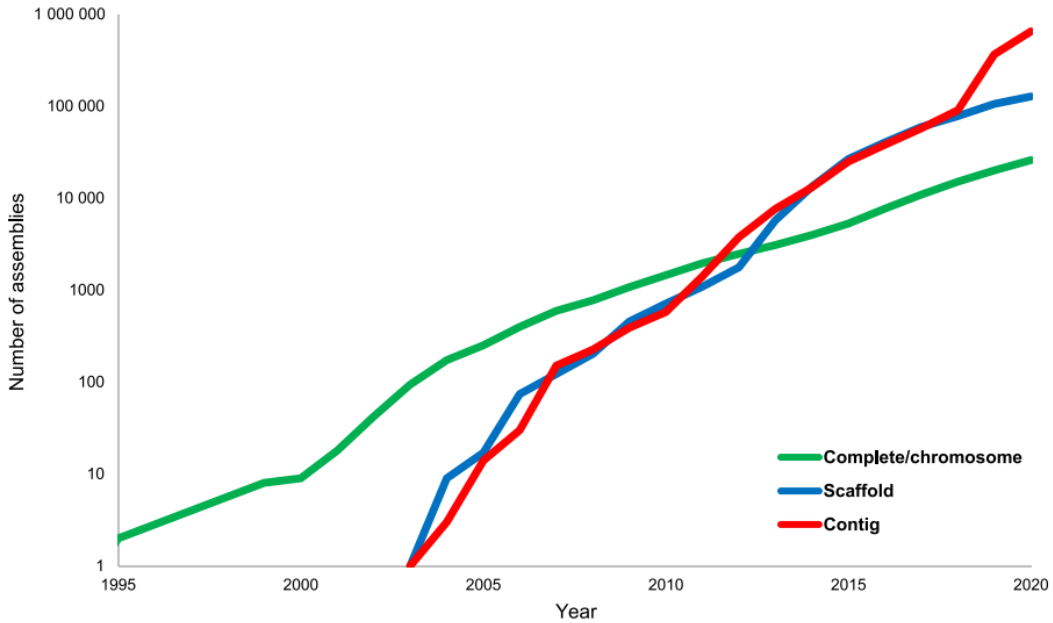
1. Introduction to Computational biology
2. Analysis of microbiome data
3. Introduction to RNAseq
4. Introduction to the R programming language – theory and concepts

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.

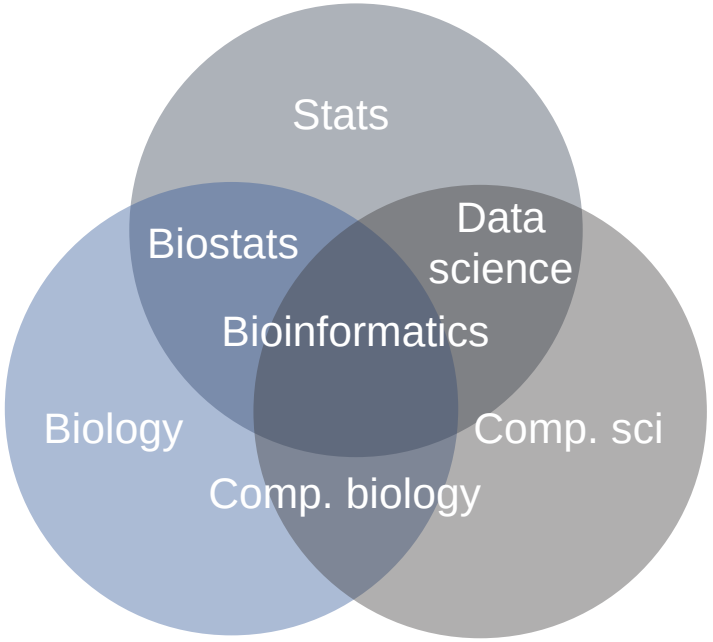


Source: https://www.genome.gov/about-genomics/fact-sheets/Sequencing-Human-Genome-cost

Koonin, Makarova and Wolf, Trends in Microbiology, 2021

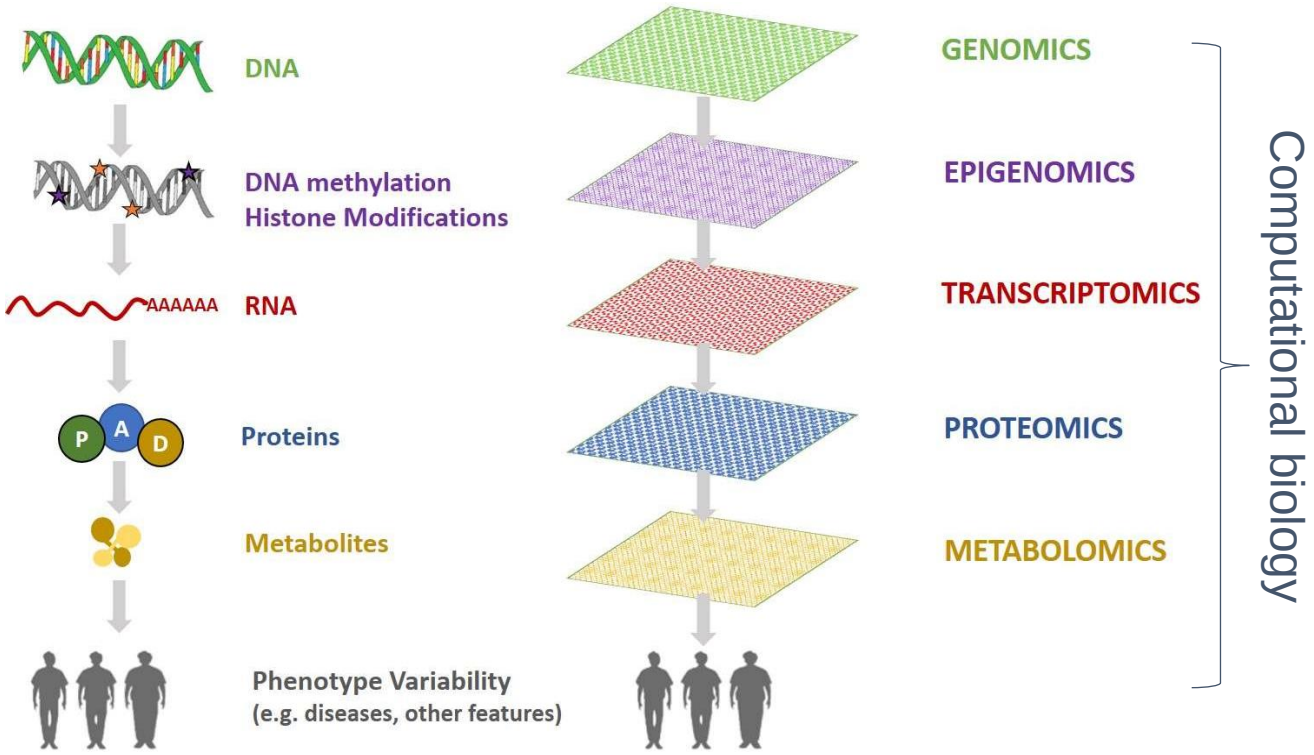
What skills are required?

- 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.

