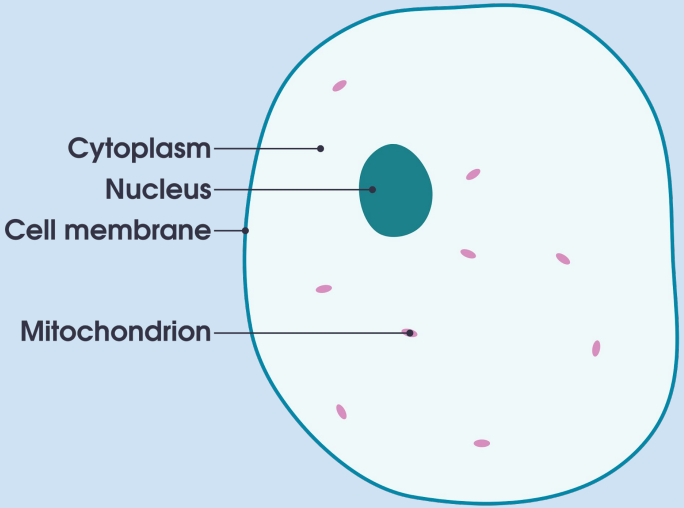
Genomics Data : Generation, Management, & Analysis

Introduction to cell biology

Central Dogma of Molecular Biology

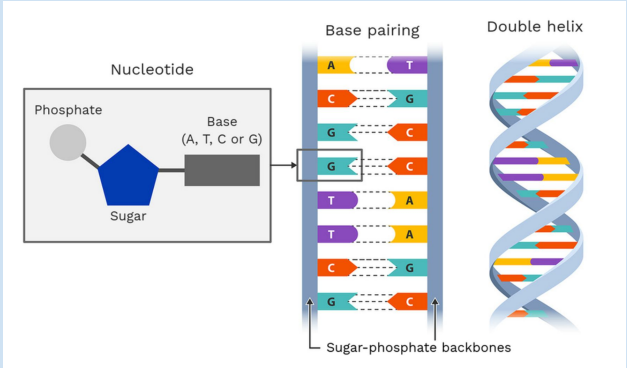
Cells

- Fundamental units of life
- **Cells store information in DNA** found in the cell nucleus
- **All cells** in a multicellular organism contain the **exact same DNA**, but cells have very different functions
- Cells “turn on” different sets of genes depending on their developmental history and environmental cues



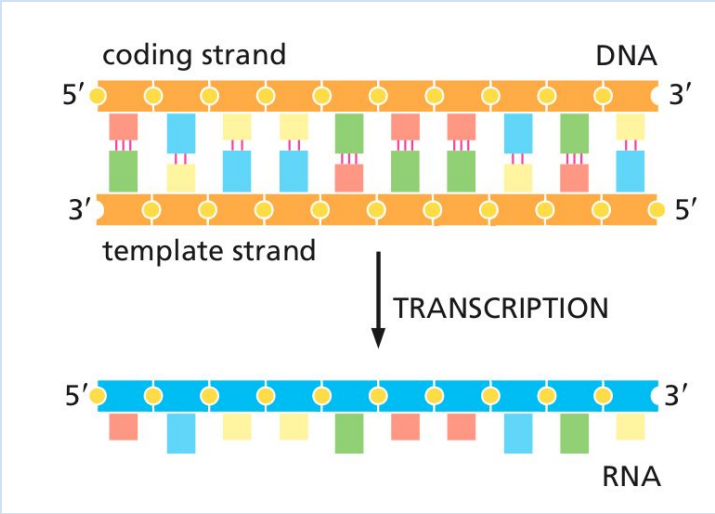
DNA and the genome

- **Nucleotide** : building block of DNA
  - The four bases are *adenine, guanine, thymine* and *cytosine* (A, G, T, C)
  - Two nucleotides bound together are called a ***base pair***
- **DNA** : double-helix with two complementary DNA strands, held together by bonds between A-T and C-G pairs
- Genetic information is carried in DNA encoded by a sequence of nucleotides
  - These are stored in chromosomes, which are single, long strands of DNA
- **Genome** : collection of all DNA in an organism
- **Gene** : section of DNA which encodes instructions to produce a specific product
  - The human genome contains around 25,000 genes



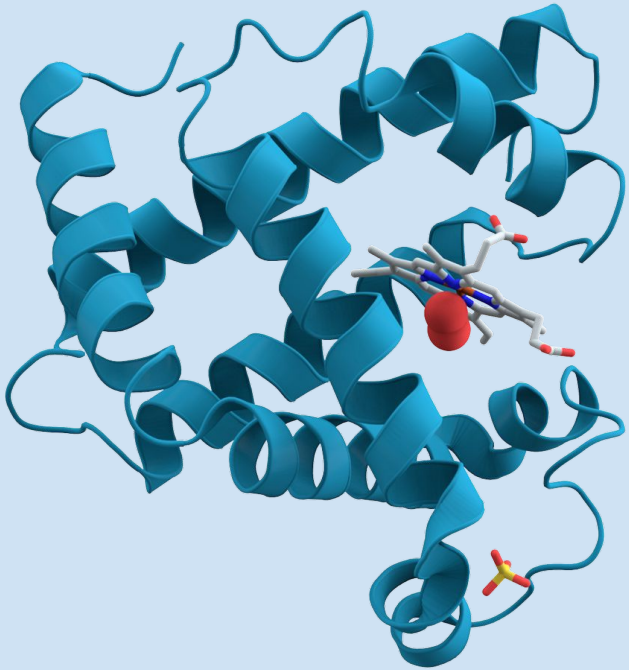
From DNA to protein : gene expression

- Genetic information flows from DNA -> RNA -> protein (central dogma)
- To *express* the information in DNA, the gene is *transcribed* into RNA
- **RNA** is similar to a single-strand version of DNA
- **messenger-RNA** molecules carry instructions for making proteins
- mRNA molecules are synthesised into protein through **translation**



Proteins

- **Proteins are complex functional molecules with diverse roles**
- Examples : hormones, enzymes, building blocks, catalysts
- Each protein has its own unique sequence of amino acids determining its shape and biological function

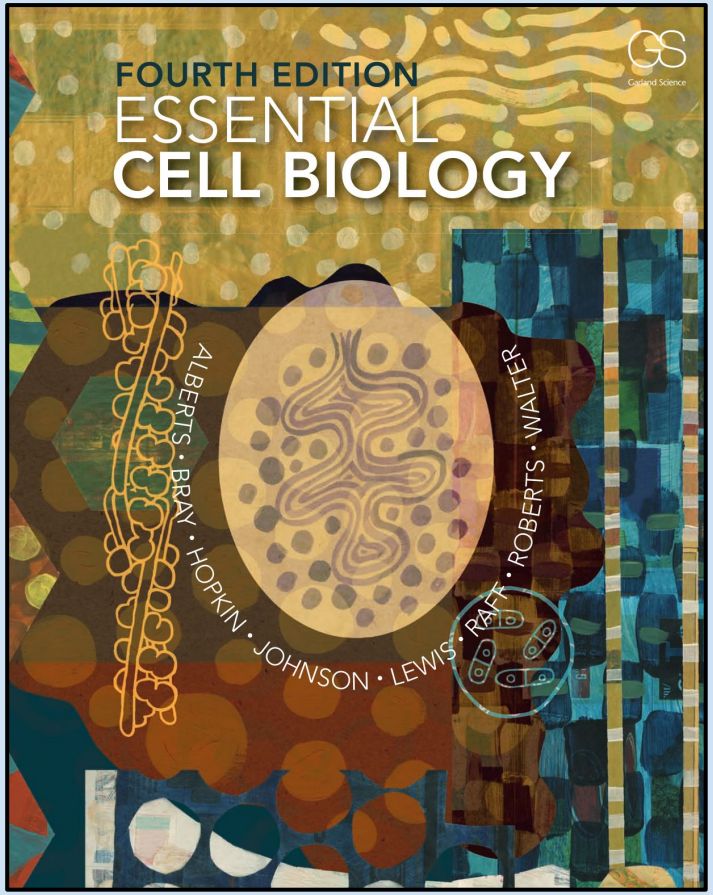


For more information

Alberts, B., Bray, D., Hopkin, K., Johnson, A. D., Lewis, J., Raff, M., Roberts, K., & Walter, P. (2013). *Essential cell biology* (4th ed.). Garland Publishing.

Free pdf online

Chapters : 1, 4, 5, 7, 8



Omics Data

General motivation

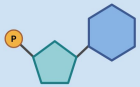
- Biological systems are complex, involving networks of interacting molecules that drive health and disease processes

- Unraveling these interactions can reveal how diseases work at the molecular level and identify biomarkers for diagnosis or treatment

- **High-throughput technologies enable the simultaneous measurement of thousands of molecules in a single biological sample, providing a comprehensive snapshot in time**

What are omics data?

-omics ≈ all parts considered collectively

	Field	Molecule	Short Description
	Genomics	DNA	Study of complete DNA sequence
	Epigenomics	DNA	Chemical changes to the genome
	Transcriptomics	RNA	Gene expression
	Proteomics	Protein	Functional proteins and their interactions
	Metabolomics	Metabolites	Small molecules resulting from cellular metabolic processes

Why are we interested in omics data?

- **Comprehensive & unbiased** : measuring all variables in a system reveals patterns missed by single-target measurements
- **Biomarker discovery :** signatures for prognosis, diagnosis, and treatment
- **Mechanistic insight :** discovery of mechanisms of treatment and disease, guiding future development

Some success stories in omics research

Genomics : gene variants (BRCA1, BRCA2) in breast cancer patients successfully identified patients who benefit most from olaparib treatment

- Tutt ANJ, et Al. Adjuvant Olaparib for Patients with *BRCA1*- or *BRCA2*-Mutated Breast Cancer. N Engl J Med. doi: 10.1056/NEJMoa2105215.

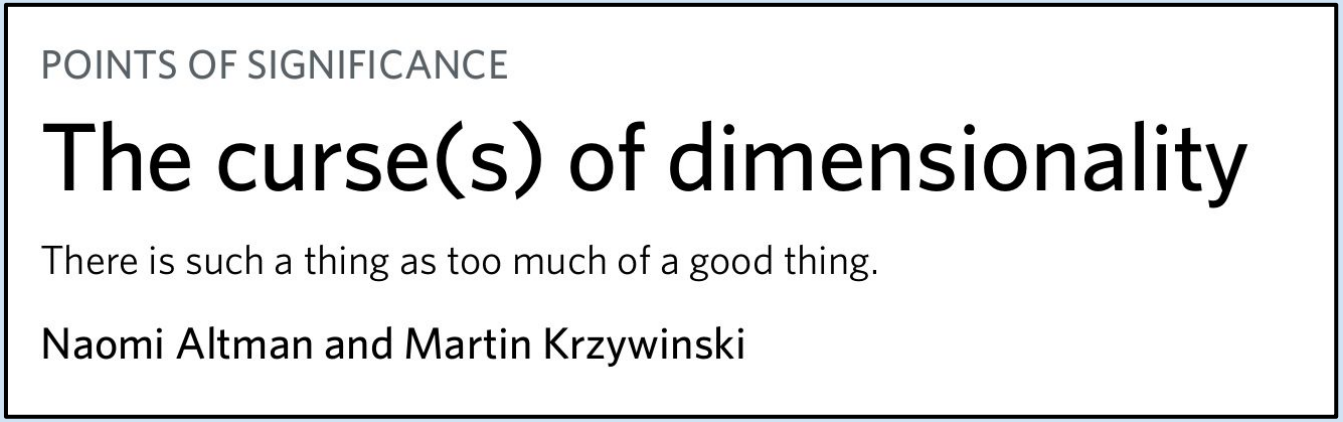
Transcriptomics : signature of 21 genes used to predict chemotherapy benefit in breast cancer patients

- Sparano JA, et al. Adjuvant Chemotherapy Guided by a 21-Gene Expression Assay in Breast Cancer. N Engl J Med. 2018 doi: 10.1056/NEJMoa1804710.

Metabolomics : screening newborns for metabolic diseases using mass spectrometry

- Chace DH, et al. Use of tandem mass spectrometry for multianalyte screening of dried blood specimens from newborns. Clin Chem. 2003 doi: 10.1373/clinchem.

Why are omics data hard to analyse?


POINTS OF SIGNIFICANCE

The curse(s) of dimensionality

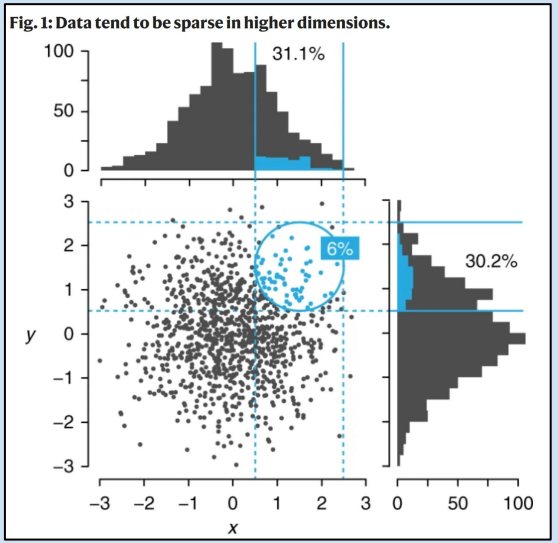
There is such a thing as too much of a good thing.

Naomi Altman and Martin Krzywinski

Altman, N., Krzywinski, M. The curse(s) of dimensionality. *Nat Methods* **15**, 399–400 (2018). https://doi.org/10.1038/s41592-018-0019-x

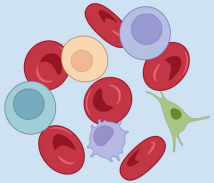
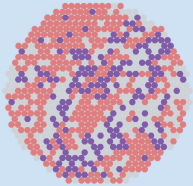
The many curses of high-dimensionality

1. **Data sparsity :** higher dimension -> data occupies more volume -> true signal “drowned” in dimension
2. **Multicollinearity :** redundant information when variables can be expressed as combinations of other variables
3. **Inflated false discovery rates :** more statistical tests -> more false positives dominate
4. **Overfitting :** Flexible prediction models exploit random patterns in the data which contain no true signal

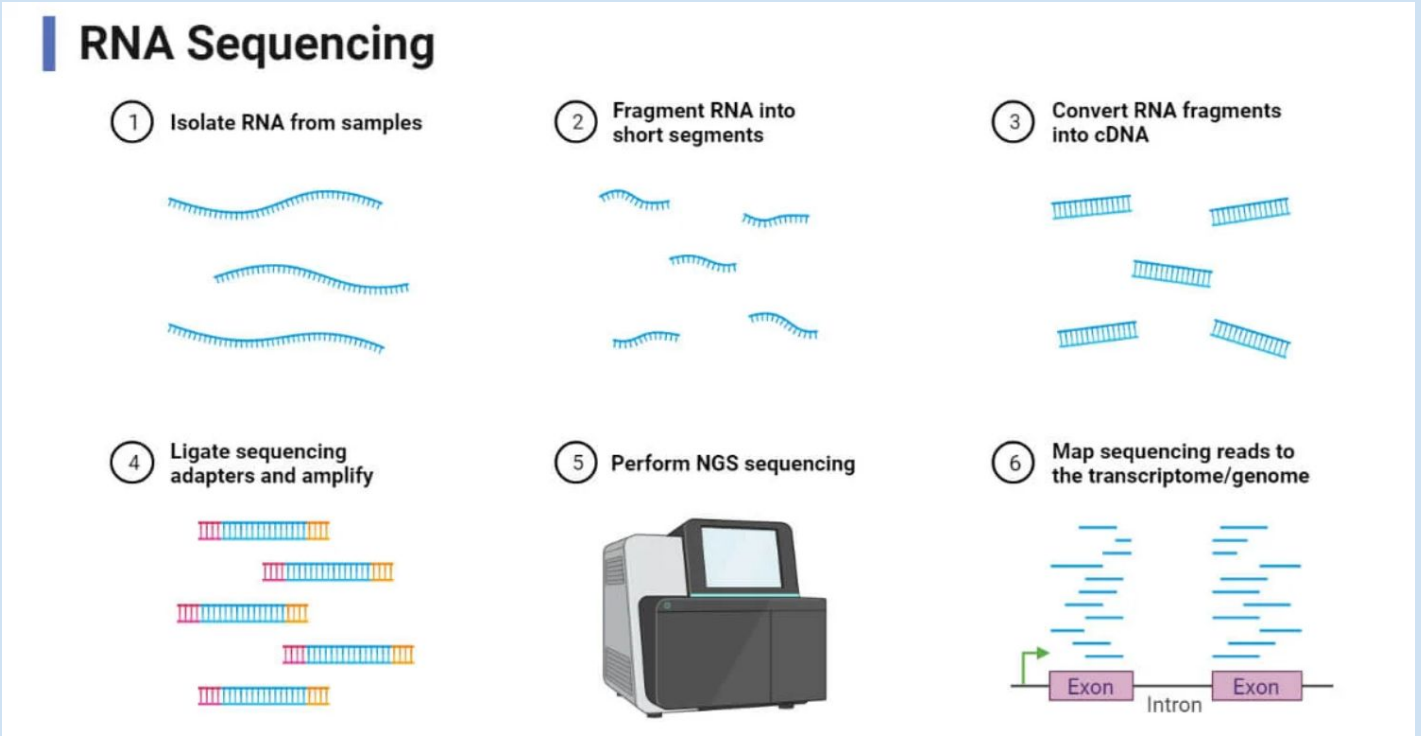


Transcriptomics data

Types of transcriptomic data

	Bulk	Single-cell	Spatial
			
Description	Aggregated expression in mixed cell types	Expression of individual cells	Expression of individual cells and their location in a tissue sample
Strengths	Cheap, simple analysis, good for large effect sizes	High resolution, insight into cell heterogeneity and rare cell types	Preserves biological structure, spatial interactions
Limitations	Rare cell types expression drowned out	More expensive, complex analysis	Most expensive, complex generation and analysis
Cost per sample	$200-$500	$1,000-$3,000	$5,000-$10,000

RNA-seq data generation



What does the data look like?

Genes →

Samples ↓

	ENSG00000000003	ENSG00000000005	ENSG00000000419	ENSG00000000457
Berry_London_Sample1	5	0	681	699
Berry_London_Sample2	14	0	596	809
Berry_London_Sample3	1	0	212	294
Berry_London_Sample4	6	0	619	665
Berry_London_Sample5	4	0	324	322
Berry_London_Sample6	6	0	297	251
Berry_London_Sample7	8	0	719	787
Berry_London_Sample8	18	0	855	872

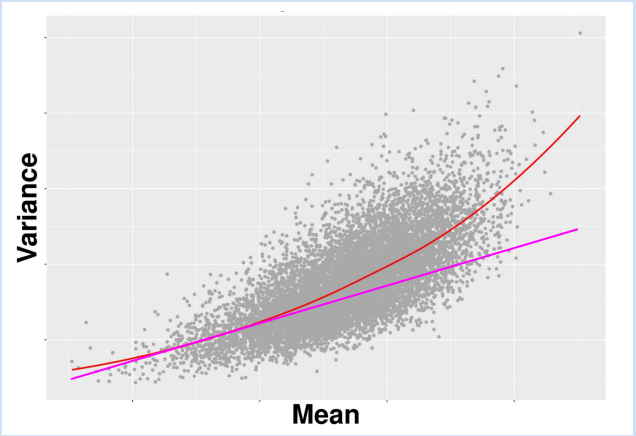
Sequencing depth and coverage

- **Coverage :** percentage of the transcriptome which is detected by sequencing
- **Sequencing depth / library size :** total number of reads in a sample

Detecting rare transcripts requires high sequencing depths!

Statistical problems in RNA-seq data

- **Curses of dimensionality**
  - Sparsity, multicollinearity, inflated false positives, overfitting
- **Count data requires special methods**
  - Discrete, non-negative
- **Heteroskedastic & overdispersed**
  - Mean depends on variance
  - Variance typically greater than mean
- **Large dynamic range & Inflated in zeros**
  - Typical range : 0 - 2,000,000
  - 0 is over-represented compared to rest of range due to unexpressed genes

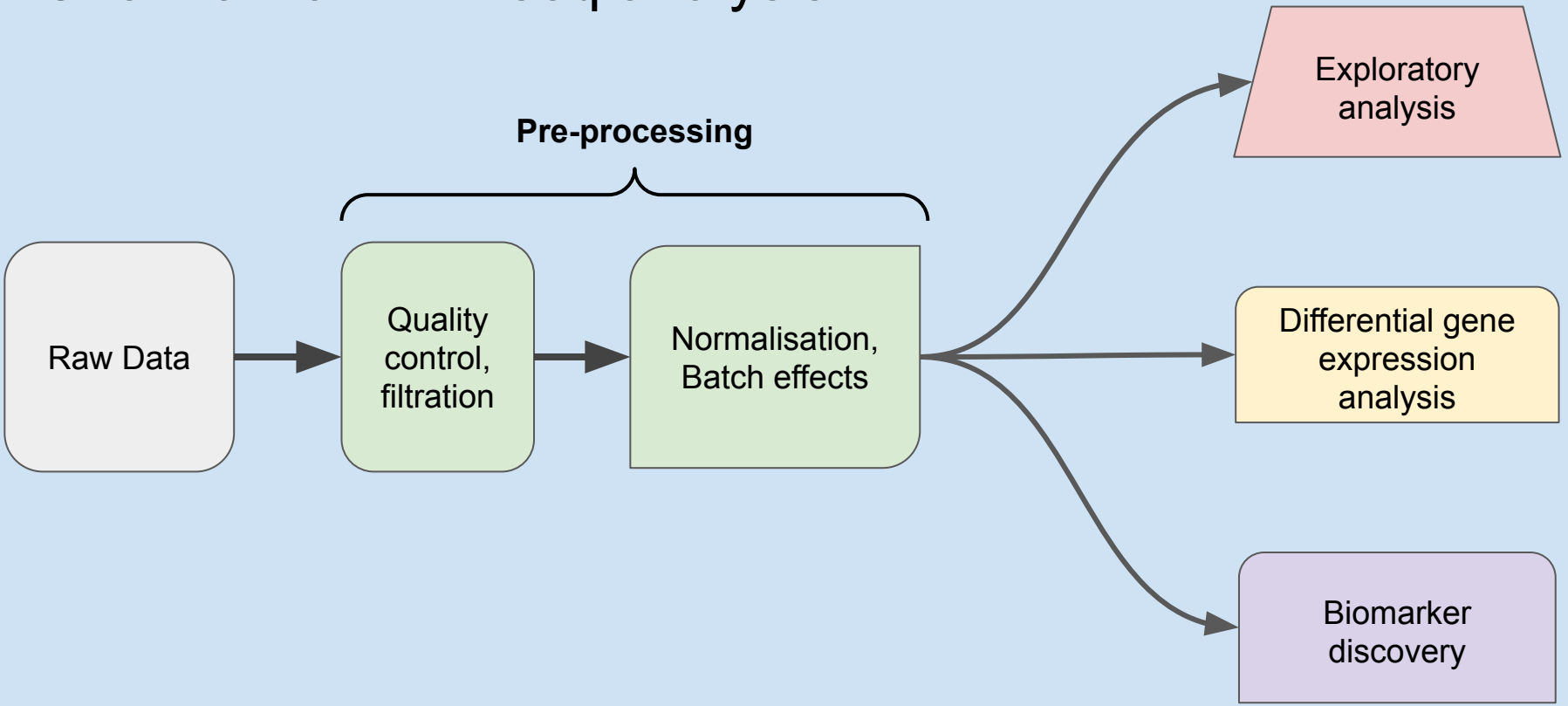


Mean vs variance for genes (simulated data)

