Phillips, A., Unwin, R. D., Hubbard, S., & Dowsey , A. (2021). Uncertainty aware protein-level quantification and differential expression analysis of proteomics data with seaMass. Manuscript submitted for publication. In *Statistical methods for proteomics* (Methods in Molecular Biology). Springer.

Early version, also known as pre-print

Uncertainty aware protein-level quantification and differential expression analysis of proteomics data with seaMass

Alexander M Phillips, Richard D Unwin, Simon J Hubbard and Andrew W Dowsey

Abstract
seaMass is an R package for protein-level quantification, normalisation and differential expression analysis of proteomics mass spectrometry data after peptide identification, protein grouping and feature-level quantification. Using the concept of a blocked experimental design, seaMass can analyse all common discovery proteomics paradigms including label-free (e.g. Waters Progenesis input), SILAC (e.g. MaxQuant input), isotope labelling (e.g. SCIEX ProteinPilot iTraq and Thermo ProteomeDiscoverer TMT input) and data-independent acquisition (e.g. OpenSWATH-PyProphet input), and is able to scale to studies with hundreds of assays or more. By utilising hierarchical Bayesian modelling, seaMass assesses the quantification reliability of each feature and peptide across assays so that only those in consensus influence the resulting protein group quantification strongly. Similarly, unexplained variation in each individual assay is captured, providing both a metric for quality control and automatic down-weighting of suspect assays. To achieve this, each protein group-level quantification outputted by seaMass is accompanied by the standard deviation of its posterior uncertainty. seaMass integrates a flexible differential

Alexander M Phillips
Department of Electrical Engineering Electronics and Computational Biology Facility, Faculty of Health and Life Sciences, University of Liverpool, e-mail: a.m.phillips@liverpool.ac.uk

Richard D Unwin
Stoller Biomarker Discovery Centre and Division of Cancer Sciences, School of Medical Sciences, Faculty of Biology, Medicine and Health, The University of Manchester, Manchester, M13 9NQ, United Kingdom e-mail: r.unwin@manchester.ac.uk

Simon J Hubbard
School of Biological Sciences, Faculty of Biology, Medicine and Health, University of Manchester, Oxford Road, Manchester M13 9PT, UK e-mail: simon.hubbard@manchester.ac.uk

Andrew W Dowsey
Department of Population Health Sciences and Bristol Veterinary School, Faculty of Health Sciences, University of Bristol, Bristol, BS8 2BN, United Kingdom e-mail: andrew.dowsey@bristol.ac.uk

expression analysis subsystem with false discovery rate control based on the popular MCMCglmm package for Bayesian mixed-effects modelling, and also provides uncertainty-aware principal components analysis. We provide a description for using seaMass to perform an end-to-end analysis using a real dataset associated with a published clinical proteomics study.

Key words
quantitative proteomics, protein quantification, Bayesian modelling, differential expression analysis, false discovery rate control

1 Introduction

seaMass (https://github.com/biospi/seamass) is an R package which provides a complete protein quantification, normalisation and differential expression pipeline for discovery mass spectrometry data, after prior identification and feature-level quantification. In particular, it is expected that protein grouping has been performed, so that each "protein" to be quantified represents a "protein group" of accessions that cannot be unambiguously identified given the peptide identification evidence. seaMass consists of three main components: seaMass-sigma, which performs raw protein group-level quantification from peptide and feature-level mass spectrometry data; seaMass-theta, which performs protein group-level normalisation across assays (label-free runs or iTraq/TMT/SILAC channels); and seaMass-delta, which performs differential expression analysis and false discovery rate (FDR) estimation. All three of these procedures use Bayesian hierarchical mixed-effects modelling in order to estimate the uncertainty of the estimated quantities including: peptide and protein group quantifications; normalisation effects; and differential expression fold change estimates.

The mixed-effects modelling employed by seaMass-sigma includes so-called "random effects" to account for variability at multiple levels: the variability of peptides across samples (for example, due to poor or variable digestion) and the variability of measurements across assays (due to contamination or matrix, for example). seaMass-sigma wraps this model within an empirical Bayes procedure that borrows strength across the population of protein groups: it uses those protein groups with a large number of peptides and measurements to estimate informative prior distributions for the distribution of the variance of peptides and features across all protein groups.

By estimating the uncertainty of each peptide, each peptide's contribution to the final protein group quantification estimate can be weighted according to their inferred variance, such that highly variable peptides have a smaller contribution to the overall protein group quantification. Similarly, where peptides are observed via multiple features each feature has its variance estimated so that more variable features contribute less to the peptide-level quantifications.

This quantification uncertainty is propagated from the feature-level through the peptide- and protein group-levels up to the differential expression estimates. seaMass wraps external methods which leverage this additional uncertainty information to provide robust significance testing.

seaMass also captures assay-specific variation not explained by variation at the peptide or feature levels. In this way unreliable assays are identified and flagged during processing, and their contribution towards differential expression analysis and principal components analysis can be automatically down-weighted.