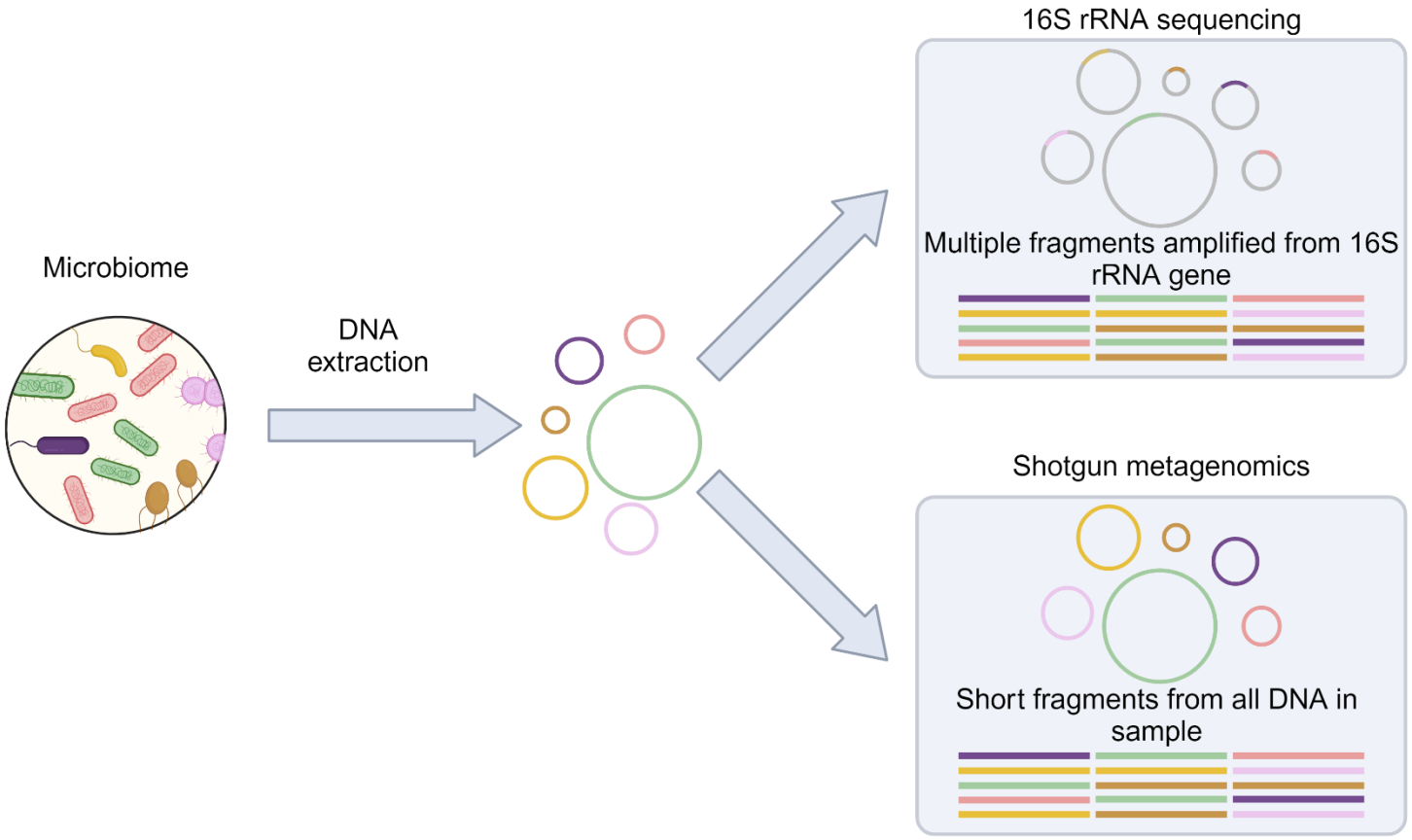


Source: https://comics.dcv.fct.unl.pt/resources/

Lecture Outline

1. Introduction to Computational biology
2. Analysis of microbiome data
3. Introduction to RNAseq
4. Introduction to the R programming language – theory and concepts

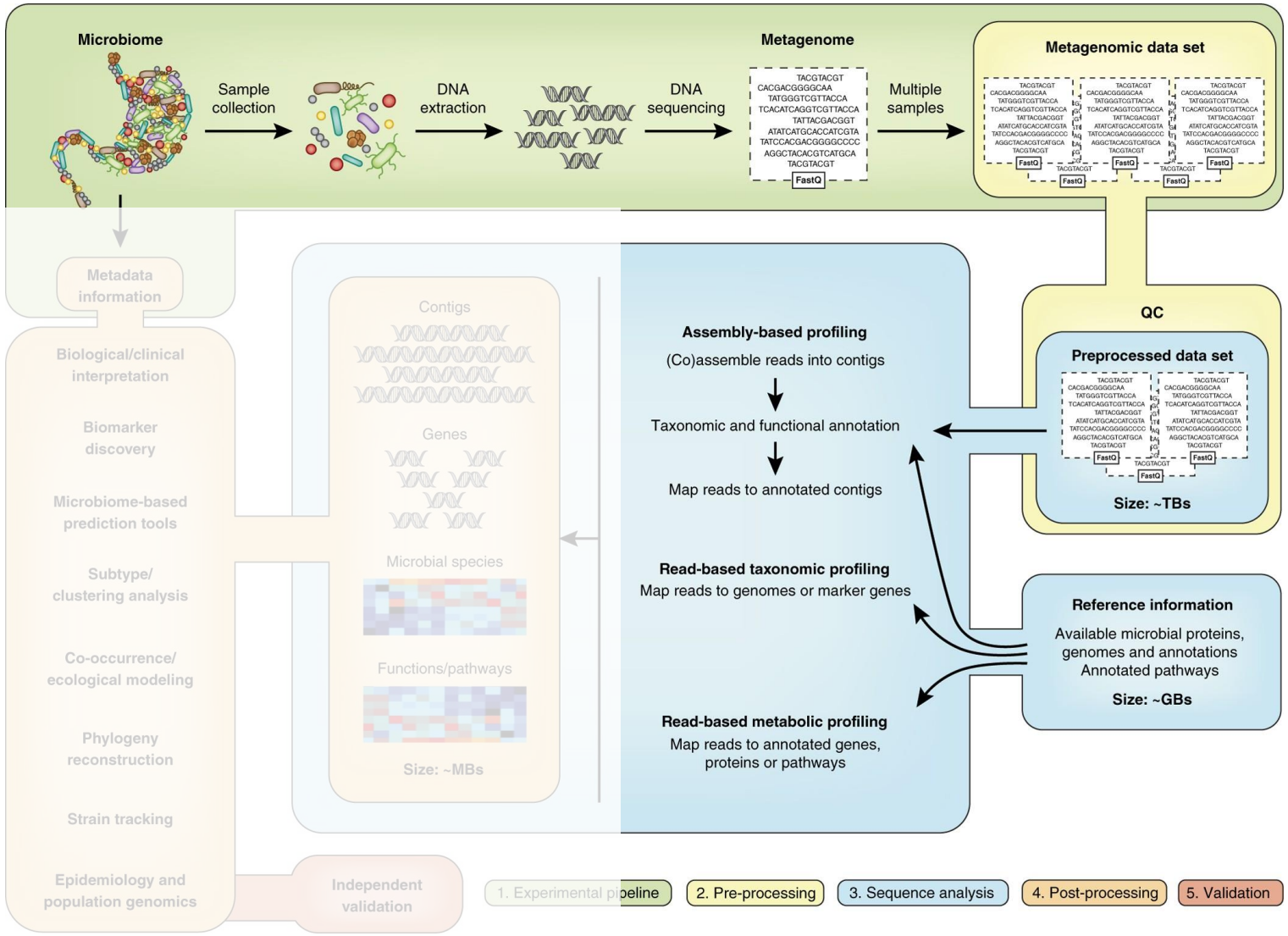
Methods of profiling microbiomes



- Targeted sequencing (16S rRNA)
- Maximum genus-level

- Sequencing all DNA
- Species level and gene content

Metagenomic data processing



Quince *et al*., 2017. Nature Biotechnology

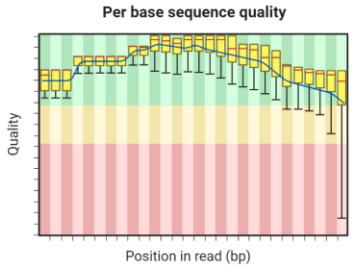
Institute for Stroke and Dementia Research (ISD)

Quality control

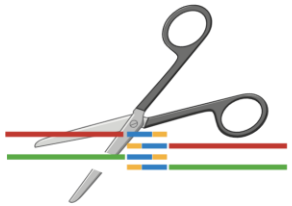
- Read trimming
  - Removal of primers/Adapters
  - Truncate and filter low quality reads

- Host removal
  - Removal of host (e.g. human or mouse reads)