The red line shows the non-linear, increasing relationship (heteroskedasticity)

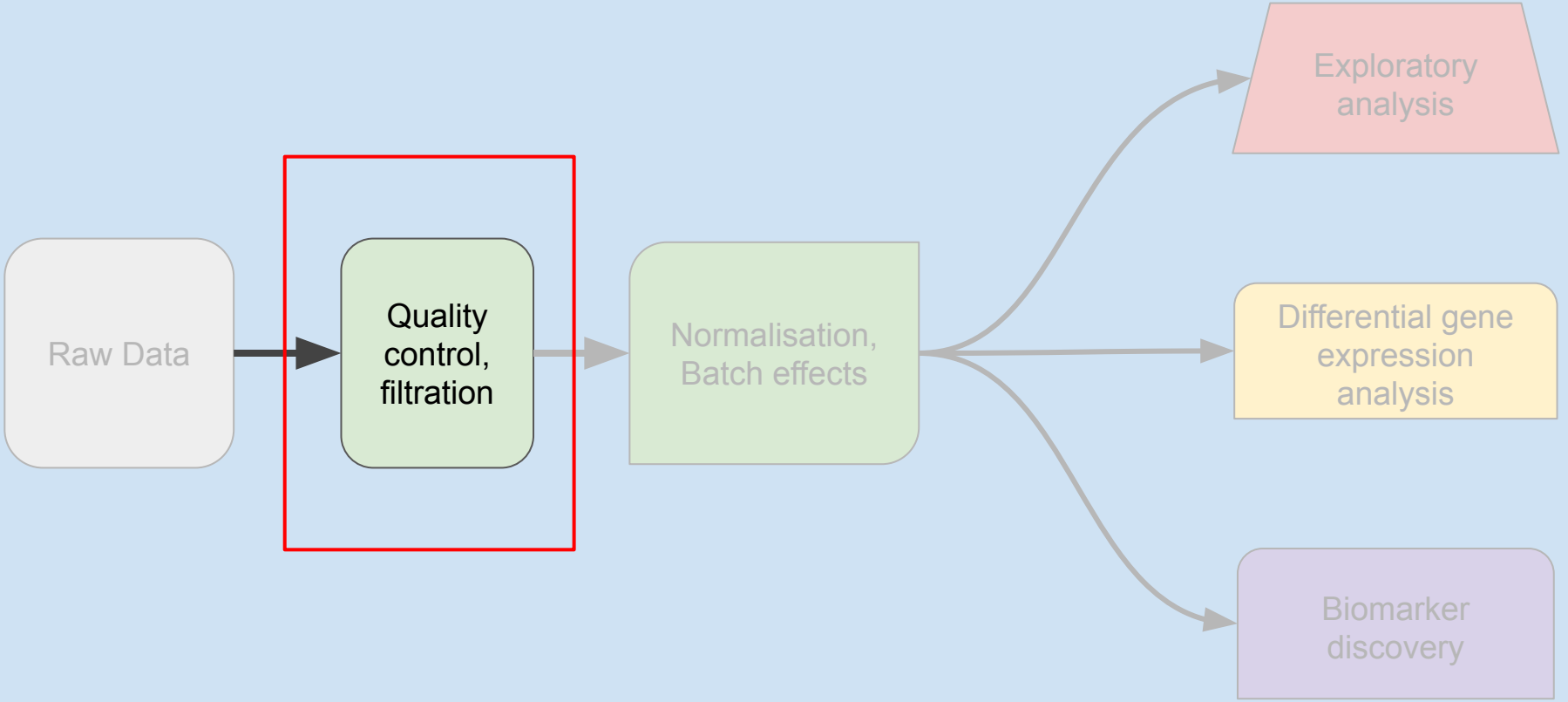
The pink line is where mean = variance, demonstrating overdispersion

Overview of RNA-seq analysis



Pre-processing of RNA-seq Data

Quality Control & Filtration


Raw Data
Quality control, filtration
Normalisation, Batch effects
Exploratory analysis
Differential gene expression analysis
Biomarker discovery

Quality control basics

- **RIN (RNA integrity number)**
  - Number 1-10 giving the quality of sequenced RNA
  - Typically samples with low RIN are removed from study
  - A threshold of RIN >=8 is common
- **Sequencing depth / library size**
  - Number of reads in a sample
  - Higher depth = more precise detection of weakly expressed genes
  - Samples with lower than 1,000,000 reads could be removed

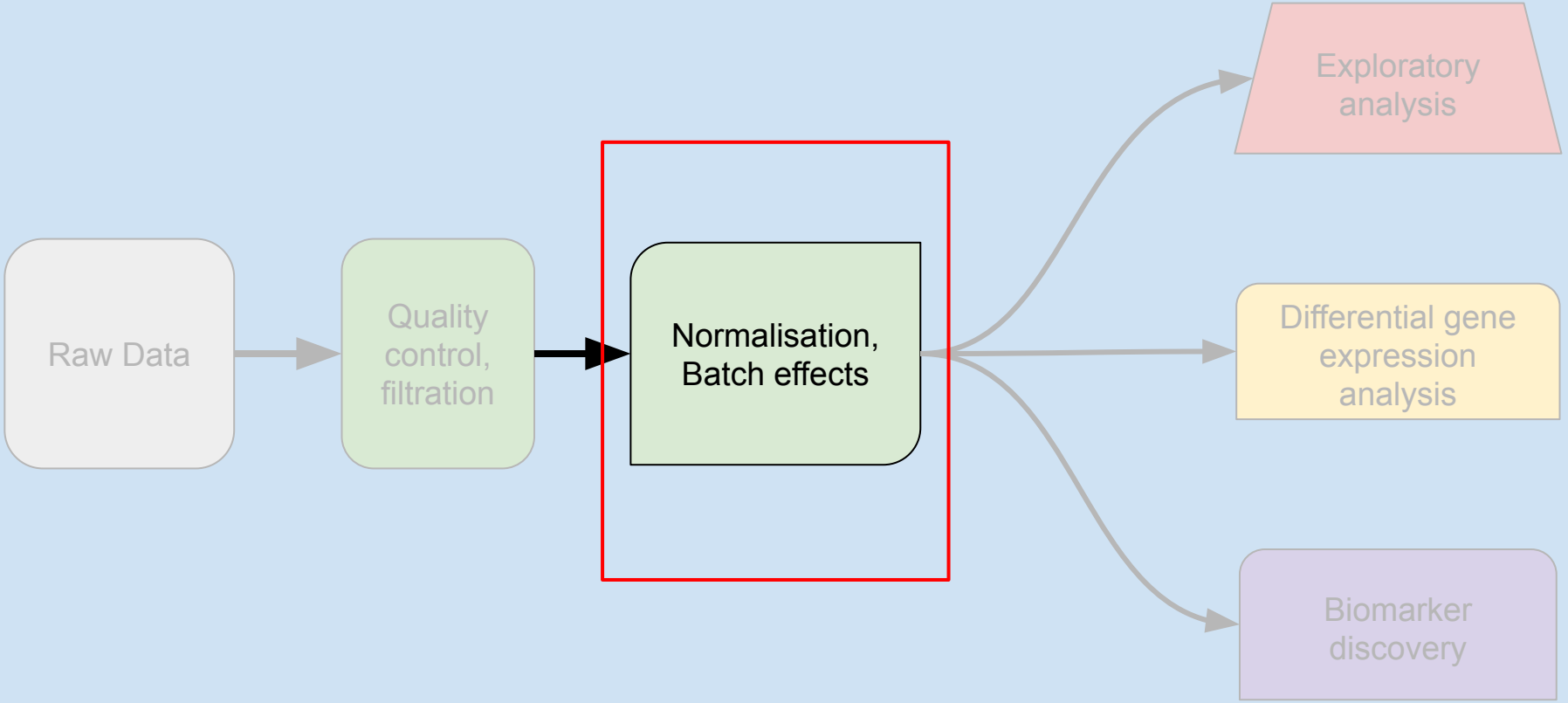
Filtration

We may choose to filter out genes who are uninteresting or who complicate the analysis of the data. This includes

- **Constant genes :** genes whose expression does not vary across samples. This includes unexpressed genes.
- **Non-protein-coding genes**
- **Weakly expressed genes :** detection of these genes is less precise
    - Thresholds are often arbitrary e.g. only keep genes who are expressed in more than 5 samples

Pre-processing of RNA-seq Data

Normalisation


Raw Data
Quality control, filtration
Normalisation, Batch effects
Exploratory analysis
Differential gene expression analysis
Biomarker discovery

Why do we normalise count data?

- Sequencing yields counts which mix two sources of information
  - **True biological abundance of each gene**
  - Technical effects
- Normalisation seeks to remove technical effects

Sample	Exposure	Gene 1	Gene 2	Gene 3	Library Size
A	Yes	300	300	400	1,000
B	No	600	600	800	2,000
True effect?		No	No	No	

No gene is differentially expressed, but more reads were sampled in B

A simple solution

Idea : use ***normalisation factors*** equal to the library size

- Entries are now relative abundance of each gene

Sample	**Exposure**	**Gene 1**	**Gene 2**	**Gene 3**	**Library Size**
A	Yes	0.3	0.3	0.4	1
B	No	0.3	0.3	0.4	1
True effect?		No	No	No	

Counts as proportions of total reads sampled

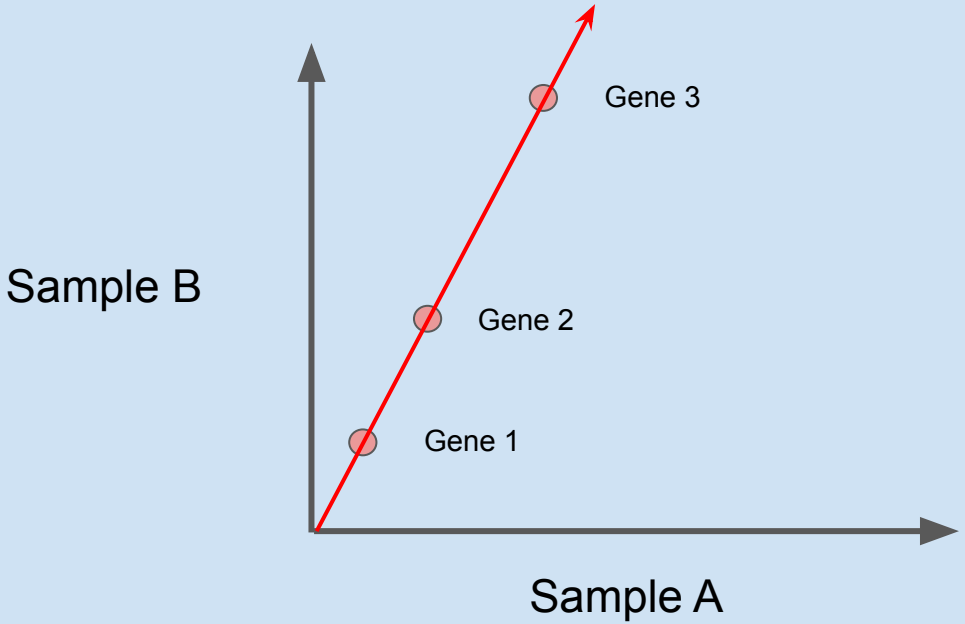
Counts-per-million

- Counts-per-million (CPM) multiplies these proportions by 1,000,000
  - Often this is accompanied by a log2 transformation (log2-CPM)

Sample	Exposure	Gene 1	Gene 2	Gene 3	Library Size
A	Yes	300,000	300,000	400,000	1,000,000
B	No	300,000	300,000	400,000	1,000,000
True effect?		No	No	No	

Counts in CPM

Graphical representation of CPM



Slope of red line =
normalisation factor

A problem with CPM

- **Di*fferentially abundant* genes** can change the relative abundance of genes with no true effect

Sample	**Exposure**	**Gene 1**	**Gene 2**	**Gene 3**	**Library Size**
A	Yes	300	300	1400	2,000
B	No	300	300	400	1,000
True effect?		No	No	Yes	

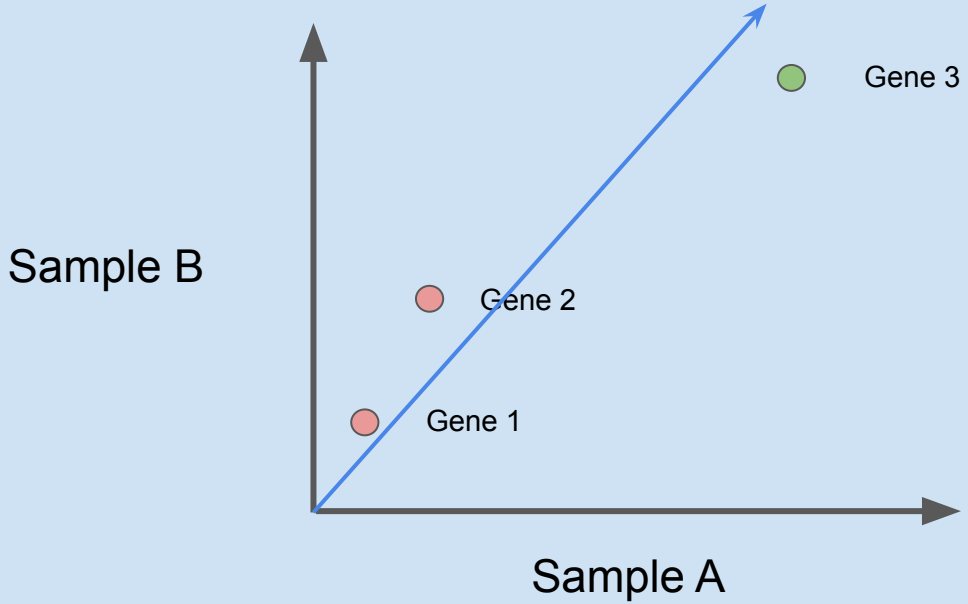
A problem with CPM

Sample	Exposure	Gene 1	Gene 2	Gene 3	Library Size
A	Yes	150,000	150,000	700,000	1,000,000
B	No	300,000	300,000	400,000	1,000,000
True effect?		No	No	Yes	

Counts in CPM

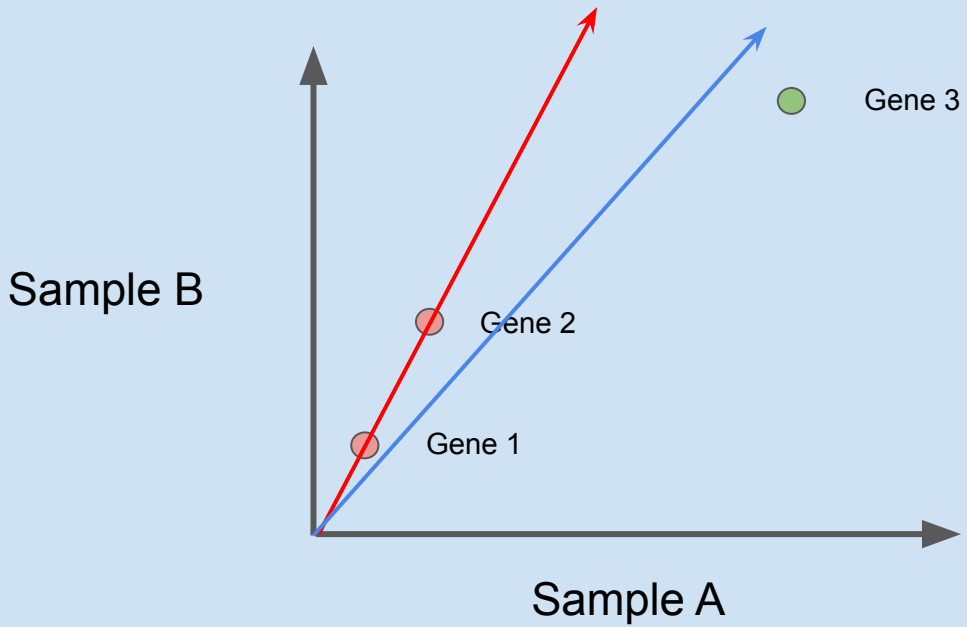
Genes 1 and 2 do not change expression, but appear artificially lower because gene 3 dominates the sample

Graphical representation of the problem with CPM



Slope of blue line = normalisation factor (biased by over-expression of gene 3)

Graphical representation of the problem with CPM



Ideally, we would like to estimate the red line (normalisation factors while excluding gene 3)

Trimmed-mean-of-M-Values normalisation

TMM estimates normalisation factors while ***trimming*** (i.e. excluding) extreme values

- It **assumes that most genes are not differentially abundant**
- It calculates normalisation factors while excluding extremely high or low values
  - These normalisation factors are called *M-values, and represent the ratio of relative abundances between exposure groups*

$$M_g = \frac{Count_{g,A}}{library_A} \div \frac{Count_{g,B}}{library_B}$$

Sample	Exposure	Gene 1	Gene 2	Gene 3	Library Size
A	Yes	0.15	0.15	0.7	1,000
B	No	0.3	0.3	0.4	2,000
M-values		0.5	0.5	1.75	
True effect?		No	No	Yes	

Counts as proportions of total reads sampled

Since the M-value of gene 3 is extreme, we exclude it when calculating the normalisation factors

- Normalisation factor = (0.5 + 0.5) / 2 = 0.5
- To normalise each count, we multiply the counts by the normalisation factors

Sample	Exposure	Gene 1	Gene 2	Gene 3	Library Size
A	Yes	300	300	1400	2,000
B	No	300	300	400	1,000
True effect?		No	No	Yes	

Raw counts

TMM normalisation

Sample	Exposure	Gene 1	Gene 2	Gene 3
A	Yes	150	150	700
B	No	150	150	200
True effect?		No	No	Yes

Counts with TMM normalisation

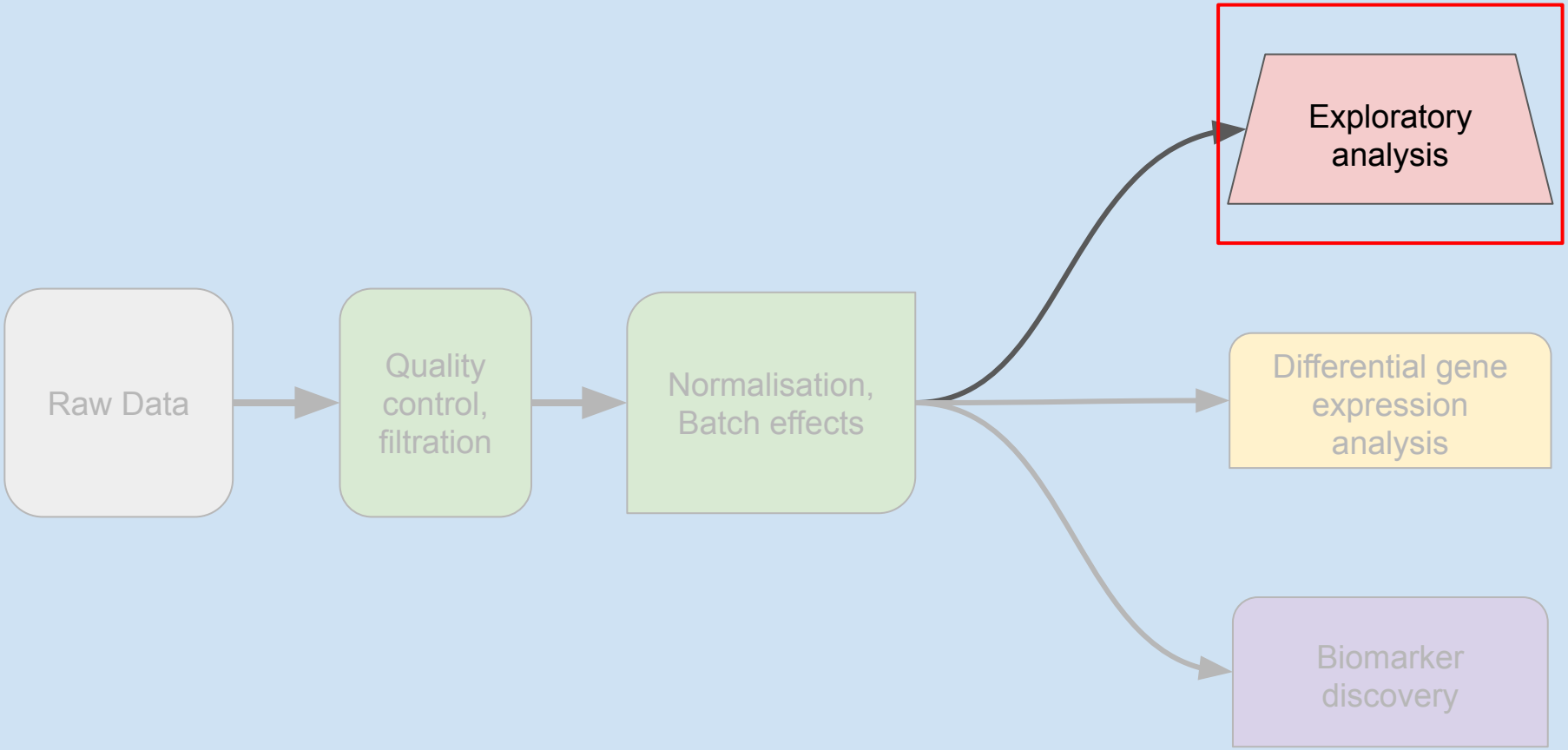
Now genes 1 and 2 appear the same while gene 3 still appears differentially abundant

Summary on normalisation

- Count data combines biological effects with technical effects
- Normalisation is a first step to isolate the biological effects
- **CPM** normalises based on **library size**
    - However, when there is a true effect present, this normalisation can bias the unaffected genes
- **TMM-normalisation** seeks to fix this by **excluding genes with extreme** relative abundances

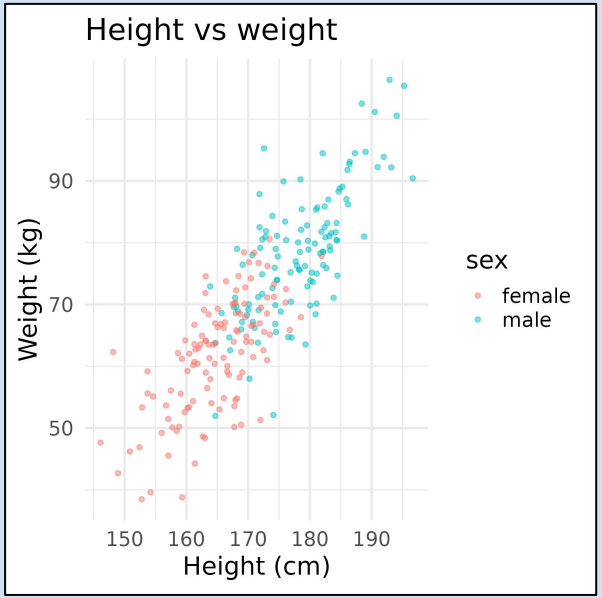
Robinson MD, Oshlack A. A scaling normalization method for differential expression analysis of RNA-seq data. Genome Biol. 2010;11(3):R25. doi: 10.1186/gb-2010-11-3-r25.