The model fitting is performed with Bayesian Markov chain Monte Carlo (MCMC) sampling using the MCMCglmm [1] R package. Multiple MCMC "chains" are fit for each protein group. False Discovery Rate (FDR) estimation is similarly provided by the ashr [2][3] R package.

2 Material

2.1 Data Type

The input data generally consists of tabulated data in either comma-separated or tab-separated values from a number of different preprocessing software (*see* Subheading 2.2).

2.2 Data Format

seaMass has functions for reading data from each of the following formats:

1. SCIEX ProteinPilot
2. Thermo ProteomeDiscoverer
3. Waters Progenesis QI
4. MaxQuant [4]
5. OpenSWATH

For ProteinPilot, seaMass requires the `PeptideSummary.txt` file output by ProteinPilot. For ProteomeDiscoverer, seaMass requires the `PSMs.txt` file output by ProteomeDiscoverer. For data output by the Progenesis QI software, seaMass requires the `pep_ion_measurements.csv` file. For data output by MaxQuant, seaMass requires both the `evidence.txt` and `proteinGroups.txt` files. For OpenSWATH, seaMass takes in either the output of PyProphet or TRIC. Import routines for other formats can be implemented on request.

2.3 Hardware Requirements

seaMass can be run on either a desktop computer or on a high-performance computing cluster. This tutorial focuses on running seaMass on a desktop machine. The number of samples to be analysed determines the memory requirements of the software; at least 16GB is preferable. Multiple CPU cores can be utilised, though this will increase the memory footprint.

2.4 Software Requirements

1. Either of Linux, macOS or Windows operating systems.
2. A recent version of the R software; for version 1.0.2.0 of seaMass, version 4.0.4 or higher of R is required.

2.5 Software Installation

To install seaMass, enter the following into the R console:

```
> install.packages("devtools")
> devtools::install_github("biospi/seaMass",
    ref = "v1.0.2.0", dependencies = TRUE)
```

which will install the devtools R package before downloading and installing seaMass v1.0.2.0 and its dependencies (*see* **Note 1**). The following R packages should be installed in this process:

```
ashr, data.table, bit64, doRNG, doSNOW, egg, emmeans,
extraDistr, FactoMineR, filelock, fitdistrplus, fst, ggplot2,
ggrepel, gridExtra, igraph, MCMCglmm, plotly, rmarkdown,
R.utils, utf8, uuid, zip
```

Additionally, to download the example dataset used in this tutorial, the `osfr` package is required, which can be installed by running:

```
> install.packages("osfr")
```

3 Methods

This section details the typical workflow of using seaMass to perform analysis of a quantitative proteomics dataset by walking through the process using data associated with a clinical study on Alzheimer's disease (AD) progression, which was first analysed using an earlier version of seaMass in [5] (*see* **Note 2**). Tissue samples from multiple brain regions were collected from the brains of eighteen subjects: nine AD-affected patients (S1–S9) and nine age- and sex-matched controls (S10–S18). For each brain region a pooled reference sample, R, was created by mixing equal amounts of each of the eighteen samples together. Each region was processed as a separate experiment of three iTRAQ 8-plexes. Mass spectrometry analysis was then performed using a SCIEX QSTAR Elite Q-TOF instrument. Peak extraction, peptide identification, protein grouping and iTRAQ reporter quantification was performed using ProteinPilot v4.0. The `PeptideSummary.txt` file from ProteinPilot's output provides the quantitative feature-level data which is input into seaMass.

Here, to illustrate the robustness of seaMass, we analyse the middle temporal gyrus brain region that was excluded from the original publication as the proteomics data for this region did not pass quality control. Notes throughout this section provide guidance for how the example data may be substituted for data from other sources.

3.1 Loading seaMass

First, load the seaMass package in R by entering into the R console:

```
> library(seaMass)
```

3.2 Data Loading

1. The mass spectrometry data from the Alzheimer's disease study is openly available online and can be downloaded using the `osfr` R package by inputting the following into the R console:

```
> osfr::osf_download(osfr::osf_retrieve_file("https://osf.io/vqcgz/"),
    conflicts = "skip", verbose = T, progress = T)
```

   which will download the mass spectrometry data for the middle temporal gyrus.

