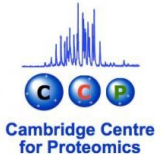


UNIVERSITY OF CAMBRIDGE
Bioinformatics Training

Analysis of expression proteomics data in R

December 2024

The Instructors

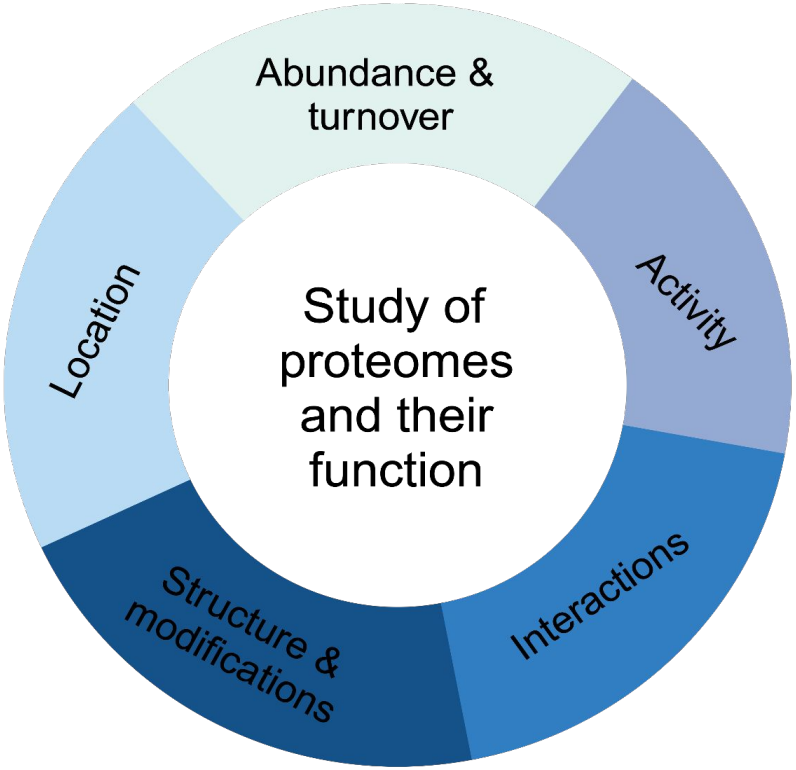
Alistair Hines (he/him)
PhD Candidate & Senior Scientist AstraZeneca
Cambridge Centre for Proteomics
University of Cambridge & Open University

Lisa Breckels (she/her)
Computational Biologist & Postdoctoral Research Associate
Cambridge Centre for Proteomics
Department of Biochemistry
University of Cambridge

Tom Smith (he/him)
Bioinformatician
MRC Laboratory of Molecular Biology (LMB)
Cambridge

Oliver Crook (he/him)
MRC Career Development & Todd-Bird Junior Research Fellow
Kavli Institute for NanoScience Discovery
University of Oxford

Applications of proteomics



Why use proteomics?

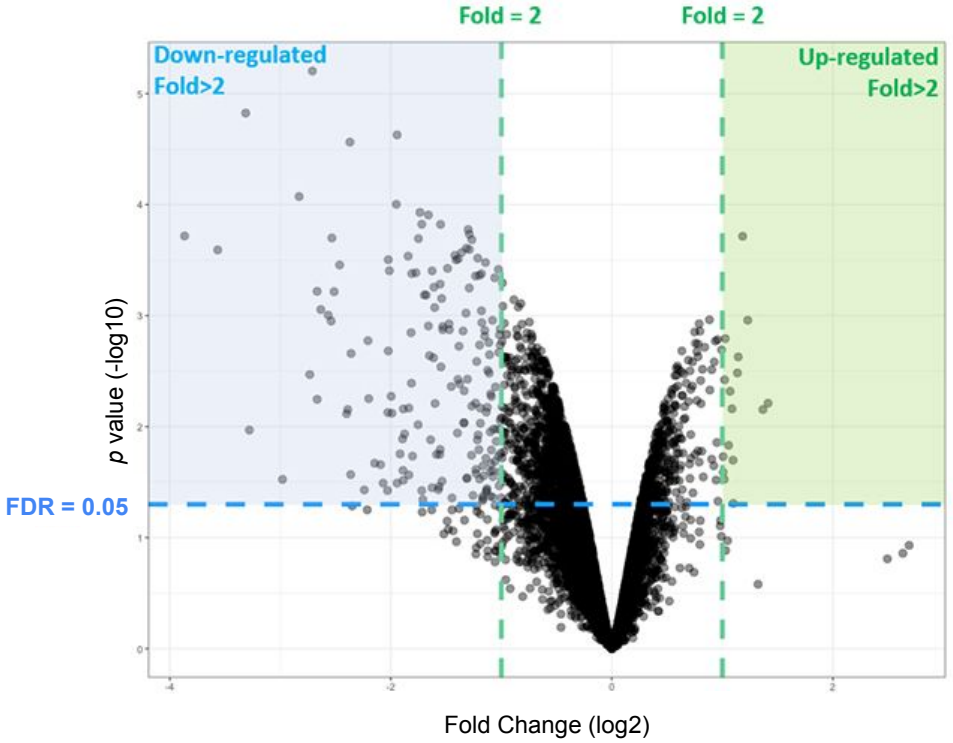
- Proteins are the effectors of the cell
- Phenotypic diversity cannot be completely explained at the genomic, epigenomic or transcriptomic level
- High throughput proteomics methods allow us to study thousands of proteins within a system at once

Expression proteomics

Expression proteomics = taking quantitative measurements of protein abundance across different samples

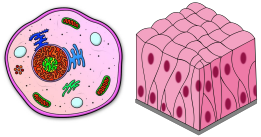
Aim: identify proteins that have differential abundance between samples

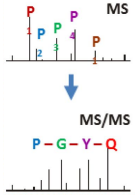
- Protein abundance is determined by synthesis and degradation
- Changes in protein abundance between conditions can be used to infer important biological aspects of the relative conditions e.g., disease

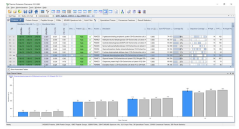


Identifying proteins via bottom-up MS

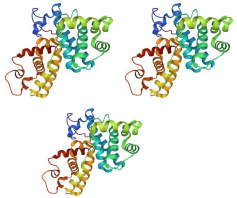
A successful expression proteomics experiment relies on proteins being both **identified** and **quantified**

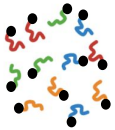


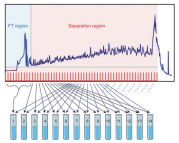




Sample(s) → Extract proteins → Proteolytic digestion → Label peptides → Sample clean-up → Offline fractionation → LC-MS/MS → Database search → Data analysis

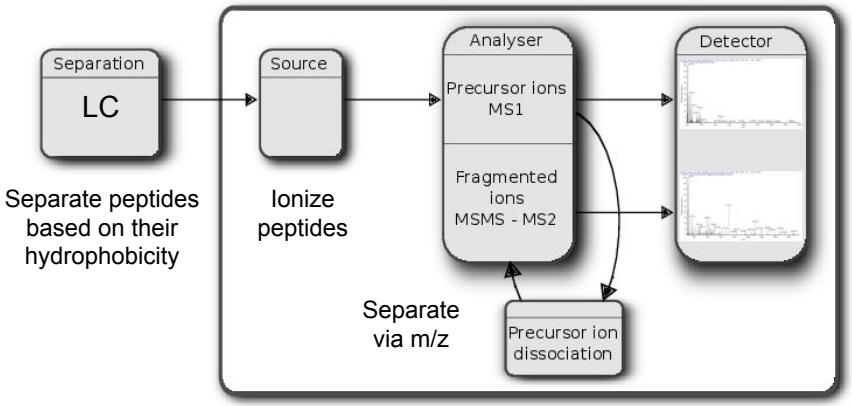


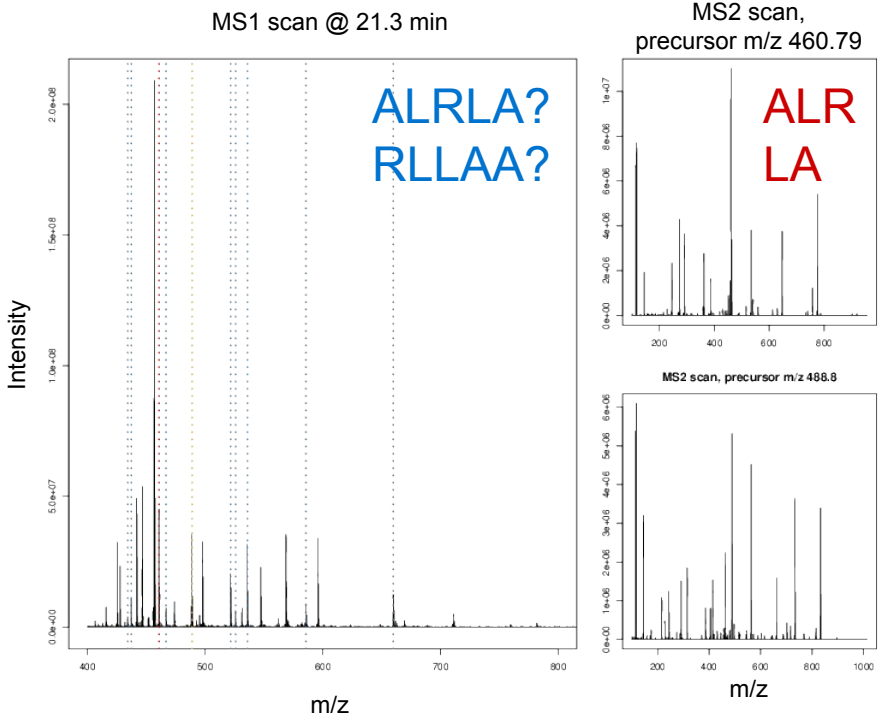




Steps completed during experimental sample preparation can influence downstream analysis

Identifying peptides via bottom-up MS





- **MS1:** retention time (inside LC column), mass-to-charge ratio, intensity
- **MS2 (MS/MS):** peptide sequence
- Tandem mass spectrometry (LC-MS/MS)

Figures modified from Gatto & Loriot, 2023

DDA vs. DIA

Data-dependent acquisition (DDA)
Top N most abundant precursor ions selected for fragmentation

Data-independent acquisition (DIA)
All precursor ions in a pre-determined m/z range selected for fragmentation

- Fewer missing values
- Less bias towards abundant peptides
- Only feasible more recently - specific programmes developed to deconvolute the complex spectra
- Still limited multiplexing capacity

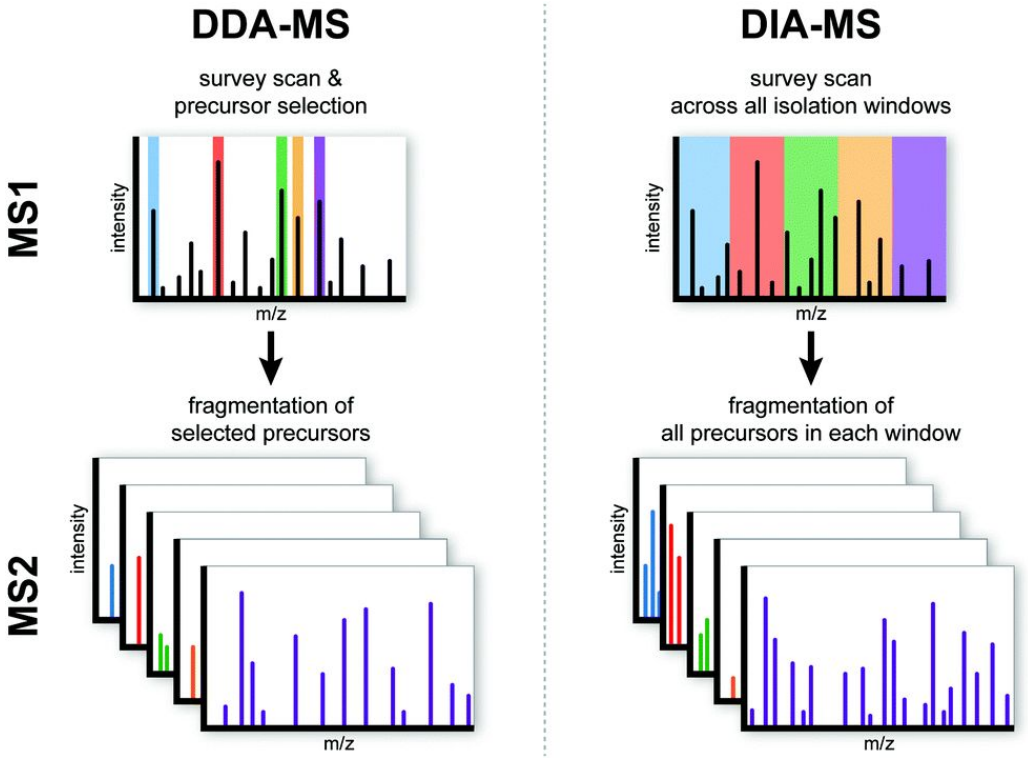
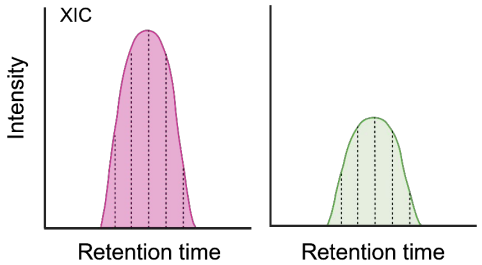


Figure from Krasny & Huang, 2020

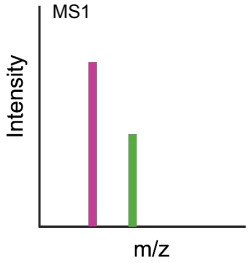
Quantifying peptides via bottom-up MS

Label-free quantification (MS1)



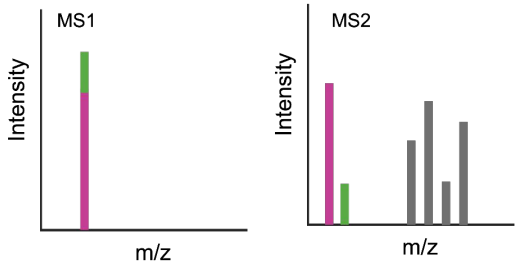
- Samples run independently on MS
- Extracted ion chromatograms (XIC)
- Peaks for a precursor ion across different retention times integrated
- Area under curve used to calculate peptide abundance

Metabolic labelling - SILAC (MS1)



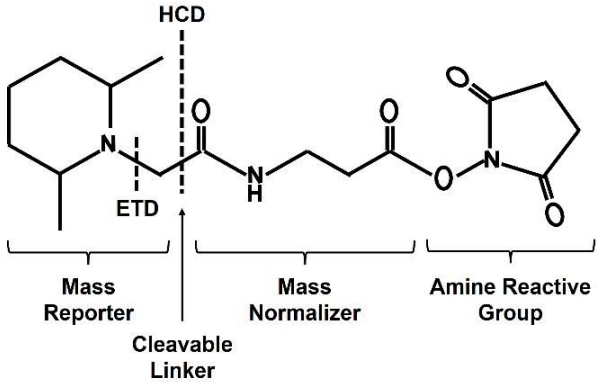
- Cells grown in media containing ‘heavy’ and ‘light’ amino acids
- Proteins from samples are pooled and run together on MS
- Each protein has a heavy and light peak corresponding to the sample from which it was derived
- Ratiometric analysis

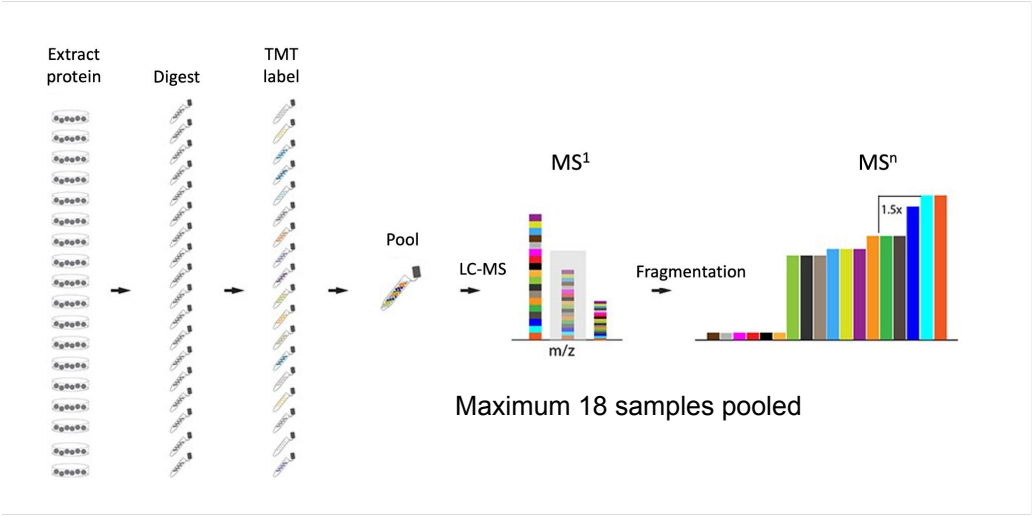
Chemical labelling - TMT (MS2)



- Peptides chemically labelled with isobaric tags
- Peptides from different samples are pooled and run on MS together
- Indistinguishable at MS1
- Reporter ions released at MS2 (or MS3)
- Reporter ion intensity used as abundance