Exploratory Analysis :
Principal Component Analysis


Raw Data
Quality control, filtration
Normalisation, Batch effects
Exploratory analysis
Differential gene expression analysis
Biomarker discovery

Variability is information

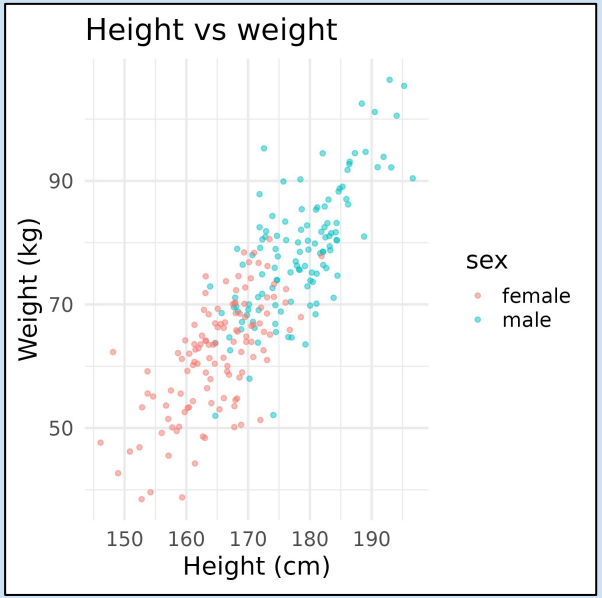
- **Statistics is all about exploring this variability!**



A lot of the variation in height and weight is *explained* by sex

Variability is information

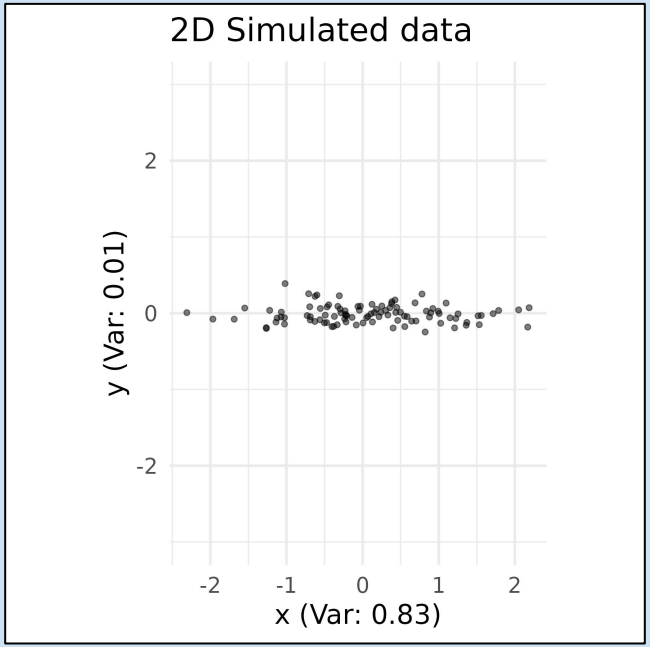
- **Statistics is all about exploring this variability!**



Visual inspection is limited to <3 dimensions

- We cannot explore 1000s of variables in this way
- **PCA** gives an **efficient** way to **represent** the variability in the data in only a **few dimensions**!

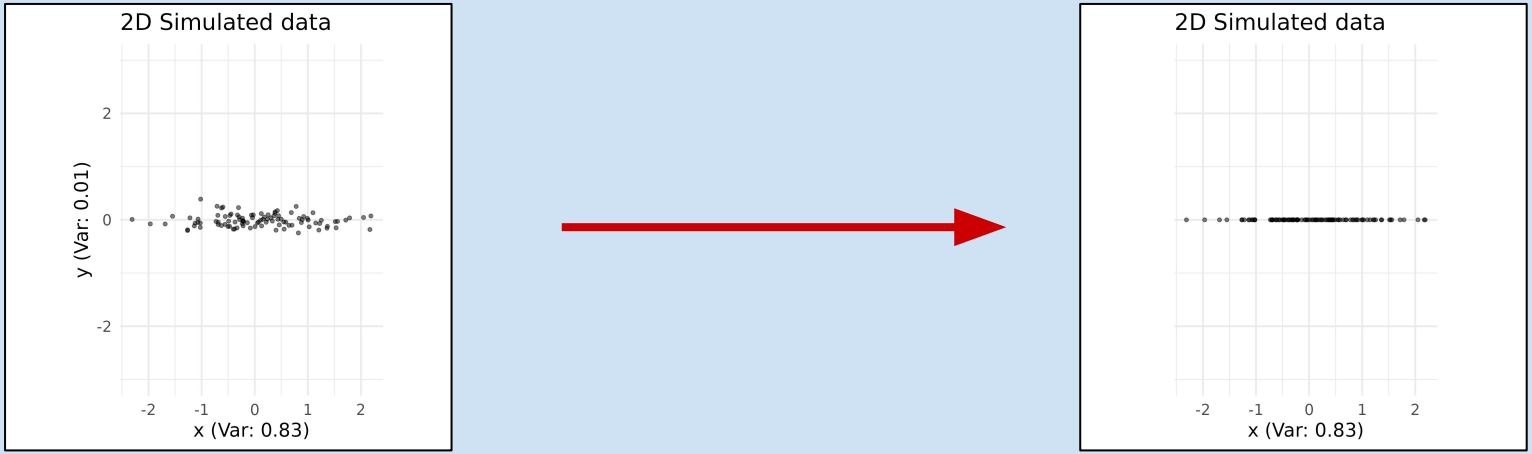
An illustrative example



Could we represent this data in a lower dimension without losing much information?

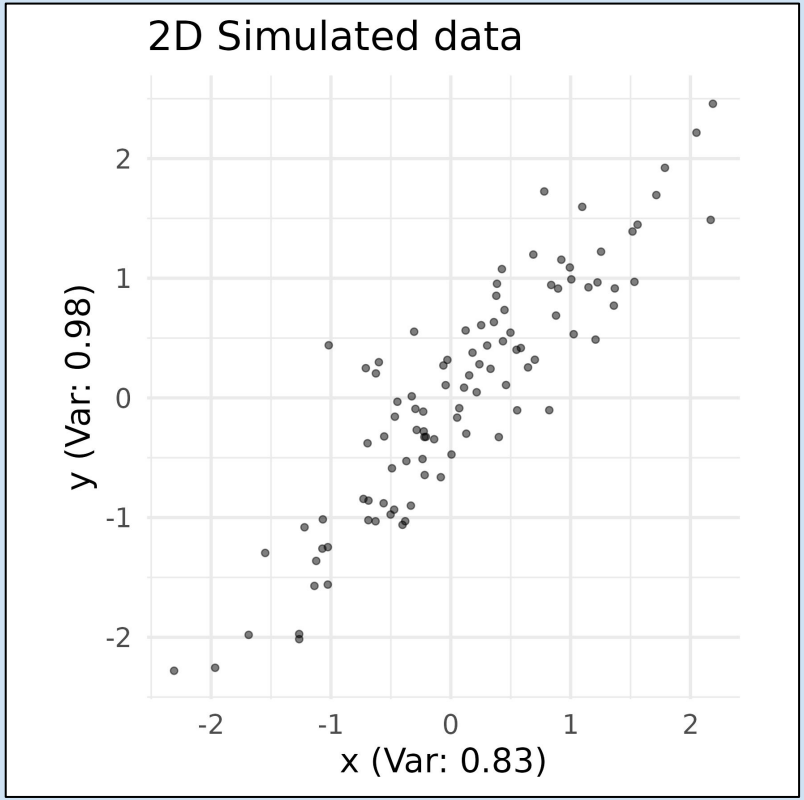
An illustrative example

- **Variability is information!**
- **Most of the information** is contained in the **x-coordinates**
  - The proportion of variability explained by x is 0.83/(0.83+0.01) = 99%
- Could we discard the y coordinates and represent the data in 1D?



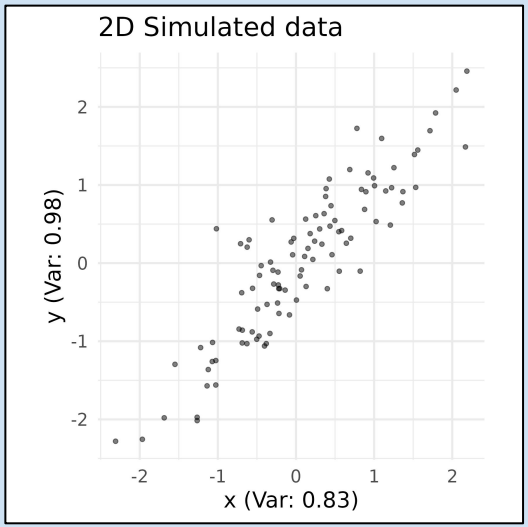
A new example in 2D

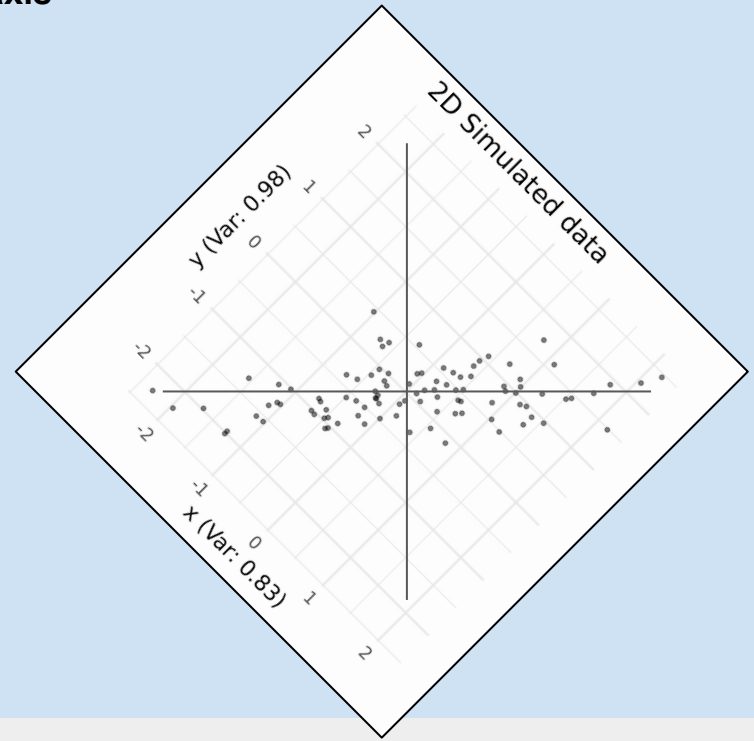
- This time, data varies across both x and y
- However, x and y are correlated - they contribute similar information
- **Could we represent this 2D data in 1D** without losing much information?



Intuition of PCA

- If we rotate our axes, we have a similar situation to the first example…
- **Most of the variability is now captured by the horizontal axis**
- This is what PCA does!



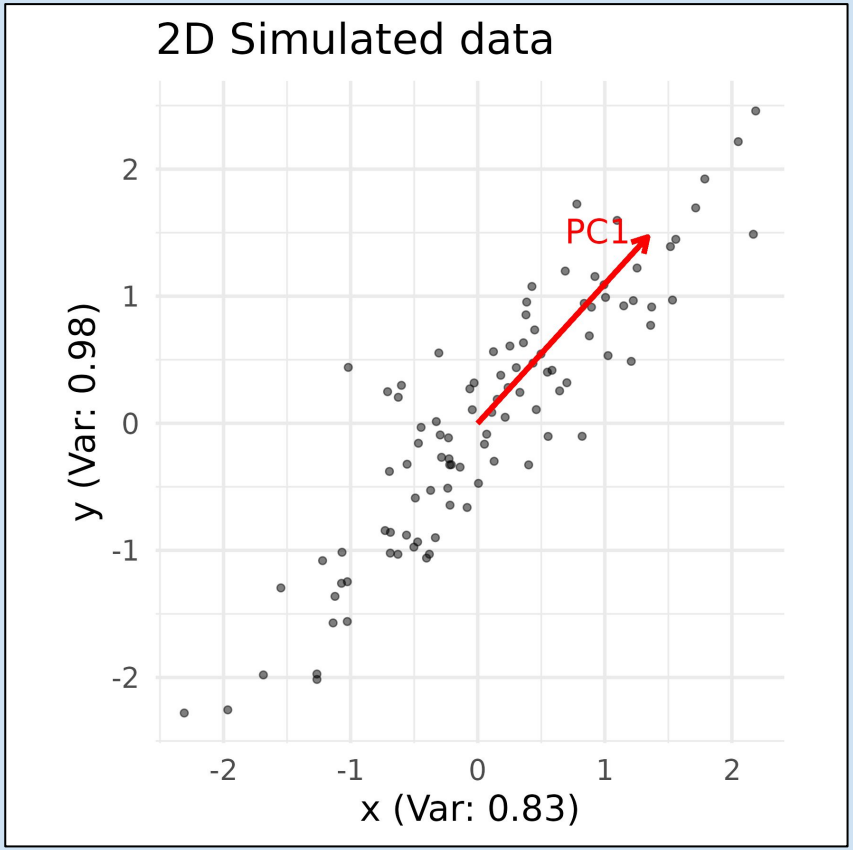


Intuition

PCA finds the directions of greatest variability

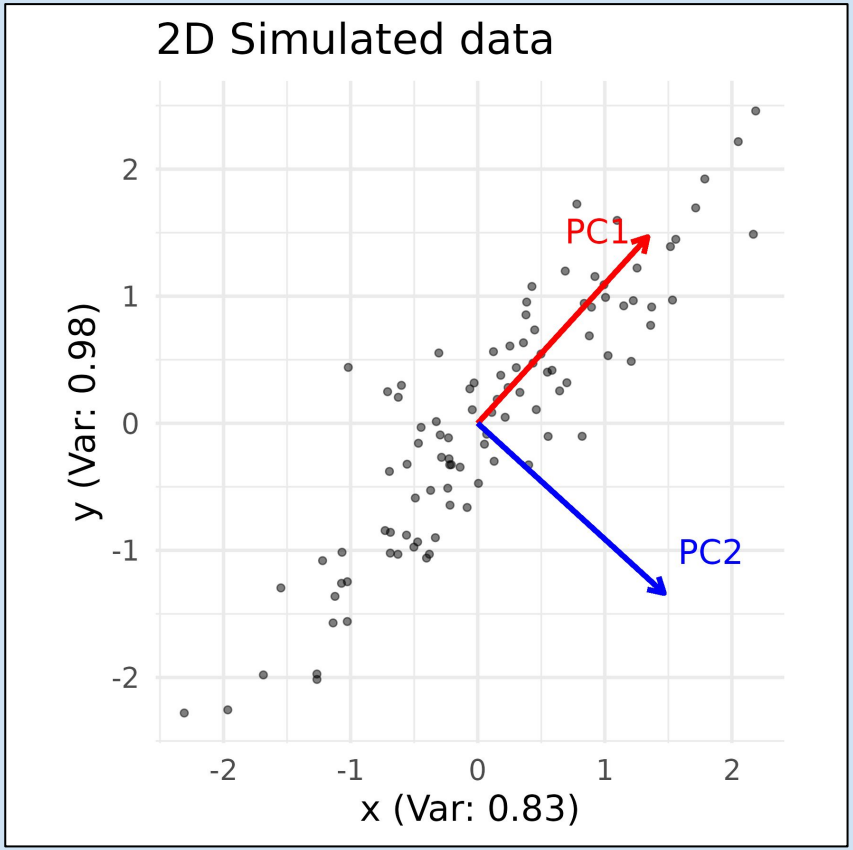
Principal components = these directions

The first principal component (PC1) = direction of most variability



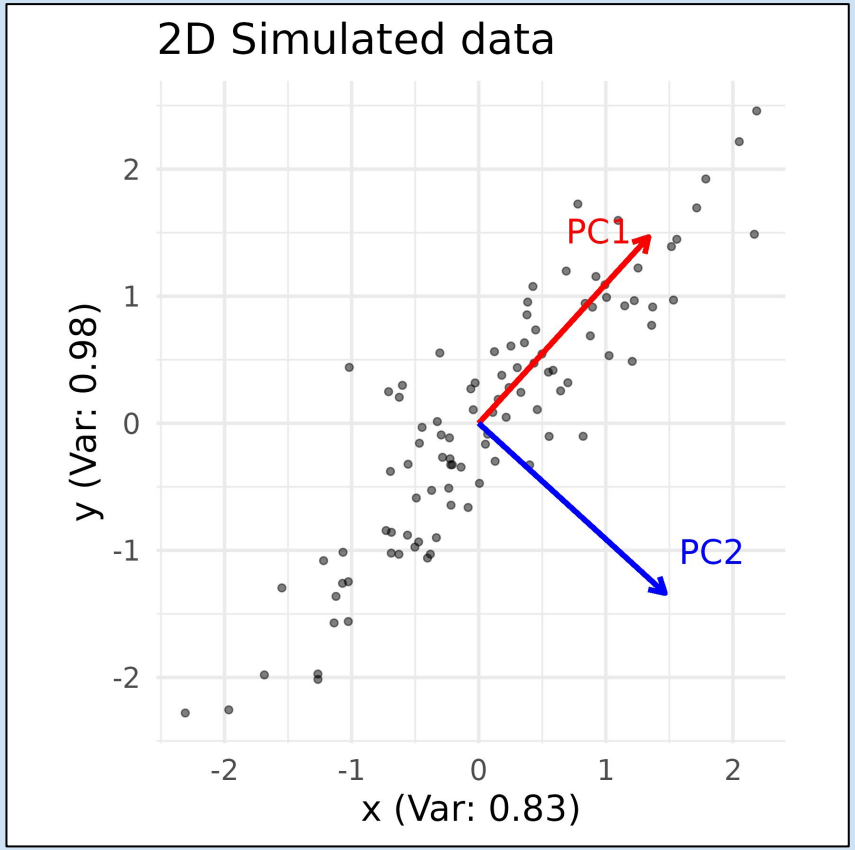
Intuition

- The second principal component (PC2) is **orthogonal to PC1** and is the direction of the **second most variability**
- These 2 redefined axes are equivalent to “rotating the data” as we showed



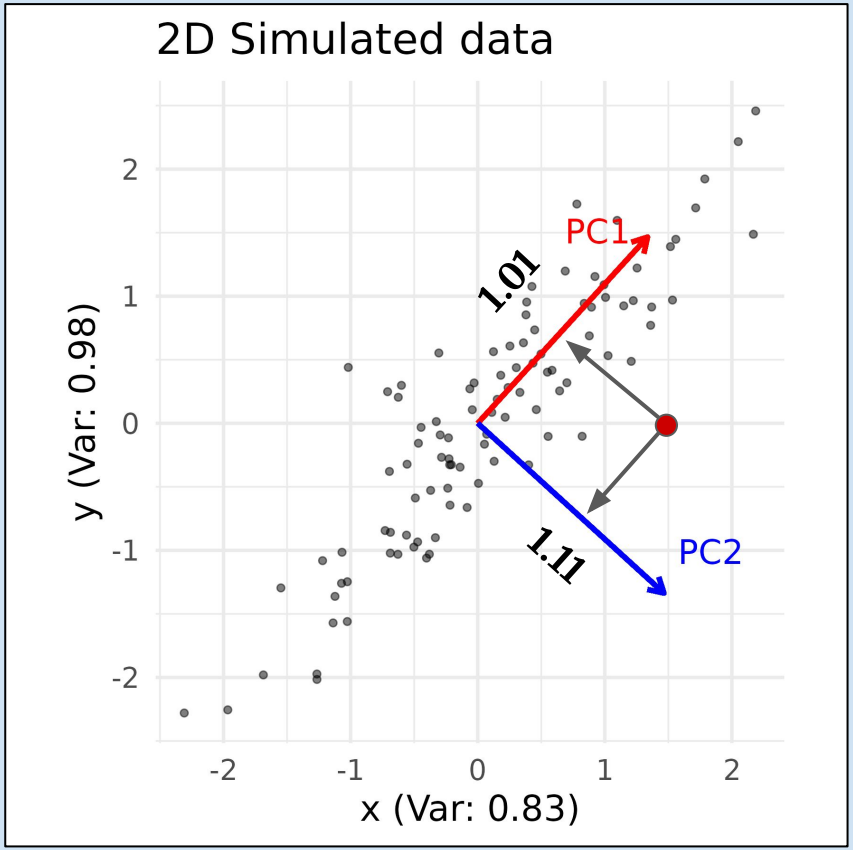
How were the PCs formed?

- **Weighted combinations** of the x and y coordinates
- In this example :
  - PC1 = 0.67x + 0.74y
  - PC2 = 0.74x - 0.67y



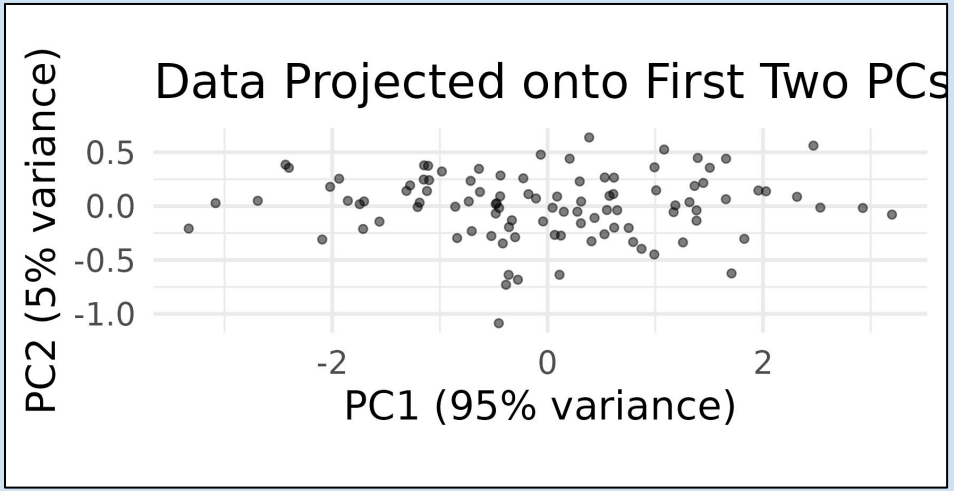
How to ***project*** the data onto the PCs?

- PC1 = 0.67x + 0.74y
- PC2 = 0.74x - 0.67y
- Imagine a point with **(x,y) = (1.5,0)**
- Its coordinates on the 2 principal axes are **(PC1, PC2) = (1.01, 1.11)**



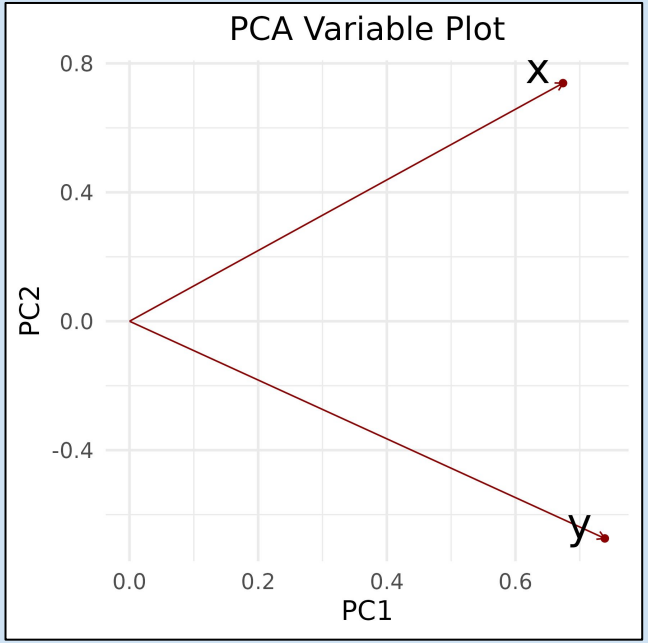
Samples projected onto PCs

- We can represent our data on our new “principal axes”
  - i.e. rotating our original data to new axes
  - **Important :** the data stays 2-dimensional
- **Almost all of the variability** in the data is explained by the points’ value on **PC1**
  - Maybe we could discard PC2 and just represent the data on PC1?