2. For data processed using ProteinPilot, seaMass requires the output file to perform protein group quantification. We specify the location of the PeptideSummary file and import it into the R environment, before using seaMass's `import_ProteinPilot` function (*see* **Note 3**) to extract the feature-level data into a data frame that seaMass can use for subsequent processing:

```
> file <- "PeptideSummary_MiddleTemporalGyrus.txt"
> data <- import_ProteinPilot(file)
```

3.3 Fractionation

1. For data which has been fractionated, it is necessary to specify which fractions belong to which runs. Firstly, generate a skeleton run table:

   ```
   > data.runs <- runs(data)
   ```

2. Next, we assign runs to each fraction:

   ```
   > data.runs$Run[1:68] <- "A"
   > data.runs$Run[69:152] <- "B"
   > data.runs$Run[153:222] <- "C"
   ```

   In this instance, fractions 1 through 68 belong to run A, 69 through 152 to run B, and 153 through 222 to run C.
3. This fractionation information is then merged back to the imported data:

   ```
   > runs(data) <- data.runs
   ```

3.4 Experimental Design

1. We can now create a skeleton design matrix from our data:

   ```
   > data.design <- new_assay_design(data)
   ```

2. The biological sample associated with each assay can optionally be renamed. The distinction between technical and biological replicates can be made; In this instance, the pooled sample “R” is assigned to six different assays as six technical replicates of the same sample. Each of the biological samples S1-S18 are also assigned to separate assays (*see* **Note 4**):

   ```
   > data.design$Sample <- factor(c(
       "R","R","S1","S3","S7","S12","S17","S10",
       "R","R","S2","S6","S9","S13","S15","S18",
       "R","R","S4","S5","S8","S11","S14","S16"
     ))
   ```

   The assays, in this case corresponding to each iTRAQ channel in each of the three runs, can also be similarly renamed through `data.design$Assay` (*see* **Note 5**).
3. The condition to which each assay belongs is assigned, the `levels` argument can be used to determine which conditions are to be compared. Here, we specify that “Ct” is the first and therefore baseline condition. The pooled reference assays “R” should also be excluded from differential expression analysis:

   ```
   > data.design$Condition <- factor(c(
       NA,NA,"AD","AD","AD","Ct","Ct","Ct",
       NA,NA,"AD","AD","AD","Ct","Ct","Ct",
       NA,NA,"AD","AD","AD","Ct","Ct","Ct"
     ), levels = c("Ct", "AD"))
   ```

4. For experiments where runs are performed in batches or across multiple instruments, it may be desirable to split the assays into multiple "blocks" (*see* **Note 6**); for iTRAQ and TMT experiments with multiple runs, seaMass automatically splits each multiplex out into a separate block.
5. Additional covariates can be added to the experimental design at this stage by adding additional columns to the `data.design` table.
6. Finally, "reference weights" can be assigned to specify reference assays. Conventionally, replicated pooled sample assays are used as reference assays in each block so that protein group quantifications can be standardised in relation to them for direct comparison across blocks. As seaMass-theta allows for multiple reference samples per block, to standardise to the average of the two pooled sample assays in each block, the reference weights are set as:

   ```
   > data.design$RefWeight <- c(
       1,1,0,0,0,0,0,0,
       1,1,0,0,0,0,0,0,
       1,1,0,0,0,0,0,0
     )
   ```

   For a blocked experimental design where each condition is represented in each block, seaMass also allows standardisation using a suitable weighted average of the samples themselves, so that no pooled samples are necessary. For example, to standardise to the average of the AD and Ct samples (*see* **Note 7**):

   ```
   > data.design$RefWeight <- c(
       0,0,1,1,1,1,1,1,
       0,0,1,1,1,1,1,1,
       0,0,1,1,1,1,1,1
     )
   ```

7. The complete experimental design can be viewed by typing `data.design` into the R console. The complete table for the example dataset is shown in Table 1

3.5 Protein Group Quantification and Normalisation

1. After the initial setup and addition of experimental design, protein group quantification can be performed by running `seaMass_sigma`, which takes as input the feature-level data. Optionally, the output directory can be specified using the `path` argument. The experimental design table can also be supplied; while not required at this stage, it will be used to add design metadata to the automatically generated plots:

   ```
   > fit.sigma <- seaMass_sigma(
       data,
       data.design,
   ```

Table 1 Experimental Design Table for the Middle Temporal Gyrus Dataset

Run	Channel	Assay	RefWeight	Sample	Condition
A	113	R1	1	R	<NA>
A	114	R2	1	R	<NA>
A	115	S1	0	S1	AD
A	116	S3	0	S3	AD
A	117	S7	0	S7	AD
A	118	S12	0	S12	Ct
A	119	S17	0	S17	Ct
A	121	S10	0	S10	Ct
B	113	R3	1	R	<NA>
B	114	R4	1	R	<NA>
B	115	S2	0	S2	AD
B	116	S6	0	S6	AD
B	117	S9	0	S9	AD
B	118	S13	0	S13	Ct
B	119	S15	0	S15	Ct
B	121	S18	0	S18	Ct
C	113	R5	1	R	<NA>
C	114	R6	1	R	<NA>
C	115	S4	0	S4	AD
C	116	S5	0	S5	AD
C	117	S8	0	S8	AD
C	118	S11	0	S11	Ct
C	119	S14	0	S14	Ct
C	121	S16	0	S16	Ct

```
    path = "MiddleTemporalGyrus",
    control = sigma_control(nthread = 8)
  )
```