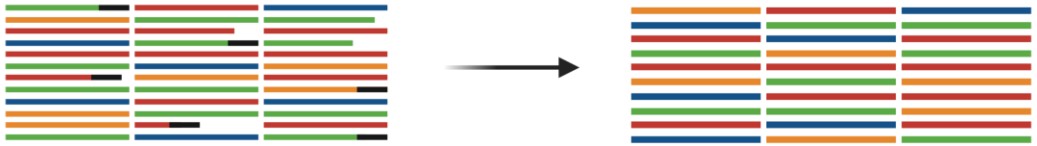
Quality control



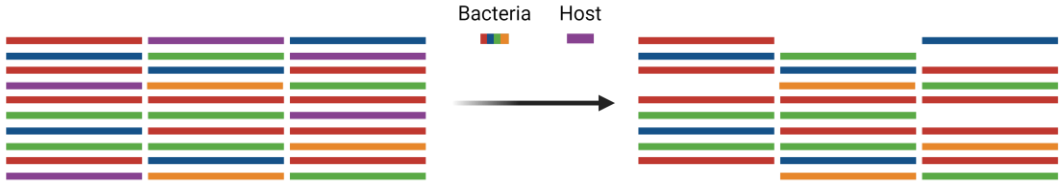
Adapter trimming and filtering



Truncate and filter poor quality reads



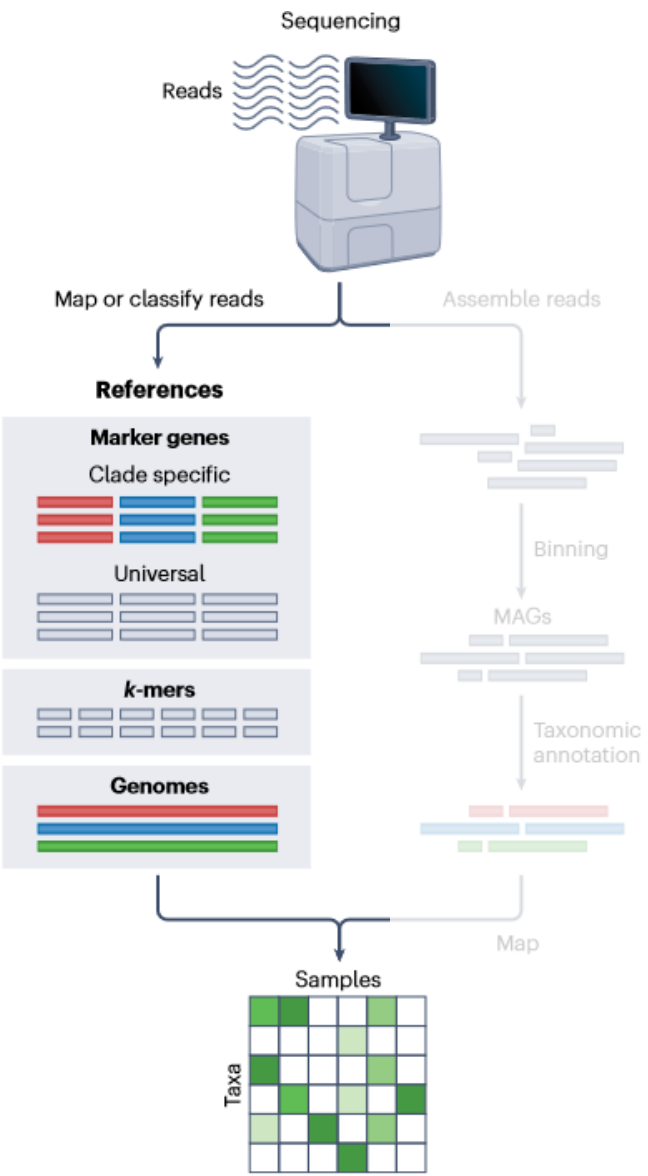
Filter reads aligning to host genome



Taxonomic and Functional profiling

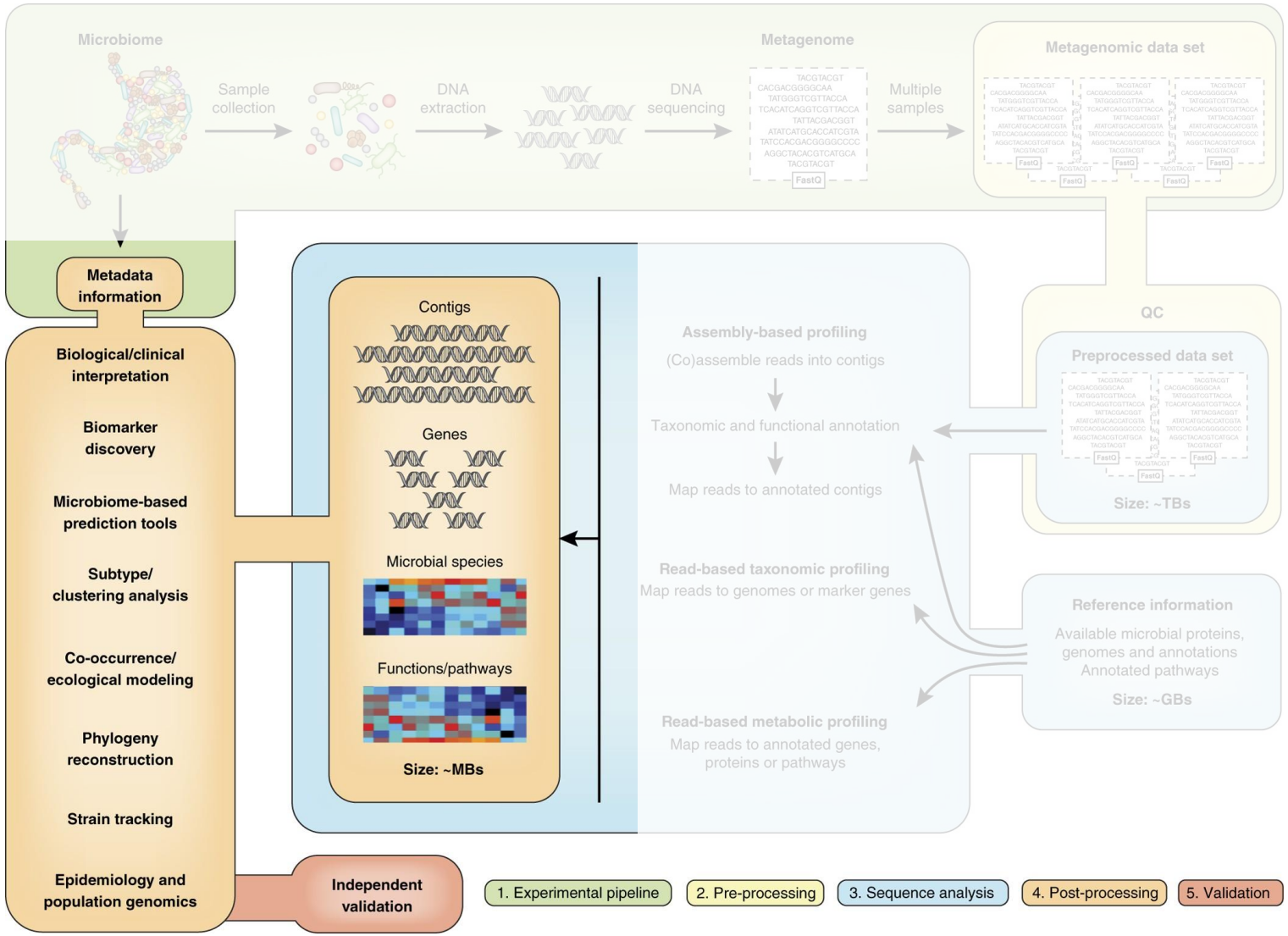
- Assembly
  - Assembly – stitching reads together to create contiguous sequence and binning into genomes.
  - Database-free
  - Computationally intensive

- Reference-based (read-mapping)
  - Mapping reads against database
  - Less computationally intensive
  - Database-dependent



Pinto and Bhatt 2024, Nature Reviews Genetics

Metagenomic data analysis



Quince *et al*., 2017. Nature Biotechnology

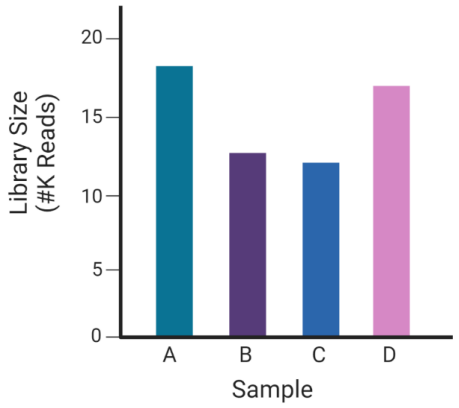
Institute for Stroke and Dementia Research (ISD)

Normalisation – data transformations

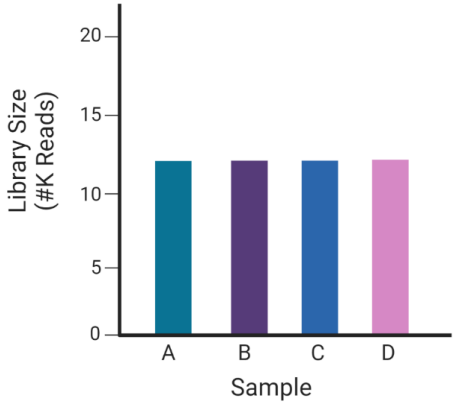
- Biological and technical variation lead to different library sizes between samples.
  - Must be controlled for to limit erroneous conclusions

- Most commonly used methods are rarefaction or relative abundance.