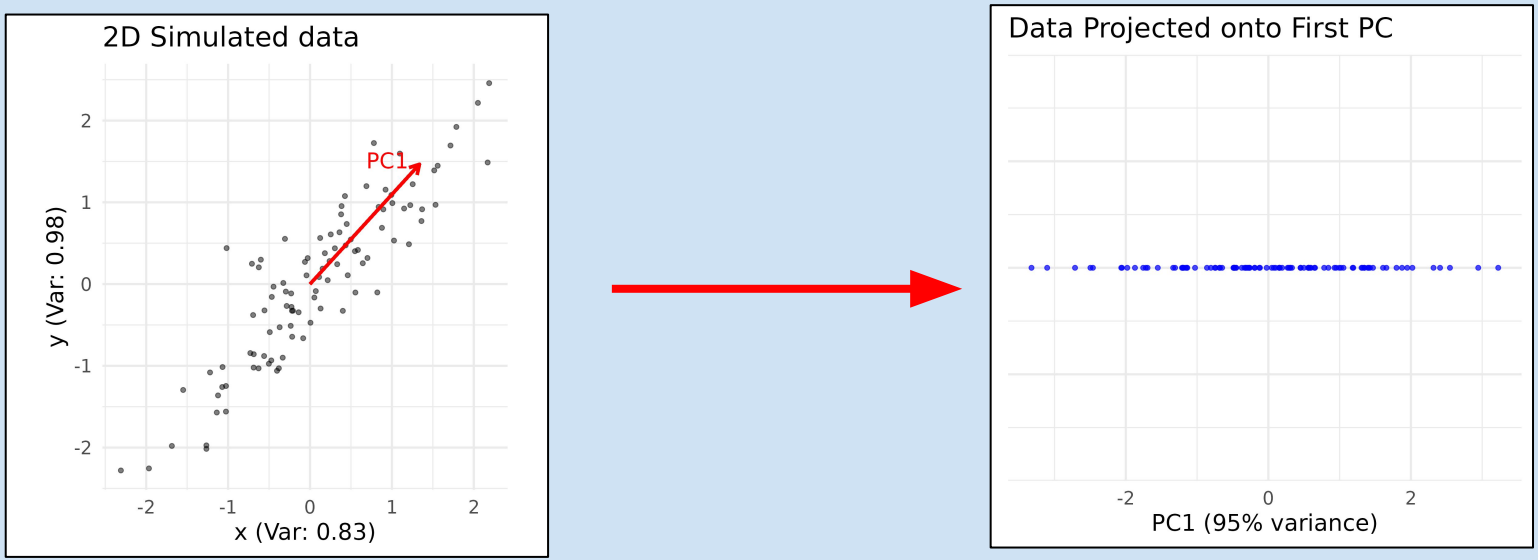
Variables contribution to PCs

- Common to explore how variables contribute to the PCs
- This shows us the **contributions** (weights) of each variable to each PC, e.g. here
  - PC1 = 0.67x + 0.74y
  - PC2 = 0.74x - 0.67y
- **Length : strength of contribution** to PC
- **Direction :** how the variable relates to each PC
  - Variables in same direction are positively correlated
  - Variables in opposite direction negatively correlated
  - Orthogonal variables uncorrelated

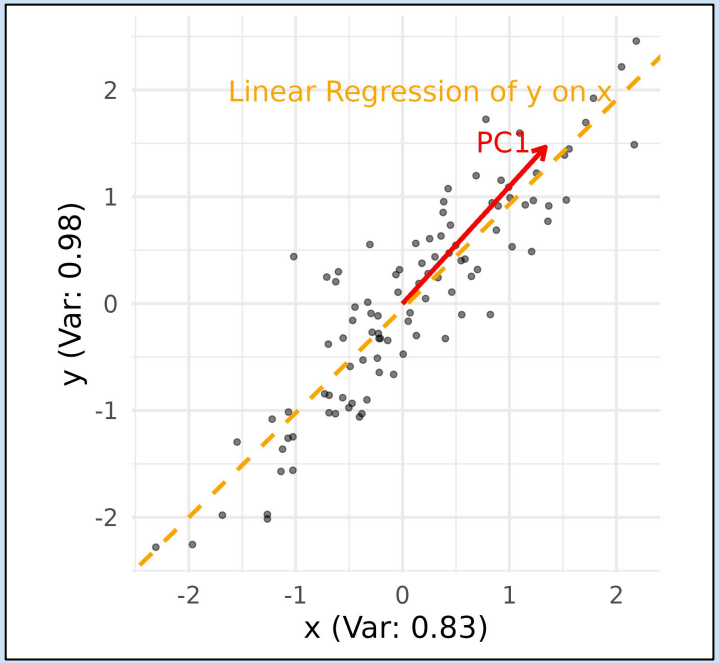


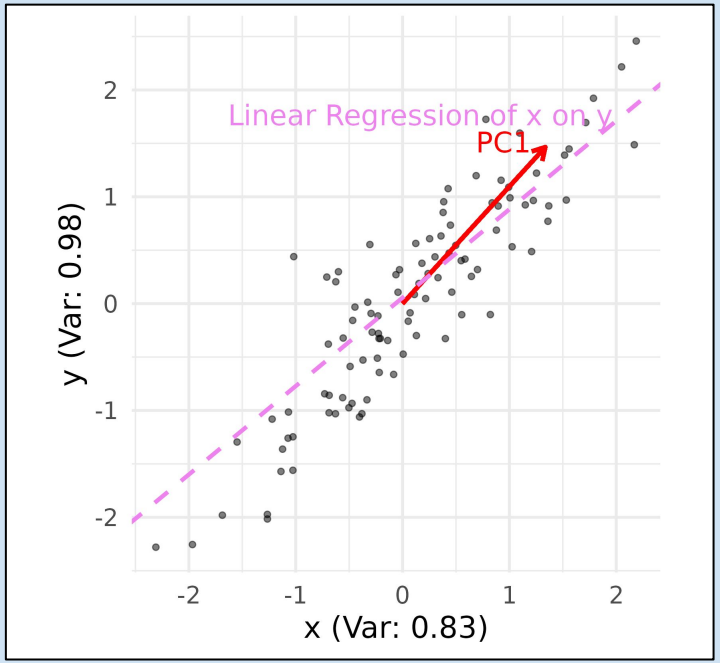
Can we reduce the dimension?

- We can discard PC2 and only represent the data points on PC1, keeping 95% of the variability in the data explained
- We have now ***reduced the dimension*** of our data from 2 to 1



Note : PCs are NOT linear regression!





What happens in higher dimensions?

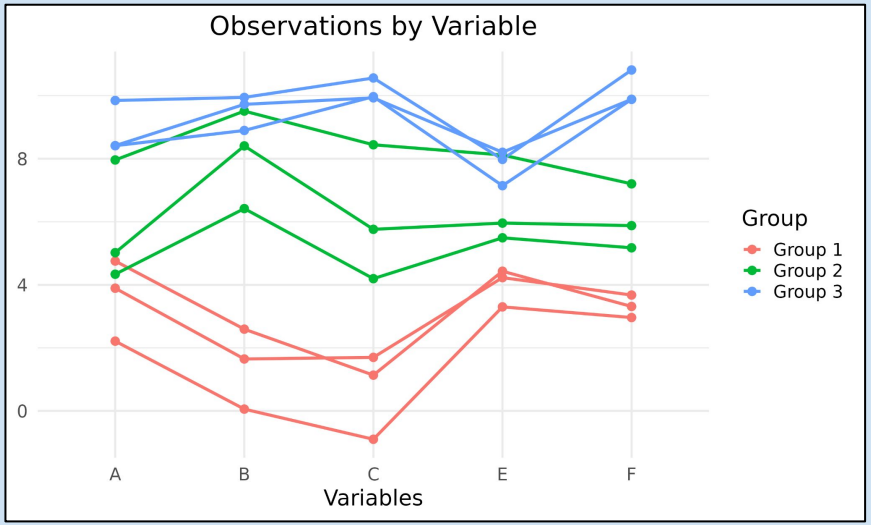
If we start with p-variables, PCA will compute p-components as follows

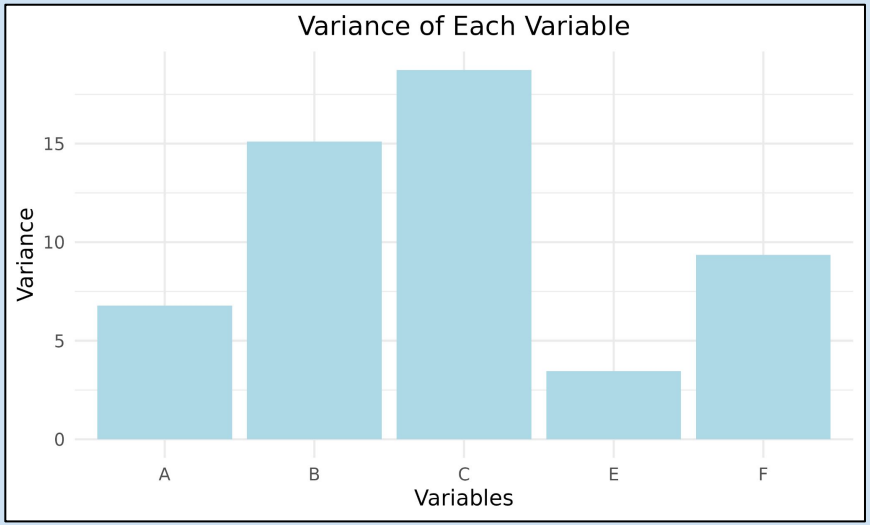
1. PC1 : direction of most variability
2. PC2 : **orthogonal** (i.e. at a right angle) to PC1, direction of most variability
3. PC3 : **orthogonal** to PC1 AND PC2, direction of most variability
4. …
5. PC(p) : **orthogonal** to PC1,...,PC(p-1), direction of most variability

So, **PCA does not directly reduce the dimension**, it just concentrates the variability in the first few variables (PCs)

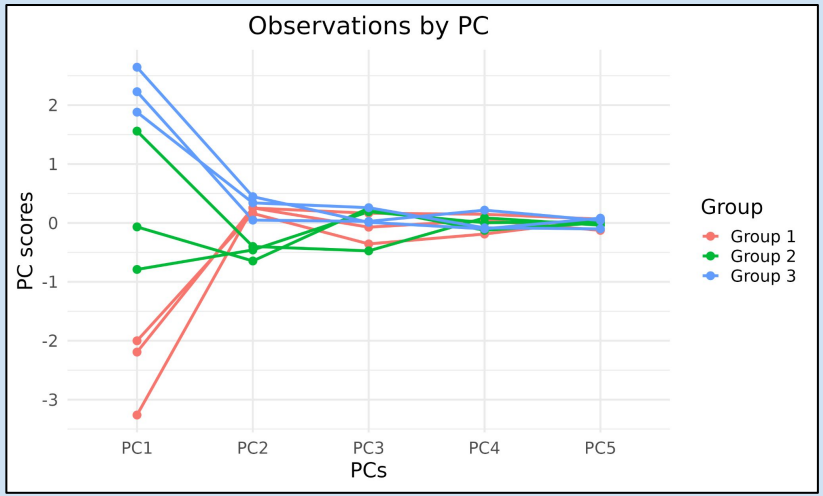
An example in higher-dimension

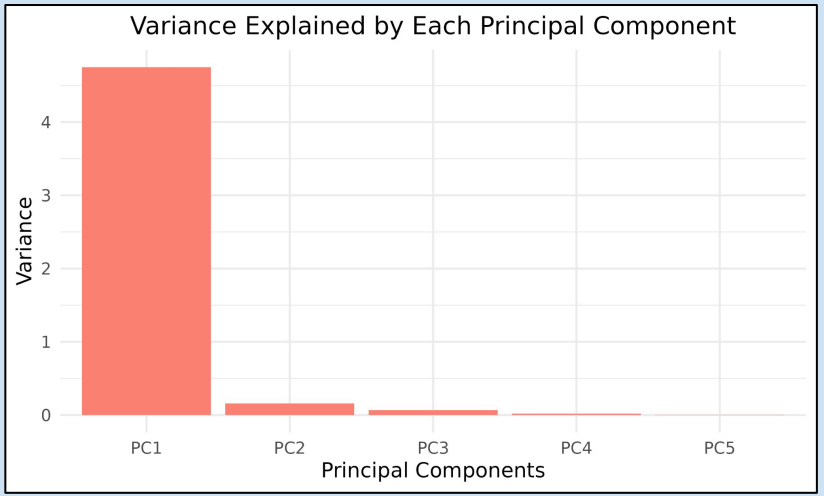
- 9 observations of 5 variables
- 3 categorical groups which explain a lot of the variance





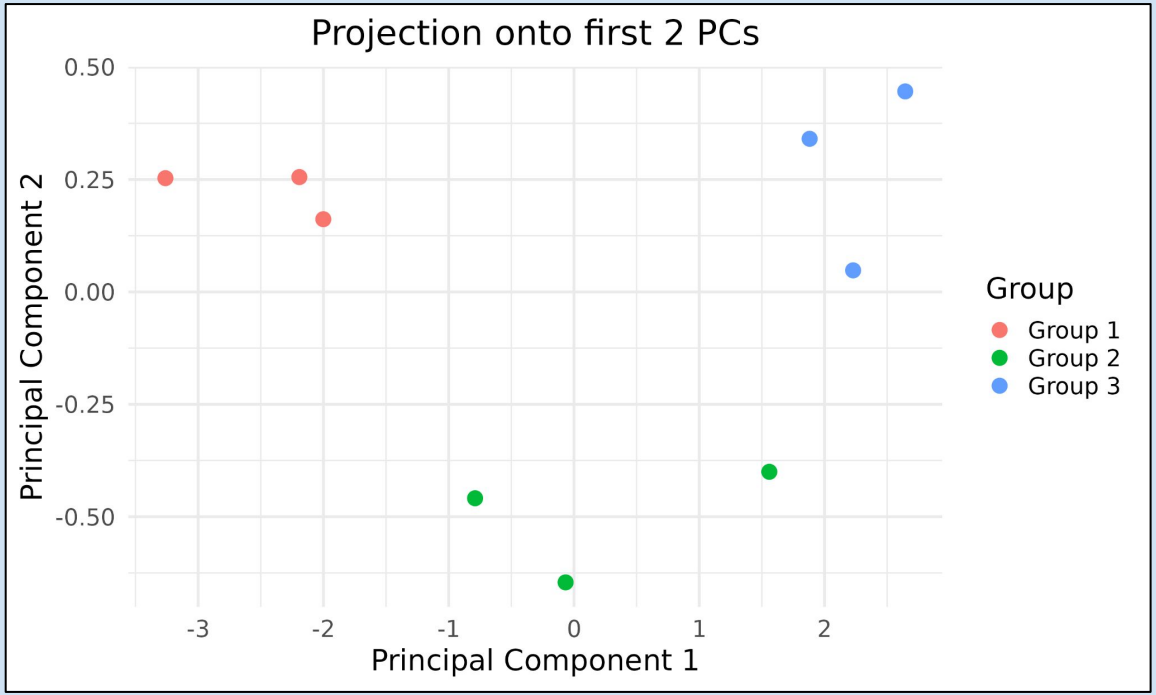
Observations after PCA





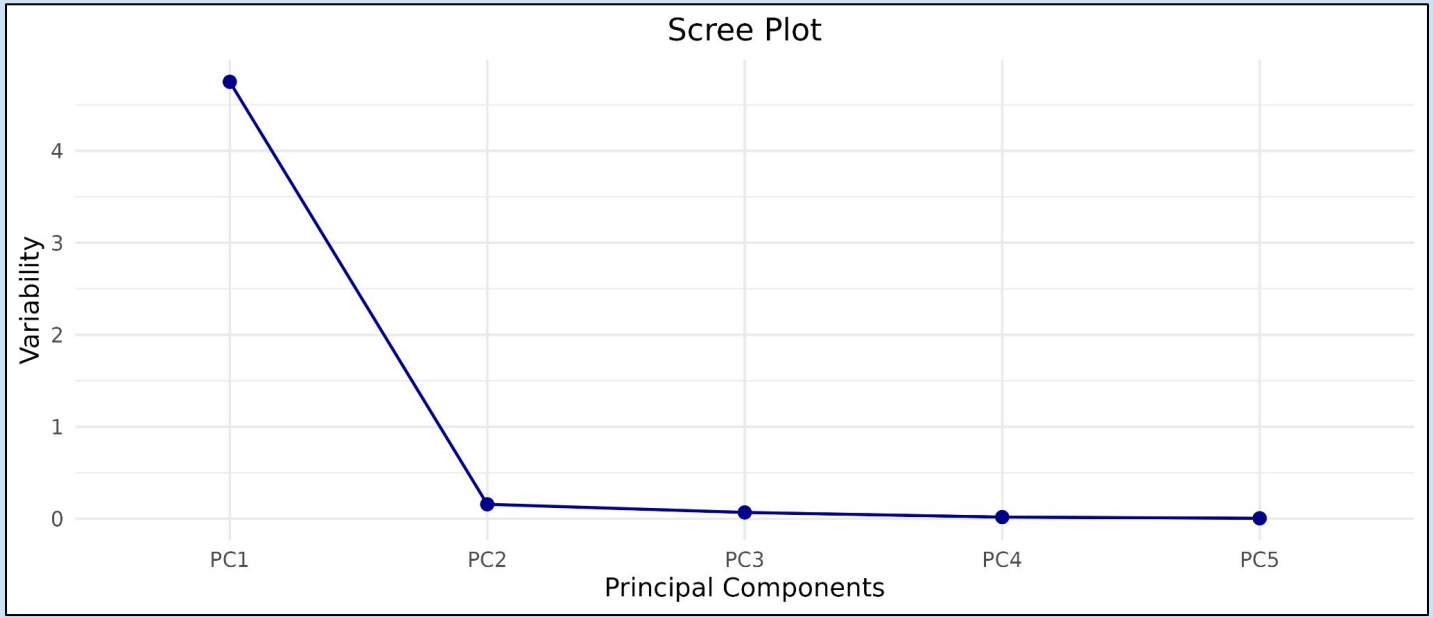
Projection of observations on the first 2 PCs

- PCA can help identify clusters
- The groups ***cluster together*** when plotted on the first 2 PCs



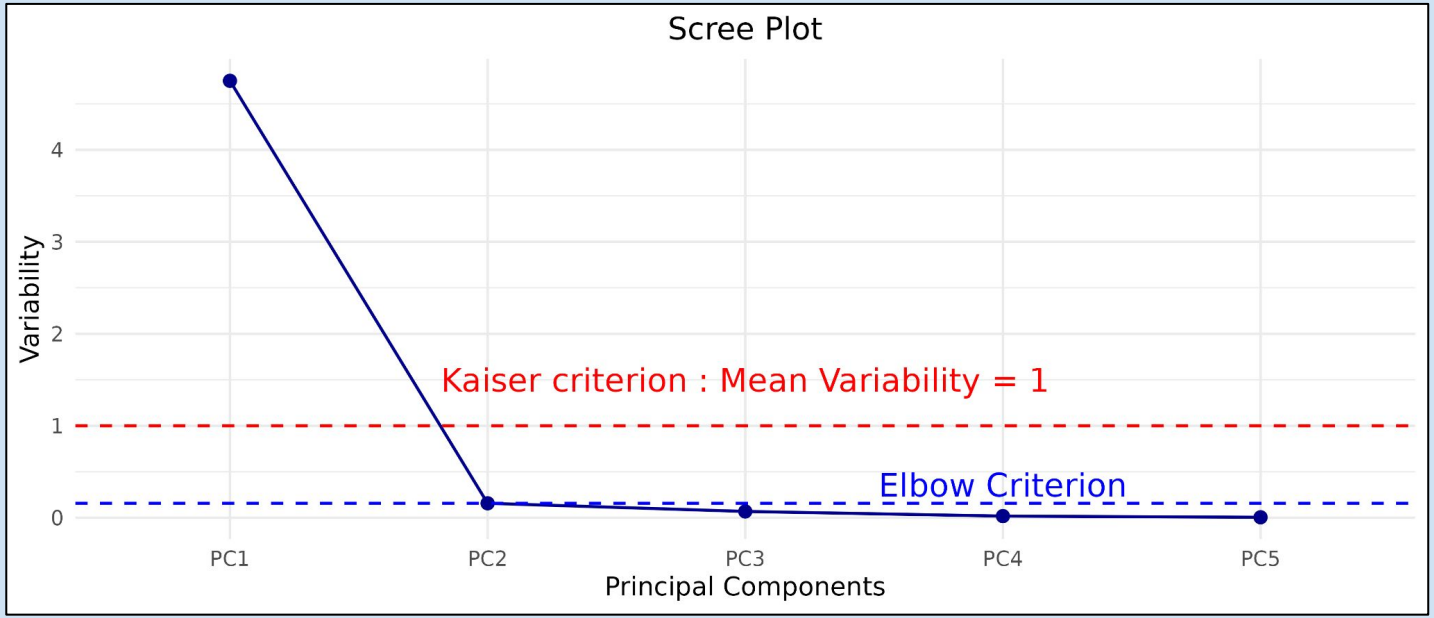
How to choose how many PCs to keep?

- Plot variability of ordered PCs



How do we choose how many PCs to keep?

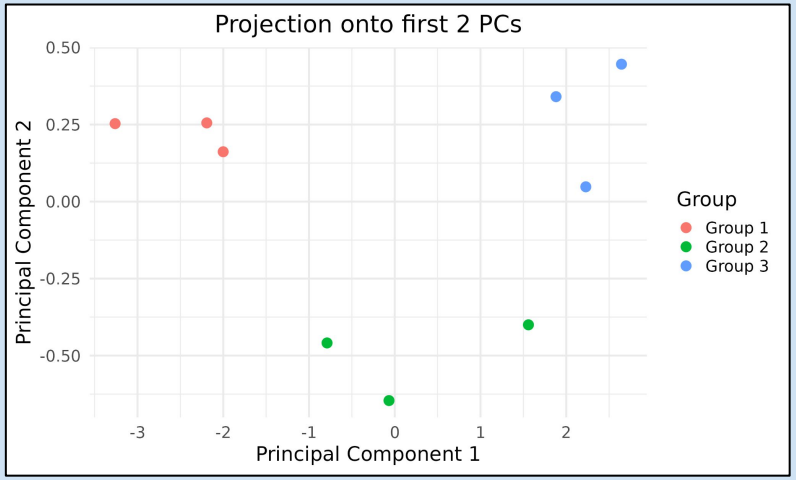
- **Kaiser criterion :** Keep PCs accounting for more variability than average
- **Elbow/ Catell criterion :** Find a visual “elbow” in the scree plot



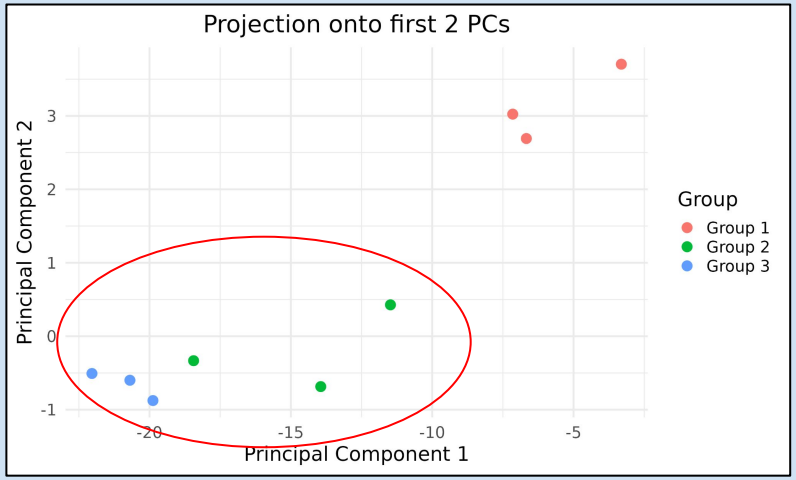
PCA is NOT scale independent!