2. After `seaMass_sigma` is finished, the derived raw protein group quantifications can be normalised within blocks and standardised across blocks by executing `seaMass_theta`:

```
  > fit.theta <- seaMass_theta(
    fit.sigma,
    norm.groups = top_groups(fit.sigma)
  )
```

   The `seaMass_theta` normalisation model derives a normalisation factor for each assay as to minimise protein group-level variance for the maximum number of protein groups. For computational efficiency, by default only high-quality protein groups are examined. If the user has some prior knowledge of the subset of protein groups to normalise against, this can be specified by supplying a subset of `groups(fit.sigma)$Group` to the `norm.groups` option instead.

3. Configuration of seaMass processing is achieved through the `sigma_control` and `theta_control` objects. On multi-core systems with sufficient amounts of

RAM, multiple CPU threads can be used; the number of threads can be specified as the `nthread` option to `sigma_control` (*see* **Note 8**).

3.6 Protein Group Quantification Output

1. seaMass outputs a set of convenient CSV files of the results in the `csv` sub-folder of the output folder `"MiddleTemporalGyrus"` . Alternatively, results can be output within R. For instance, a table of normalised protein group quantifications can be output using:

```
> proteinQuants <- group_quants(fit.theta)
```

2. In the resulting `data.frame` , seaMass outputs each protein group quantification `m` (the posterior mean) along with its uncertainty `s` (the posterior standard deviation). Subsequently, these protein group quantifications can be outputted to, for example, a CSV file:

```
> write.csv(proteinQuants, file = "my_proteinQuants.csv")
```

3.7 seaMass-sigma and seaMass-theta Plots Output

seaMass outputs a full HTML report with a rich set of interactive plots as a zip archive in the output directory (*see* **Note 9**). Below, we instead use the R package to output specific plots. For some of the following examples we will generate the plots for a single protein group, sp|P09211|GSTP1_HUMAN.

Each violin in the violin plots, such as in Figure 1, spans the 90% interval of probable values for that variable (the Bayesian 90% credible interval), with the median value represented as a vertical bar. Left of the median the girth of the violin reduces in size, representing the local FDR for the variable being that value or less (posterior probability that the variable is that value or more), with the violin truncated at 5% local FDR. Conversely, right of the median the girth also reduces, representing the local FDR for the variable being that value or more, and is truncated similarly. Wider violins therefore represent more uncertain estimations.

1. Local FDR violin plots showing the inferred normalisation factors ("assay means") and unexplained assay variation ("assay standard deviations") can be generated by typing in the R console:

```
> g1 <- plot_assay_means(fit.theta, output = "ggplot")
> g2 <- plot_assay_stdevs(fit.sigma, output = "ggplot")
> g <- gridExtra::grid.arrange(g1, g2)
> ggplot2::ggsave("assay_means_stdevs.pdf", g,
    width = 7, height = 7)
```