Tandem Mass Tag (TMT) labelling





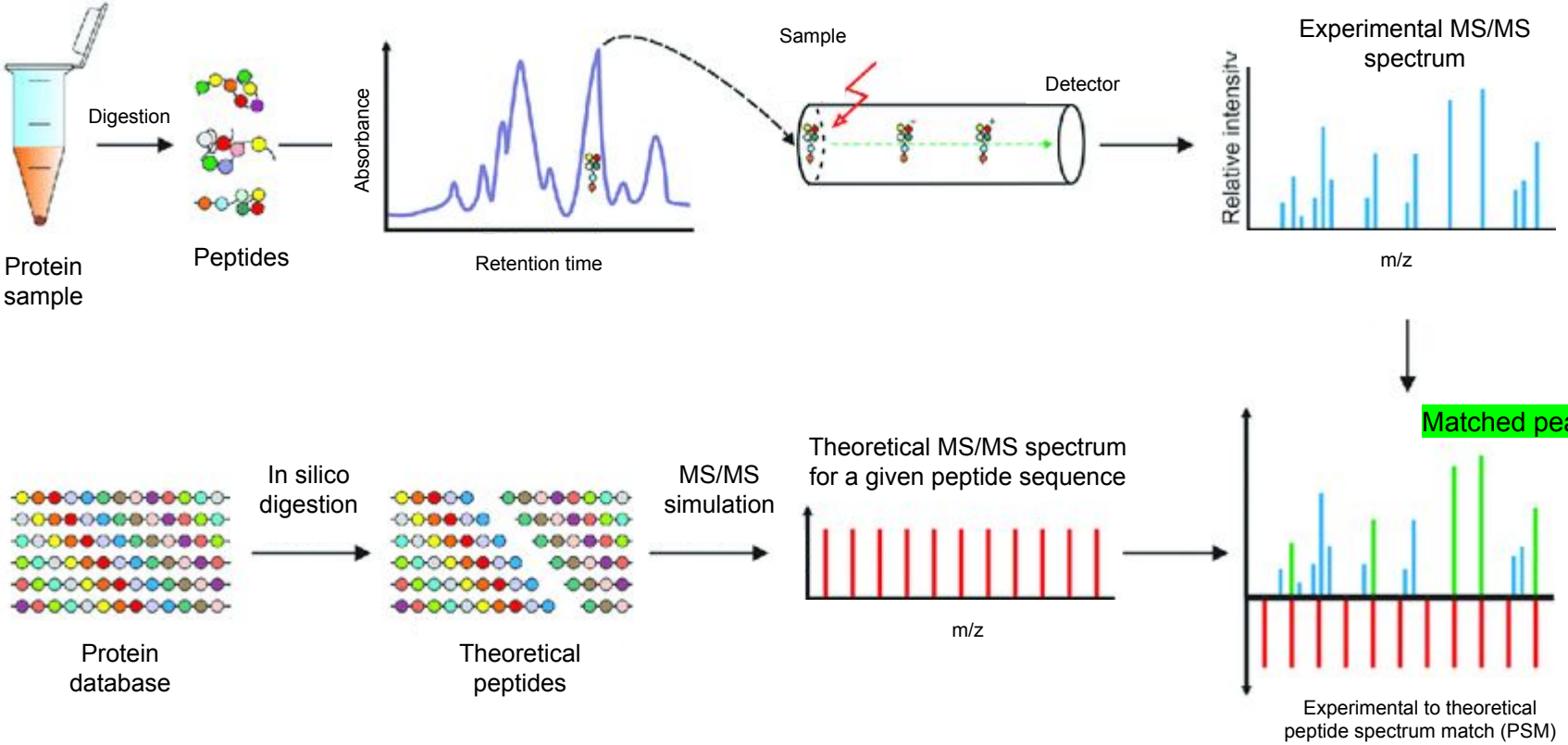
- Set of isobaric labels (same total mass) which covalently attach to peptide N-termini and lysine (K) residues
- Not distinguishable at MS1 but HCD fragmentation releases reporter ions of differential mass at MS2 or MS3
- Relative quantification of reporter ions corresponds to relative peptide quantification across samples

Advantage: Multiplexing samples reduces MS time and variability, particularly stochastic selection of precursors in DDA

Summary (1)

- Expression proteomics involves the global identification and quantification of proteins
- Comparison of protein abundance between conditions can be used to infer important molecular players for particular biological processes e.g., disease
- Bottom-up MS involves the digestion of proteins and analysis of peptides
- MS data can be acquired by data-dependent or -independent acquisition
- Quantification can be achieved by label-free or label-based methods

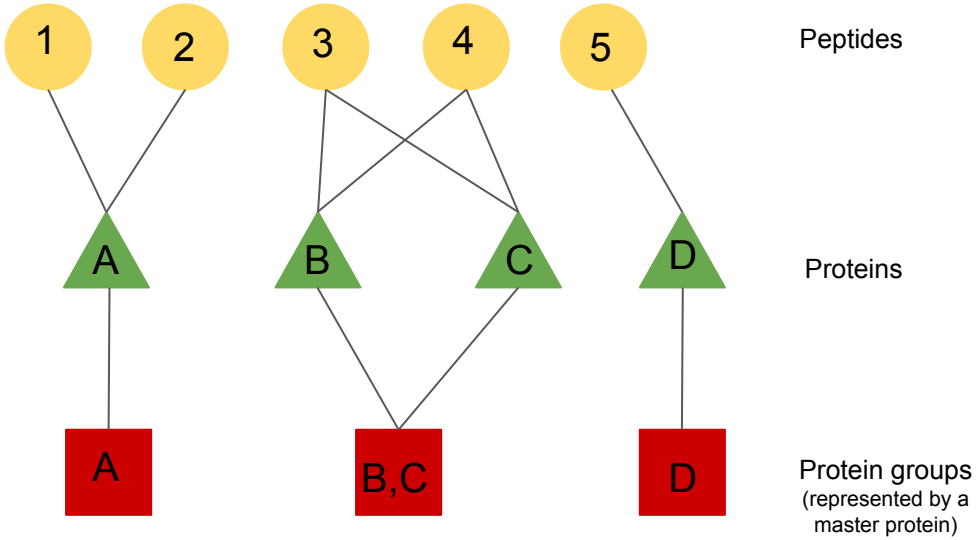
Database search of raw MS data



Peptide spectrum match (PSM) = a match between theoretical and observed mass spectra that can link an observed mass spectra to its potential peptide sequence

Figure from Chunganova et al., 2017

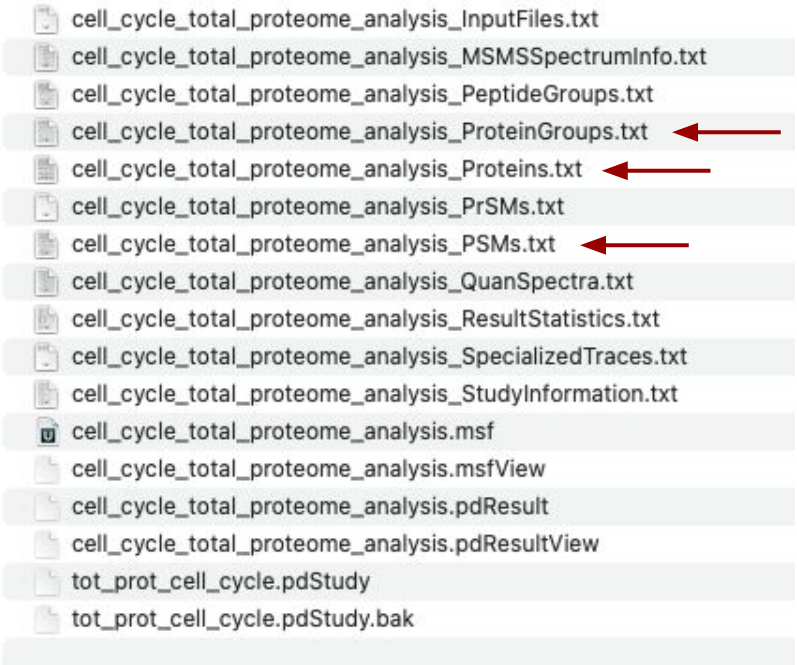
From peptide to protein



Protein inference is an complex and widely researched topic:

- We have identified and quantified peptide sequences based on PSMs
- We want to infer information about proteins
- A peptide sequence may be **unique** to only one protein in our database
- Or a peptide sequence may be **shared** between multiple proteins in our database
- Proteins that only have shared peptides and cannot be unambiguously identified become **protein groups**
- A **master protein** is selected to represent each protein group

Database search of raw MS data



What do we get out of a database search?

- A list of peptide spectrum matches (PSMs)

Depending on the third party software used, we can also get:

- Peptide and protein level data aggregated from the original PSM data
- Summary statistics and optional analysis tools

Recommend completing your data analysis from the lowest possible data level (PSM or peptide depending on quantification method)

Summary (2)

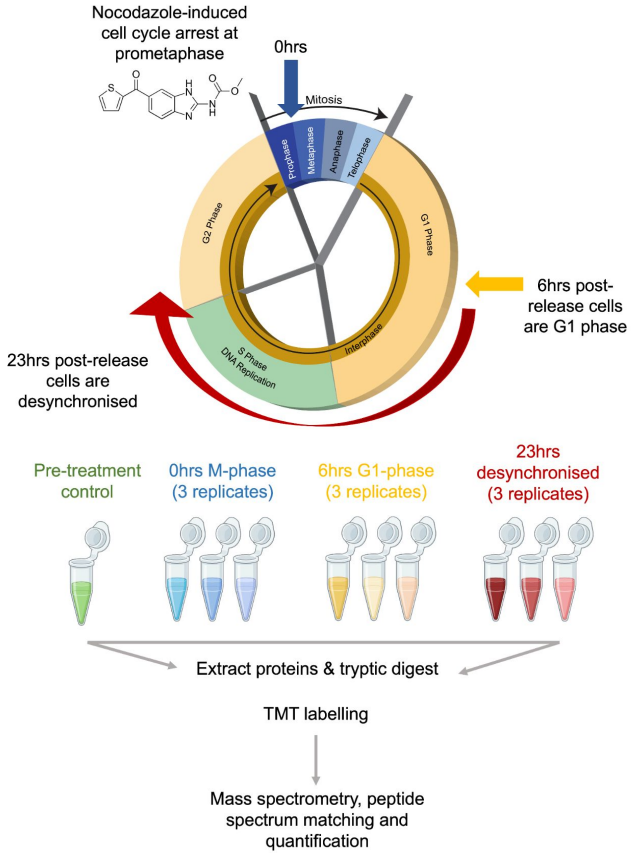
- The raw MS spectra are processed using third-party software, most commonly via a database search
- The direct output of a database search is a list of peptide spectrum matches (PSMs), which represent links between raw mass spectra and peptide sequences
- We get PSM, peptide and protein level outputs from the database search
- Taking the lowest possible data level output provides maximum user control and understanding

Use-case expression proteomics data

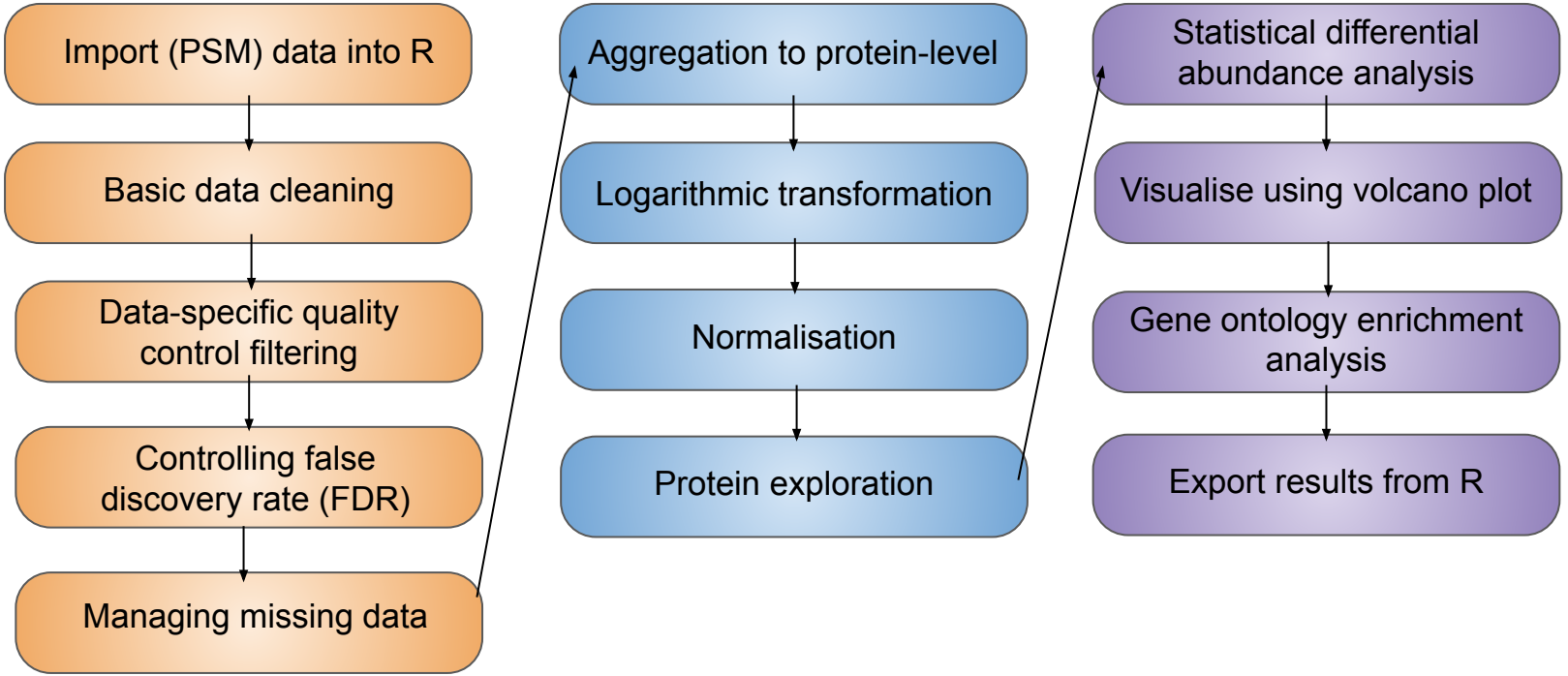
Analysis of expression proteomics data case-study (Queiroz et al., 2019):

- Data dependent acquisition
- TMT labelling and quantification
- Raw data processed via a database search using Proteome Discoverer v3.0
- Analysis will start from the PSM level

Aim: Identify proteins with differential abundance across cell cycle stages



In this workshop…



Glossary

Key terms that you should understand for this workshop:

Precursor ion = the original parent ion representing an ionized form of the entire peptide sequence.

Fragment ion = an ion produced by fragmentation of the precursor ion, thus only representing a fraction of the original peptide sequence.

MS1 spectrum = raw mass spectrum produced by the separation of precursor ions based on their mass-to-charge ratio (m/z). Each peak represents a precursor ion at a particular m/z and with an associated intensity.

MS2 (MS/MS) spectrum = raw mass spectrum produced by the separation of fragment ions based on their mass-to-charge ratio (m/z). Each peak corresponds to a fragment ion derived from the same precursor ion.

Peptide spectrum match (PSM) = A match made between a theoretical mass spectrum for a given peptide sequence and an observed experimental spectrum, thus linking a raw mass spectrum to its predicted peptide sequence

Tandem mass tag (TMT) = a type of peptide label which can be used for relative quantification of peptides across samples. Quantification is measured at the MS2 or MS3 level.

References and further information

Case-study data:

- Queiroz, R.M.L., Smith, T., Villanueva, E., Marti-Solano, M., Monti, M., Pizzinga, M., Mirea, D.-M., Ramakrishna, M., Harvey, R.F., Dezi, V., Thomas, G.H., Willis, A.E. & Lilley, K.S. (2019) Comprehensive identification of RNA–protein interactions in any organism using orthogonal organic phase separation (OOPS). *Nature Biotechnology*. 37 (2), 169–178. doi:10.1038/s41587-018-0001-2.

Mass spectrometry-based proteomics:

- Dupree, E.J., Jayathirtha, M., Yorkey, H., Mihasan, M., Petre, B.A. & Darie, C.C. (2020) A Critical Review of Bottom-Up Proteomics: The Good, the Bad, and the Future of This Field. *Proteomes*. 8 (3), 14. doi:10.3390/proteomes8030014.
- Jiang, Y., … Meyer, J.G. (2024) Comprehensive Overview of Bottom-Up Proteomics Using Mass Spectrometry. *ASC Measurement Science Au*. doi:10.1021/acsmeasuresciau.3c00068
- Obermaier, C., Griebel, A. & Westermeier, R. (2021) Principles of protein labeling techniques. In: A. Posch (ed.). *Proteomic Profiling: Methods and Protocols*. Methods in Molecular Biology. New York, NY, Springer US. pp. 549–562. doi:10.1007/978-1-0716-1186-9_35.

Data analysis workflow:

- Hutchings, C., Dawson, C.S., Krueger, T., Lilley, K.S. & Breckels, L.M. (2024) *A Bioconductor workflow for processing, evaluating, and interpreting expression proteomics data*. doi:10.12688/f1000research.139116.2.
- Rainer, L.G., Sebastian Gibb, Johannes (n.d.) *Chapter 5 Quantitative data | R for Mass Spectrometry*. https://rformassspectrometry.github.io/book/sec-quant.html.

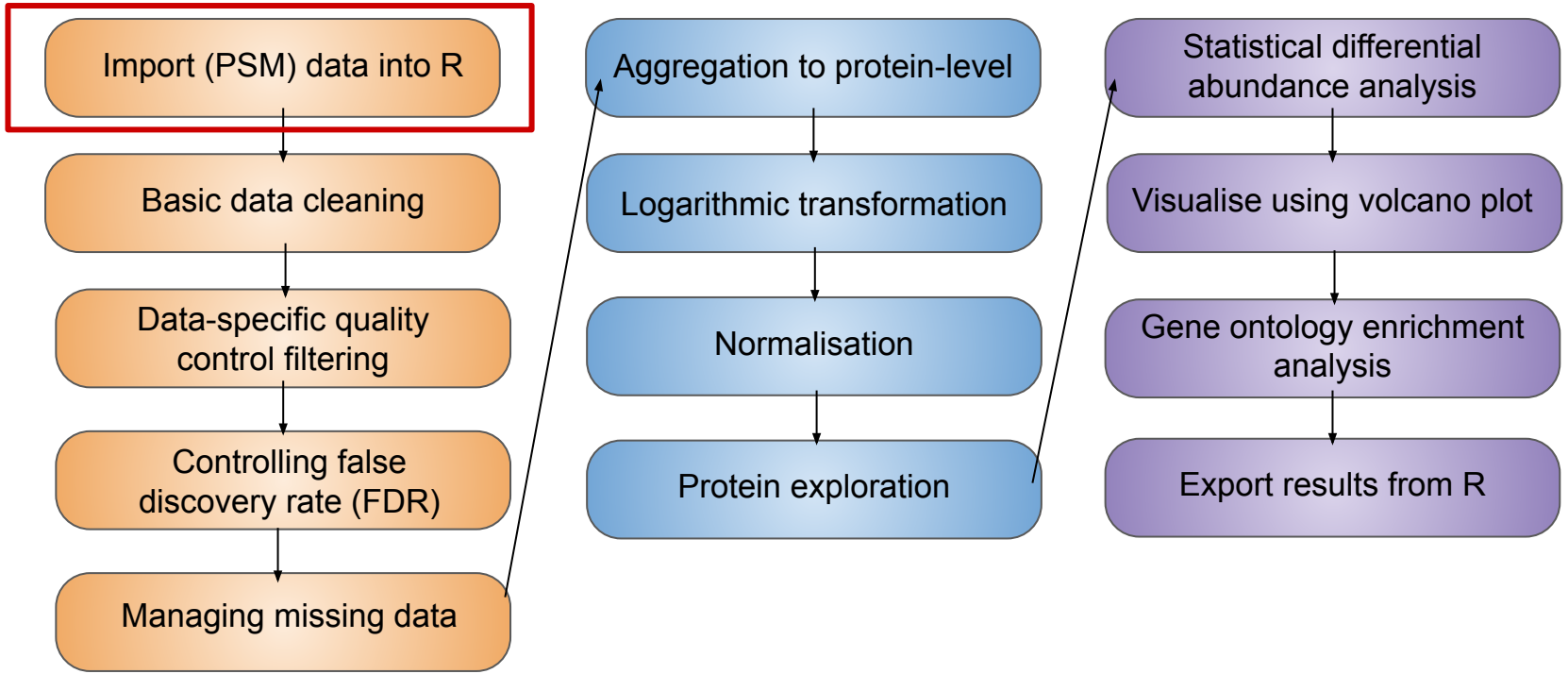
Cambridge Centre for Proteomics

UNIVERSITY OF CAMBRIDGE
Bioinformatics Training

Lesson 2: Import and Infrastructure

December 2024

Data analysis overview



How is our data organised?