- **Scaled data :** all variables have variance 1
- **Centered data :** all variables have mean 0
- Always center and scale your data before PCA

Centered and scaled data

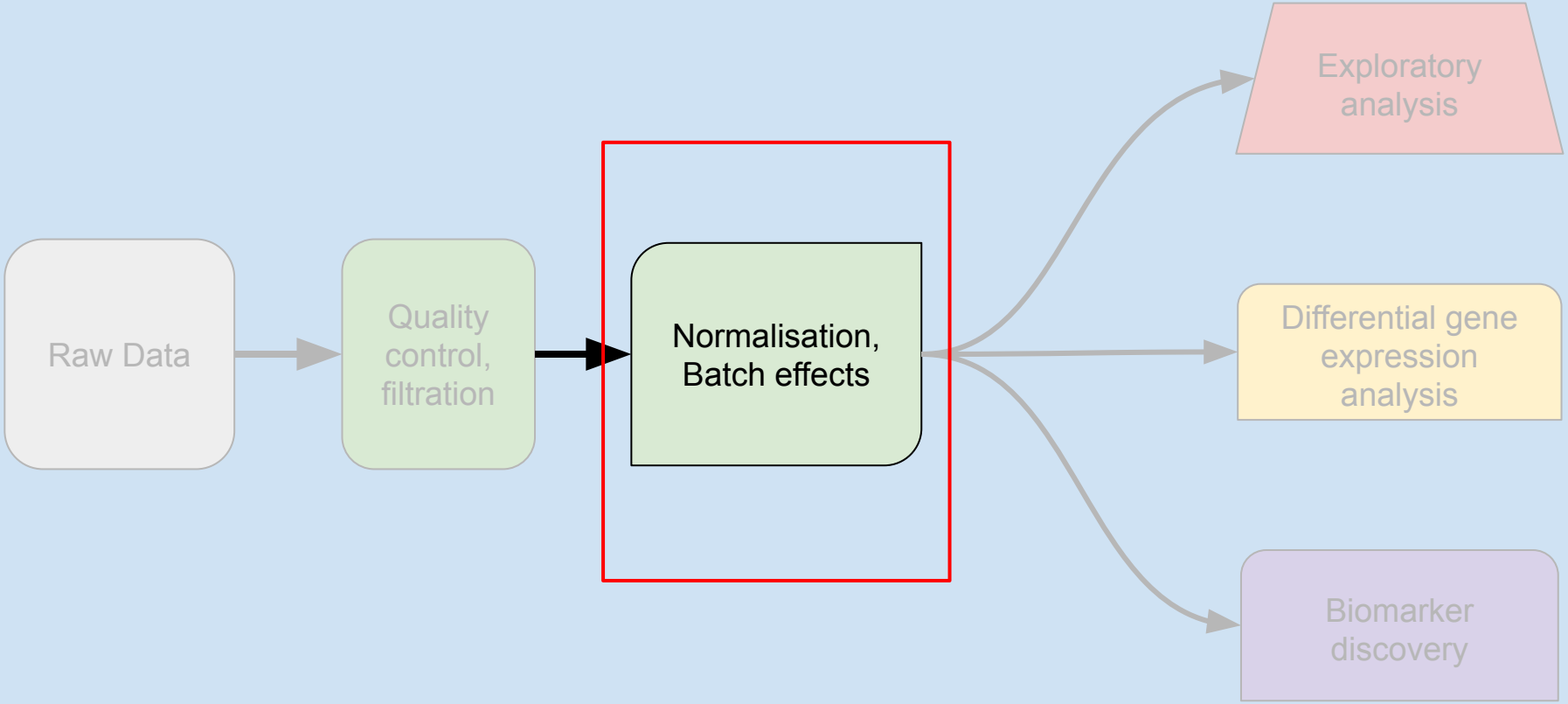


Raw data



Pre-processing of RNA-seq Data

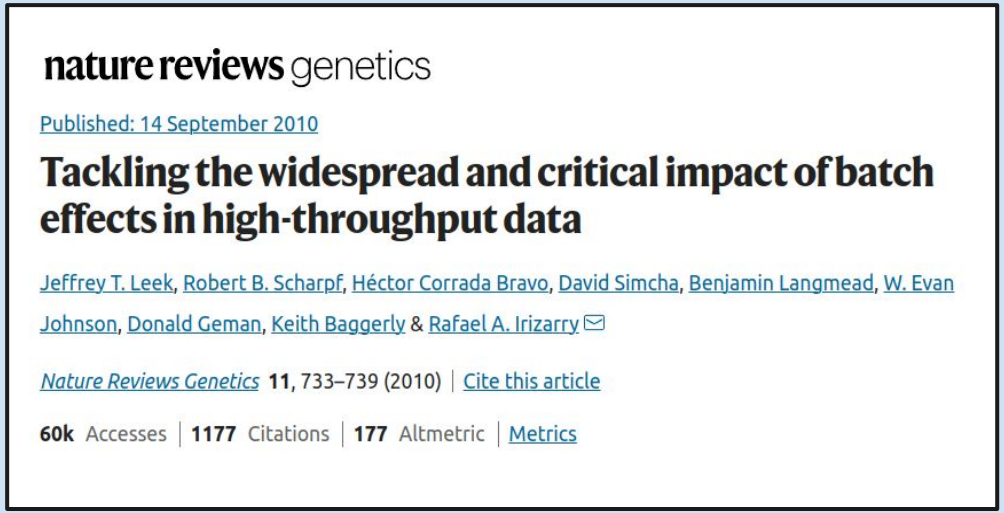
Batch effects


Raw Data
Quality control, filtration
Normalisation, Batch effects
Exploratory analysis
Differential gene expression analysis
Biomarker discovery

Batch Effects

Effects in the data from **non-biological sources**

- Experiment date
- Instrument used
- Atmospheric conditions
- Technician
- Laboratory



nature reviews genetics

Published: 14 September 2010

Tackling the widespread and critical impact of batch effects in high-throughput data

Jeffrey T. Leek, Robert B. Scharpf, Héctor Corrada Bravo, David Simcha, Benjamin Langmead, W. Evan Johnson, Donald Geman, Keith Baggerly & Rafael A. Irizarry

Nature Reviews Genetics **11**, 733–739 (2010) | Cite this article

60k Accesses | **1177** Citations | **177** Altmetric | Metrics

Unaccounted batch effects can dramatically alter the validity of findings

How to detect and correct batch effects?

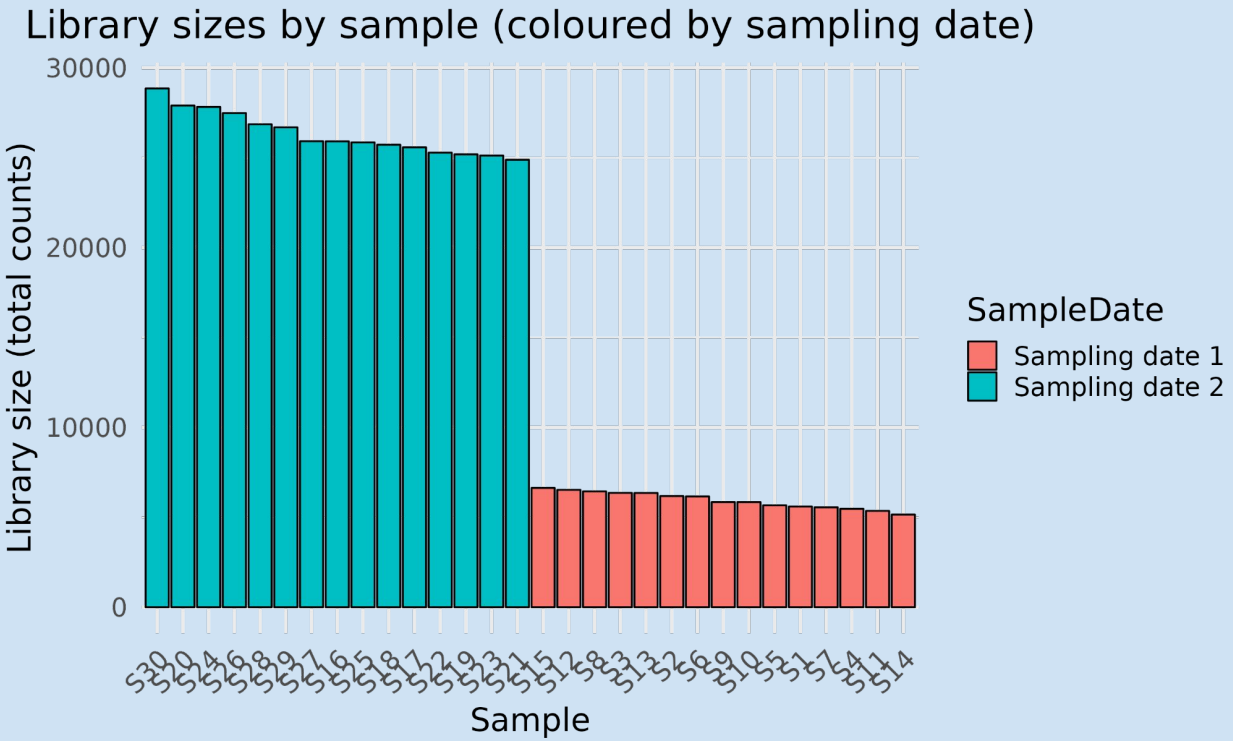
Detection

- Exploratory analysis!
  - Visualise data as a function of batch variables
  - Principal variance component analysis

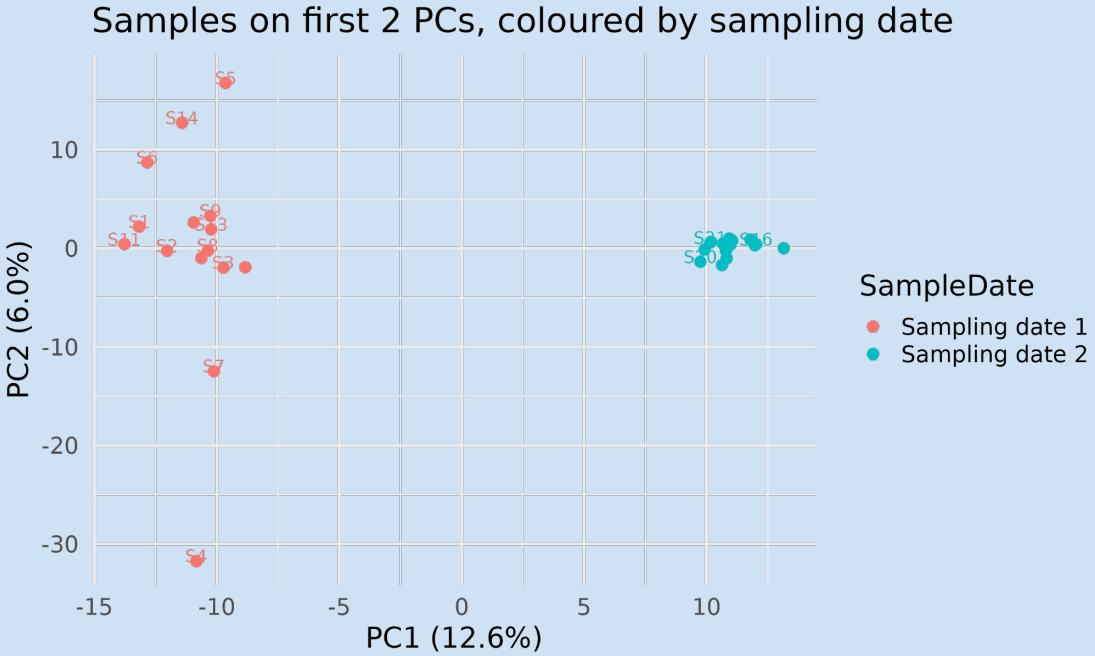
Correction

- Estimate and remove batch effect prior to analysis : e.g. **ComBat-Seq package** in R
  - No detail given here but we will practice this in the practical
- Adjust during analysis by considering batch variable as a covariate

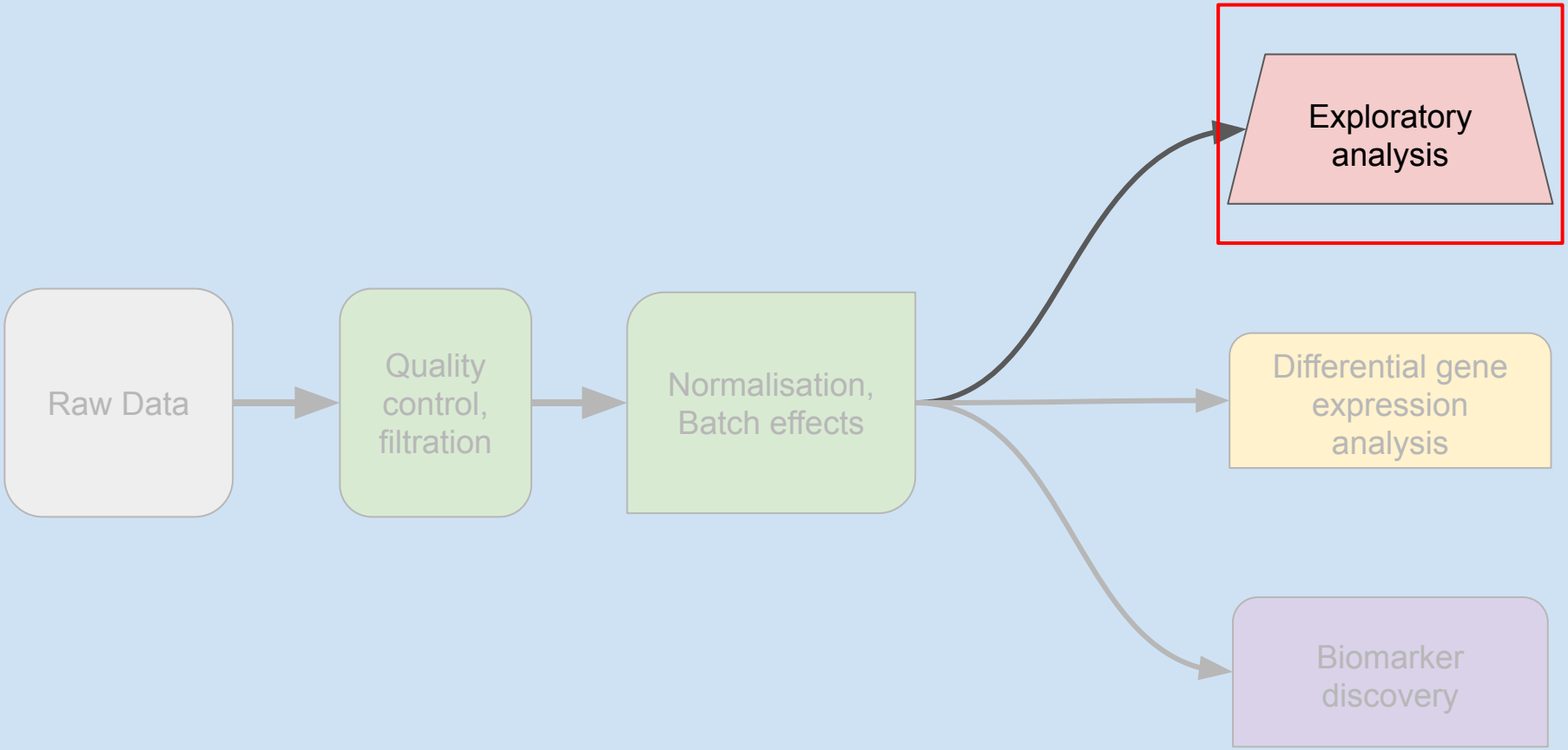
Example of a batch effect



Detecting batch effect with PCA



Exploratory Analysis :
Hierarchical Clustering

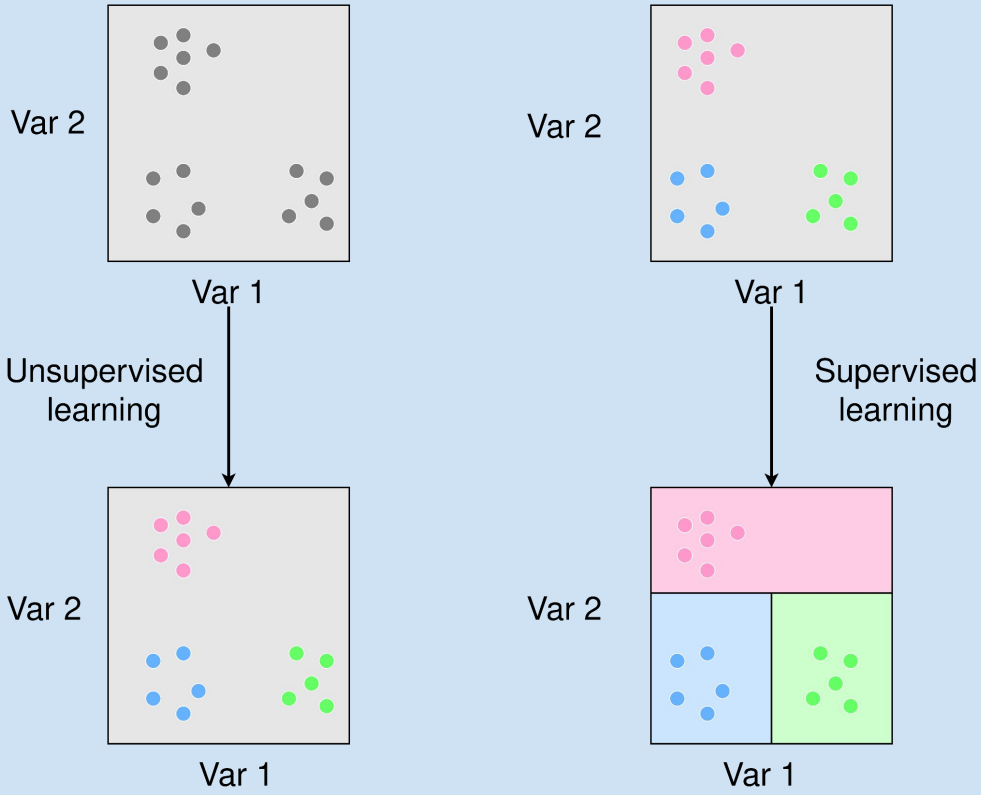

Raw Data
Quality control, filtration
Normalisation, Batch effects
Exploratory analysis
Differential gene expression analysis
Biomarker discovery

Unsupervised learning

- Exploring patterns in data
- No “correct” response
- PCA, clustering

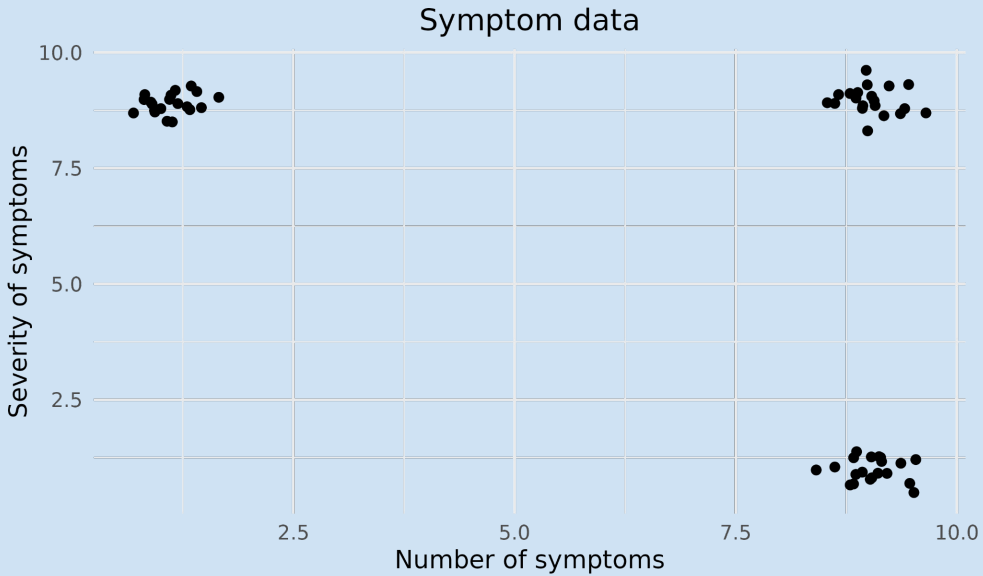
Supervised learning

- Learn the relationship between predictors and a response
- Linear regression, random forest,...