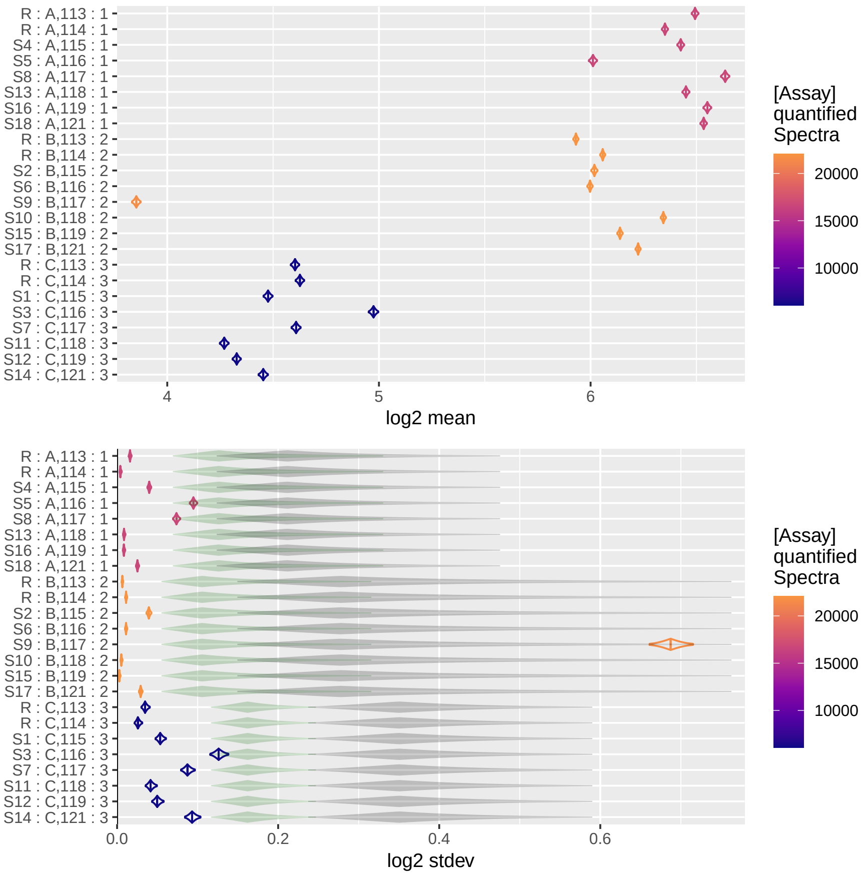

Fig. 1 Local FDR violin plots of assay means and standard deviations for the example dataset. Top: The estimated assay means are coloured by the number of quantified spectra for that iTraq 8-plex. It can be seen that the assays in iTRAQ 8-plex run C (block 3) are generally under-exposed relative to the assays in runs A and B, except for sample S9 in run B which is even more under-exposed. Bottom: Similarly, the estimated assay standard deviations are shown, together with the inferred distribution of "explained" peptide standard deviations (green violins) and feature standard deviations (grey violins). As a rule of thumb, unexplained assay variation should be substantively lower than explained variation, and both should be less than the fold change of differential expression you hope to discover. Hence here it illustrates that sample S9 in run B is a significant potential quality control problem - as a result seaMass-delta will automatically down-weight this assay's contribution downstream.

In the above, setting the `output` option to `"plotly"` generates interactive plots, whereas setting it to `"ggplot"` generates static plots potentially more suitable for constructing publication figures. The PDF output of this code is shown in Figure 1.

2. Principal Components Analysis (PCA) plots are generated for each block and for the experiment as a whole automatically. These PCA plots down-weight

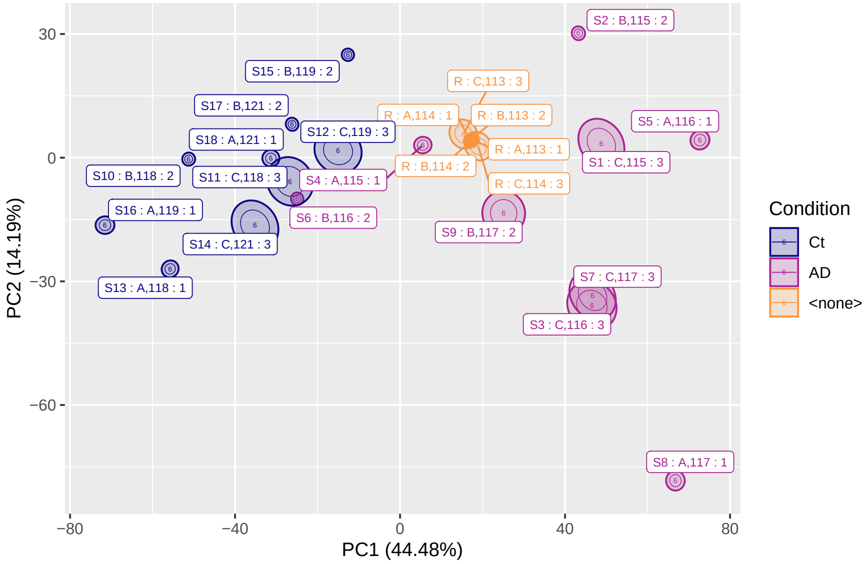

Fig. 2 Robust Principal Component Analysis plot for the example dataset. Each assay in the experiment appears as an ellipse, coloured by condition. Contours are shown for each assay indicating the uncertainty in quantifications for that assay. Here it can be seen that the potential issues with run C and sample S9 are reflected in larger uncertainty ellipses.

poorly quantified assays and protein groups, and are subsequently augmented with ellipses indicating the 95% and 68% posterior regions of uncertainty in the principal components for each assay. These can be used to determine whether any assays exhibit more variation than the others, which would be indicative of issues in sample preparation for example.

```
> g <- plot_robust_pca(fit.theta, colour = "Condition",
    fill = "Condition", shape = 6, output = "ggplot")
> ggplot2::ggsave("robust_pca.pdf", g,
    width = 7, height = 5)
```

The PDF plot generated for the example data is shown in Figure 2.

3. Plots of the raw and normalised protein group quantifications for any particular protein group can be generated by running:

```
> g <- plot_group_quants(fit.theta,
    "sp|P09211|GSTP1_HUMAN", output = "ggplot")
> ggplot2::ggsave("group_quants.pdf", g,
    width = 7, height = 4)
```

The PDF plot generated for sp|P09211|GSTP1_HUMAN is shown in Figure 3.