Start with *“cell_cycle_total_proteome_analysis_PSMs.txt”*

Sequence	Annotated Sequence	...	Number of Proteins	Master Protein Accessions	Abundance 126	Abundance 127N	Abundance 127C	Abundance 128N	Abundance 128C	Abundance 129N	Abundance 129C	Abundance 130N	Abundance 130C	Abundance 131
REEEMR	rEEEmR	...	2	P49755; Q15233	22.3	48.6	15.1	26.8	39.0	21.6	41.6	38.7	21.2	34.7
QQNGTASSR	qQnGTASSR	...	1	Q92917	19.8	31.3	14.1	19.5	32.1	29.2	37.1	41.2	30.0	34.9
CHMEENQR	cHmEENQR	...	1	Q96TC7	2.4	7.9	1.2	2.3	4.1	1.0	3.6	7.0	3.1	2.7
QACQER	qAcQER	...	1	P26358	2.2					2.3	3.1		1.7	
HSEAATAQR	hSEAATAQR	...	1	Q14103	2.2	4.2	2.2	1.6	5.7	1.4		3.6	4.0	4.0
QQQQQQQHQQPNR	qQQQQQQHQQPNR	...	1	Q6Y7W6		1.7								2.7
TANREECR	tANREEcR	...	1	Q4G0J3	20.7	35.9	7.1	25.0	36.1	22.5	33.3	32.3	20.9	33.2
QRQEEEAR	qRQEEEAR	...	2	P55036	7.0	14.5	1.9	5.7	12.3	13.4	13.7	14.5	15.3	10.2
QEAAQSR	qEAAQSR	...	1	Q96A33	8.6	14.9	2.9	10.4	10.7	10.8	20.0	11.8	8.7	12.6
AQANEQR	aQANEQR	...	1	O00291	4.4	7.7		7.2	7.7	6.7	2.5	7.5	7.6	3.3
TSGDTNAR	tSGDTNAR	...	1	Q8N1F7	17.7	31.2	6.5	11.3	16.2	23.9	31.8	35.0	19.7	34.9
QQQQQQQHQQPNR	qQQQQQQHQQPnR	...	1	Q6Y7W6	2.1	6.0	1.4	3.2	3.3	6.9	7.1	2.3	3.1	8.0

- Data in obtained after quantification and identification of the raw MS files
- Third party software e.g. Proteome Discoverer, MaxQuant, FragPipe etc returns tabular data

How is our data organised?

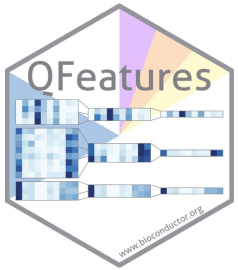
We find the data contains two types of columns:

Sequence	Annotated Sequence	...	Number of Proteins	Master Protein Accessi	Abundance 126	Abundance 127N	Abundance 127C	Abundance 128N	Abundance 128C	Abundance 129N	Abundance 129C	Abundance 130N	Abundance 130C	Abundance 131
REEEMR	rEEEmR	...	2	P49755; Q15233	22.3	48.6	15.1	26.8	39.0	21.6	41.6	38.7	21.2	34.7
QQNGTASSR	qQnGTASSR	...	1	Q92917	19.8	31.3	14.1	19.5	32.1	29.2	37.1	41.2	30.0	34.9
CHMEENQR	cHmEENQR	...	1	Q96TC7	2.4	7.9	1.2	2.3	4.1	1.0	3.6	7.0	3.1	2.7
QACQER	qAcQER	...	1	P26358	2.2					2.3	3.1		1.7	
HSEAATAQR	hSEAATAQR	...	1	Q14103	2.2	4.2	2.2	1.6	5.7	1.4		3.6	4.0	4.0
QQQQQQQHQQPNR	qQQQQQQHQQPNR	...	1	Q6Y7W6		1.7								2.7
TANREECR	tANREEcR	...	1	Q4G0J3	20.7	35.9	7.1	25.0	36.1	22.5	33.3	32.3	20.9	33.2
QRQEEEAR	qRQEEEAR	...	2	P55036	7.0	14.5	1.9	5.7	12.3	13.4	13.7	14.5	15.3	10.2
QEAAQSR	qEAAQSR	...	1	Q96A33	8.6	14.9	2.9	10.4	10.7	10.8	20.0	11.8	8.7	12.6
AQANEQR	aQANEQR	...	1	O00291	4.4	7.7		7.2	7.7	6.7	2.5	7.5	7.6	3.3
TSGDTNAR	tSGDTNAR	...	1	Q8N1F7	17.7	31.2	6.5	11.3	16.2	23.9	31.8	35.0	19.7	34.9
QQQQQQQHQQPNR	qQQQQQQHQQPnR	...	1	Q6Y7W6	2.1	6.0	1.4	3.2	3.3	6.9	7.1	2.3	3.1	8.0

Feature meta data

Quantitative data

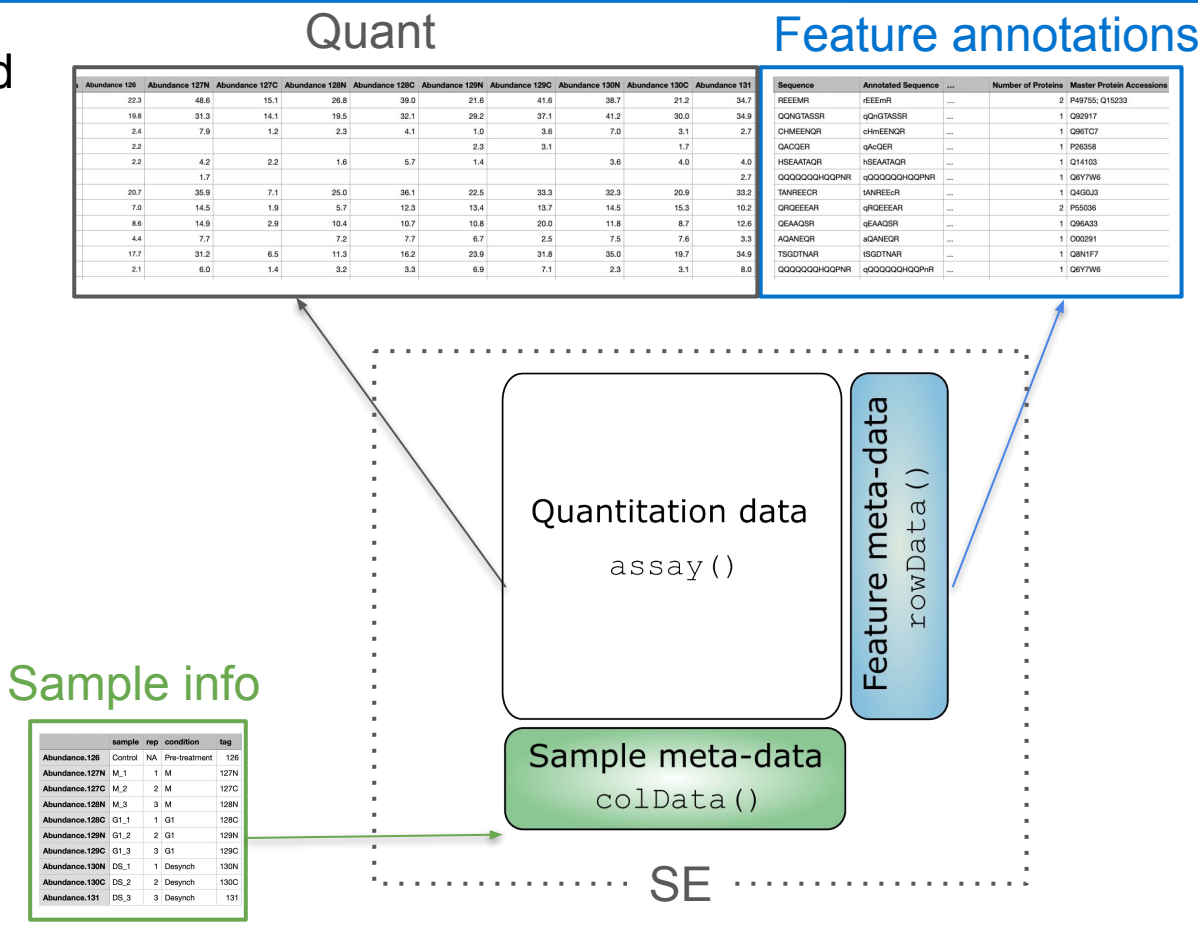
Dedicated data structures in R/Bioconductor



- Data in proteomics (and other omics') is complex
- Developers define specialised data containers to store omics' data
- Bioconductor packages are a set of R packages that facilitate reproducible analysis of omics' data
- We will be using the SummarizedExperiment and QFeatures packages

(1) The SummarizedExperiment class

- Provides a dedicated structure to store your MS data
- A SE contains 3 different parts
  1) assay slot (quant)
  2) rowData slot (feature annotations)
  3) colData slot (sample info)



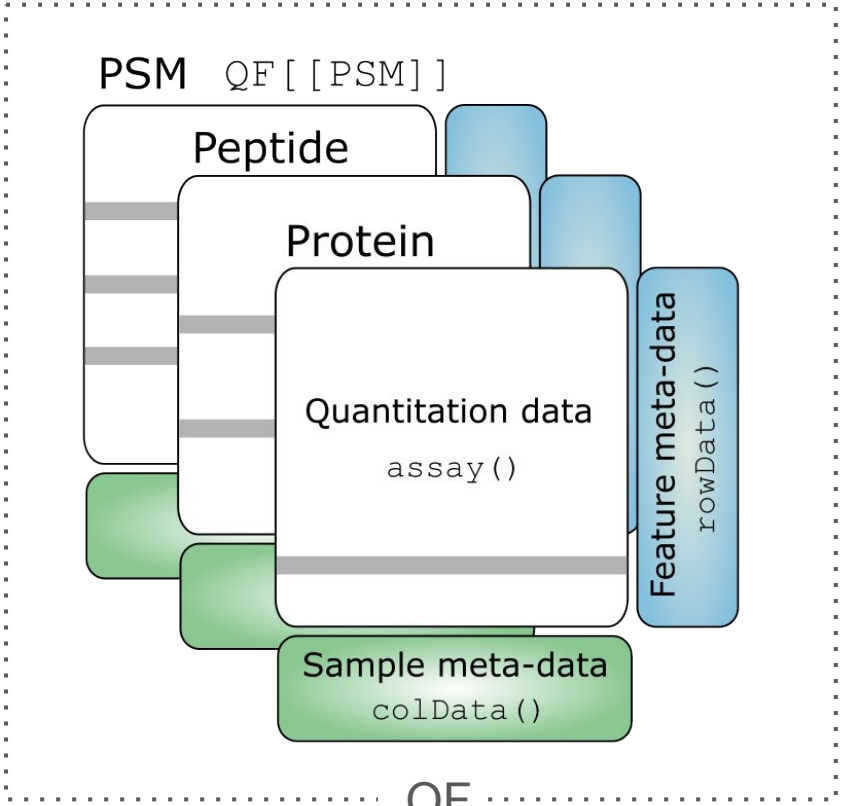
	Abundance 126	Abundance 127N	Abundance 127C	Abundance 128N	Abundance 128C	Abundance 129N	Abundance 129C	Abundance 130N	Abundance 130C	Abundance 131
	22.3	48.6	15.1	26.8	39.0	21.6	41.6	38.7	21.2	34.7
	19.8	31.3	14.1	19.5	32.1	29.2	37.1	41.2	30.0	34.9
	2.4	7.9	1.2	2.3	4.1	1.0	3.6	7.0	3.1	2.7
	2.2					2.3	3.1		1.7	
	2.2	4.2	2.2	1.6	5.7	1.4		3.6	4.0	4.0
		1.7								2.7
	20.7	35.9	7.1	25.0	36.1	22.5	33.3	32.3	20.9	33.2
	7.0	14.5	1.9	5.7	12.3	13.4	13.7	14.5	15.3	10.2
	8.6	14.9	2.9	10.4	10.7	10.8	20.0	11.8	8.7	12.6
	4.4	7.7		7.2	7.7	6.7	2.5	7.5	7.6	3.3
	17.7	31.2	6.5	11.3	16.2	23.9	31.8	35.0	19.7	34.9
	2.1	6.0	1.4	3.2	3.3	6.9	7.1	2.3	3.1	8.0

	Sequence	Annotated Sequence	...	Number of Proteins	Master Protein Accessions
	REEEMR	rEEEmR	...	2	P49755; Q15233
	QQNGTASSR	qQnGTASSR	...	1	Q92917
	CHMEENQR	cHmEENQR	...	1	Q96TC7
	QACQER	qAcQER	...	1	P26358
	HSEAATAQR	hSEAATAQR	...	1	Q14103
	QQQQQQQHQQPNR	qQQQQQQHQQPNR	...	1	Q6Y7W6
	TANREECR	tANREEcR	...	1	Q4G0J3
	QRQEEEAR	qRQEEEAR	...	2	P55036
	QEAAQSR	qEAAQSR	...	1	Q96A33
	AQANEQR	aQANEQR	...	1	O00291
	TSGDTNAR	tSGDTNAR	...	1	Q8N1F7
	QQQQQQQHQQPNR	qQQQQQQHQQPnR	...	1	Q6Y7W6

	sample	rep	condition	tag
Abundance.126	Control	NA	Pre-treatment	126
Abundance.127N	M_1	1	M	127N
Abundance.127C	M_2	2	M	127C
Abundance.128N	M_3	3	M	128N
Abundance.128C	G1_1	1	G1	128C
Abundance.129N	G1_2	2	G1	129N
Abundance.129C	G1_3	3	G1	129C
Abundance.130N	DS_1	1	Desynch	130N
Abundance.130C	DS_2	2	Desynch	130C
Abundance.131	DS_3	3	Desynch	131

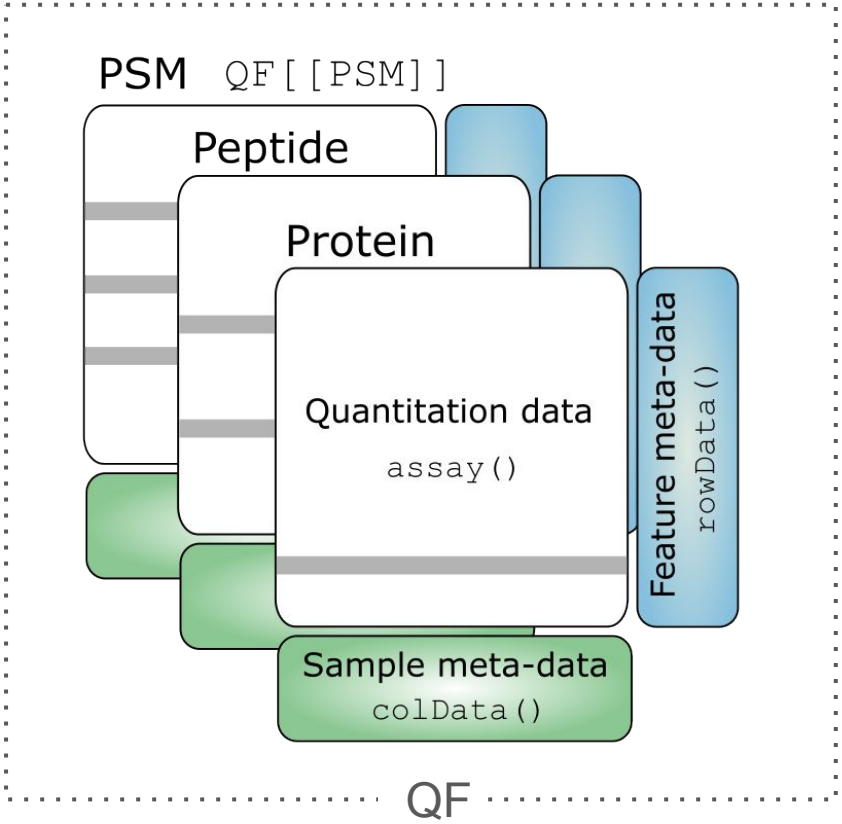
(2) The QFeatures class

- A list of SummarizedExperiments (SEs)
- Can store data across different levels (PSM, peptide and protein, …)
- Explicit links maintained between levels
- Index using R list nomenclature [[double brackets]]



Data import

Let’s open RStudio and get started!