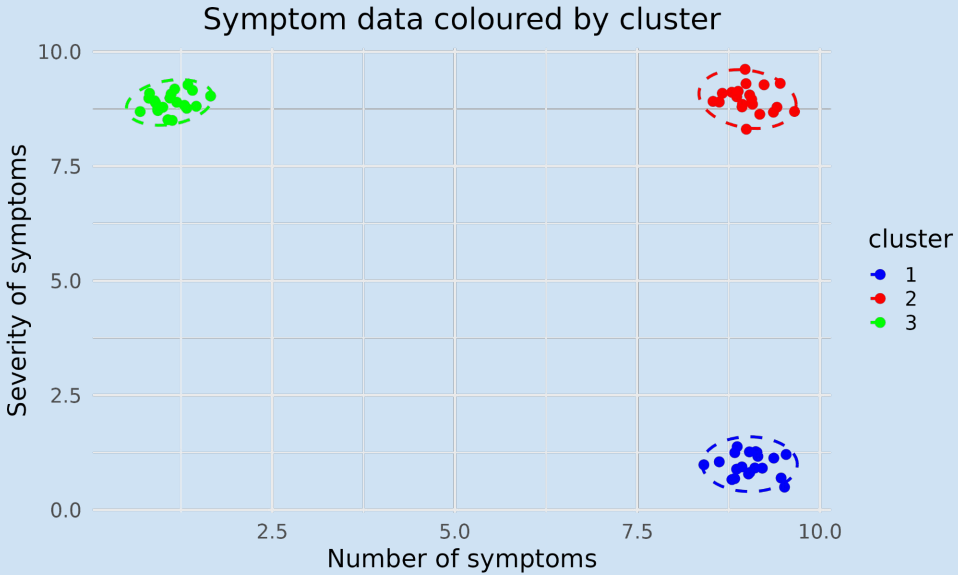
Motivation

***Unsupervised learning* to infer groups** (clusters) from data



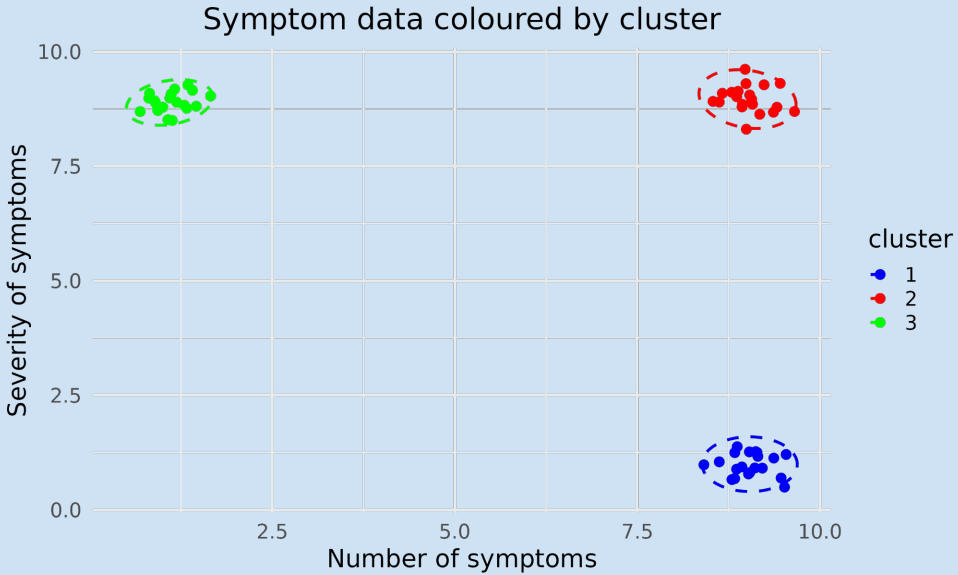
Motivation

***Unsupervised learning* to infer groups** (clusters) from data



Motivation

***Unsupervised learning* to infer groups** (clusters) from data



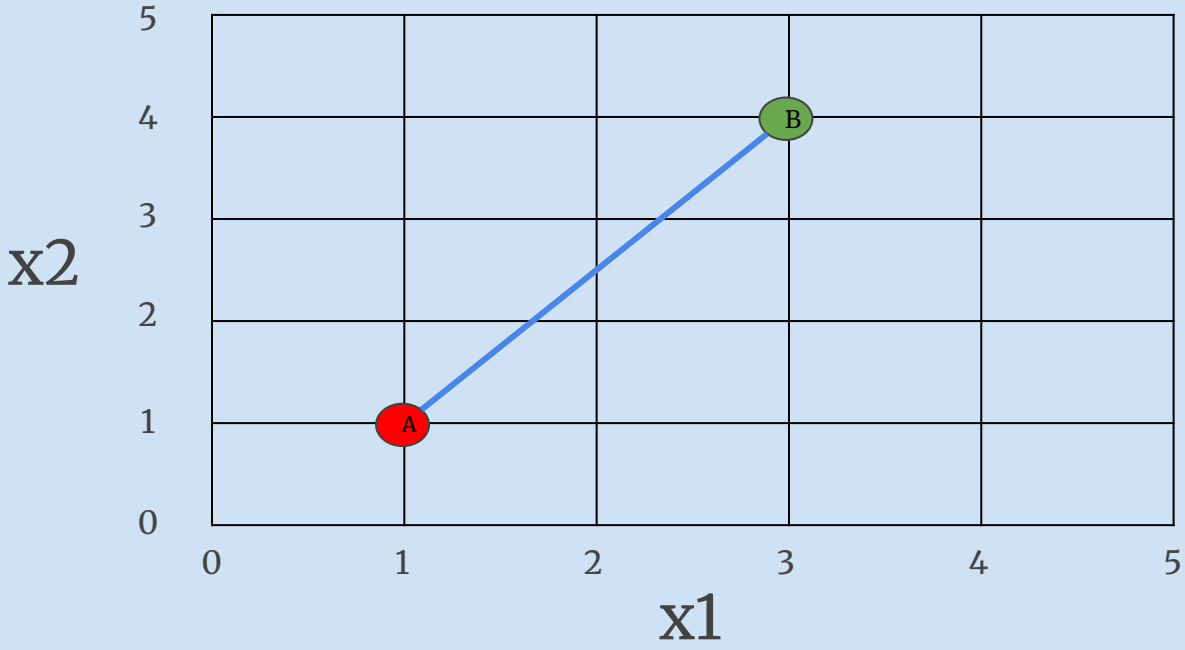
- Data within a cluster is similar
- Clusters are different from each other

What does *similar* mean?

Similarity between data is formalised with ***distance metrics***, e.g.

- Euclidean distance
- Manhattan distance
- Maximum distance
- and many more…

Euclidean distance in 2 dimensions

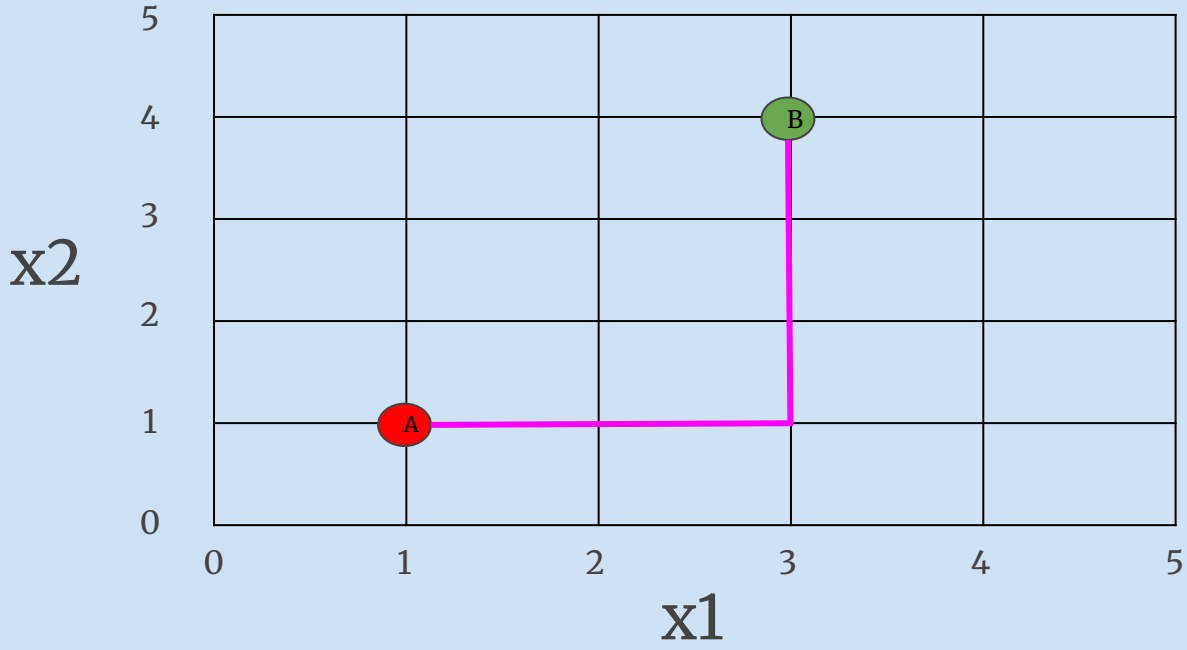


$$d_{euclidean} = \sqrt{\Delta x_1^2 + \Delta x_2^2}$$

Euclidean distance in *p* dimensions

$$d_{euclidean} = \sqrt{\Delta x_1^2 + \Delta x_2^2 + ... + \Delta x_p^2}$$

Manhattan distance



$$d_{mannhattan} = \Delta x_1 + \Delta x_2$$

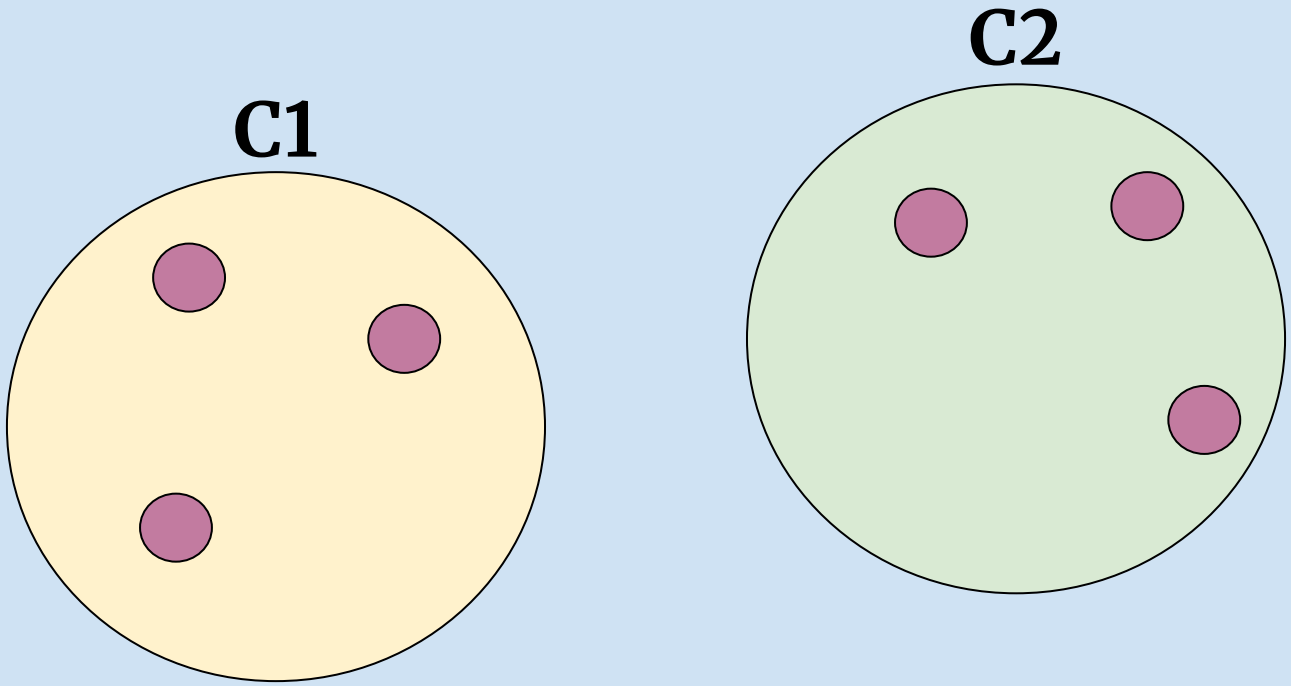
Manhattan distance in *p* dimensions

$$d_{mannhattan} = \Delta x_1 + \Delta x_2 + ... + \Delta x_p$$

Maximum distance

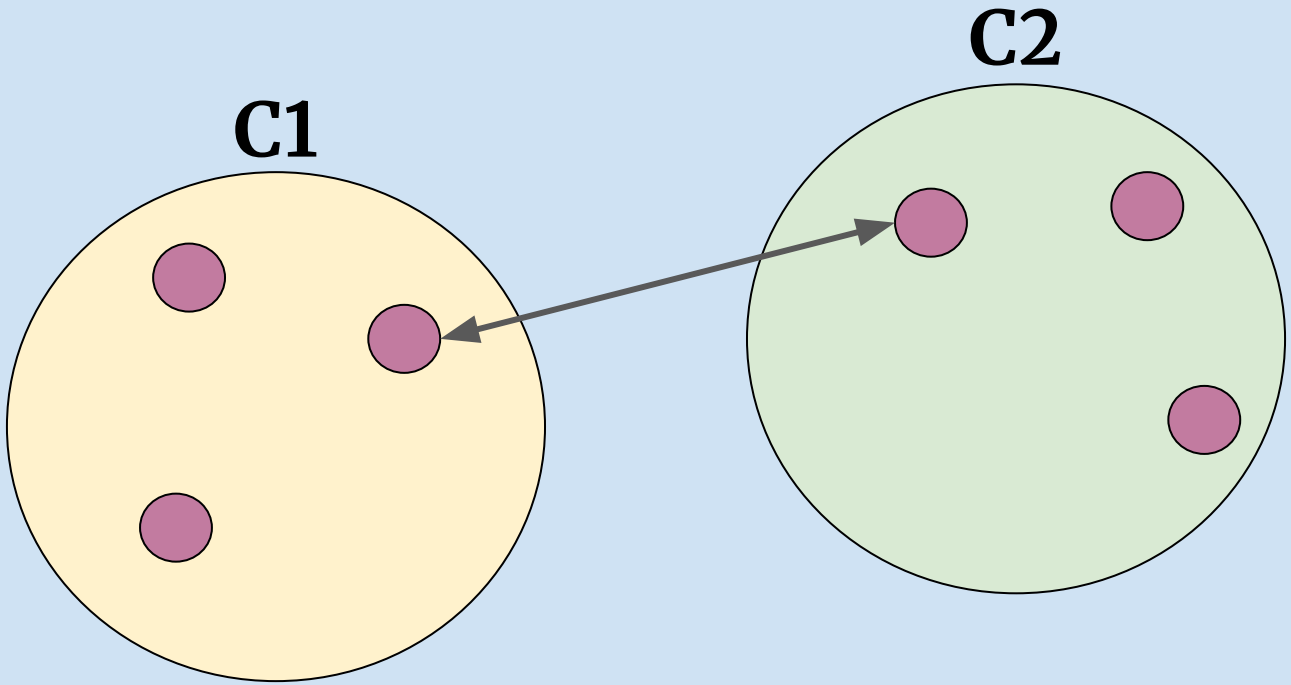
$$d_{max} = max(\Delta x_1, \Delta x_2, ..., \Delta x_p)$$

How can we define distance between clusters?



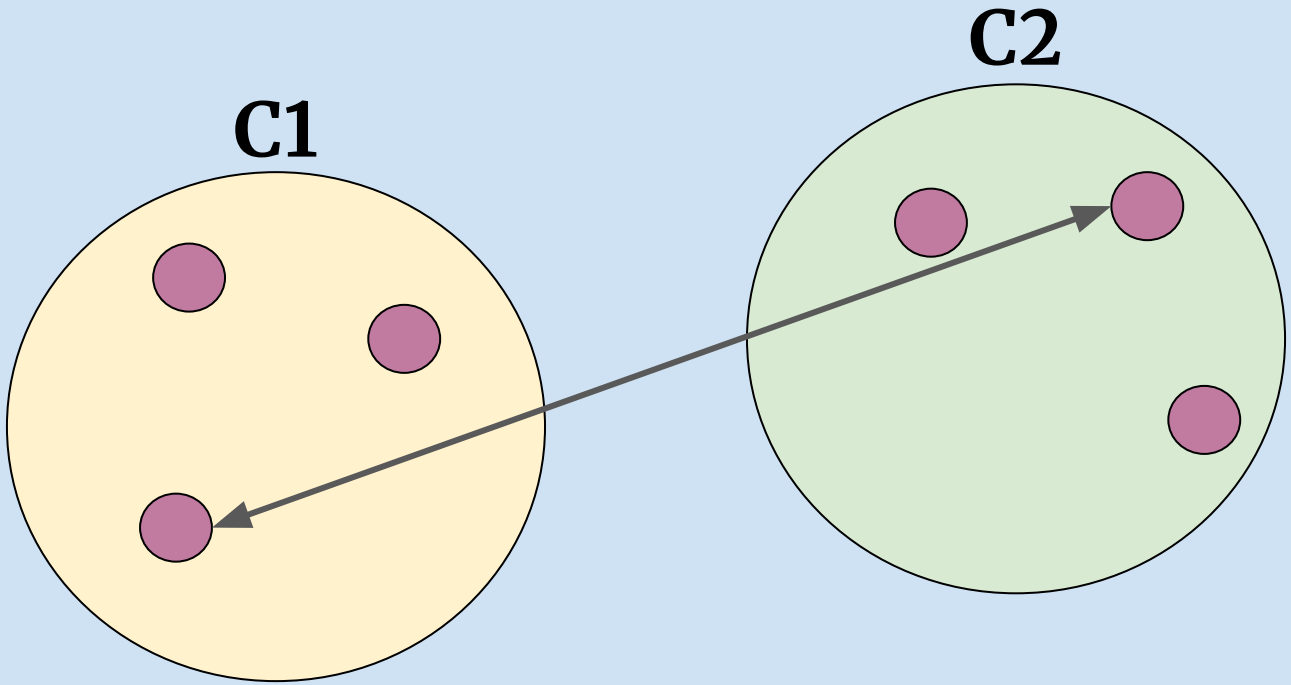
Single linkage

Distance between closest points



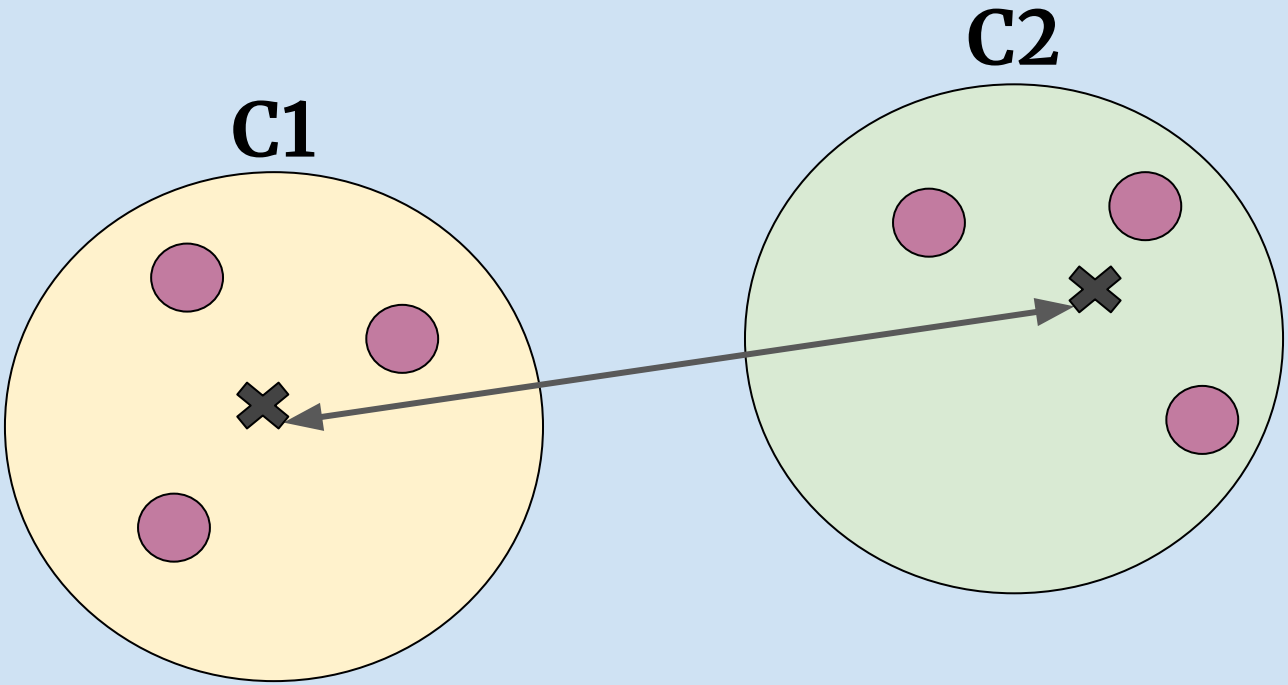
Complete linkage

Distance between farthest points



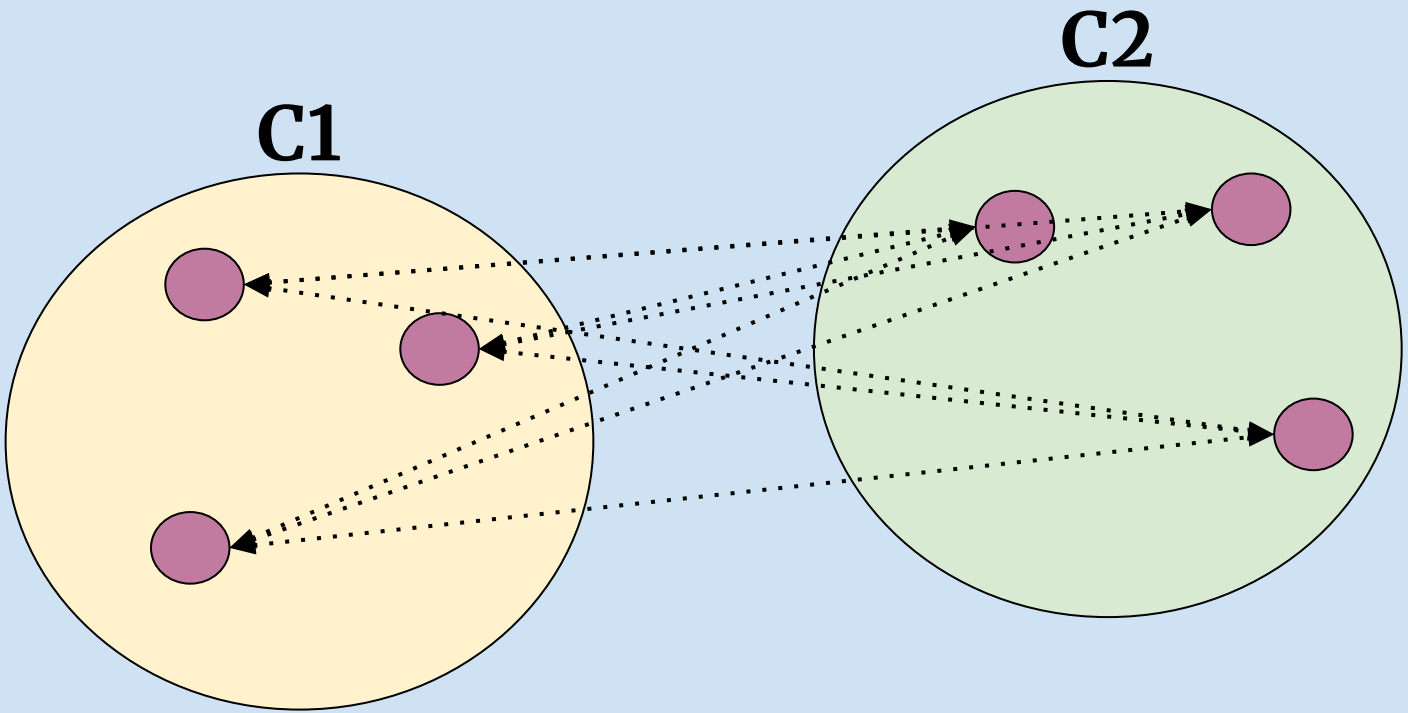
Centroid linkage

Distance between cluster “centers”



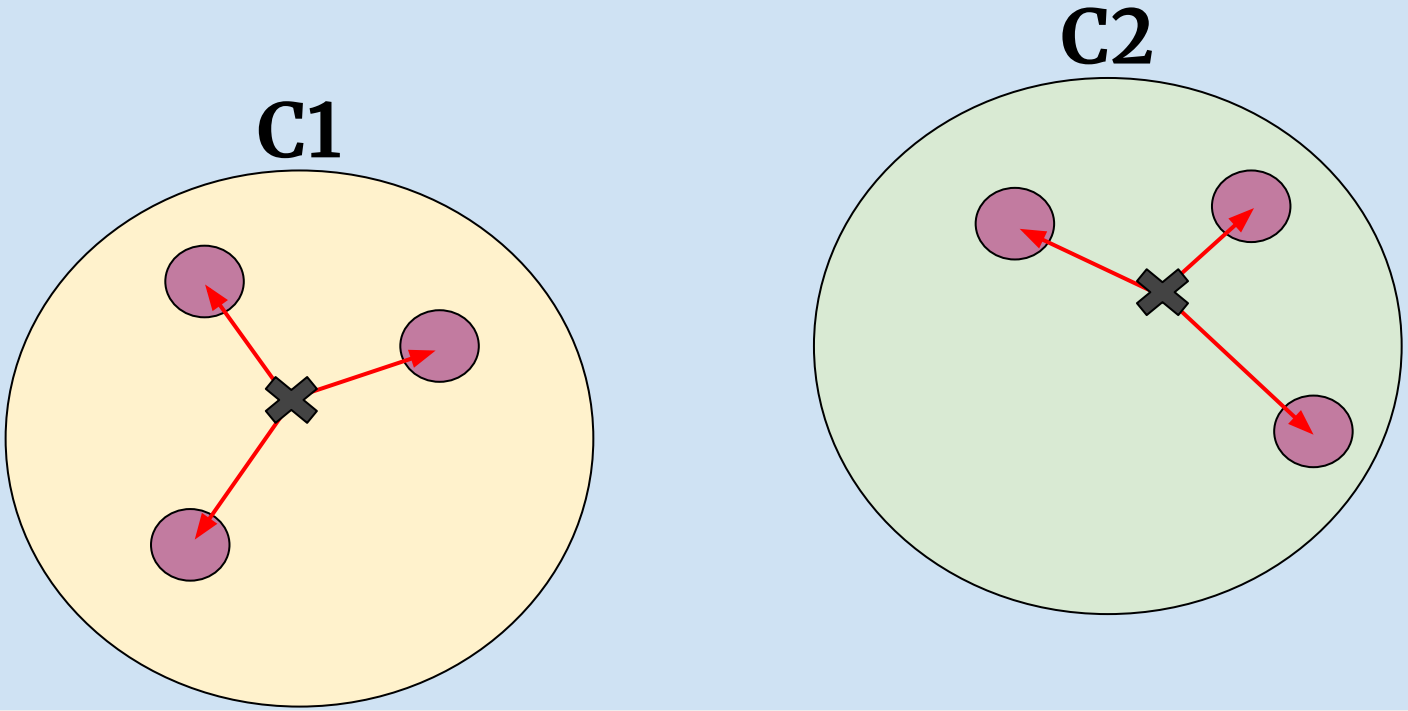
Average linkage

Average distance between all points



Ward’s method

Average within-cluster variance



What is *Hierarchical* clustering?