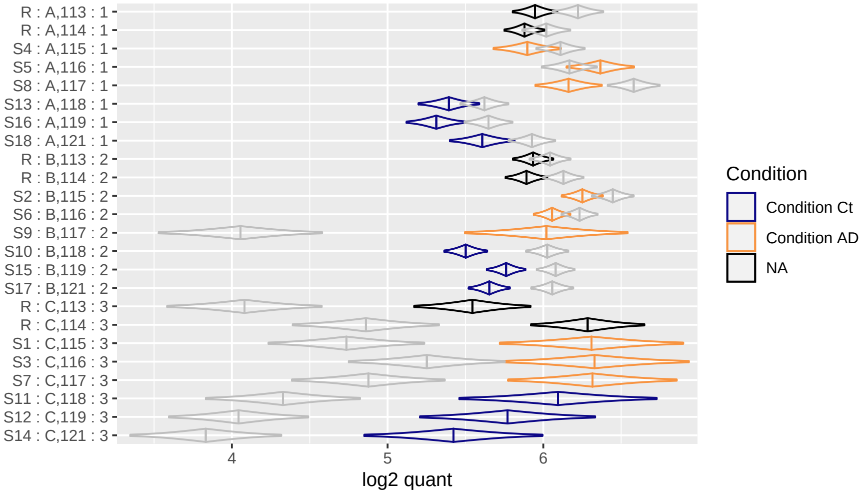

Fig. 3 Local FDR violin plots of the raw (grey) and normalised (coloured by condition) protein group-level quantifications for the protein sp|P09211|GSTP1_HUMAN in the example dataset. Note that the quantifications for run C and sample S9 are more uncertain than the rest, which is propagated down-stream to the seaMass-delta differential expression analysis phase.

4. We can also visualise how the peptide-level quantification estimates differ from the protein-level quantifications. seaMass calls these “component deviations”:

```
> g <- plot_component_deviations(fit.sigma,
    "sp|P09211|GSTP1_HUMAN", output = "ggplot")
> ggplot2::ggsave("component_deviations.pdf", g,
    width = 10, height = 12)
```

   The plot generated for the protein sp|P09211|GSTP1_HUMAN in the example Alzheimer’s disease dataset is shown in Figure 4.

5. Plots of the mean intensity and standard deviation of each peptide observed for a particular protein group can be generated by typing:

```
> g1 <- plot_component_means(fit.sigma,
    "sp|P09211|GSTP1_HUMAN", output = "ggplot")
> g2 <- plot_component_stdevs(fit.sigma,
    "sp|P09211|GSTP1_HUMAN", output = "ggplot")
> g <- gridExtra::grid.arrange(g1, g2)
> ggplot2::ggsave("component_means_stdevs.pdf", g,
    width = 7, height = 4)
```

   The plot generated for the protein sp|P09211|GSTP1_HUMAN in the example Alzheimer’s disease dataset is shown in Figure 5.

6. Similar to the peptide-level plots we can also generate violin plots of the feature-level (“measurement"") mean intensities and standard deviations:

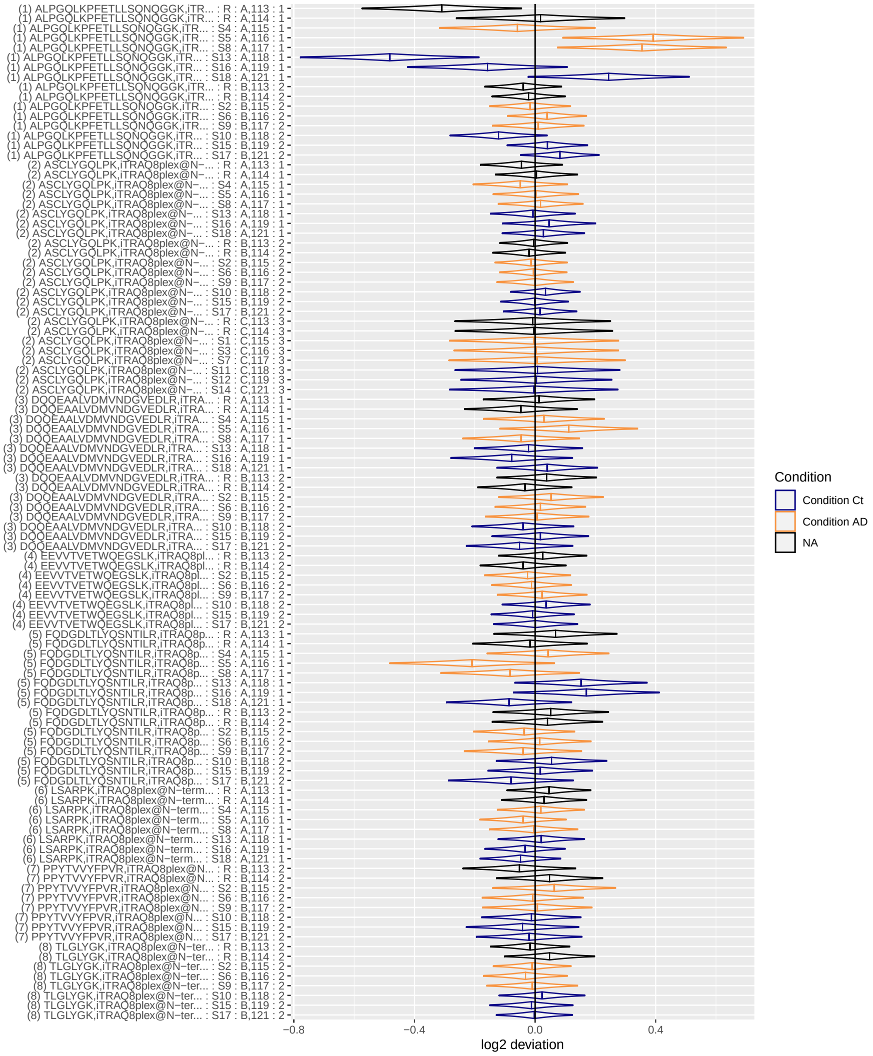

Fig. 4 Local FDR violin plots showing the peptide-level deviations from the parent protein group quantification for the protein group sp|P09211|GSTP1_HUMAN. The difference is notable particularly for the top peptide in the plot, which could be due to a systematic technical issue or be indicative of a differingly expressed proteoform.

```
> g1 <- plot_measurement_means(fit.sigma,
    "sp|P09211|GSTP1_HUMAN", output = "ggplot")
> g2 <- plot_measurement_stdevs(fit.sigma,
    "sp|P09211|GSTP1_HUMAN", output = "ggplot")
> g <- gridExtra::grid.arrange(g1, g2)
> ggplot2::ggsave("measurement_means_stdevs.pdf", g,
```

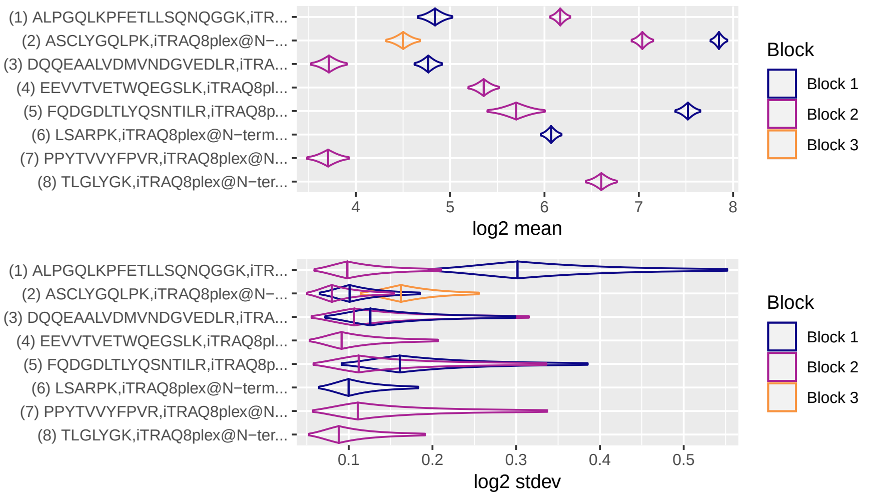

Fig. 5 Local FDR violin plots showing peptide-level means and standard deviations for the protein sp|P09211|GSTP1_HUMAN in the example dataset. Peptide-level means are a weighted average of feature-level mean intensities. Each feature is weighted by its precision hence more variable features contribute less to the peptide-level quantification. Peptide precisions affect the overall protein group-level quantification similarly. Here the top peptide in the plot is particularly variable in block 1 (run A).

```
        width = 7, height = 9)
```

The generated PDF is shown in Figure 6.

3.8 Differential Expression and FDR Estimation

1. The differential expression analysis and FDR estimation component of seaMass, seaMass-delta, can be run on the resulting seaMass-theta fit object:

```
   > fit.delta <- seaMass_delta(fit.theta)
```

2. seaMass-delta will proceed to fit a differential expression model to the normalised protein quantification estimates generated by seaMass-theta. By default, this differential expression model is equivalent to performing a Welch's t-test for each pair-wise comparison of defined conditions. Different differential expression models can be configured by supplying additional arguments to `seaMass_delta` (*see* **Note 10**).
3. FDR estimation is then performed using the ashr R package [3]. Ash uses an empirical Bayes approach to perform "adaptive shrinkage" on the estimated log2 fold changes generated by seaMass-delta and harnesses the extra uncertainty information provided by seaMass to estimate the distribution of log2 fold changes