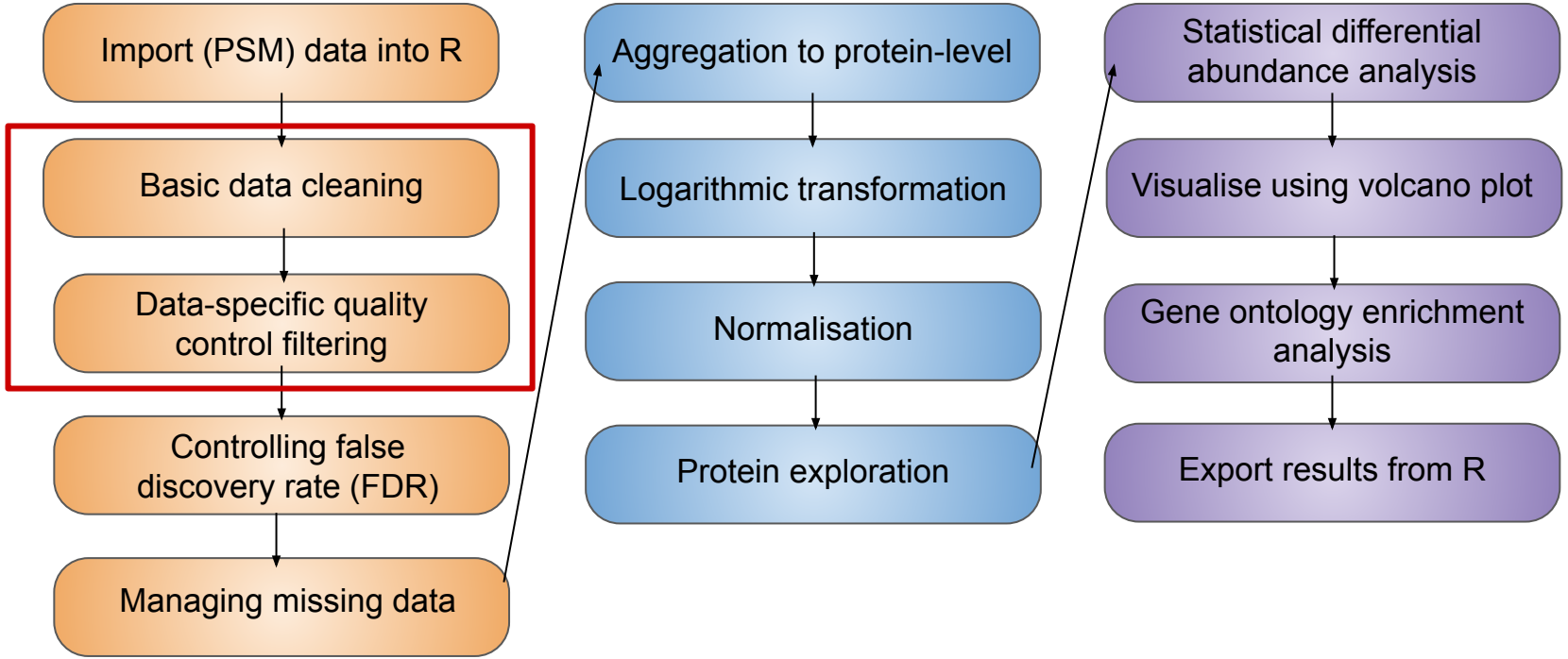
Cambridge Centre for Proteomics

UNIVERSITY OF CAMBRIDGE
Bioinformatics Training

Lesson 3: Data cleaning

December 2024

In this workshop…



Overview

1. Create a copy of the raw data

2. Basic data cleaning - common to most expression proteomics experiments

3. Data-specific quality control cleaning - tailored to the data set and experimental design

Which PSMs should we use?

Basic data cleaning: common to the majority of expression proteomics experiments

Does the PSM correspond to a protein of interest? Does the PSM have quantitative data?

Remove:

1. PSMs without a protein assignment
2. PSMs without quantitative data
3. PSMs from a contaminant protein

*Some software may output decoy PSMs - these need to be removed too

Are we confident that the PSM is associated with the correct peptide and protein sequence?

Remove:

1. PSMs that are not rank 1
2. PSMs that are not unambiguous
3. PSMs that are not unique

*False discovery rate also used to ensure that we only keep confident identifications. We will do this later at the protein level.

Which PSMs should we use?

Data specific filtering: quality control steps with vary and are data-dependent

Is the raw mass spectrum of high quality? Can we be confident in the quantification we derive from it?

TMT data processed with Proteome Discoverer:

1. Co-isolation interference (%)
2. Reporter ion signal-to-noise ratio
3. Synchronous precursor selection mass matches (SPS-MM %)

- Many quality control parameters
- Differ depending on the quantification method and software used for database search
- Need to look at the available parameters and see which thresholds are appropriate for individual dataset
- Trade-off between quality and quantity

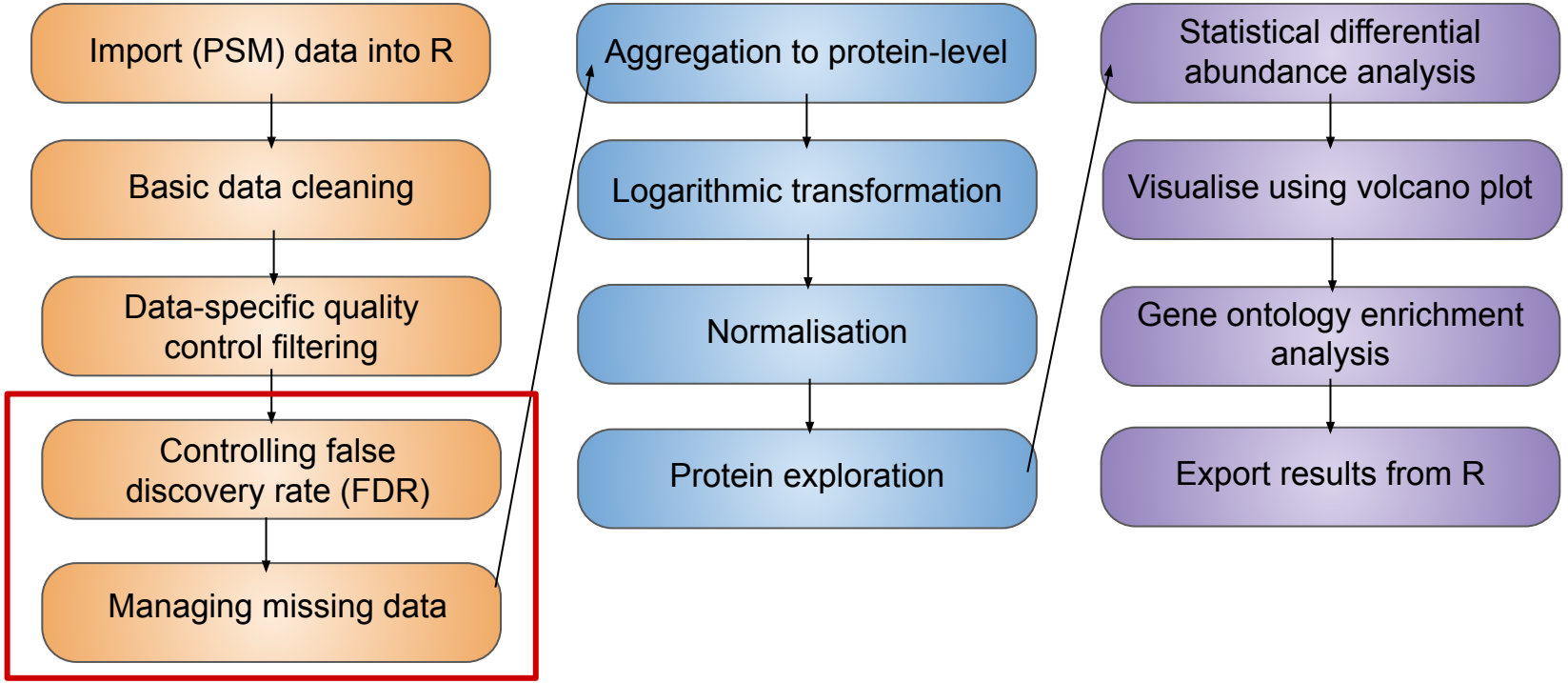
Cambridge Centre for Proteomics

UNIVERSITY OF CAMBRIDGE
Bioinformatics Training

Lesson 3: Data cleaning

December 2024

In this workshop…



Overview

1. The target-decoy approach to false discovery rate (FDR) control
2. Why and how we control the false discovery rate (FDR) at protein level
3. Exploring missing values in the data
4. Removing features (PSMs) with high proportion of missing values
5. Potential use of imputation where absolutely necessary

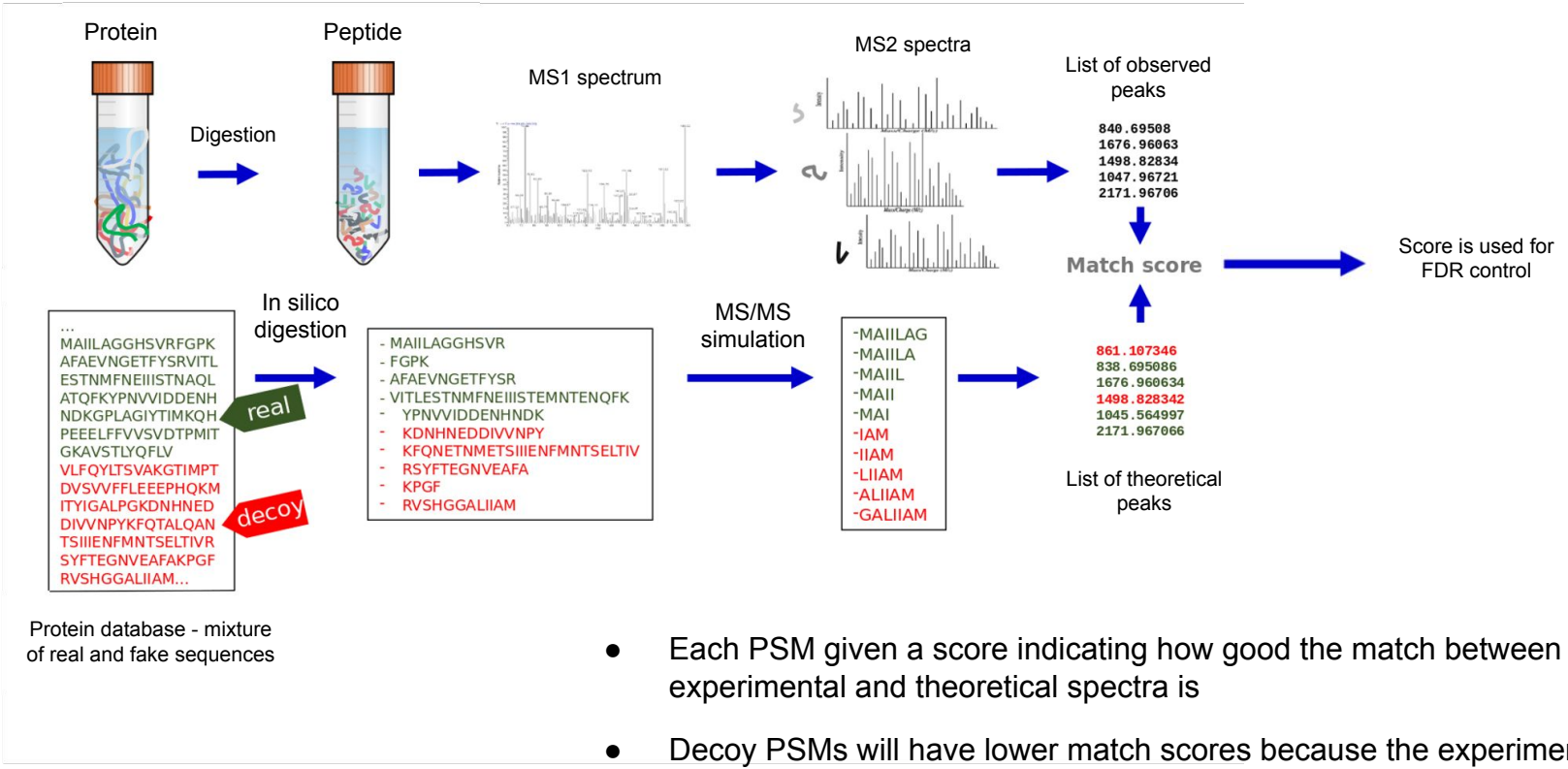
False Discovery Rate (FDR)

False discovery rate (FDR) = the proportion of false positive identifications in the data

- Each mass spectra can have multiple PSMs
- Only a few PSMs are a correct match between the raw spectra and peptide sequence
- Need to control for false positives

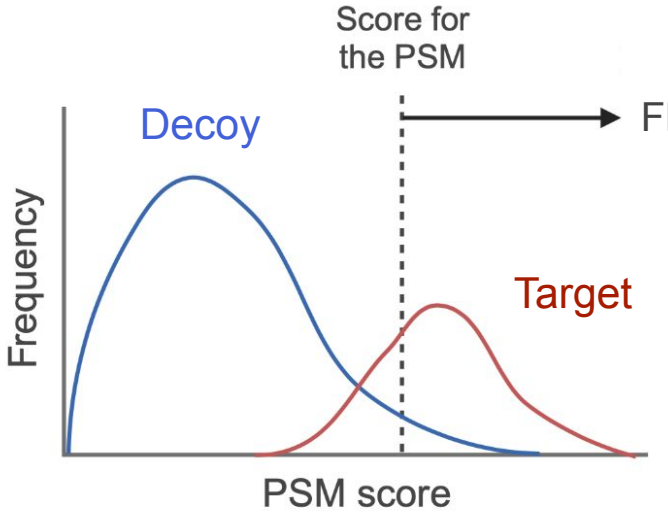
	Peptide sequence matches	Peptide sequence does not match
PSM reported	True positive	False positive
PSM not reported	False negative	True negative

FDR control via target-decoy search



- Each PSM given a score indicating how good the match between experimental and theoretical spectra is
- Decoy PSMs will have lower match scores because the experimental spectra for decoy peptides do not exist

PSM FDR



FDR = (2 x Decoy) / (Target + Decoy)

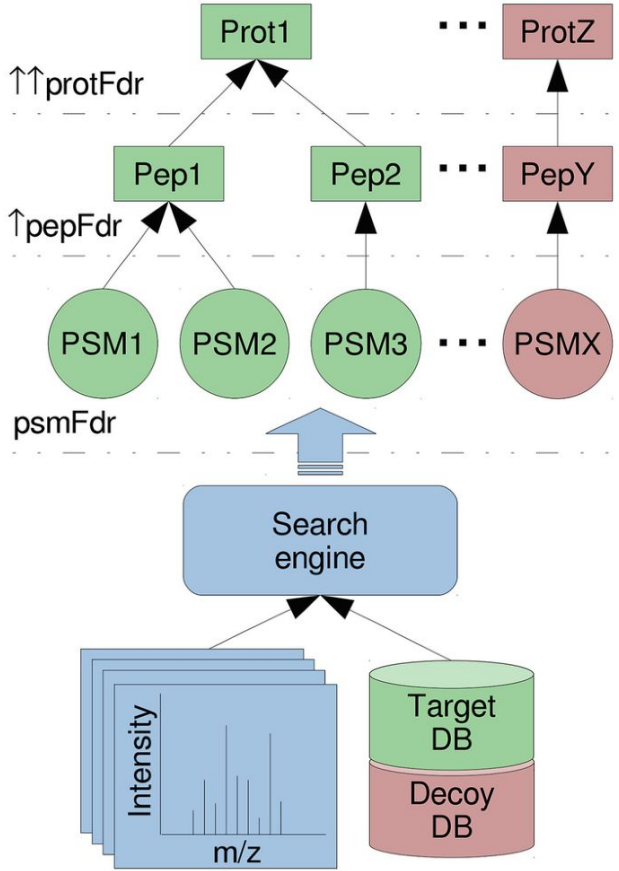
- During the database search, the rank 1 PSMs are used to calculate the FDR for a given PSM score threshold
    - PSM to decoy peptides must be false

- Keeping only PSMs with an FDR < 0.01 means that 1% of the remaining PSMs are false positives

Protein FDR

Why can't we just filter on PSM FDR?

- We infer proteins based on multiple peptide sequences supported by multiple PSMs
- FDR is amplified as we move upward from PSM to peptide to protein as TP PSMs accumulate in TP peptides and proteins, whereas FP PSMs are randomly distributed
- Software should also provide peptide and protein level FDR values



Missing values

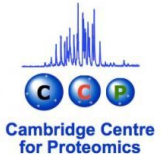
Not every PSM is quantified in every sample

AU	AV	AW	AX	AY	AZ	BA	BB	BC	BD
Abundance 126	Abundance 127N	Abundance 127C	Abundance 128N	Abundance 128C	Abundance 129N	Abundance 129C	Abundance 130N	Abundance 130C	Abundance 131
22.3	48.6	15.1	26.8	39	21.6	41.6	38.7	21.2	34.7
19.8	31.3	14.1	19.5	32.1	29.2	37.1	41.2	30	34.9
2.4	7.9	1.2	2.3	4.1	1	3.6	7	3.1	2.7
2.2					2.3	3.1		1.7	
2.2	4.2	2.2	1.6	5.7	1.4		3.6	4	4
	1.7								2.7
20.7	35.9	7.1	25	36.1	22.5	33.3	32.3	20.9	33.2
7	14.5	1.9	5.7	12.3	13.4	13.7	14.5	15.3	10.2
8.6	14.9	2.9	10.4	10.7	10.8	20	11.8	8.7	12.6
4.4	7.7		7.2	7.7	6.7	2.5	7.5	7.6	3.3
17.7	31.2	6.5	11.3	16.2	23.9	31.8	35	19.7	34.9
2.1	6	1.4	3.2	3.3	6.9	7.1	2.3	3.1	8
	2.8						1.5		

- Experimental designs vary in number of missing quant values
- Missing values are denoted differently - empty, NA, 0, inf
- QFeatures denotes missing values as NA

Is the amount and pattern of missing data what we expect? How do we remove or replace missing values?