Algorithms which sort observations into a nested *hierarchies*

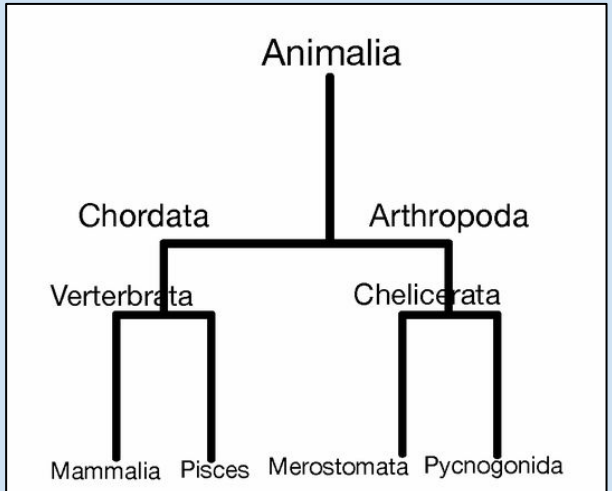
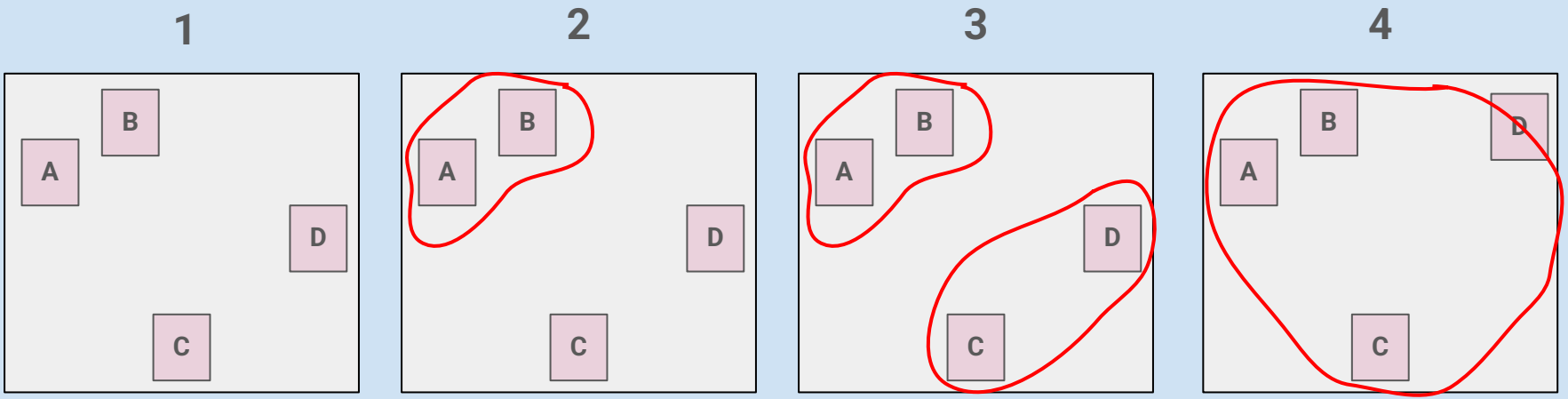


Figure 5.16: A snippet of Linnus' taxonomy that clusters organisms according to feature similarities.

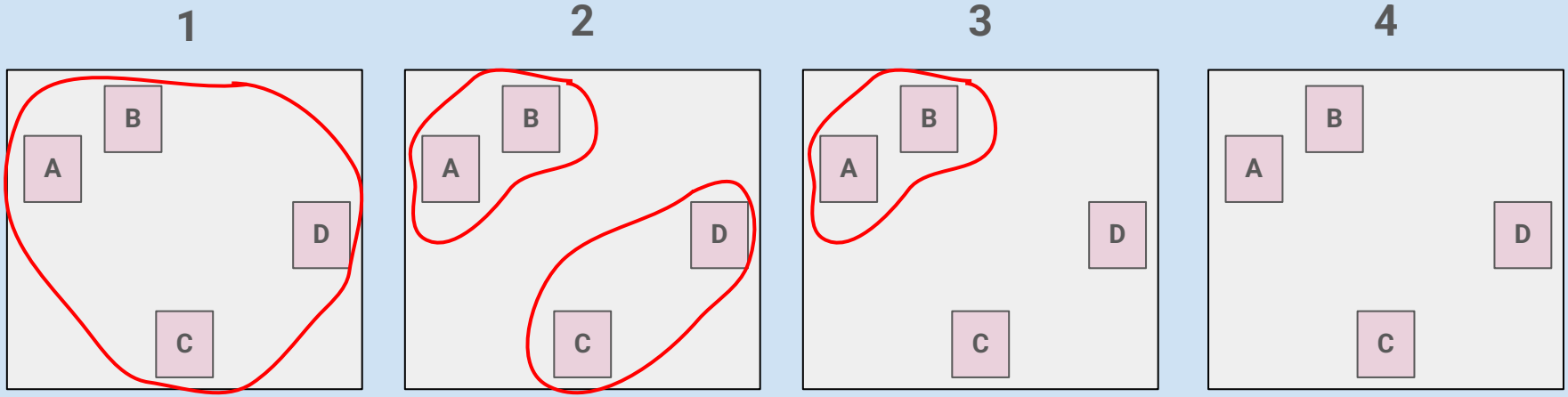
Agglomerative Hierarchical Clustering

Agglomerative = start with each point as its own cluster, finish with all points in one cluster



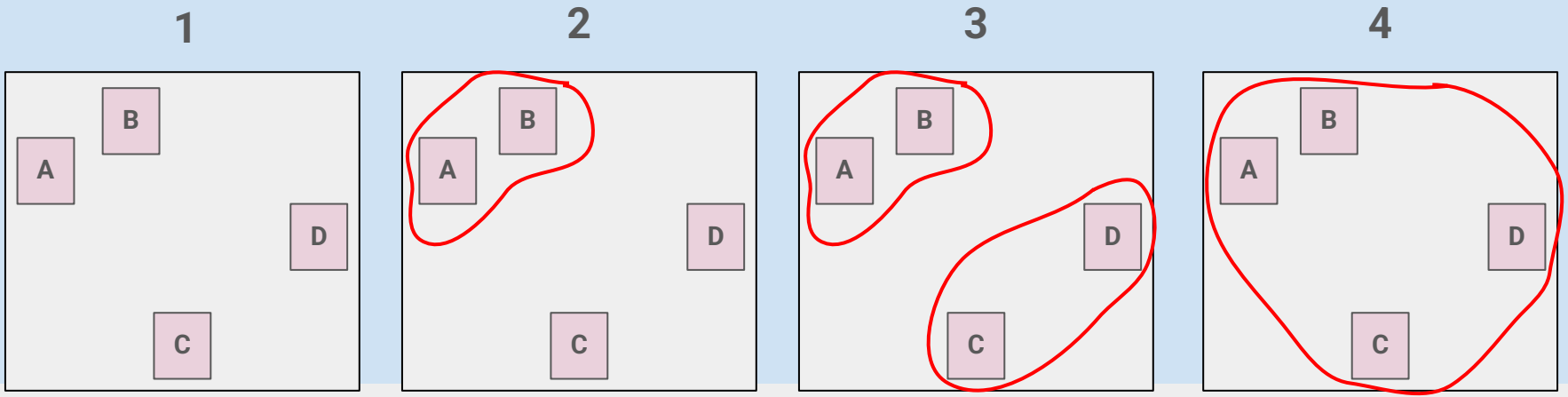
Divisive Hierarchical Clustering

Divisive = start with all points in one cluster, finish with each point as its own cluster



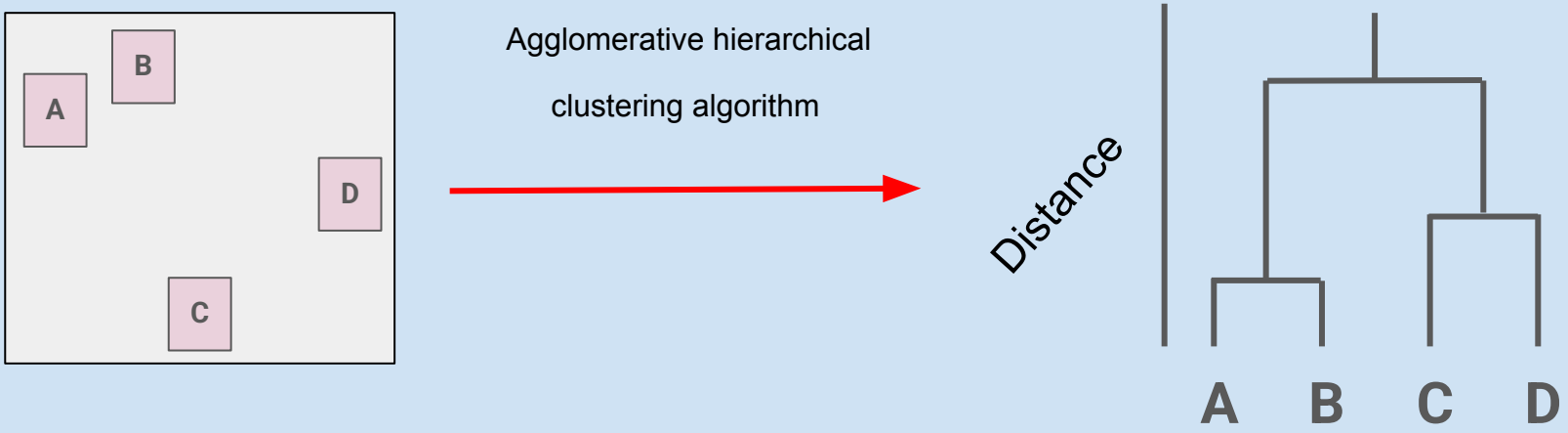
From the algorithm to clusters

- Hierarchical clustering algorithms produce a **series of iterations each defining a different set of clusters**
- We must choose at which point of the HCA to “stop” and interpret the clusters
- How could we decide this?



Dendrogram

- Visual representation of the iterations of a hierarchical clustering algorithm

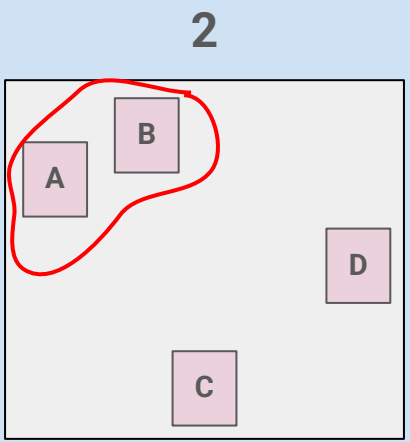


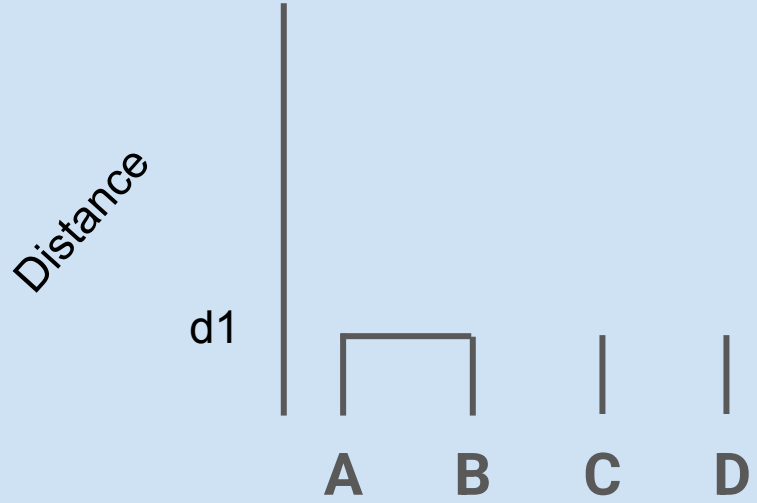
How to construct a dendrogram?

Stage 1: no clusters

How to construct a dendrogram?

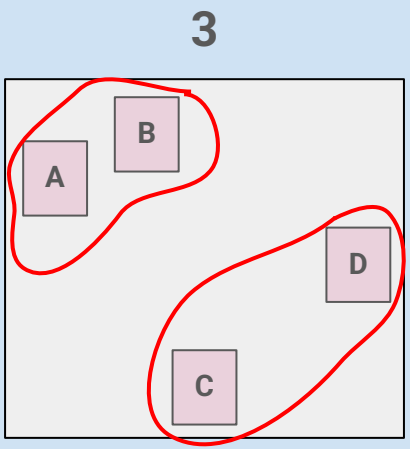
Stage 2: A and B join with some distance d1

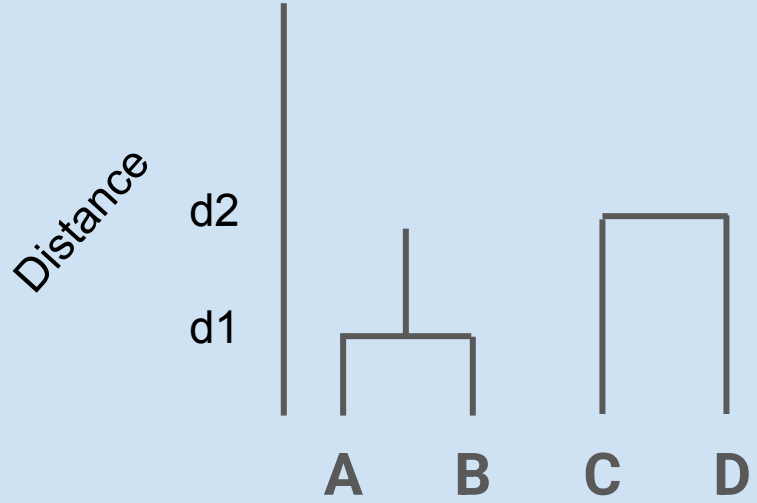




How to construct a dendrogram?

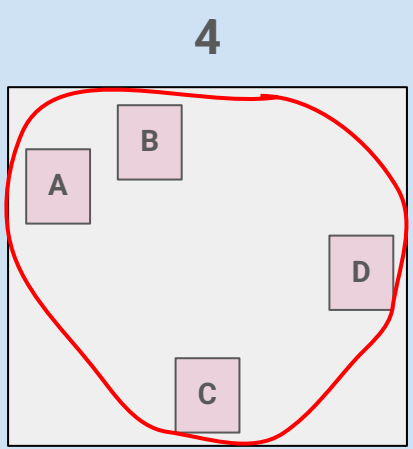
Stage 3: C and D join with some distance d2

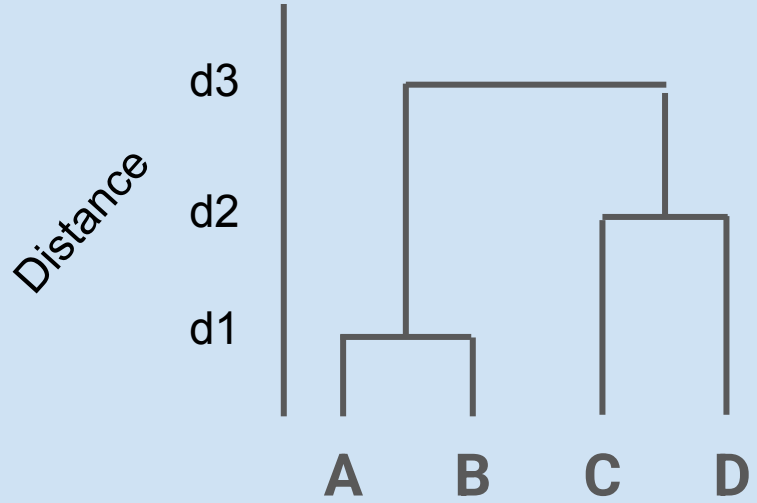




How to construct a dendrogram?

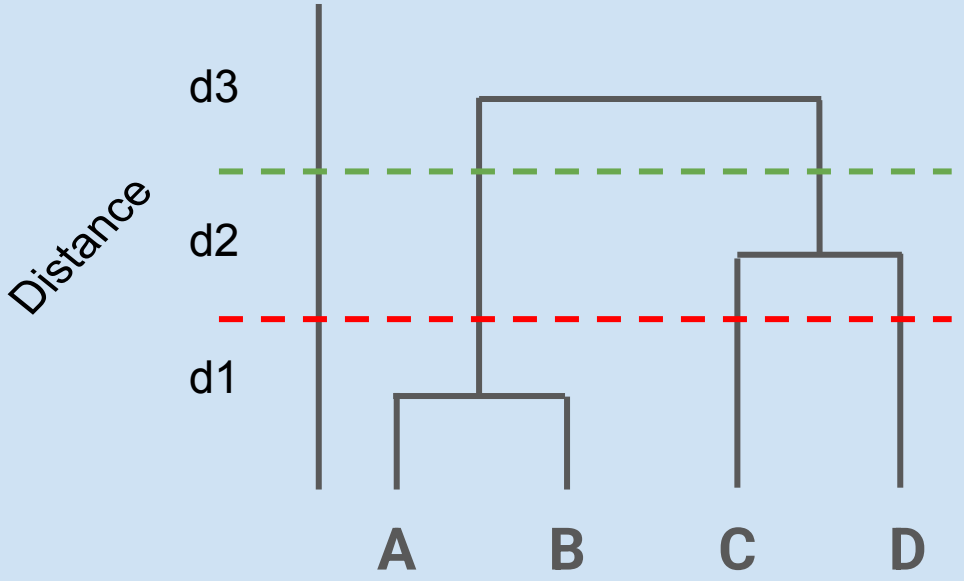
Stage 4: the cluster containing A and B joins with the cluster containing C and D with some distance d3





How to interpret dendrograms?

- Vertical axis gives **distance** at which clusters were **merged**
- ***Cutting* a dendrogram** at a given point therefore **defines a set of clusters**



Hierarchical Clustering summary

Hierarchical clustering requires 2 things

- **Distance metric** : a measure of similarity between points
- **Linkage function :** a way to measure similarity between groups

HC can be interpreted using dendrograms

- Since t**he “true” clusters are unknown** (and may not exist), hierarchical clustering is simply an **exploratory method**

Practical work

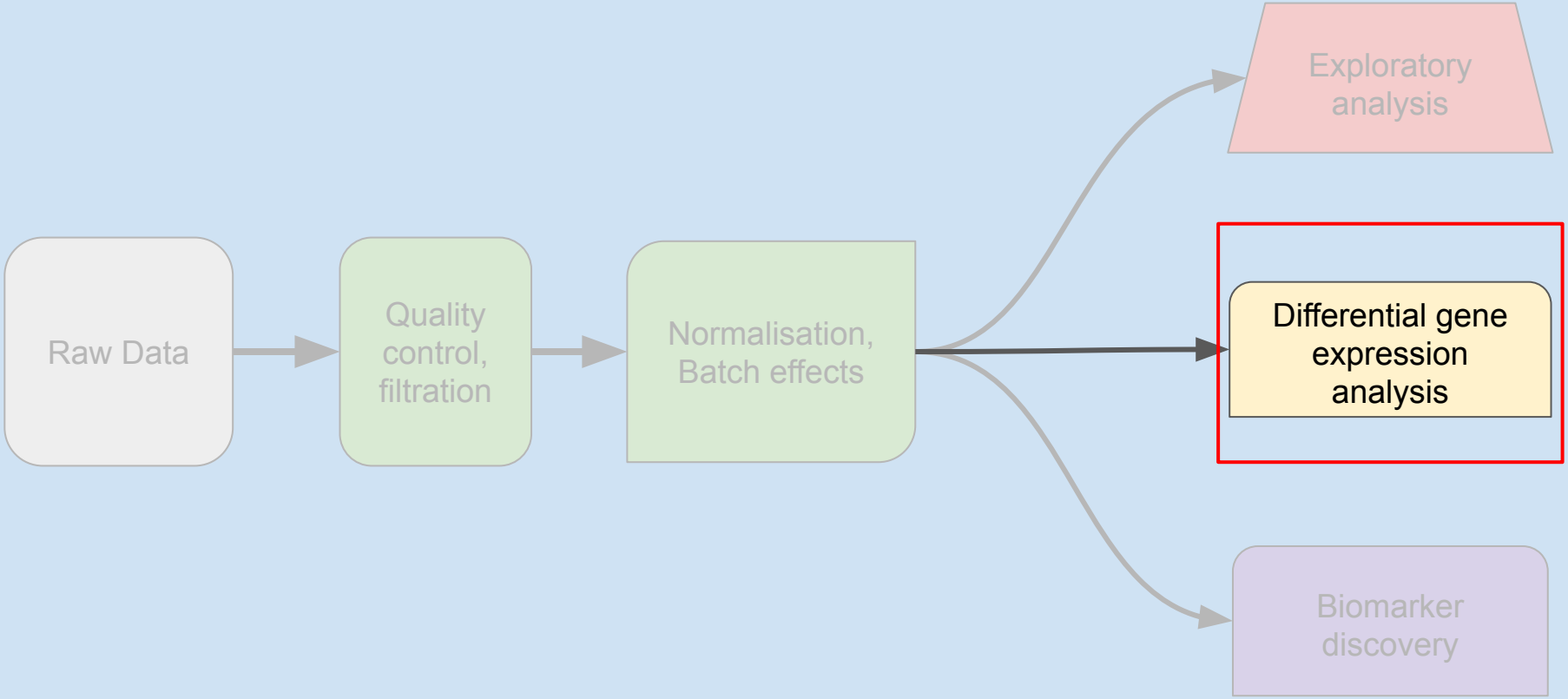
We are going to re-analyse a dataset containing transcriptomic samples from healthy patients and patients infected with mycobacterium tuberculosis, which causes tuberculosis disease. In this first practical, we will focus on

- Normalisation and batch effects
- Exploratory analysis using PCA and hierarchical clustering

Complete parts 0,1,2, and 3 from the R markdown file *PHDS_omics_GMA_2025_questions.Rmd*

Differential Gene Expression Analysis

How can we model count data?