Before normalisation



After normalisation

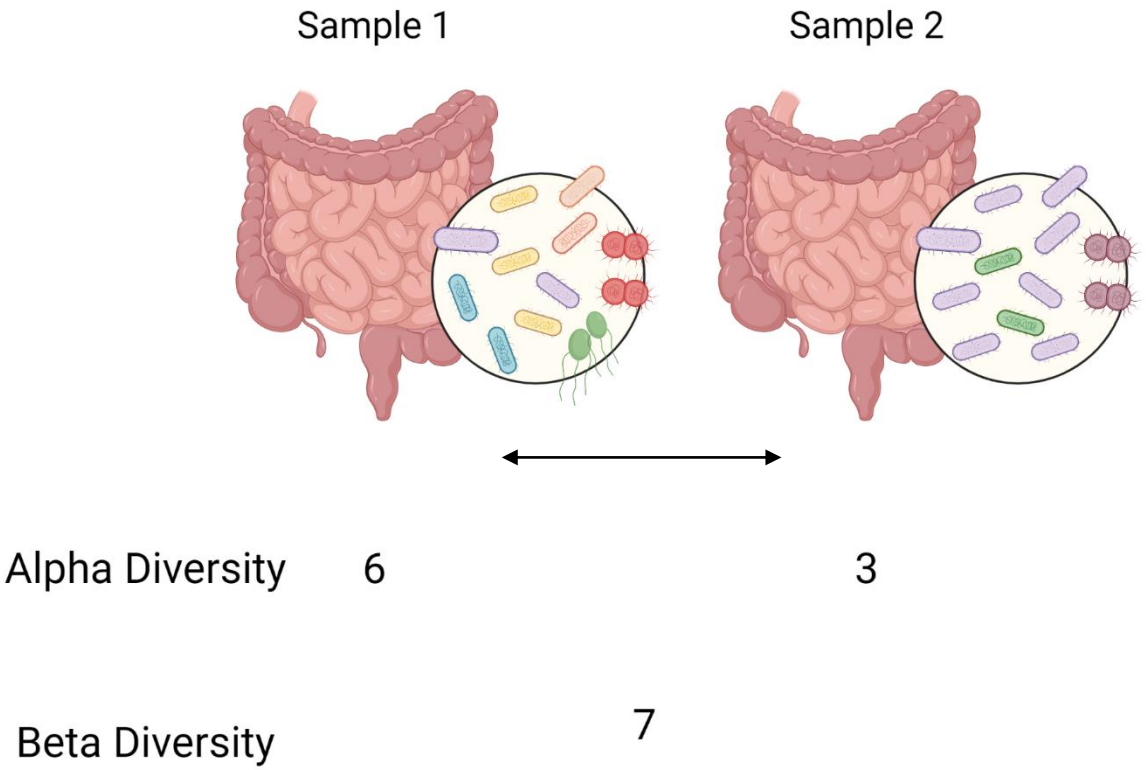


Diversity metrics

- Community level

- Alpha diversity – within sample, how many different species are present?

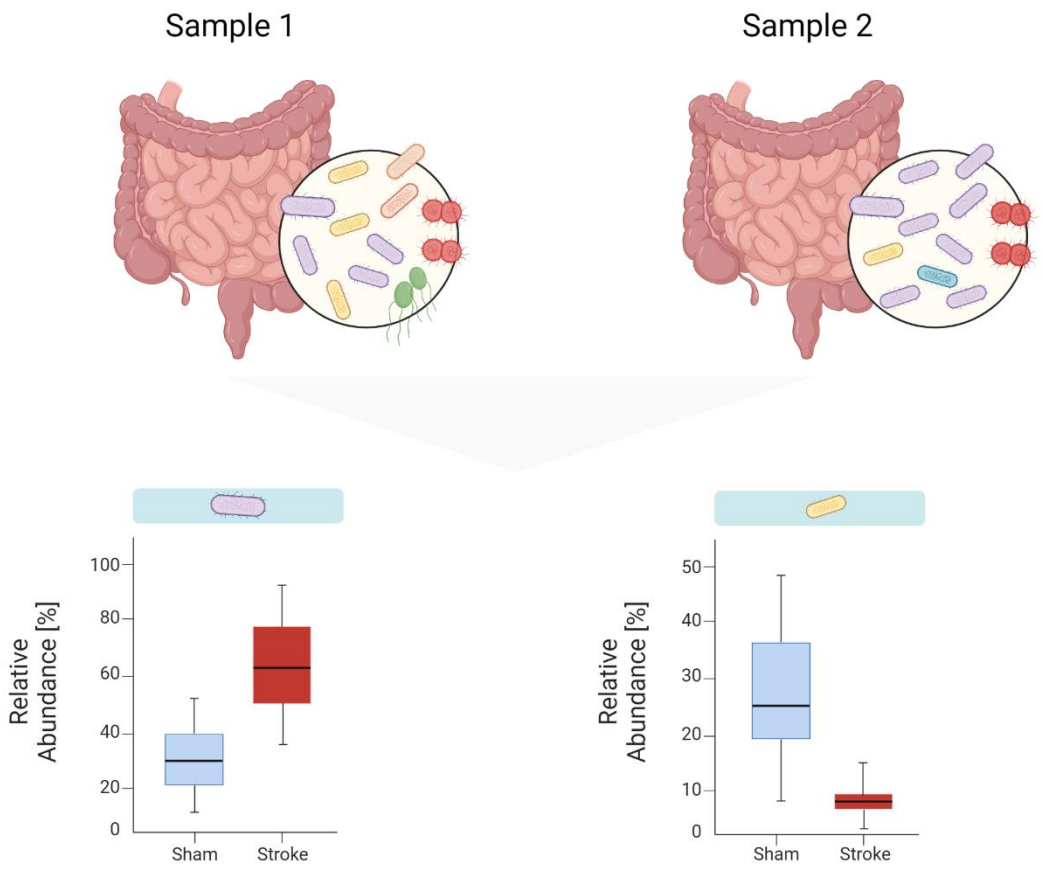
- Beta-diversity – between samples, how does the composition of species differ among samples?



Differential Abundance

- Which individual species differ between samples?

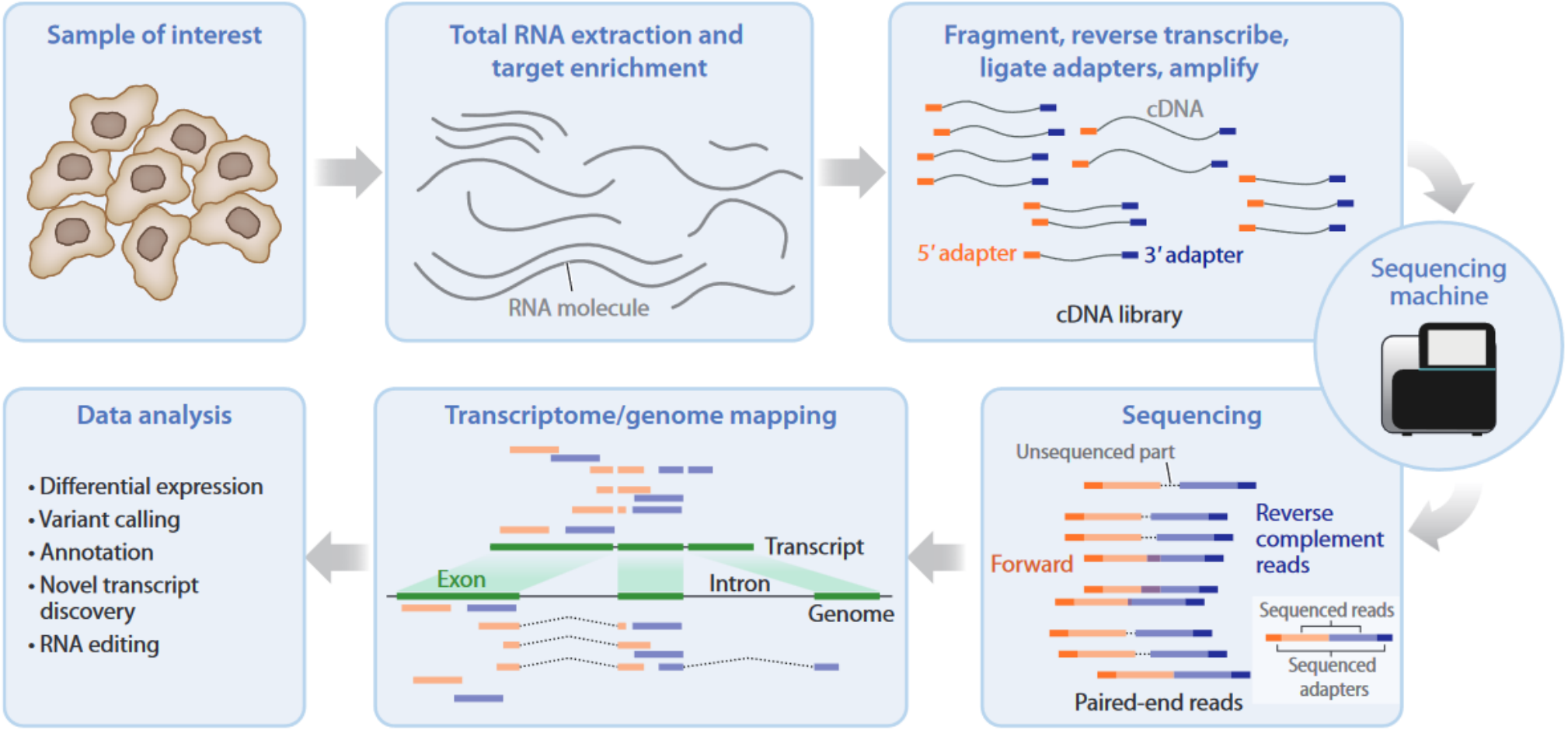
- In this example, the purple species is enriched in stroke and yellow species is depleted.