1. Explore the presence of missing values
2. Remove PSMs with excessive missing values
3. Consider whether imputation is necessary

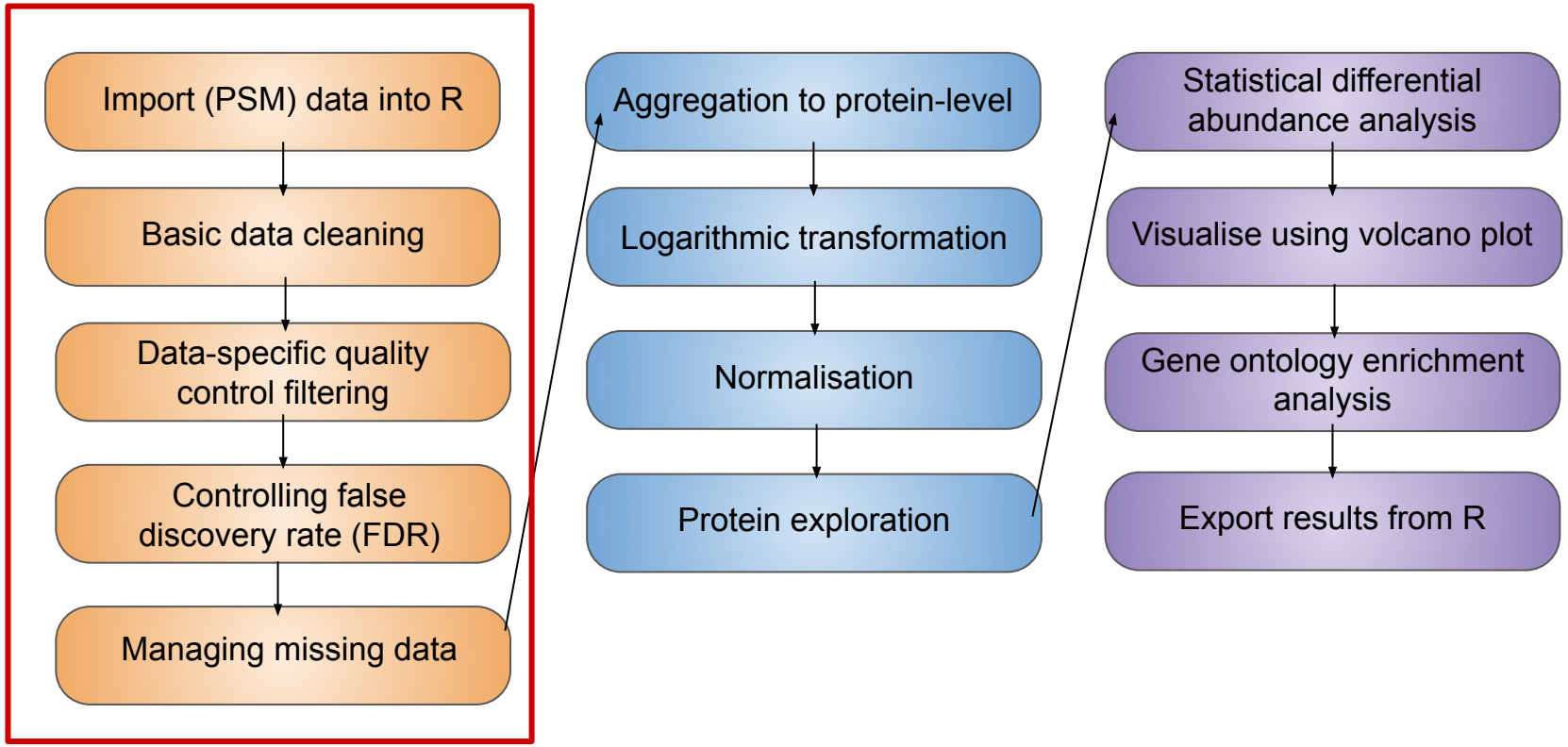


UNIVERSITY OF CAMBRIDGE
Bioinformatics Training

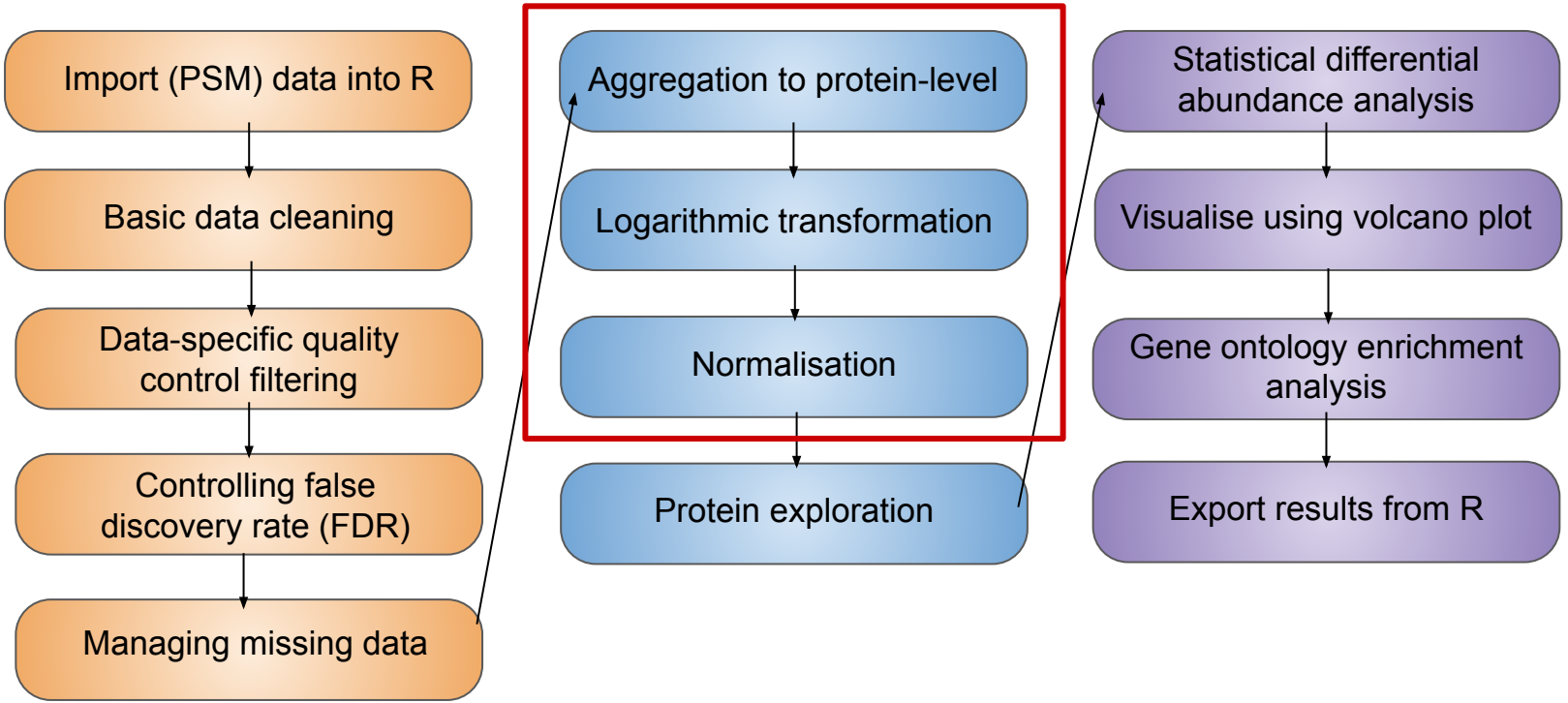
Lesson 4: Data transformation aggregation and normalisation

December 2024

Recap


Import (PSM) data into R
Basic data cleaning
Data-specific quality control filtering
Controlling false discovery rate (FDR)
Managing missing data
Aggregation to protein-level
Logarithmic transformation
Normalisation
Protein exploration
Statistical differential abundance analysis
Visualise using volcano plot
Gene ontology enrichment analysis
Export results from R

Next steps


Import (PSM) data into R
Basic data cleaning
Data-specific quality control filtering
Controlling false discovery rate (FDR)
Managing missing data
Aggregation to protein-level
Logarithmic transformation
Normalisation
Protein exploration
Statistical differential abundance analysis
Visualise using volcano plot
Gene ontology enrichment analysis
Export results from R

Lesson overview

1. Aggregation of PSMs to peptides and then to proteins

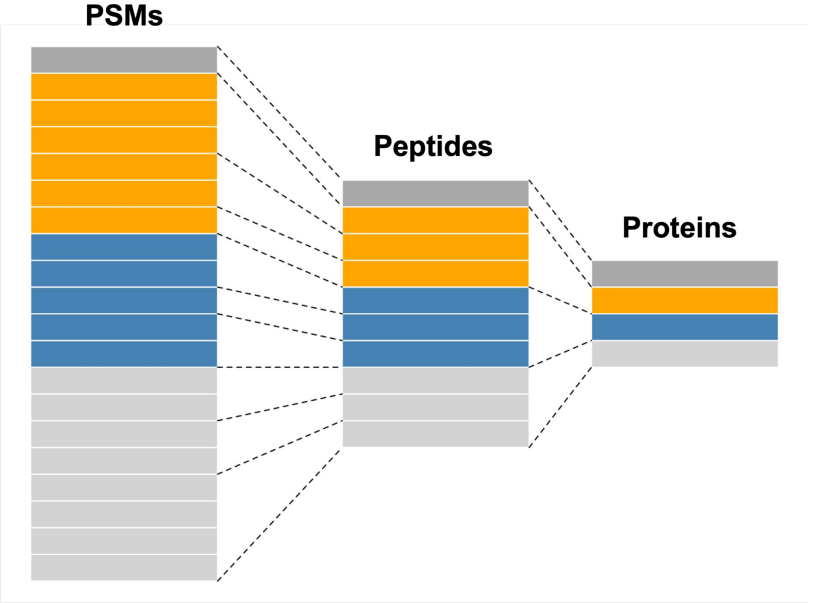
2. log2 transformation of quantitative data to generate normal (Gaussian) distribution

3. Normalisation methods to remove non-biological variation

Aggregation

1. **Summarisation** or **aggregation** is the process of summarising our PSM/peptide level intensities upwards to protein level

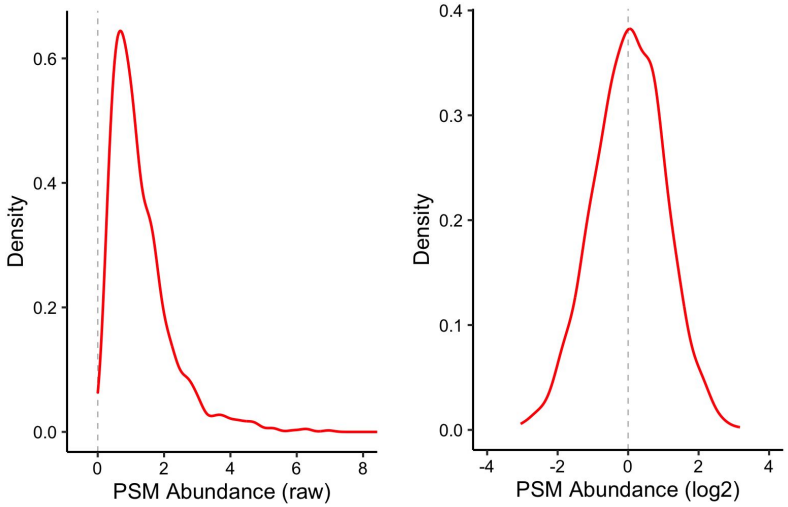
2. Many methods, e.g:
   a. Classic metrics e.g. median, sum
   b. Robust summarisation (Sticker et al. 2020) - uses a linear model to find the mean whilst correcting for different peptides with different characteristics



Data aggregation (summarisation) in bottom up proteomics

Data transformation

- PSM/peptide/protein abundance values have a skewed distribution
- To apply hypothesis testing we need a normal distribution so we log transform our data

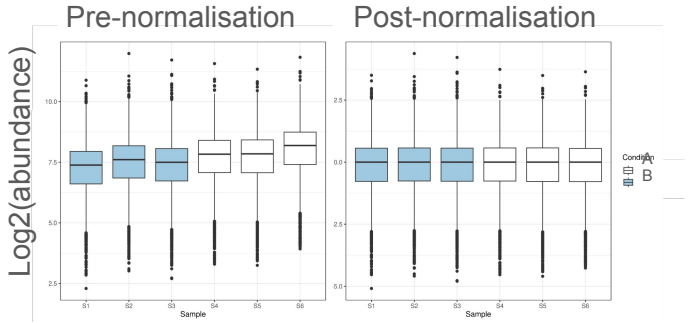


We typically use base 2 as it makes interpretation and visualisation easier as any protein that halves in abundance between conditions will have a 0.5 fold change, which translates into a log2 fold change of -1. Any protein that doubles in abundance will have a fold change of 2 and a log2 fold change of +1.

Normalisation

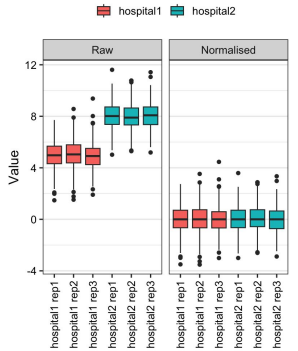
Normalisation = transforming data to:

1. Remove/minimise random, non-relevant variation
2. Retain biologically-relevant variation, unchanged.



Take care!

- Many methods exist. The most appropriate depends on the experimental design and proteomics techniques used.
- ***Normalisation can have a profound impact on the data.*** Make sure to check that assumptions are reasonable.



Top: normalisation looks reasonable

Bottom: condition-specific so normalisation across conditions may not be suitable

Where does variability come from?

Sources of variability

Biological: Differences between samples

- Within a condition: Not all cells/individuals/etc. are the same
- Between conditions: The ultimate goal of the research

Technical: Sample preparation

- Protocols and operator

Systematic: Bias in processing

- Instruments, reagents, settings

Minimising technical variability

A good experimental design considers:

Replication

- Within an experiment: Enables estimation of biological variance
- Between experiments: Any true finding should be reproducible

Comparison/control/baseline

- Need a reference value to compare to

Blocking

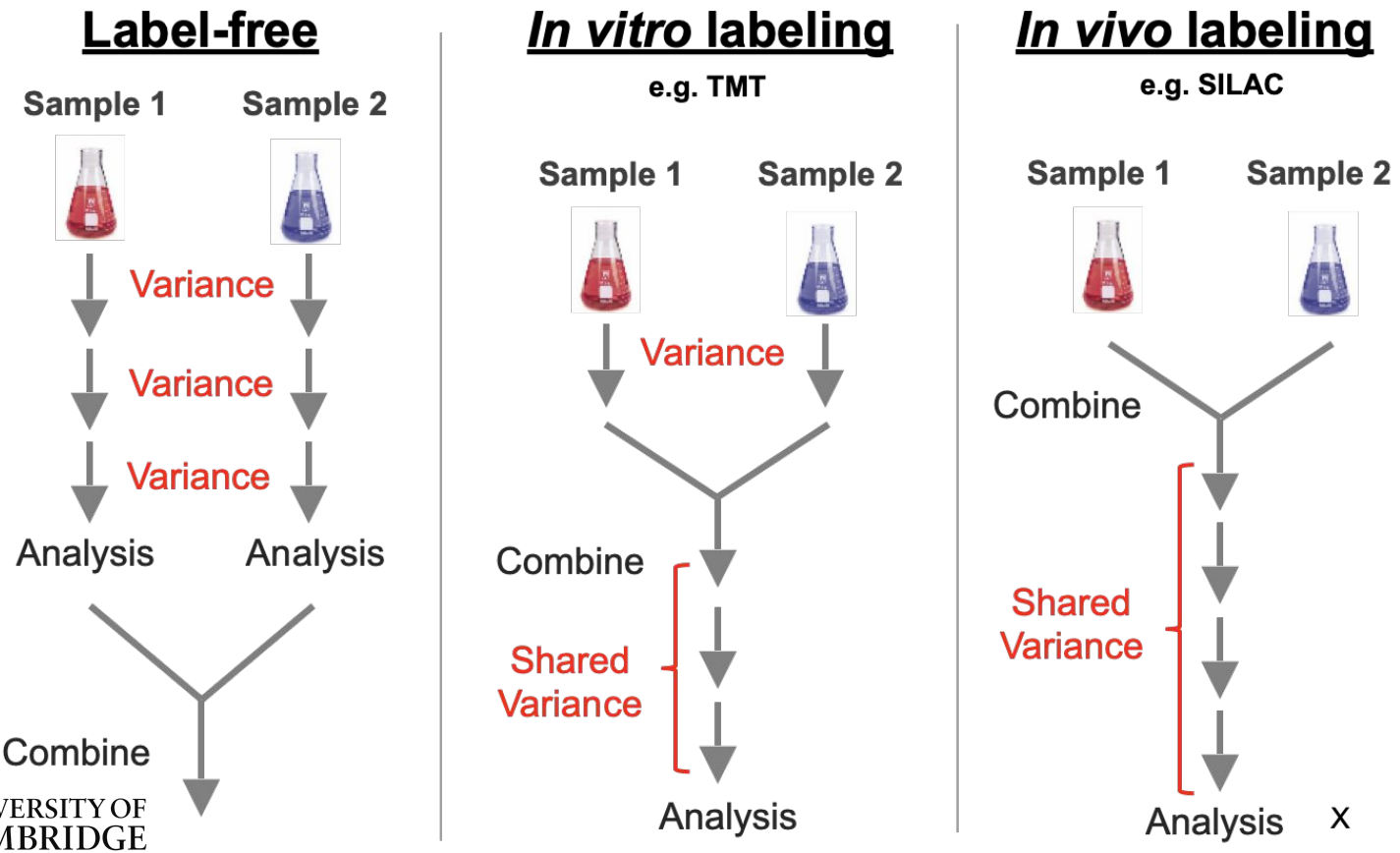
- “Remove” the contribution of nuisance factors

Randomisation

- Avoid adding systematic biases

Cambridge Centre for Proteomics

Technical variance depends on quantification method



Common normalisation methods

Ratiometric normalisation:

- Normalise to internal standards

Median normalisation (median “sweep”):

- Shift the distributions so the median values are the same

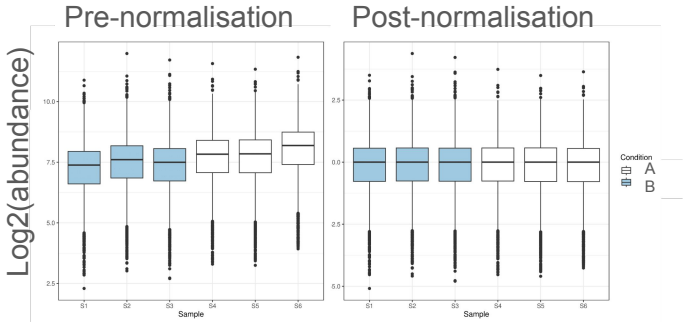
Quantile normalisation:

- Force every sample to have the exact same distribution of values

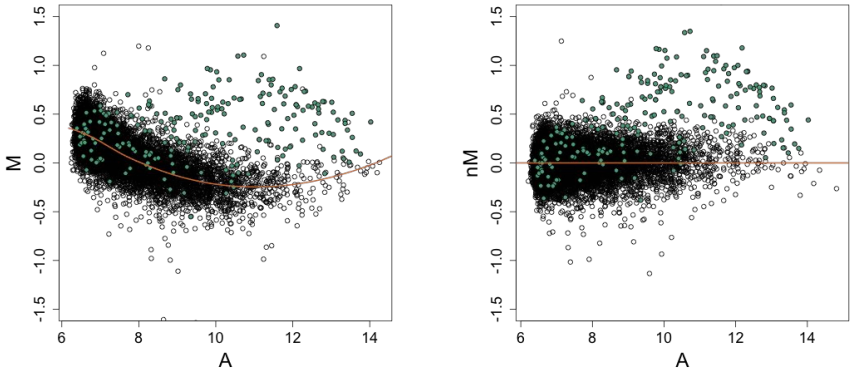
Variance stabilisation normalisation (VSN):

- Removes relationship between abundance and variance

Median



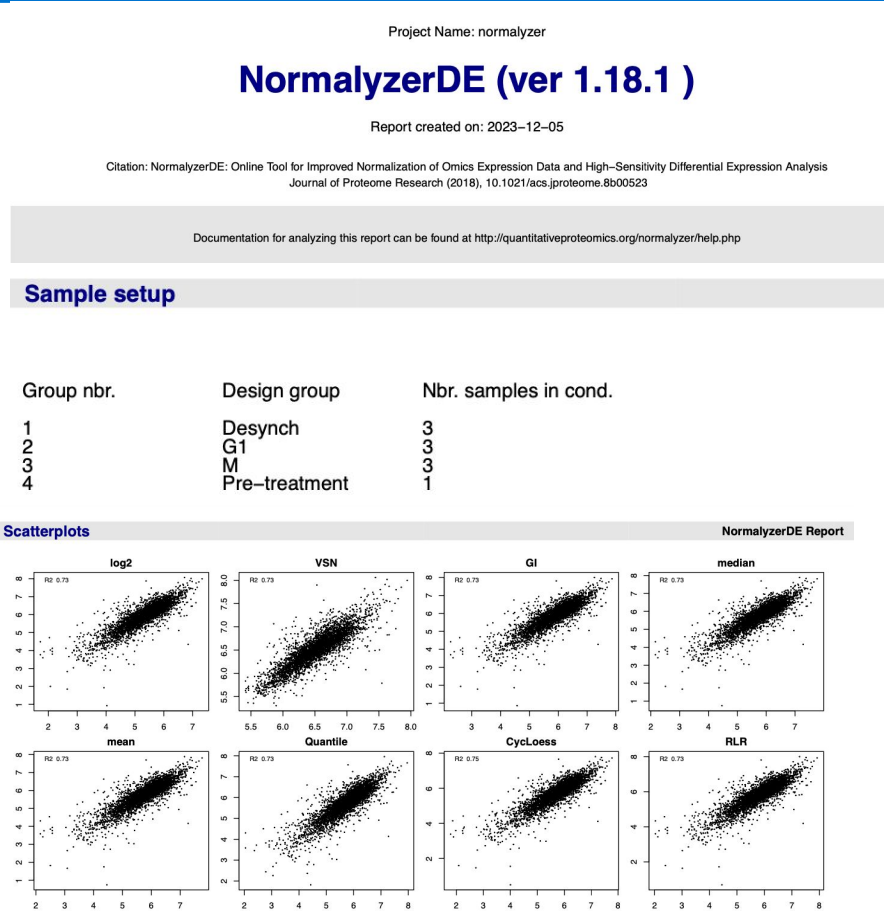
VSN



NormalyzerDE

- NormalyzerDE is an R/Bioconductor is a tool for screening of normalization methods for quantitative proteomics data
- It calculates a range of normalized matrices and performance measures
- Generates a PDF evaluation report.