Raw Data
Quality control, filtration
Normalisation, Batch effects
Exploratory analysis
Differential gene expression analysis
Biomarker discovery

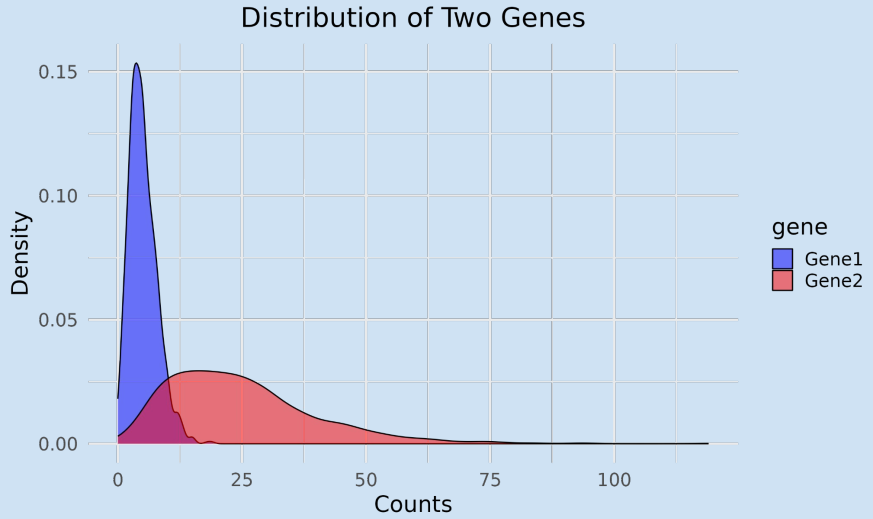
What is differential expression?

A gene is ***differentially expressed*** if there is a statistically significant change in expression between experimental conditions

- i.e. the change is greater than what would be expected due to random variation
- We need statistical tools to assess this

Reminder on RNA-seq data challenges

- Non-negative, discrete (integer) values
- Non-symmetric distribution
- Large dynamic range
- Heteroskedastic (mean depends on variance)
- Often small sample sizes
- **Noise - random, technical and biological**



How can we model count data?

- In RNA-seq, we randomly sample r reads (mRNA fragments) from a *sequencing library*
- These events *can be thought of as* independent
- If a library contains n_1 fragments corresponding to gene 1, n_2 to gene 2, etc., the library size is $n = n_1 + n_2 + ...$
- The probability of sampling a read corresponding to gene g is therefore $p_g = n_g/n$
- The mean number of counts for a gene is $\lambda_g = r \times p_g$

Poisson - a distribution for count data

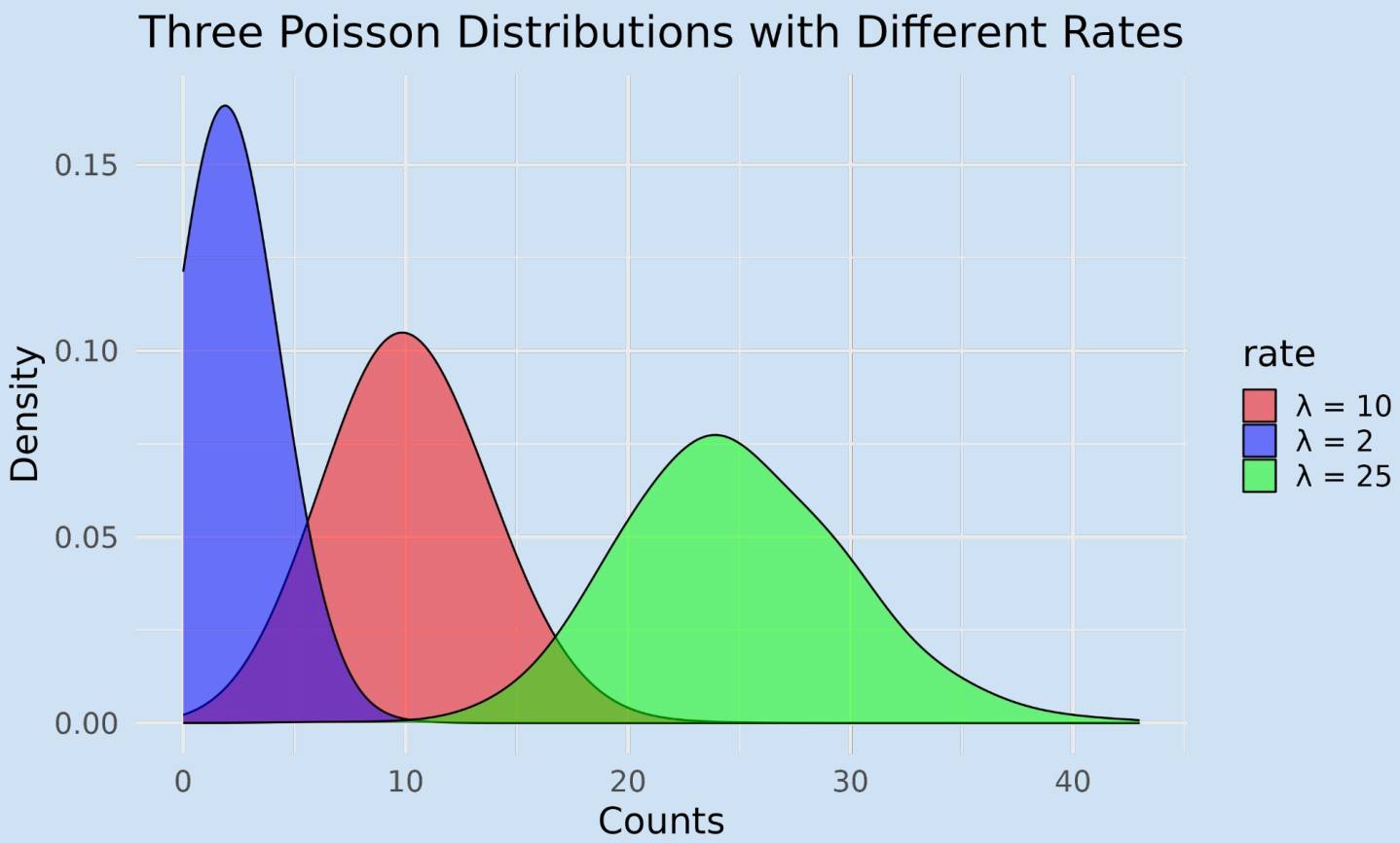
- Poisson distribution = number of events in a given period of time
- A random variable X follows a Poisson distribution with *rate parameter* $\lambda > 0$ if

$$\mathbb{P}(X = k) = \begin{cases} \dfrac{e^{-\lambda}\lambda^k}{k!}, & k = 0, 1, 2, \ldots, \\ 0, & \text{otherwise}, \end{cases}$$

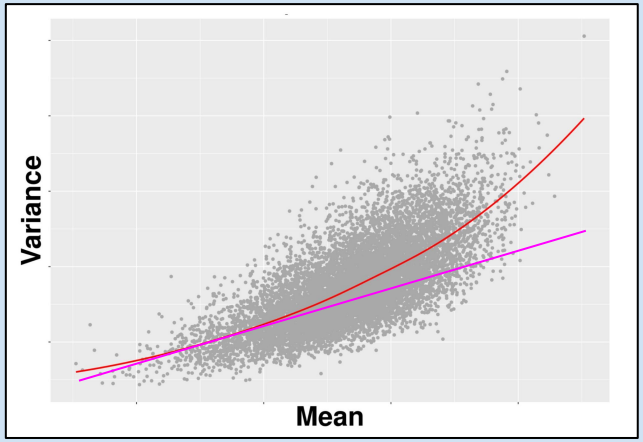
  Then $X \sim \text{Poisson}(\lambda)$.
- The expectation and variance of X are

$$\mathbb{E}[X] = \lambda, \quad \text{Var}(X) = \lambda.$$

- Could we model the counts for gene g, sample j as $y_{gj} \sim \text{Poisson}(\lambda_g)$?


Three Poisson Distributions with Different Rates
Density
0.15
0.10
0.05
0.00
0
10
20
30
40
Counts
rate
λ = 10
λ = 2
λ = 25

Reminder : RNA-seq is *overdispersed*



Mean vs variance for genes (simulated data)

- Pink line : mean = variance
- Red line : estimated mean-variance relationship
- **Mean-variance relationship is typically increasing and non-linear**

Negative Binomial Distribution

- Negative binomial = generalisation of Poisson with *overdispersion*
- A random variable X follows a Negative Binomial distribution with *mean* $\mu > 0$ and *dispersion* parameter $\theta > 0$, if

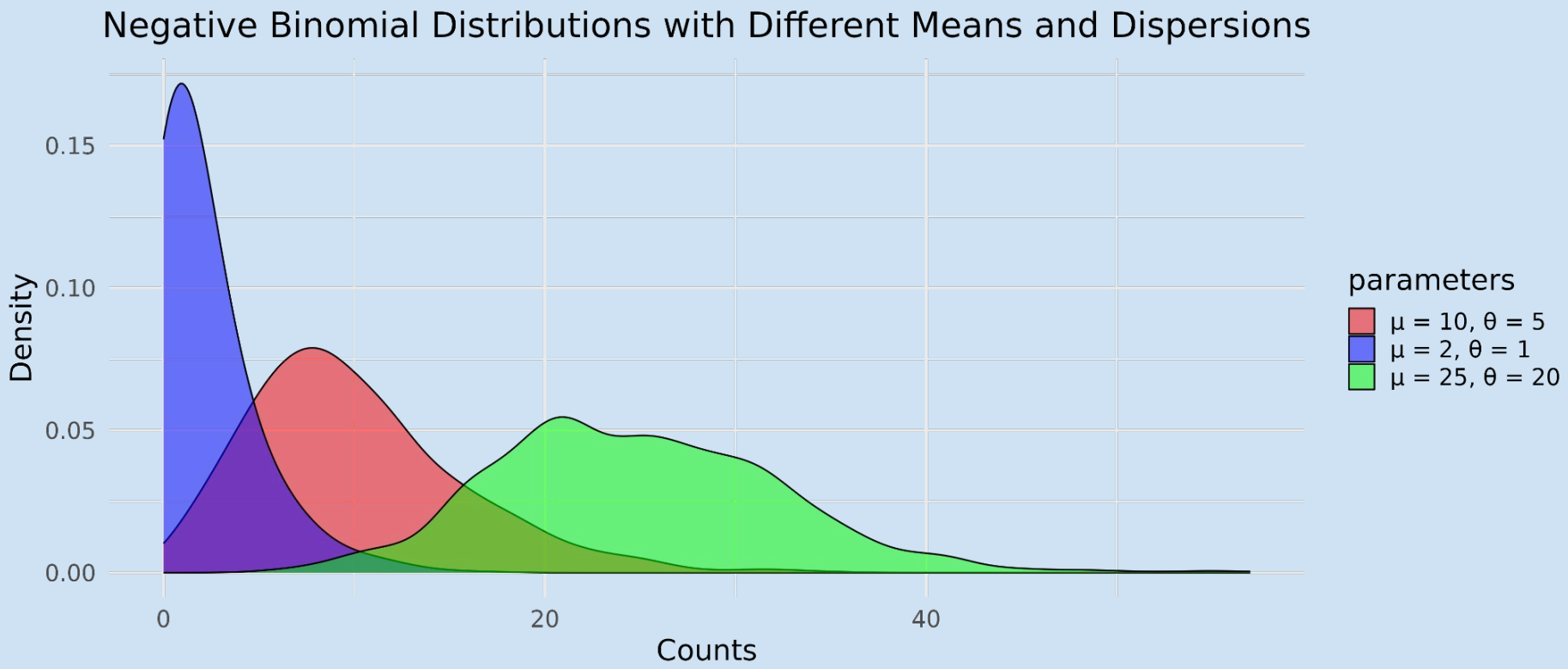
$$\mathbb{P}(X = k) = \begin{cases} \dfrac{\Gamma(k+\theta)}{k!\,\Gamma(\theta)} \left(\dfrac{\theta}{\theta+\mu}\right)^{\theta} \left(\dfrac{\mu}{\theta+\mu}\right)^{k}, & k = 0, 1, 2, \ldots, \\ 0, & \text{otherwise}, \end{cases}$$

  Then $X \sim \mathrm{NB}(\mu, \theta)$.

- The expectation and variance of X are

$$\mathbb{E}[X] = \mu, \quad \mathrm{Var}(X) = \mu + \frac{\mu^2}{\theta}.$$

- Could we model the counts for gene g, sample j as $y_{gj} \sim \mathrm{NB}(\lambda_g, \theta_g)$?


Negative Binomial Distributions with Different Means and Dispersions
Density
0.15
0.10
0.05
0.00
0
20
40
Counts
parameters
μ = 10, θ = 5
μ = 2, θ = 1
μ = 25, θ = 20

How to estimate dispersion parameters?

- Small sample size, large number of variables : difficult to estimate overdispersion for each gene

Popular packages using negative binomial modelling

- **DESeq2** : estimate mean-variance relationship across genes with regression
- **EdgeR** : borrow information across genes

Differential Gene Expression Analysis

DESeq2 and EdgeR Packages

DESeq2/EdgeR - estimating the mean

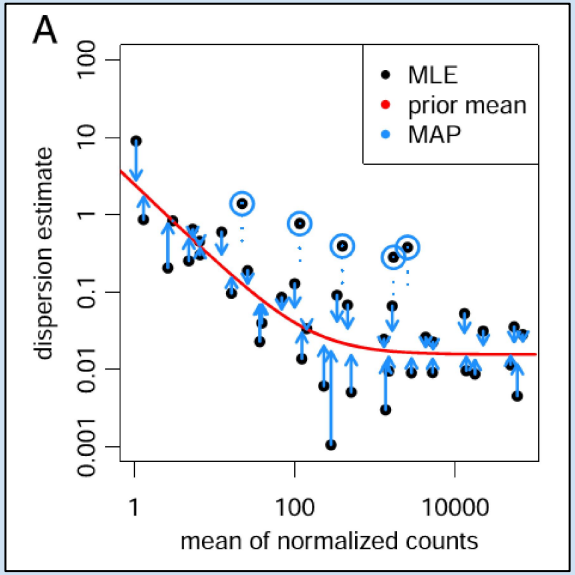
Estimating the mean

- Let x_j = experimental condition for sample j (i.e. 1 if treated, 0 if control)
- counts for gene g, sample j : $y_{gj} \sim NB(\mu_{gj}, \alpha_g)$
- mean $\mu_{gj} = s_j \times q_{gj}$ depends on
  - Sample-specific *normalisation factor* s_j
  - q_{gj} abundance of g in sample j
- Estimate $log(q_{gj}) = \beta_0 + \beta_1 x_j$

DESeq2/EdgeR - estimating the dispersion

Estimating the dispersion parameters

1. Estimate per-gene dispersion parameters with MLE
2. Estimate *mean-variance relationship* with local linear regression
3. *Shrink* per-gene estimates towards regression line (excluding large outliers)



DESeq2/EdgeR - testing

Hypothesis Testing for each gene

- Null hypothesis for gene g: $\beta_1 = 0$ (no effect of condition)
- Fit Negative Binomial GLM:

$$y_{gj} \sim NB(\mu_{gj}, \alpha_g), \quad \log(\mu_{gj}) = \log(s_j) + \beta_0 + \beta_1 x_j$$

- **DESeq2 :** Compute Wald statistic:

$$W_g = \frac{\hat{\beta}_1}{\mathrm{SE}(\hat{\beta}_1)}$$

  - Approximate p-value: $p_g = 2\Pr(|Z| > |W_g|),\ Z \sim N(0,1)$

- **EdgeR :** Quasi-likelihood F-test (detail omitted)
- Adjust p-values for multiple testing and reject null if $p < 0.05$

For more information

Love *et al. Genome Biology* (2014) 15:550
DOI 10.1186/s13059-014-0550-8

Genome Biology

METHOD Open Access

Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2

Michael I Love[1,2,3], Wolfgang Huber[2] and Simon Anders[2*]

edgeR: a Bioconductor package for differential expression analysis of digital gene expression data

Mark D. Robinson[1,2,*,†], Davis J. McCarthy[2,†] and Gordon K. Smyth[2]

Some problems with these packages

- Rely on the assumption that the raw counts are negative binomial
- Do not allow for complex experimental designs (longitudinal data, covariate correction)
- A linear model framework could fix these problems

Packages using linear model frameworks

- **Voom-limma**
- **dearseq**

Differential Gene Expression Analysis

Voom-Limma package

Voom-limma

Normalization and transformation

- Let x_j = experimental condition for sample j (e.g. 1 if treated, 0 if control)
- counts for gene g, sample j: y_{gj}
- Normalise counts for library size (e.g. TMM) to obtain scaling factors s_j
- Transform counts to log2-counts-per-million (logCPM):

$$\text{logCPM}_{gj} = \log_2\left(\frac{y_{gj}}{s_j} \cdot 10^6 + 0.5\right)$$

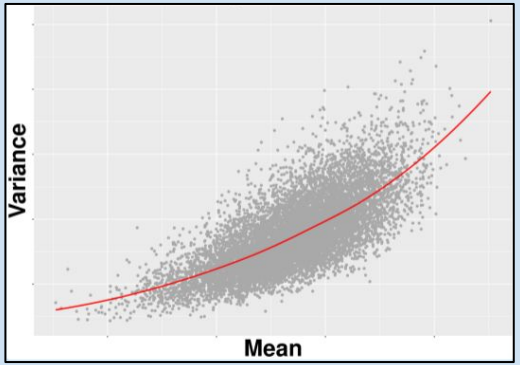
Voom-Limma

Estimating precision weights

- Compute the mean and variance of logCPM for each gene
- Estimate smooth mean–variance trend across genes
- Assign precision weights to each observation:

$$w_{gj} = \frac{1}{\hat{\mathrm{Var}}(\mathrm{logCPM}_{gj})}$$

- The higher the variance, the less the weight (i.e. noisy genes have less influence)



Voom-limma

Linear modeling and empirical Bayes

- Fit weighted linear model for each gene:

$$\text{logCPM}_{gj} \sim \beta_0 + \beta_1 x_j$$

- Use weights w_{gj} in weighted least squares
- *Shrink* estimates towards global mean

Voom-limma

Hypothesis testing

- Null hypothesis for gene g: $\beta_1 = 0$ (no effect of condition)
- Compute moderated t-statistics for $\hat{\beta}_1$
- Obtain p-values and adjust for multiple testing

Differential Gene Expression Analysis

dearseq Package

Dearseq package

Framework

- Let x_i denote covariates for sample i (e.g. condition, batch)
- Let y_{gi} be the (normalized) expression for gene g in sample i
- Working linear model for each gene:

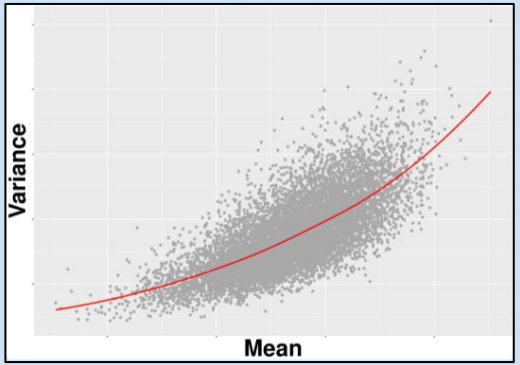
$$y_{gi} = \alpha_{g0} + X_i \alpha_g + \Phi_i \beta_g + \varepsilon_{gi},$$

where ε_{gi} has mean 0 and variance σ^2_{gi}; Φ_i contains the variable(s) of interest and β_g is the target parameter (gene is DE if $\beta_g \neq 0$).

Dearseq package

Estimating the mean–variance relationship

- Compute normalised expression
- Estimate mean-variance relationship across all genes and use as precision weights



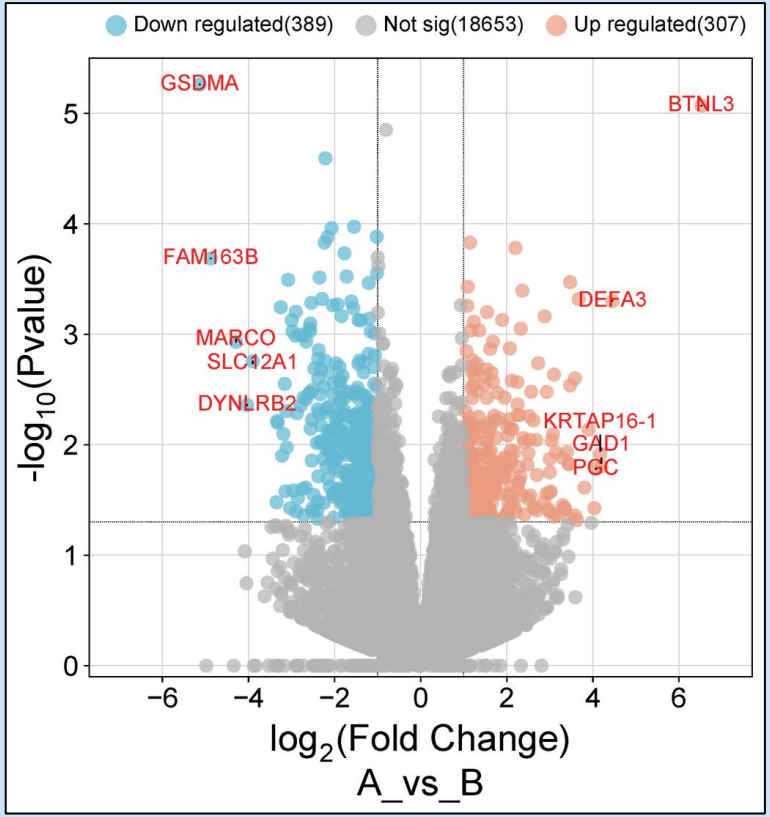
Dearseq package

Variance-component score testing

- Null hypothesis for gene g: $\beta_g = 0$
- Test with *score test*
- If n large enough : asymptotic test
- If n is small : permutation test
- Obtain p-values and adjust for multiple testing

Process	DESeq2	EdgeR	voom-limma	dearseq
Normalisation	Size factor (median of ratios)	TMM	TMM	User choice
Dispersion Estimation	**Negative binomial**, empirical Bayes shrinkage	**Negative binomial**, empirical Bayes shrinkage	Model mean-variance relationship with local linear smoothing	Model mean-variance relationship with local linear smoothing
Statistical Test	Wald Test or Likelihood ratio test	Likelihood Ratio Test or Quasi-likelihood F-test	Moderated t-test	Score test
Experimental Design	Simple	Can incorporate more complex experimental conditions	Can incorporate more complex experimental conditions + covariate correction	Can incorporate more complex experimental conditions + covariate correction + longitudinal data

Volcano plot



Conclusion on differential gene expression analysis

- Lots of different tools with the same goal
- Each tool has its own assumptions
- **Advice : try multiple methods and compare the results**

Practical work

We will now perform an analysis to investigate differentially expressed gene signatures of active tuberculosis infection

Complete part 4 from the R markdown file
PHDS_omics_GMA_2025_questions.Rmd