Lecture Outline

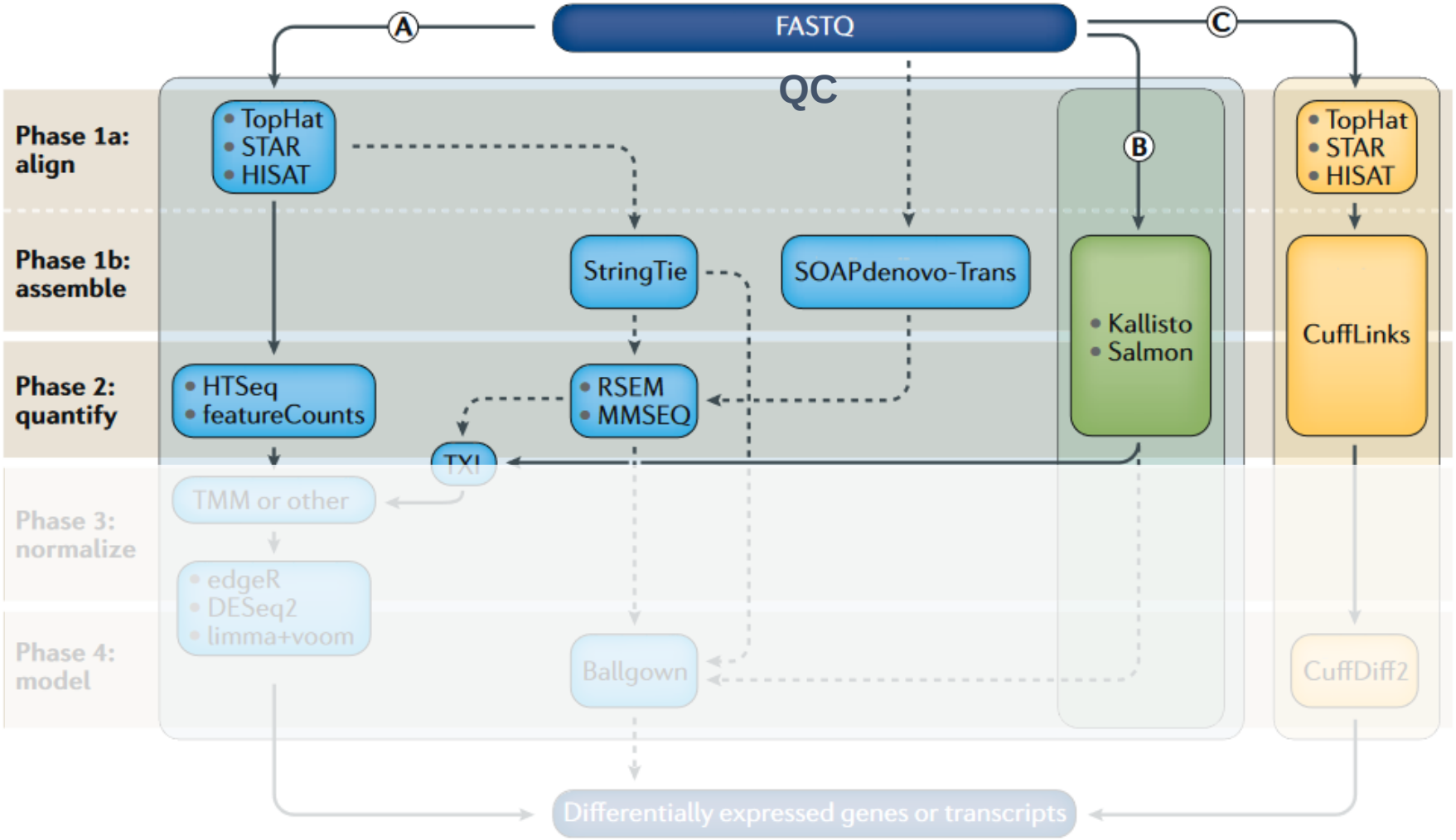
1. Introduction to Computational biology
2. Analysis of microbiome data
3. Introduction to RNAseq
4. Introduction to the R programming language – theory and concepts

RNAseq - principle



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

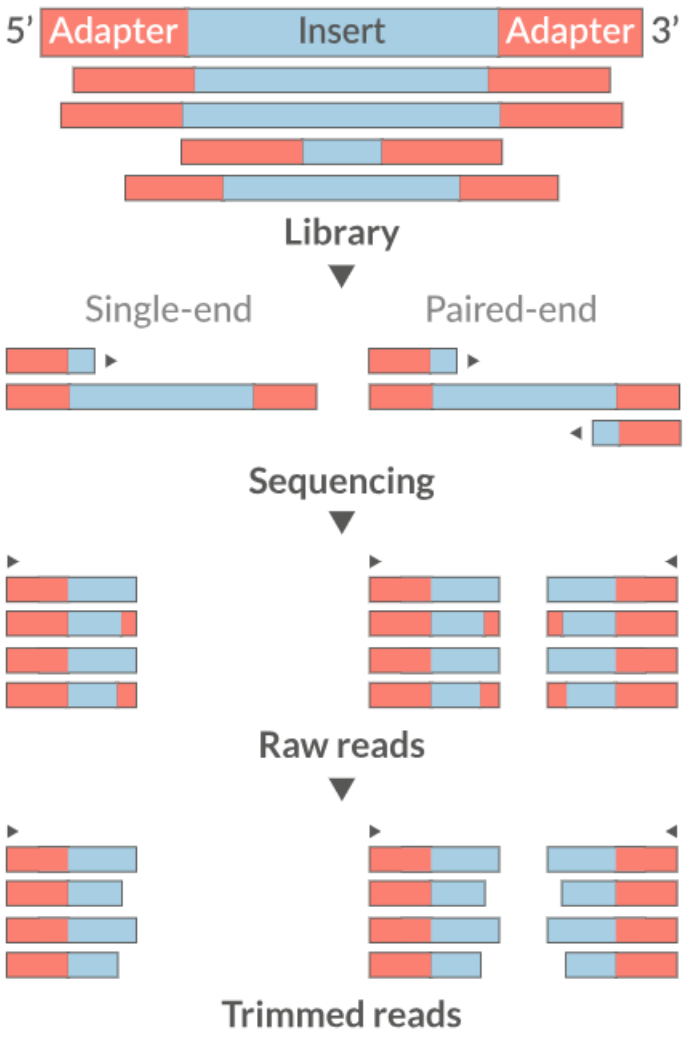
RNAseq data processing steps



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

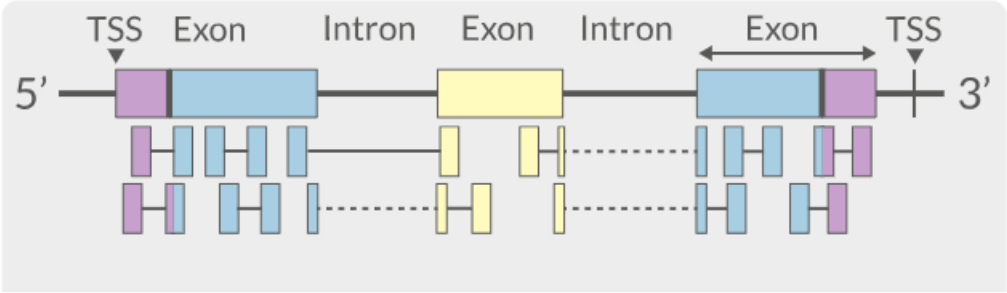


Mapping and quantification

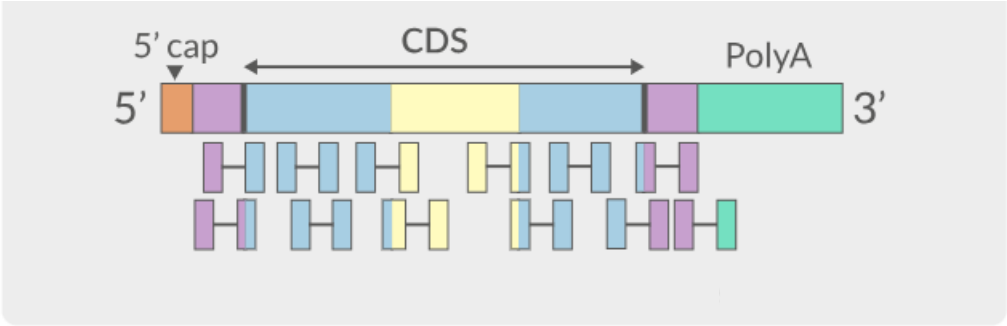
- Aligning trimmed and filtered reads to reference sequence
  - Genome (splice-aware) or transcriptome.