Extended documentation
https://normalyzerde.immunoprot.lth.se/var/www/Flask/NormalyzerDEServer/static/Normalyzer-documentation-20140422.pdf



Project Name: normalyzer

NormalyzerDE (ver 1.18.1)

Report created on: 2023–12–05

Citation: NormalyzerDE: Online Tool for Improved Normalization of Omics Expression Data and High–Sensitivity Differential Expression Analysis
Journal of Proteome Research (2018), 10.1021/acs.jproteome.8b00523

Documentation for analyzing this report can be found at http://quantitativeproteomics.org/normalyzer/help.php

Sample setup

Group nbr.	Design group	Nbr. samples in cond.
1	Desynch	3
2	G1	3
3	M	3
4	Pre–treatment	1

Scatterplots NormalyzerDE Report

Cambridge Centre for Proteomics
UNIVERSITY OF CAMBRIDGE
Bioinformatics Training

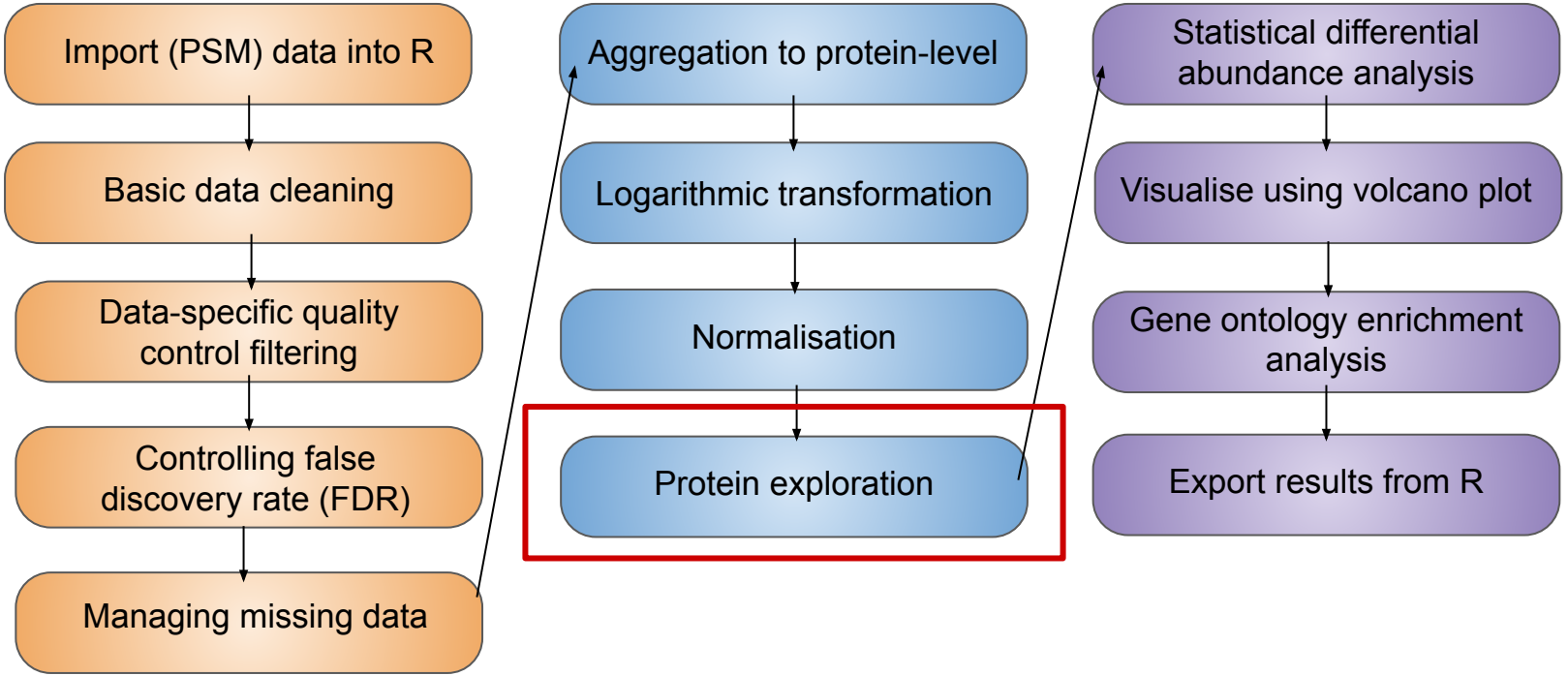
Cambridge Centre for Proteomics

UNIVERSITY OF CAMBRIDGE
Bioinformatics Training

Lesson 5: Exploration of protein data

December 2024

In this workshop…



Overview

1. Report the dimensions of a SummarizedExperiment within a QFeatures object to count final PSMs, peptides and proteins

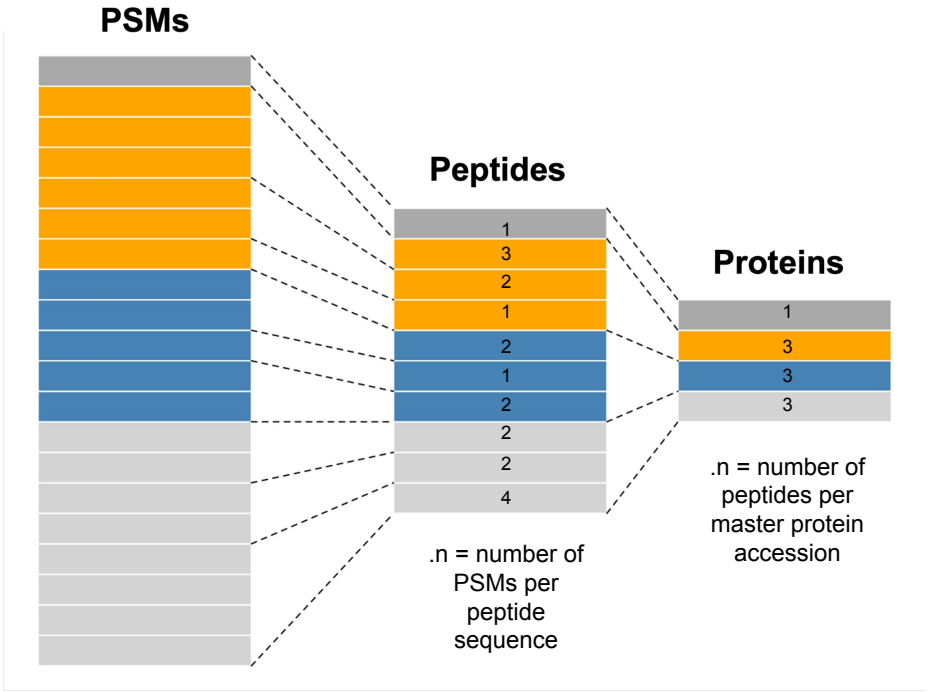
2. Be aware of the `.n` column generated by aggregateFeatures() and what this means in different contexts

3. Subset data for individual features across all data levels

4. Complete principal component analysis (PCA) to explore patterns, clusters and outliers in the data

The .n column

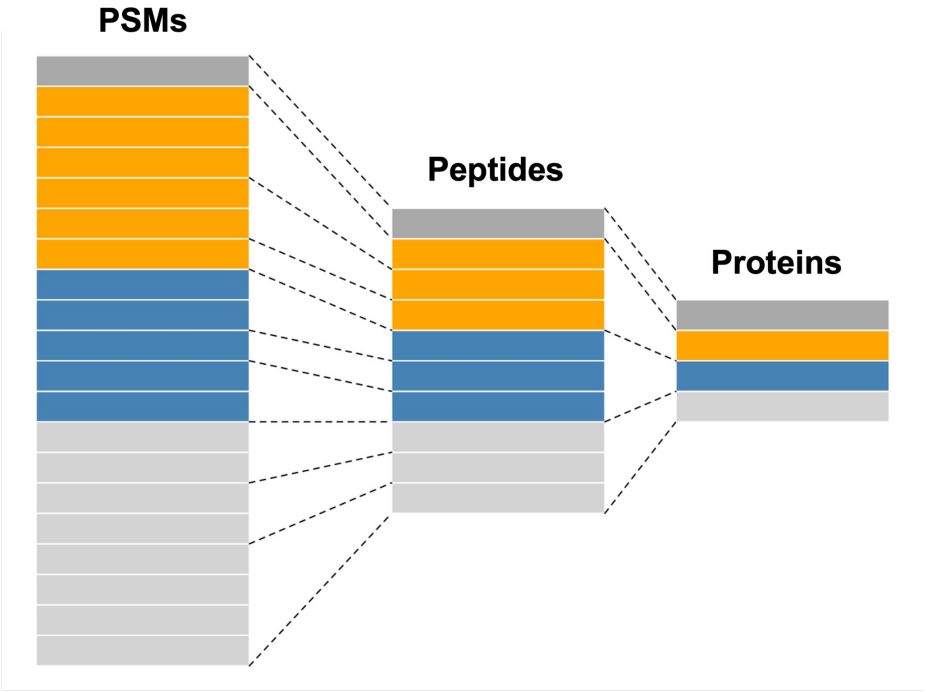
- Column created by default using aggregateFeatures() function
- .n represents the number of child features aggregated into each parent feature
- What these features are depends on what we are aggregating to and from



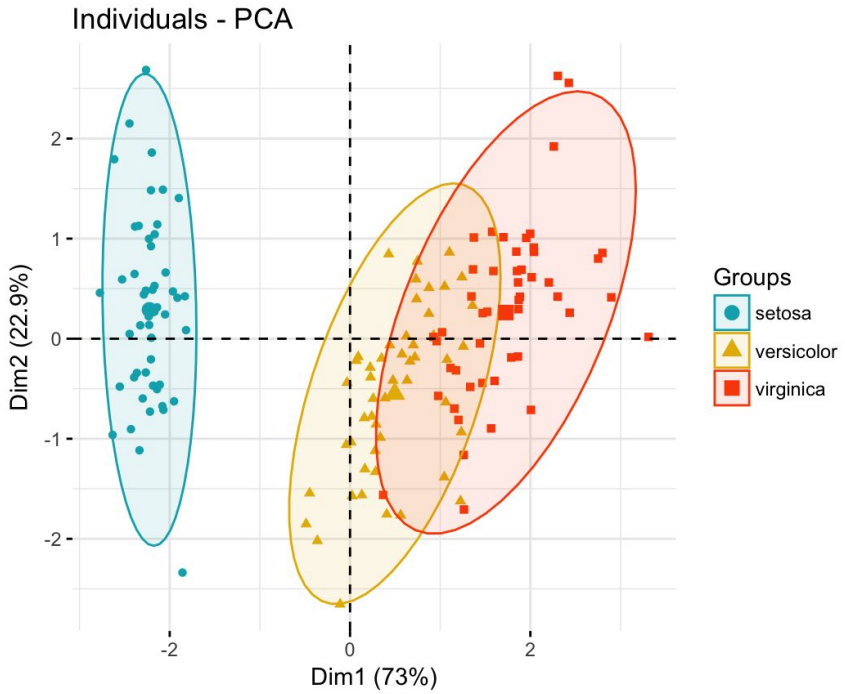
.n column has a different meaning across different experimental assays

Subset by Feature

- Easy to access all levels of information corresponding to a feature of interest
- Facilitated by the explicit links generated in a QFeatures object using aggregateFeatures
- Subset this information using subsetByFeature() function



Principal Component Analysis (PCA)

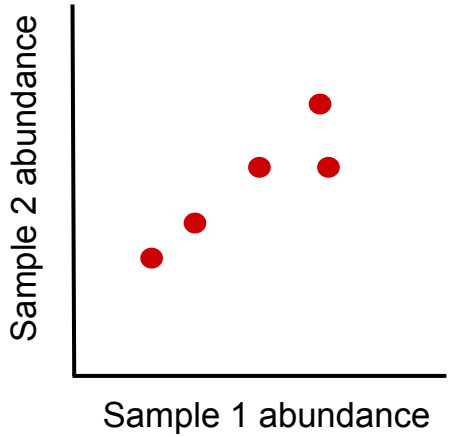


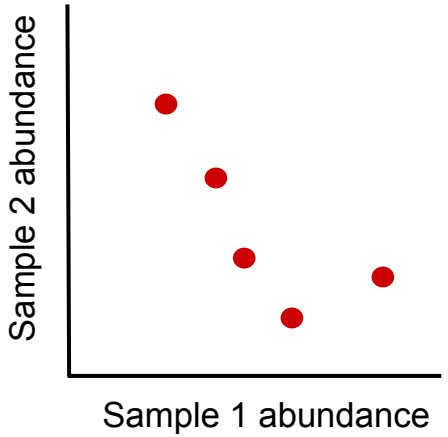
- Dimensionality reduction method to facilitate visualisation
- Use to check for data patterns and clustering as well as outlier samples
- Particularly important for experiments with multiple variables

Principal Component Analysis (PCA)

	Sample 1	Sample 2
Protein A	x	x
Protein B	x	x
Protein C	x	x
Protein D	x	x
Protein E	x	x

It is simple to visualise sample relationships in 2D data



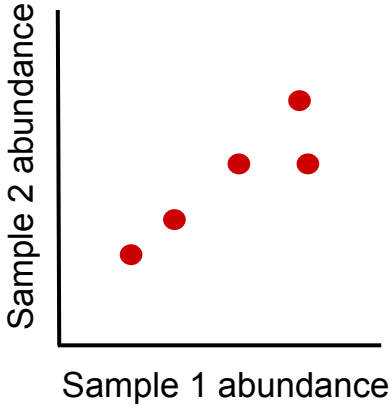


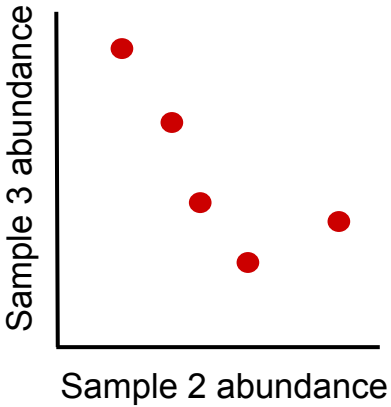
Principal Component Analysis (PCA)

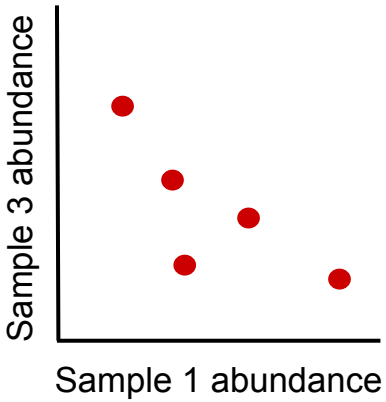
	Sample 1	Sample 2	Sample 3
Protein A	x	x	x
Protein B	x	x	x
Protein C	x	x	x
Protein D	x	x	x
Protein E	x	x	x

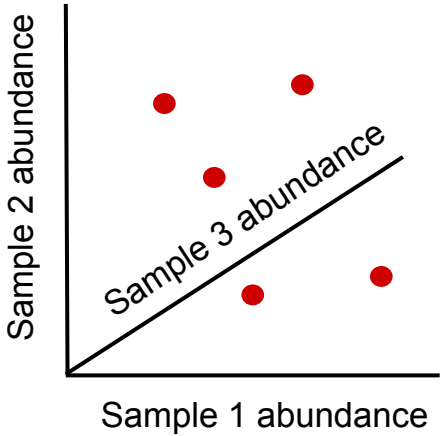
We can still visualise 3D data, but it's slightly more challenging