- Quantifying number of hits to each gene/transcript

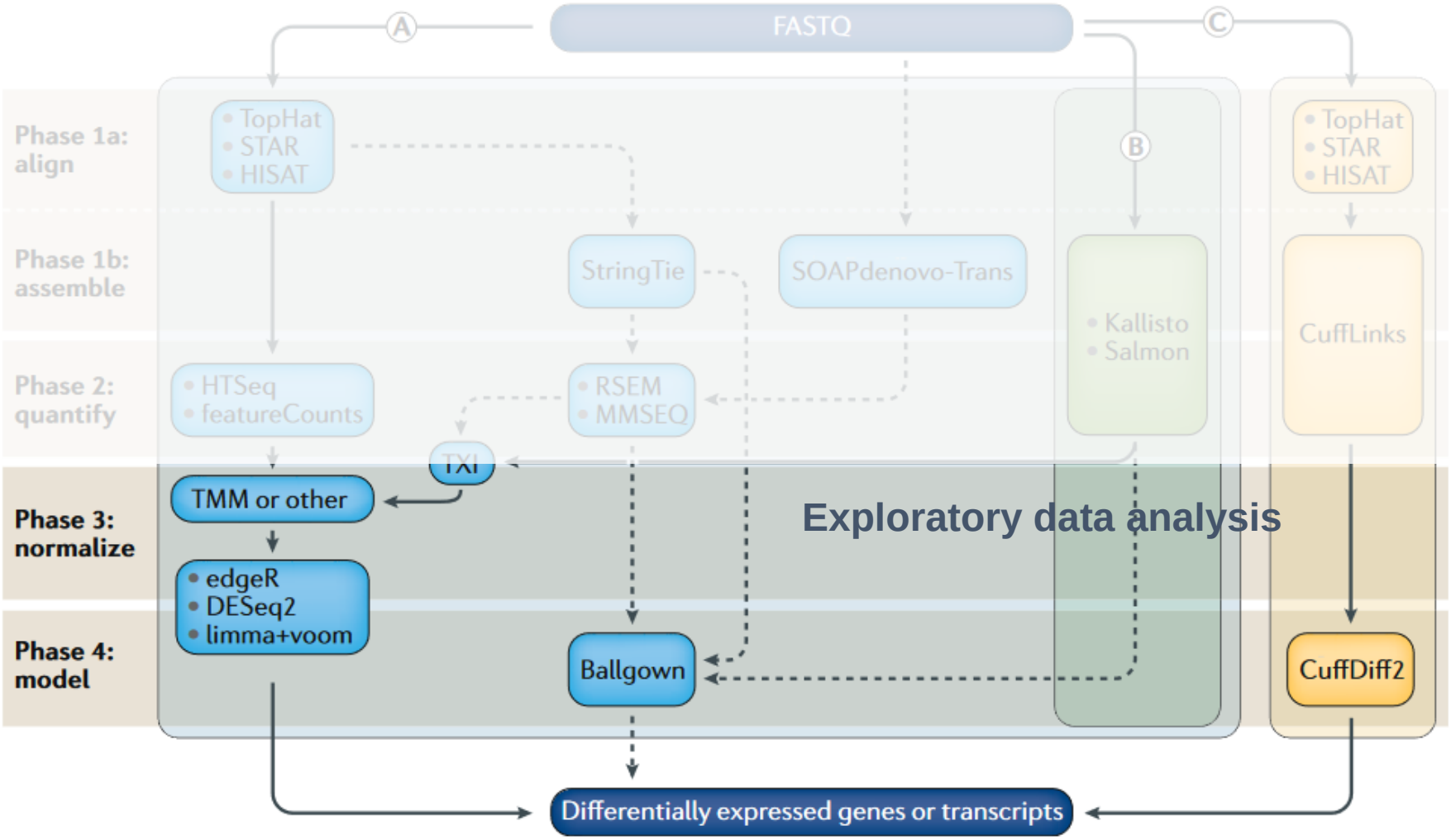
Genome



Transcriptome



RNAseq data analysis steps

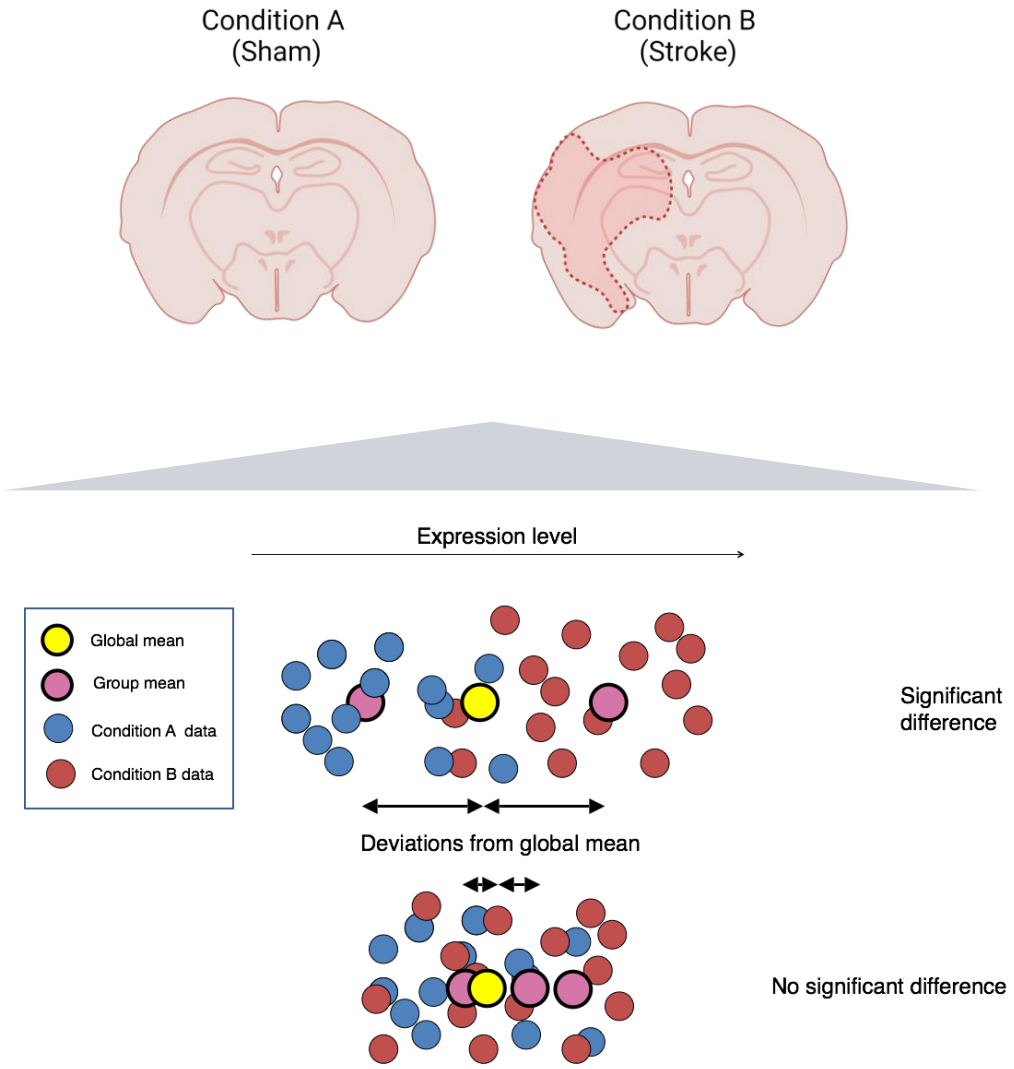


Pathway/GO analysis

Stark, Grzelak and Hadfield, 2019. Nature Reviews Genetics

Differential expression

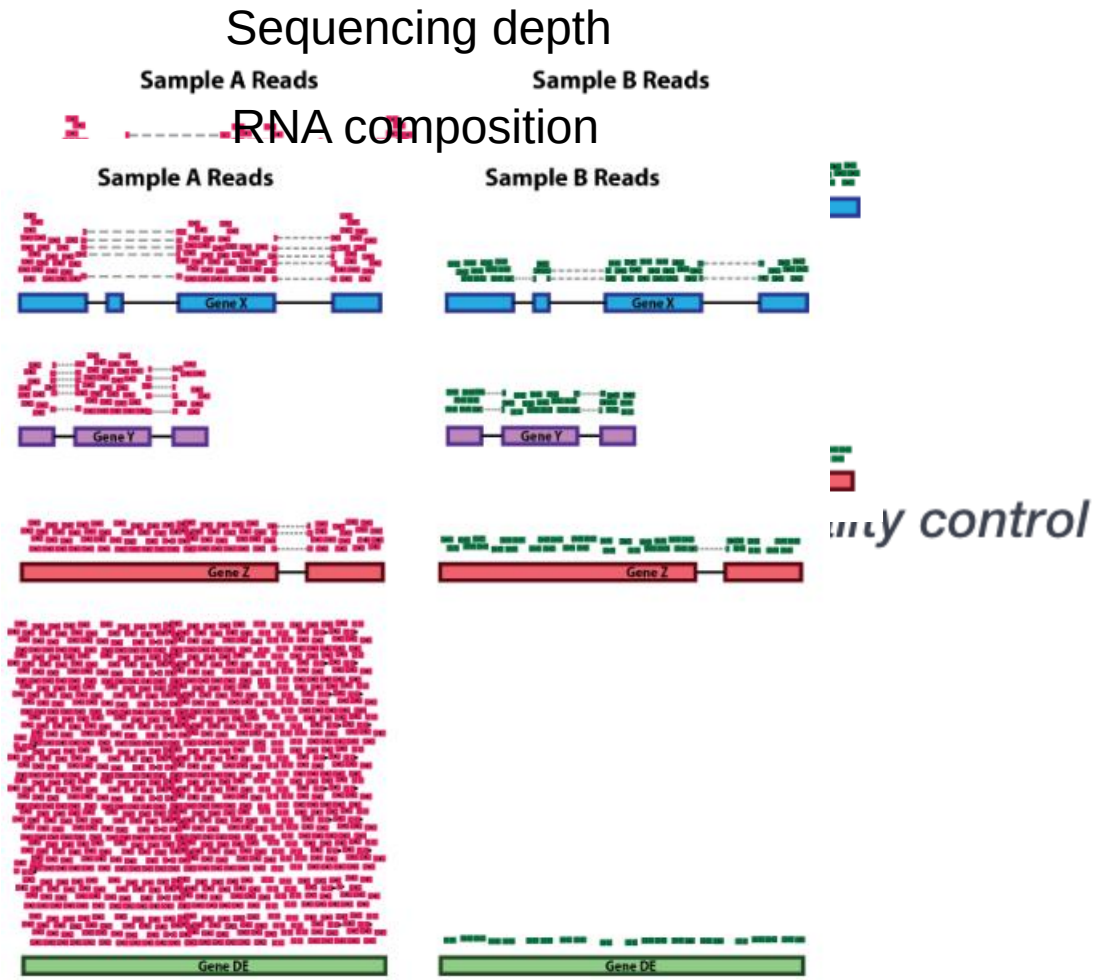
- Which genes/transcripts are different between conditions?
- Common tools include DESeq2, edgeR and limma/voom.