When we get to 4D and above these plots are of limited use

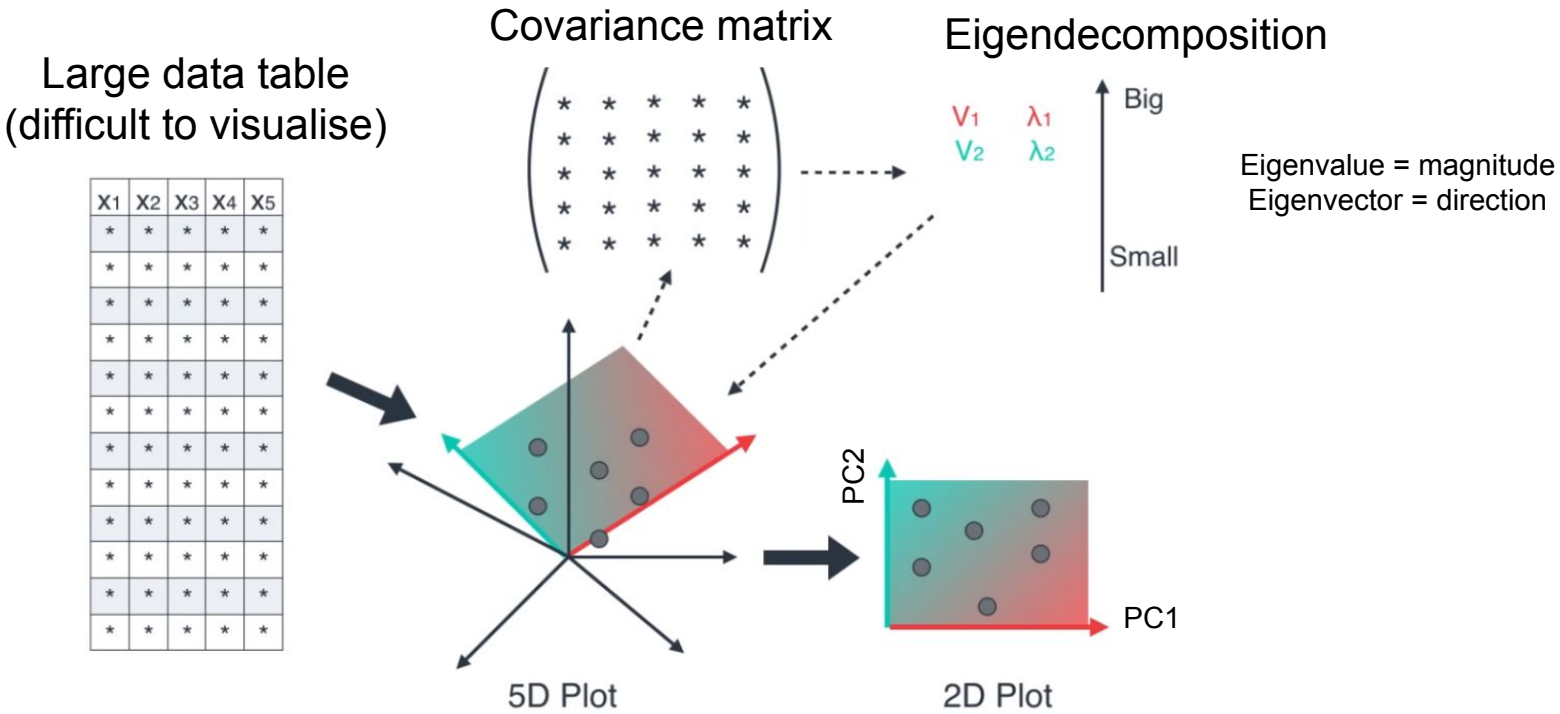


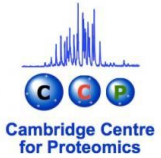






Principal Component Analysis (PCA)



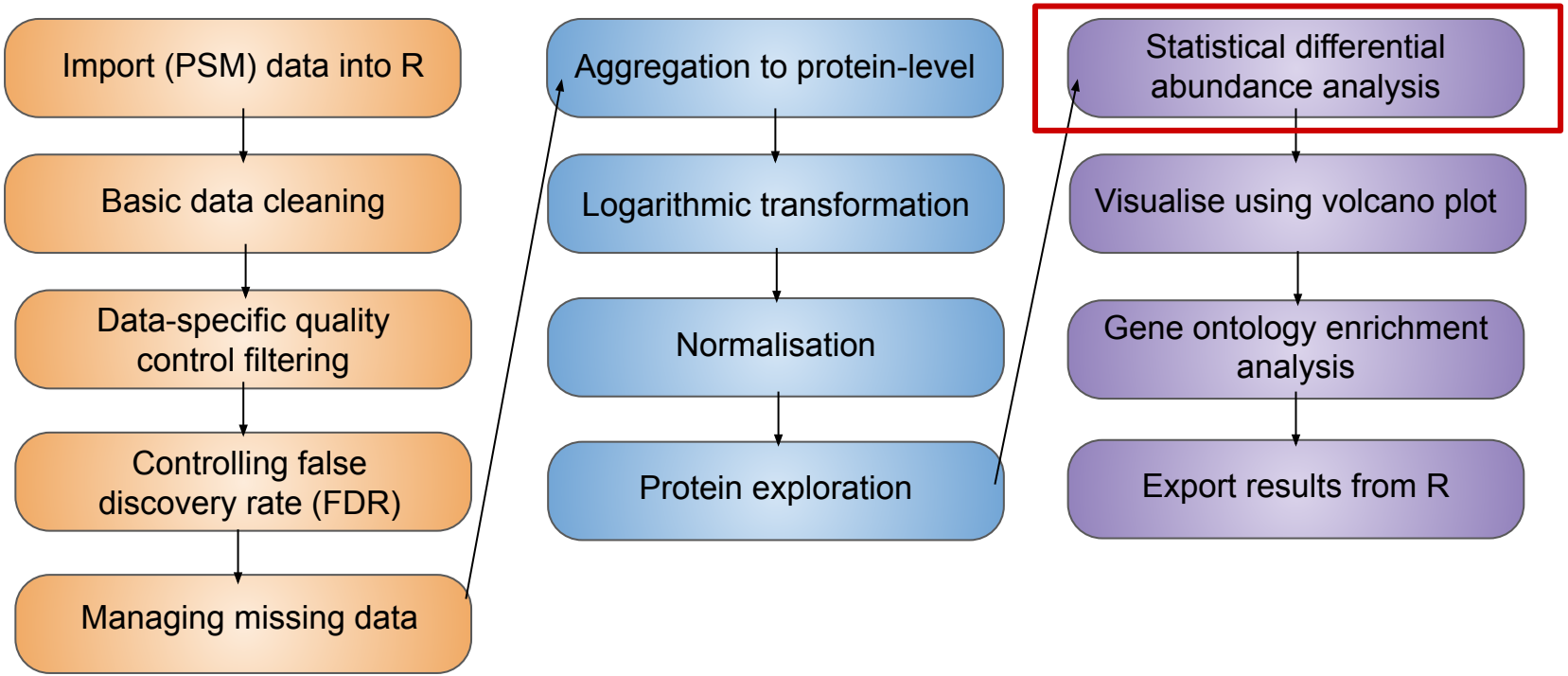


UNIVERSITY OF CAMBRIDGE
Bioinformatics Training

Lesson 6: Statistical differential expression analysis

December 2024

In this workshop…



Aims

- Model our protein abundance data using linear models

- Use these models to examine the relationship between protein abundance and cell cycle

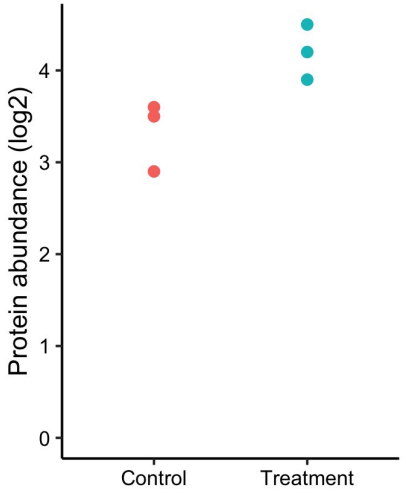
Overview

1. Defining and applying a linear model for statistical testing
2. Limma and the sharing of information between proteins
3. Model diagnostics
4. Interpreting the output of the model

Example protein abundance data

- Treatment vs control experiment:
  - 2 conditions × 3 replicates = 6 samples (columns)
- Modelling will be performed on each protein separately
  - Here, we consider a single protein for simplicity

```
> protein_abundances
  Control_1   Control_2   Control_3 Treatment_1 Treatment_2 Treatment_3
        3.5         3.6         2.9         4.5         3.9         4.2
```



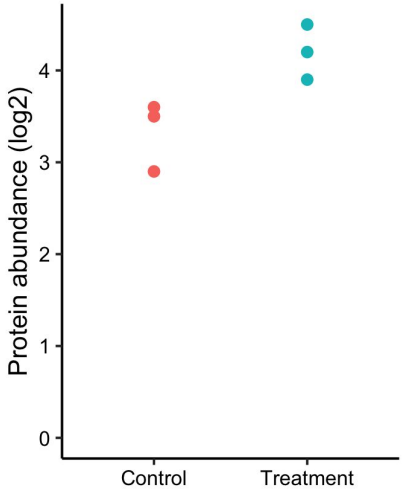
Linear model assumptions

Explanatory variables (features; conditions):

- *Linearity* = Linear relationship between conditions and abundance
- *No multicollinearity* = conditions are not correlated

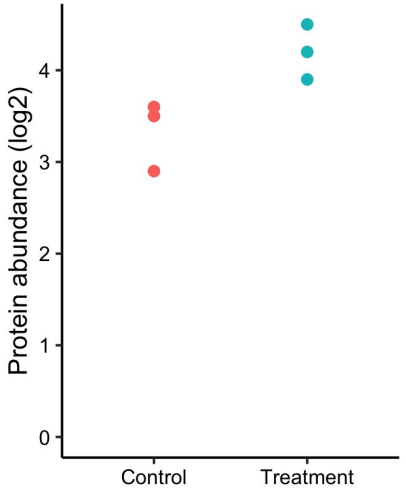
Error terms (residuals):

- *Homoscedasticity* = Equal variance
- *No autocorrelation* = Errors are independent
- *Gaussian distribution* - Will hold true if log-transformed



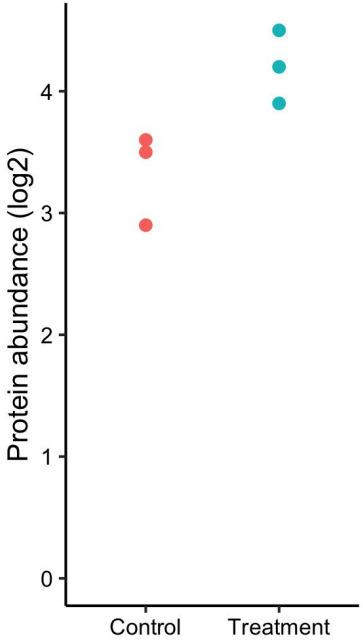
Linear modelling of protein abundances

- We want to compare the mean protein abundances between the treatments
- We can define our model using `model.matrix`

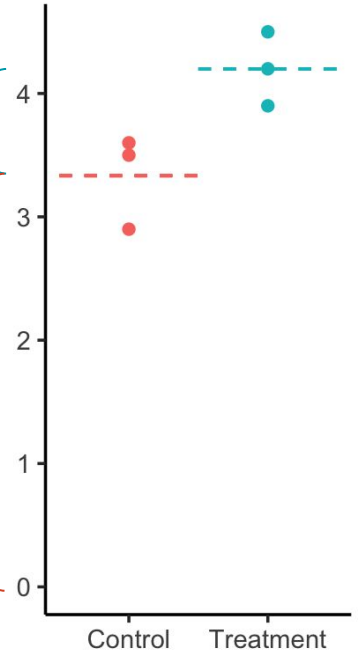



```
> protein_abundances
  Control_1   Control_2   Control_3 Treatment_1 Treatment_2 Treatment_3
        3.5         3.6         2.9         4.5         3.9         4.2
> condition
[1] "Control"   "Control"   "Control"
[4] "Treatment" "Treatment" "Treatment"
> model.matrix(~condition)
  (Intercept) conditionTreatment
1           1                   0
2           1                   0
3           1                   0
4           1                   1
5           1                   1
6           1                   1
```

Fitting our model with `lm`



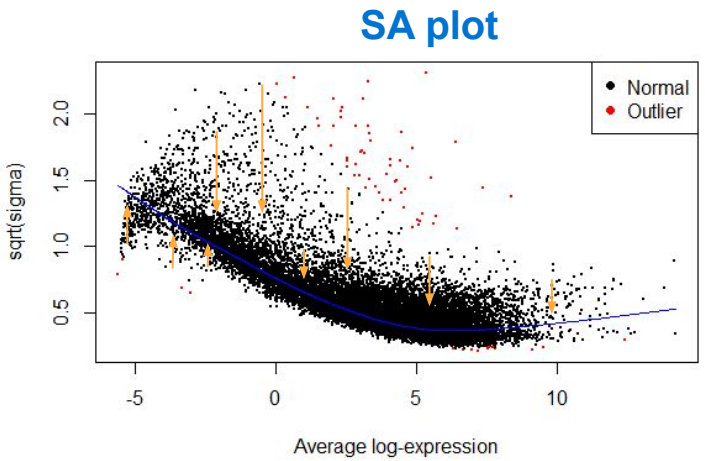

```
> lm_fit <- lm(protein_abundances ~ condition)
> summary(lm_fit)

Call:
lm(formula = protein_abundances ~ condition)

Residuals:
  Control_1   Control_2   Control_3 Treatment_1 Treatment_2 Treatment_3
  1.667e-01   2.667e-01  -4.333e-01   3.000e-01  -3.000e-01   1.142e-16

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)          3.3333     0.1972  16.903 7.18e-05 ***
conditionTreatment   0.8667     0.2789   3.108    0.036 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3416 on 4 degrees of freedom
Multiple R-squared:  0.7071,    Adjusted R-squared:  0.6339
F-statistic: 9.657 on 1 and 4 DF,  p-value: 0.03596
```



Limma - sharing information between proteins

- High-throughput experiments are often lowly replicated
  - n=3 is common in proteomics
  - Estimates of variance are likely to be inaccurate.
  - May be over or under-estimated, leading to False Positives (FP) or False Negatives (FN) in statistical testing

- Assuming variance is similar between all proteins and partly attributable to abundance, we can ‘moderate’ observed errors in the linear model towards a more likely value
  - Reduces FP and FN



Model diagnostics

- We can't determine whether the modeling is appropriate for each protein separately since we only have minimal replicates to draw conclusions from
- We can consider all proteins together to assess model suitability
- Where plots do not look as expected, may indicate inappropriate data processing or model specification

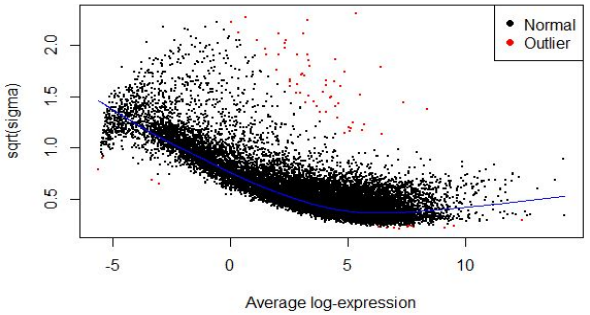
SA Plot

- Expect to see a relationship between abundance and residual standard deviation [sqrt(sigma)]

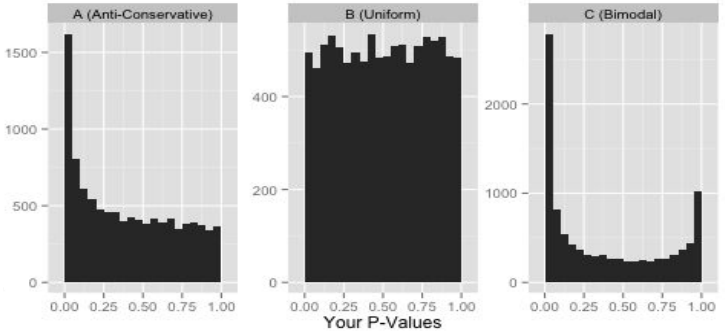
P-value histogram

- For proteins with no change in abundance, expect p-value to be drawn from uniform distribution [0-1]
- For proteins with a change in abundance, expect low p-values
- Expect p-value distribution to be a combination of the above

SA plot



Histogram of raw p-values



http://varianceexplained.org/statistics/interpreting-pvalue-histogram/

limma results

logFC: log2 fold-change
AveExpr: Average expression
t: Moderated t-statistic
P.value: Raw p-value
adj.P.Value: Adjusted p-value
B: Log-odds of differential abundance

	logFC	AveExpr	t	P.Value	adj.P.Val	B
Q9BW19	-2.273445	-0.48781151	-31.31920	3.266414e-10	6.815267e-07	14.02110
Q9NQW6	-2.049413	-0.16777937	-30.27752	4.367177e-10	6.815267e-07	13.77369
Q9ULW0	-2.163283	0.21676121	-29.57100	5.348104e-10	6.815267e-07	13.59880
P49454	-1.806194	-0.09474064	-28.54660	7.236533e-10	6.916316e-07	13.33444
O14965	-2.029577	-0.20204374	-25.55178	1.869711e-09	1.110705e-06	12.48016
P53350	-1.520595	0.06838699	-25.00830	2.247106e-09	1.110705e-06	12.31069

Overview

1. Define and apply a statistical model using the Limma package
2. Be aware of empirical Bayes moderation
3. Carry out model diagnostics
4. Interpret the output of a multi-contrast differential expression statistical model
5. Plot the results of a differential expression analysis

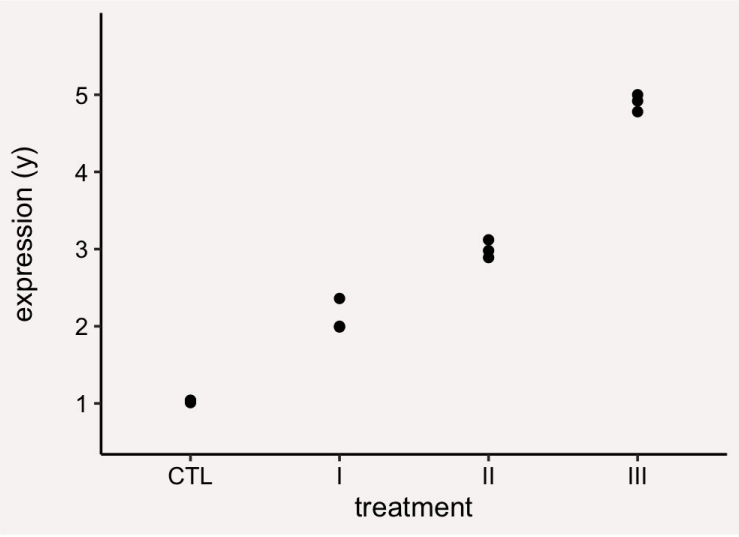
Some fake data (similar to ours)

- 4 treatments × 3 replicates = 12 samples (columns)
- Treatments: CTL, I, II, III

```
> fake_qf[["log_norm_proteins"]] %>% assay() %>% head(2)
  CTL_1 CTL_2 CTL_3  I_1  I_2  I_3 II_1 II_2 II_3 III_1 III_2 III_3
y  1.01  1.04  1.04 1.99 2.36 2.00 2.89 3.12 2.98  5.00  4.92  4.78
w  0.99  1.02  1.01 2.13 2.28 2.07 2.65 2.77 2.96  4.88  5.02  4.94
```

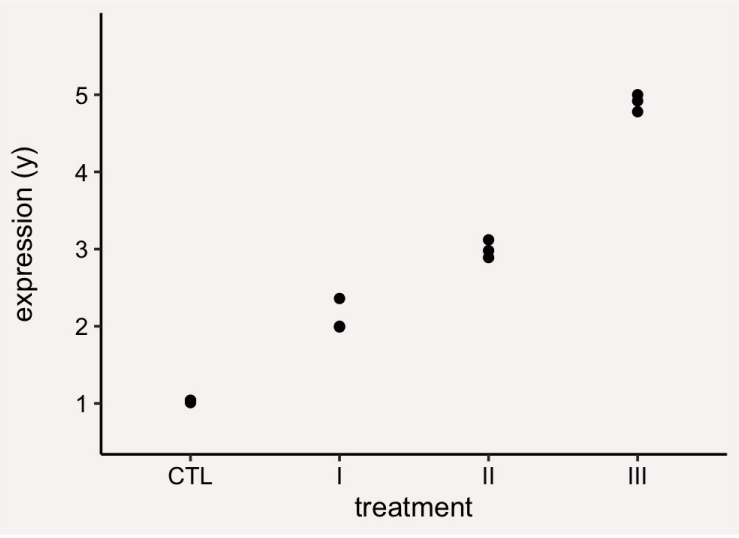
Plot protein ‘y’ abundances

	CTL_1	CTL_2	CTL_3	I_1	I_2	I_3	II_1	II_2	II_3	III_1	III_2	III_3
y	1.01	1.04	1.04	1.99	2.36	2.00	2.89	3.12	2.98	5.00	4.92	4.78



Linear modelling of protein abundances

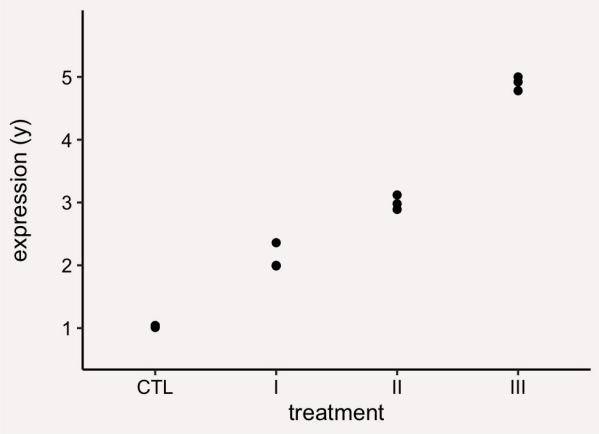
- We want to compare the mean protein abundances between the treatments
- We will model our protein abundances using a linear model (specifically a means model in this case)



Define our model using model.matrix

- How does each sample (column) correspond to the different treatments?