Differential expression - Normalization

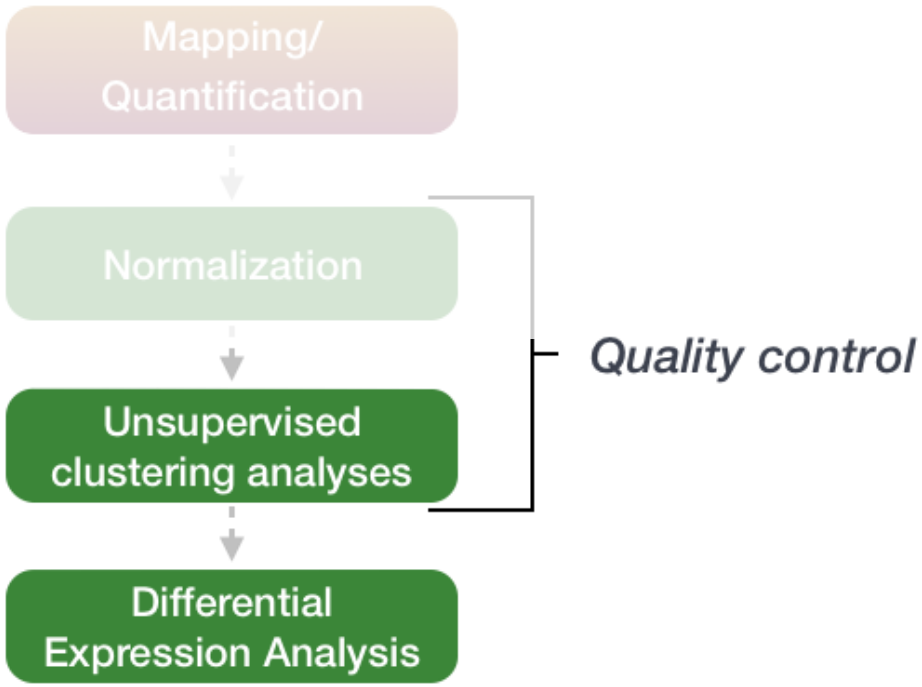
- Post-mapping –
  - Count matrix representing number of reads originating from each gene/transcript.

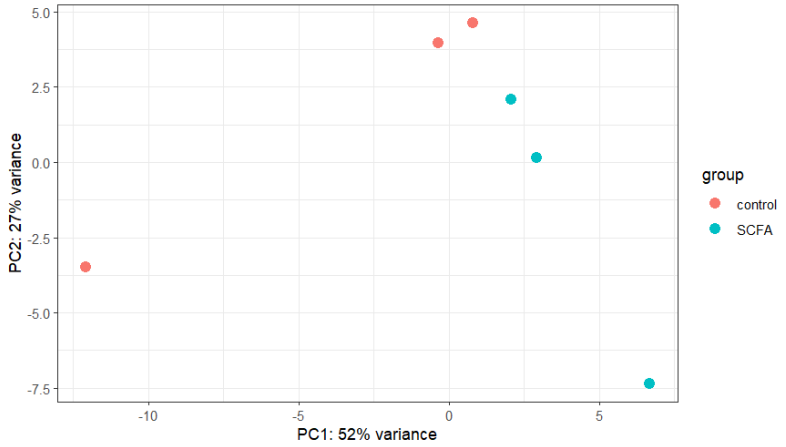
- Raw counts not comparable between samples
  - Sequencing depth and RNA composition differ.



Differential expression - Unsupervised clustering

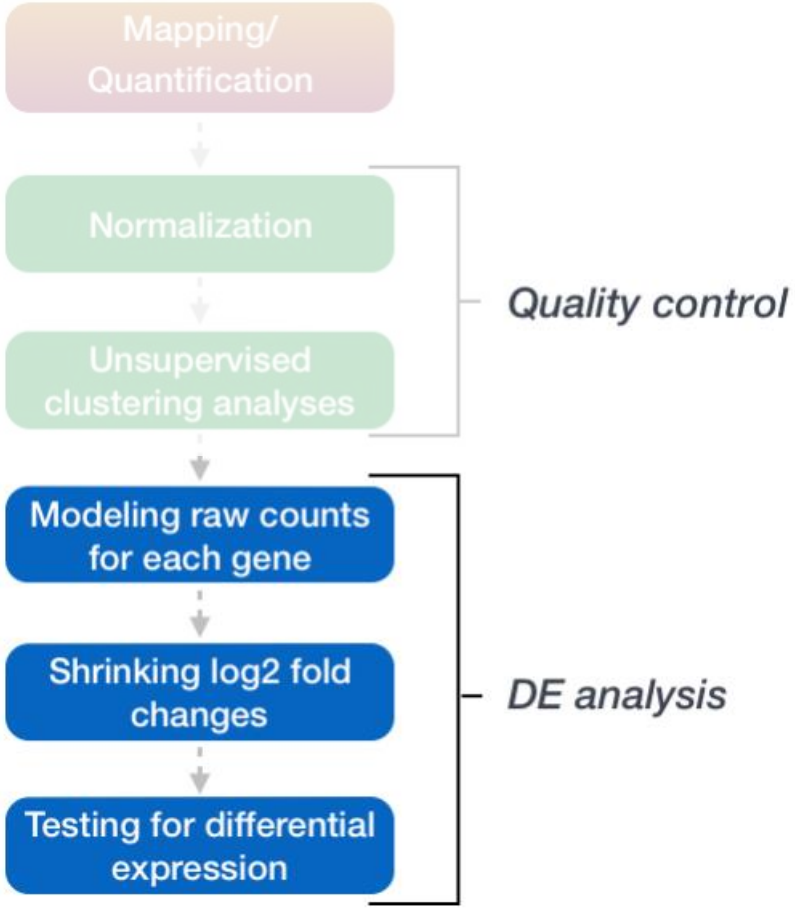
- Important to understand how similar/different samples are.
- Also, useful to examine data for outliers/confounding variables
- Principal component analysis (PCA) is a useful tool for this





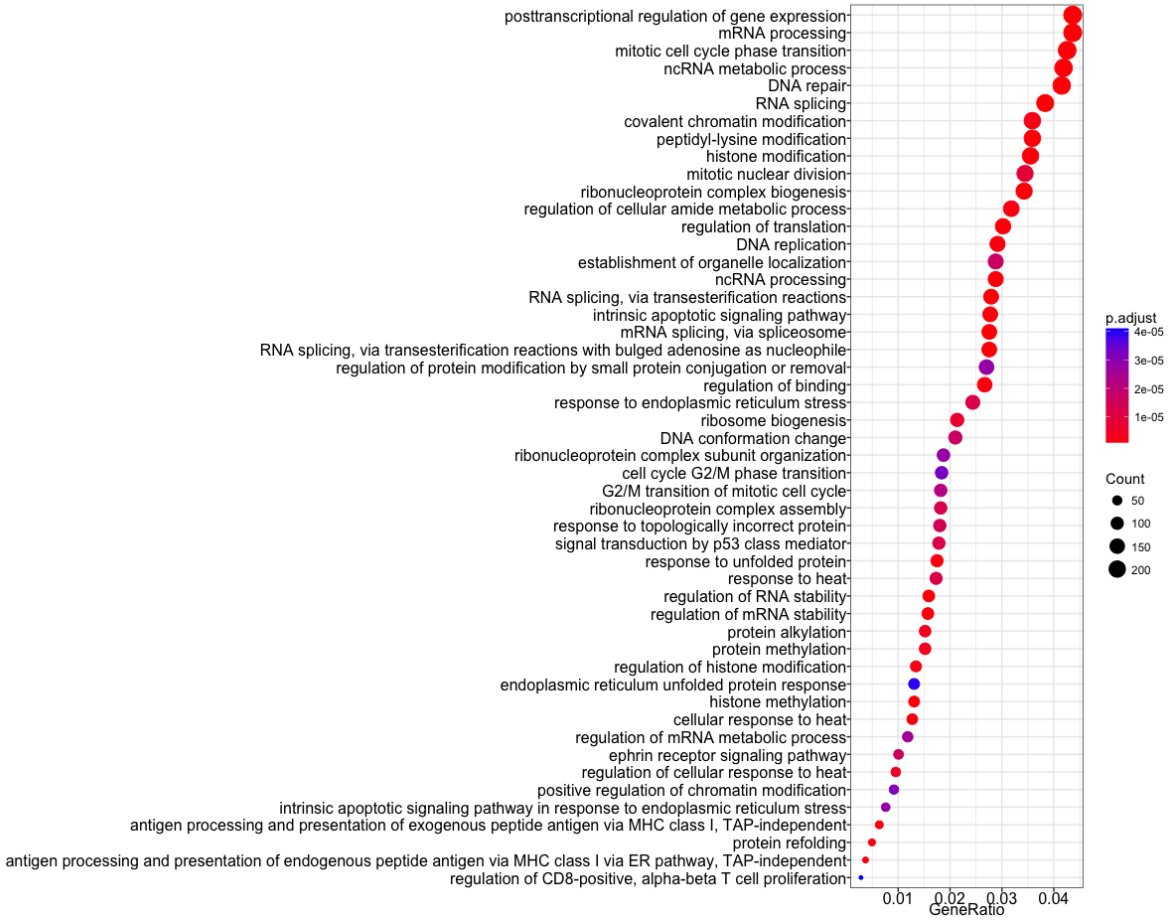
Differential expression – Identifying DEGs