	CTL_1	CTL_2	CTL_3	I_1	I_2	I_3	II_1	II_2	II_3	III_1	III_2	III_3
y	1.01	1.04	1.04	1.99	2.36	2.00	2.89	3.12	2.98	5.00	4.92	4.78



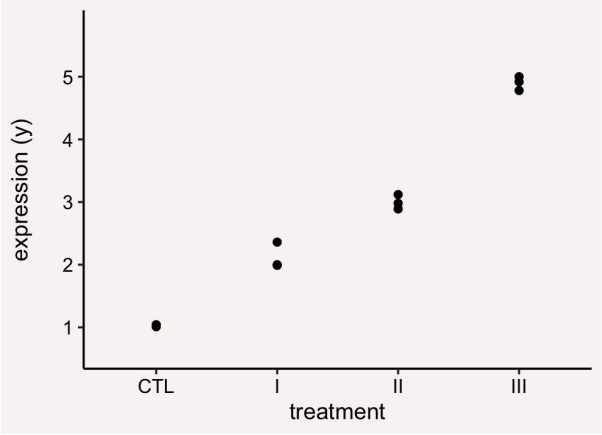
Design Matrix

```
> model.matrix(~0 + treatment)
   treatmentCTL treatmentI treatmentII treatmentIII
1             1          0           0            0
2             1          0           0            0
3             1          0           0            0
4             0          1           0            0
5             0          1           0            0
6             0          1           0            0
7             0          0           1            0
8             0          0           1            0
9             0          0           1            0
10            0          0           0            1
11            0          0           0            1
12            0          0           0            1
```

Law CW, Zeglinski K, Dong X et al. A guide to creating design matrices for gene expression experiments [version 1]. F1000Research 2020, 9:1444 (doi: 10.12688/f1000research.27893.1)

Fitting our model with lmFit

```
> fit_model <- lmFit(object = fake_data, design = m_design)
```



$$E(y) = 1.03x_0 + 2.12x_1 + 3.00x_2 + 4.90x_3$$

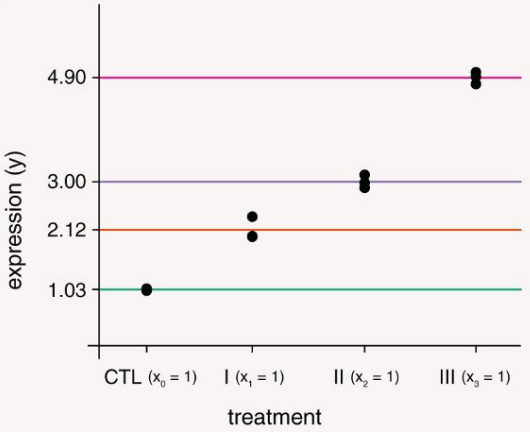
E(y)	=	1.03			=	1.03 (for control)
E(y)	=		2.12		=	2.12 (for treatment I)
E(y)	=			3.00	=	3.00 (for treatment II)
E(y)	=				4.90	= 4.90 (for treatment III)

Matrix

```
> model.matrix(~0 + treatment)
```

	treatmentCTL	treatmentI	treatmentII	treatmentIII
1	1	0	0	0
2	1	0	0	0
3	1	0	0	0
4	0	1	0	0
5	0	1	0	0
6	0	1	0	0
7	0	0	1	0
8	0	0	1	0
9	0	0	1	0
10	0	0	0	1
11	0	0	0	1
12	0	0	0	1

Plot



Law CW, Zeglinski K, Dong X et al. A guide to creating design matrices for gene expression experiments [version 1]. F1000Research 2020, 9:1444 (doi: 10.12688/f1000research.27893.1)

Define our contrasts matrix and use contrasts.fit

- Which comparisons/contrasts do we want to calculate?

Design Matrix

```
> model.matrix(~0 + treatment)
   treatmentCTL treatmentI treatmentII treatmentIII
1             1          0           0            0
2             1          0           0            0
3             1          0           0            0
4             0          1           0            0
5             0          1           0            0
6             0          1           0            0
7             0          0           1            0
8             0          0           1            0
9             0          0           1            0
10            0          0           0            1
11            0          0           0            1
12            0          0           0            1
```

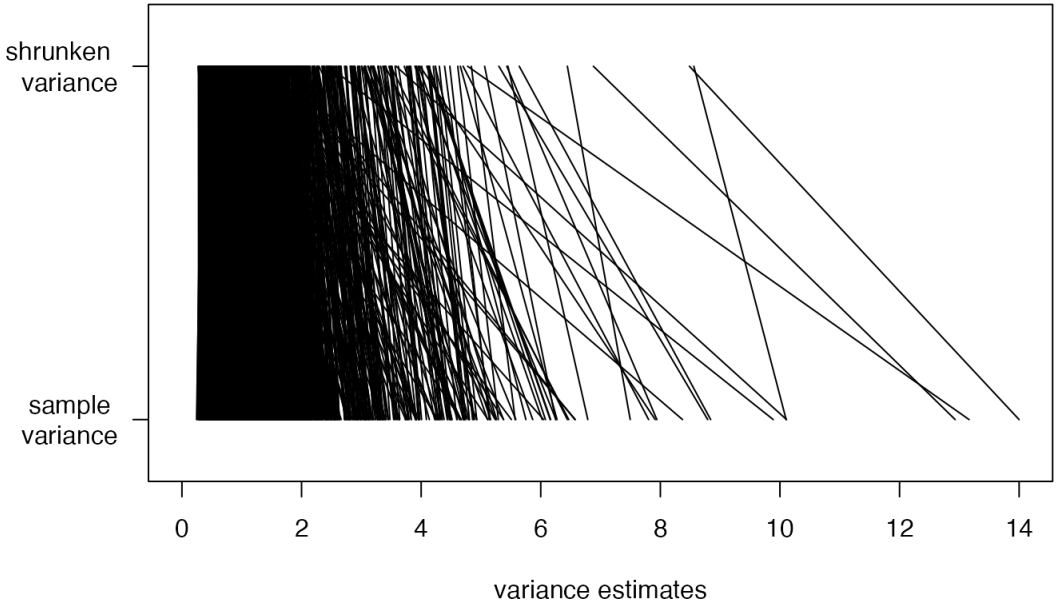
Law CW, Zeglinski K, Dong X et al. A guide to creating design matrices for gene expression experiments [version 1]. F1000Research 2020, 9:1444 (doi: 10.12688/f1000research.27893.1)

Contrast Matrix

```
> ## Specify contrasts of interest
> contrasts <- makeContrasts(CTRL_I = CTL - I,
+                            II_III = II - III,
+                            levels = m_design)
> ## Verify
> contrasts
      Contrasts
Levels CTRL_I II_III
   CTL      1      0
   I       -1      0
   II       0      1
   III      0     -1
```

- We then compute a coefficient (logFC in this case!) for each contrast

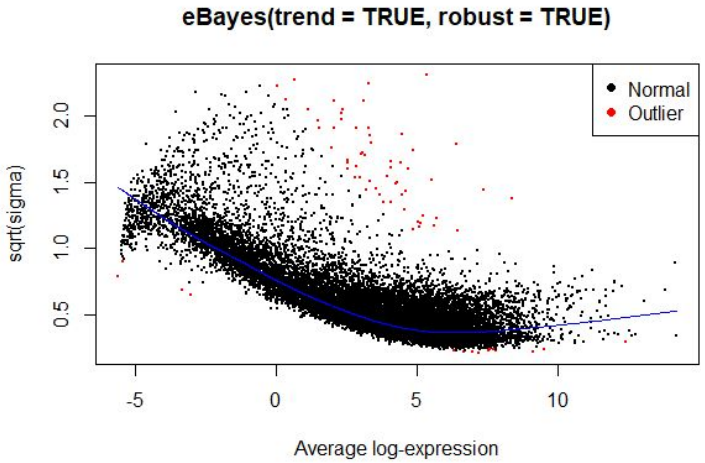
Empirical Bayes moderation with eBayes



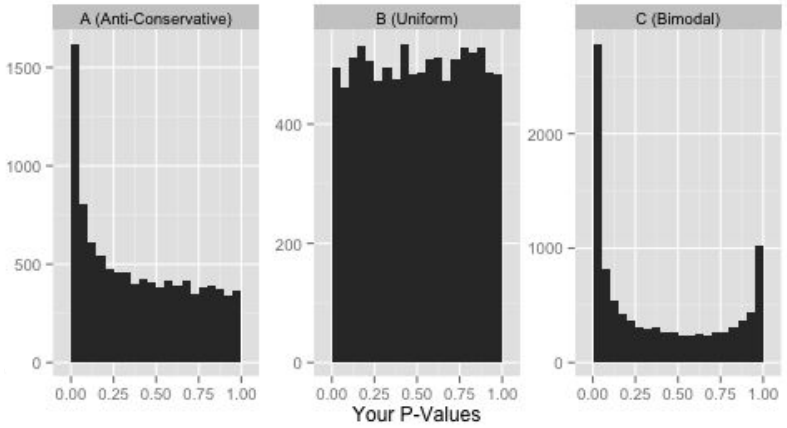
Model diagnostics

The results of our model are only valid if various assumptions are met - we check this using diagnostic plots

SA plot



Histogram of raw p-values



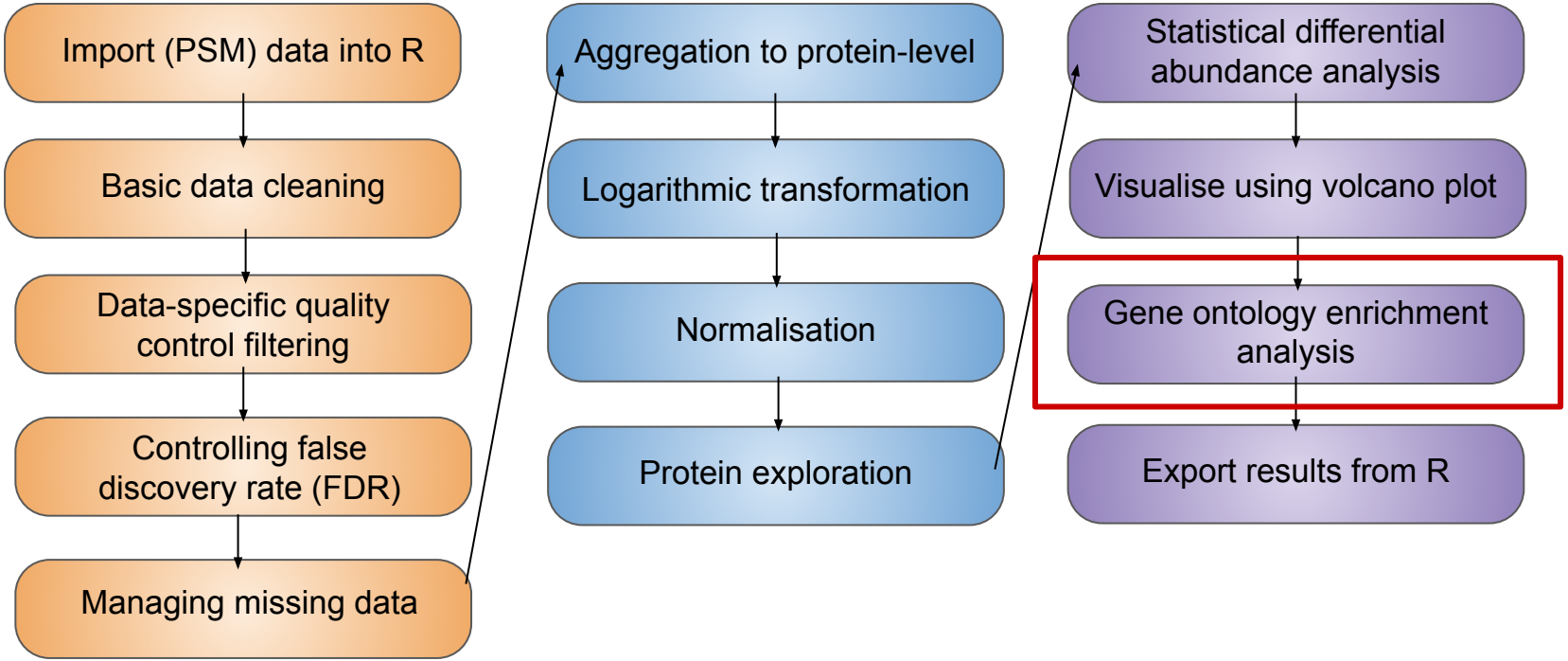
Cambridge Centre for Proteomics

UNIVERSITY OF CAMBRIDGE
Bioinformatics Training

Lesson 7: Gene Ontology enrichment analysis

December 2024

In this workshop…



Overview

1. Know what Gene Ontology terms represent
2. Understand the principle of Gene Ontology over-representation analysis using a foreground and background protein list
3. Complete Gene Ontology over-representation analysis using enrichGO in R

Biological interpretation of changes

What is the biological relevance of the proteins that change in abundance?

We want to know more about our changing proteins:

- Where are they located?
- Which other proteins do they interact with?
- What is their function?
- Which biological pathways are they involved in?

Functional enrichment analysis

Motivation: What do these proteins do?

Aim: Identify the functionalities which are observed more that expected by chance (null hypothesis)

Definition **Gene set** = Set of genes with the same function (GO terms, Hallmark, metabolic pathways, etc). Can be mapped to a set of gene products (proteins)

Approaches:

- **Over-Representation Analysis (ORA):** For the genes passing a threshold(s), is the *gene set* over-represented
- **Univariate Functional Class Scoring, e.g Gene Set Enrichment Analysis (GSEA):** Rank genes by a metric. Is the *gene set* non-randomly distributed?
- **Pathway Topology-based methods:** Uses the information about the relationships between pathway members and graph theory to identify affected pathways

Over-Representation Analysis (ORA)

L = List of genes passing threshold
G_i = Gene set i

	Sig.	Not sig.	
In gene set	131	264	G_i
Not in gene set	375	5457	
	L		

- Under the null hypothesis that there is **no association** between differential expression and membership in G_i, we can assume that the overlap is the result of random sampling.
- The probability of this overlap can be calculated using the hypergeometric distribution

ORA - foreground and background

L = List of genes passing threshold
G_i = Gene set i

Foreground: proteins passing threshold

Background (Universe): all proteins that **could have** passed the threshold

	Sig.	Not sig.
In gene set	131	264
Not in gene set	375	5457

G_i

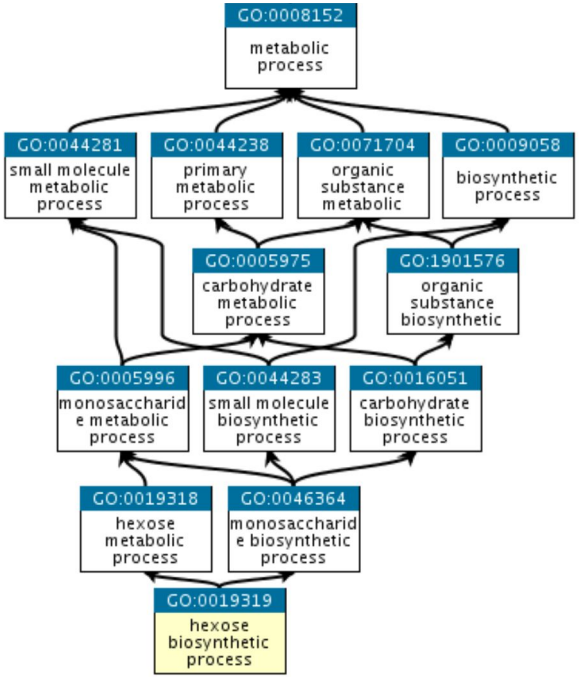
L

In most cases:

- Foreground = ‘significant’ proteins
- Background = all proteins which underwent statistical testing

Gene Ontology (GO) terms

Hierarchical terms that describe the function of genes and their protein products



Molecular function (MF) = molecular-level activities
e.g., “catalyst” or “transport” or “protein kinase activity”

Cellular Component (CC) = cellular location
e.g., “plasma membrane” or “cytoskeleton”

Biological Process (BP) = larger biological program to which a molecular function contributes
e.g., “DNA repair” or “signal transduction”

GO enrichment analysis

Given a list of proteins found to be differentially abundant in my phenotype of interest, what are the cellular components, molecular functions and biological processes involved in this phenotype?