- 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. Analysis of microbiome data
3. Introduction to RNAseq
4. Introduction to the R programming language – theory and concepts

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 w ... (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!

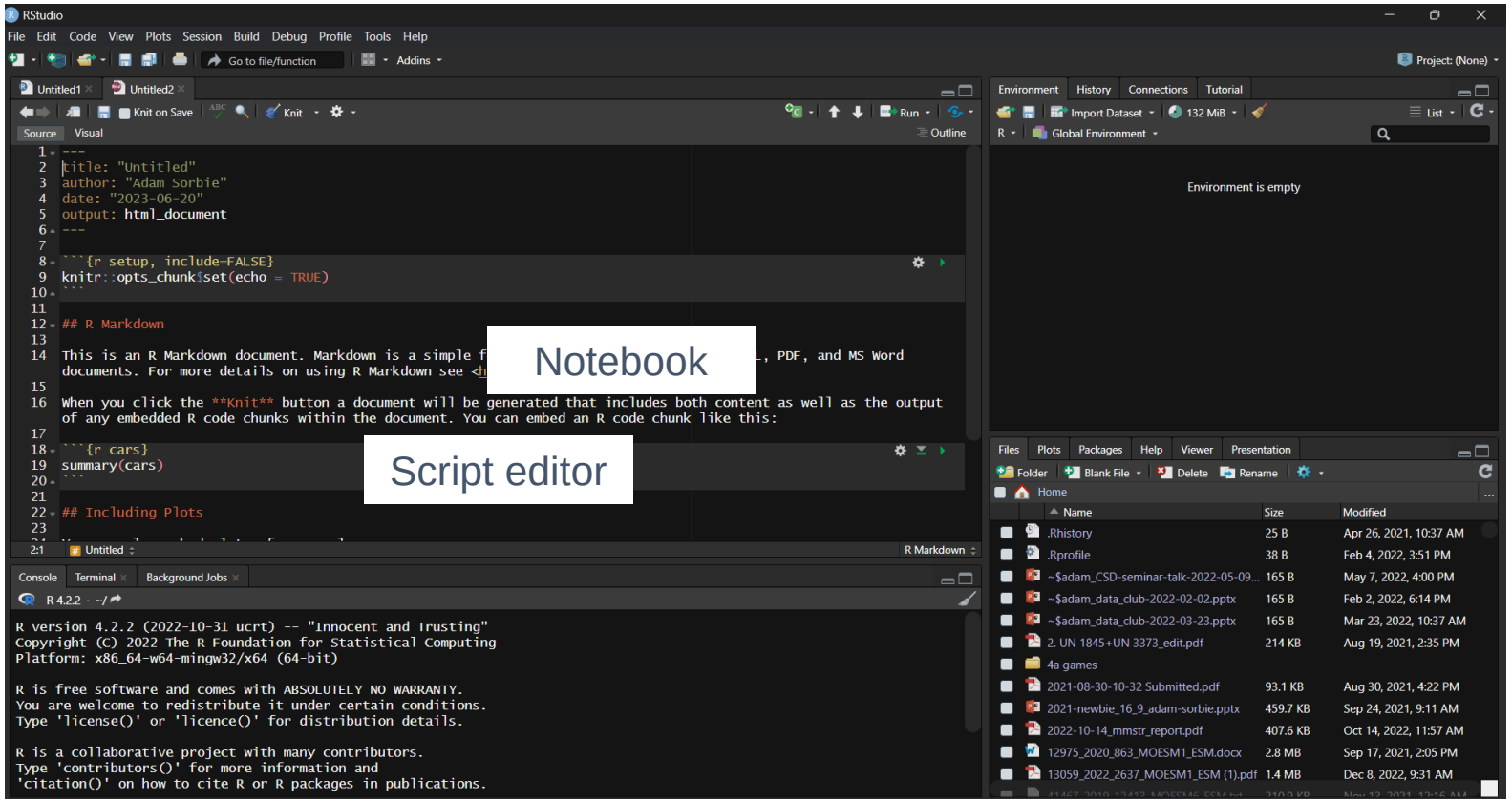
- The latest release (2023-06-16, Beagle Scouts) R-4.3.1.tar.gz, read what's new in the latest version.
- Sources of R alpha and beta releases (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are available here. Please 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

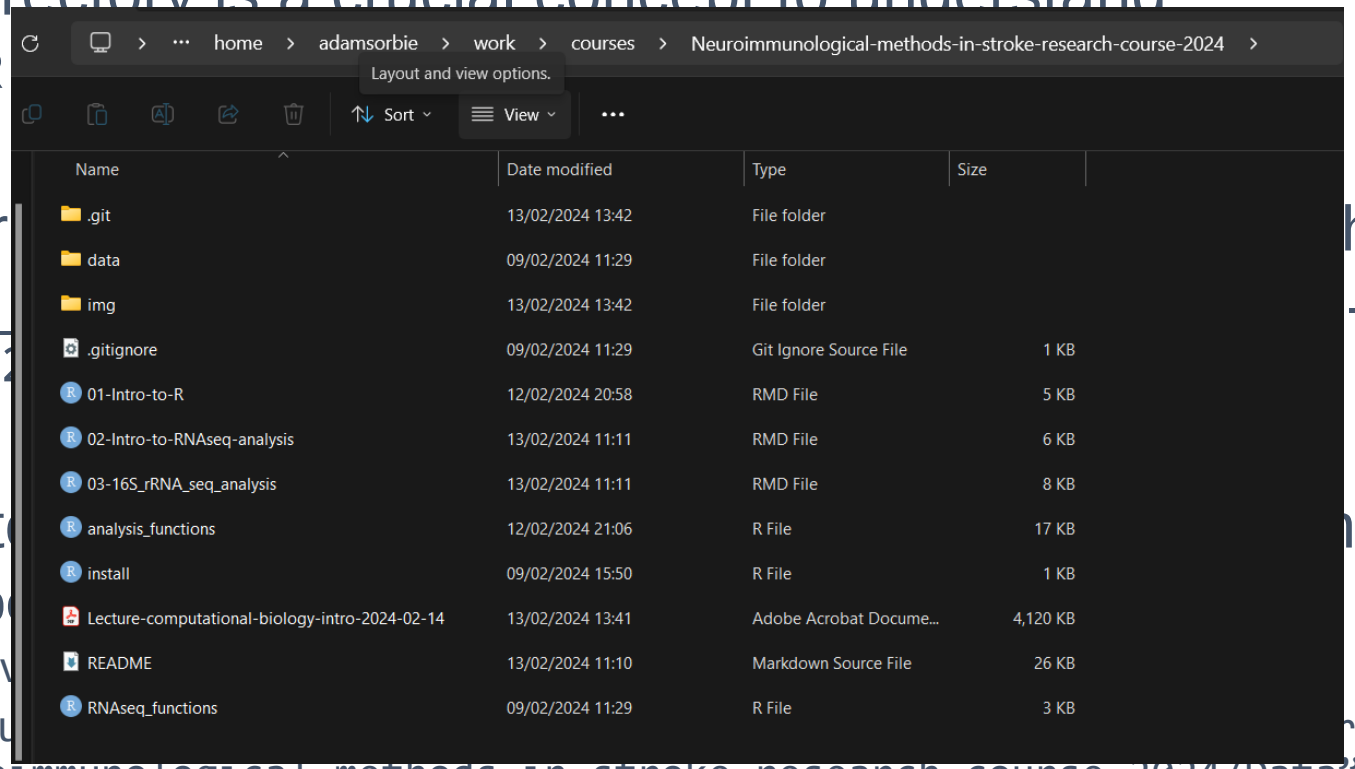


- Script editor – where you write scripts
- Console – run code interactively
- Environment – things you create/source stored here.

- 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)

Any questions?

Institute for Stroke and
Dementia Research (ISD)