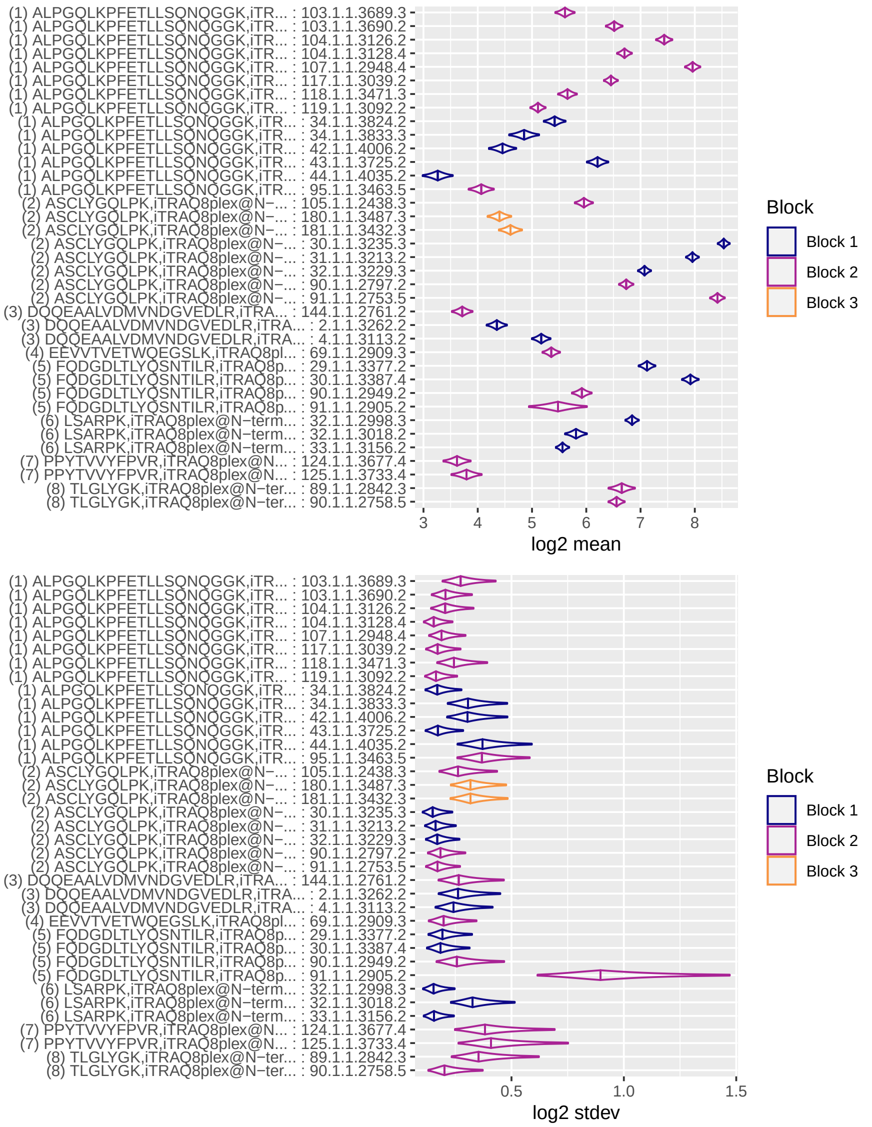

Fig. 6 Local FDR violin plots showing iTraq reporter ion-level means and standard deviations for the protein sp|P09211|GSTP1_HUMAN in the example dataset. It can be seen that reporter ion intensities range over 6 orders of magnitude and several exhibit high variance, however seaMass-sigma is able to focus its quantification on the most stable.

across the dataset and, depending on the uncertainty of each log2 fold change, moderate those estimated log2 fold changes. Protein groups for which there is

Table 2 Excerpt of the outputted results with estimated global FDR (qvalue) and posterior mean log2 fold-change and posterior standard devation of log2 fold-change for each protein (Group).

Batch	Group	qvalue	PosteriorMean	PosteriorSD
Condition.AD-Ct	sp|P09211|GSTP1_HUMAN	2.120733e-09	0.5017938	0.06695361
Condition.AD-Ct	sp|Q13228|SBP1_HUMAN	1.930646e-08	0.4427758	0.07636209
Condition.AD-Ct	sp|Q9UEY8|ADDG_HUMAN	2.679536e-06	0.3380300	0.06532386
Condition.AD-Ct	sp|P10909|CLUS_HUMAN	2.246470e-05	0.4554334	0.08949953
Condition.AD-Ct	sp|Q13510|ASAH1_HUMAN	3.719622e-05	0.4149335	0.09259445
Condition.AD-Ct	sp|Q16643|DREB_HUMAN	7.131150e-05	-0.4427733	0.10096924
Condition.AD-Ct	sp|Q9NSD9|SYFB_HUMAN	1.099772e-04	-0.4478407	0.10202492
Condition.AD-Ct	sp|Q99497|PARK7_HUMAN	1.526100e-04	0.3767814	0.07312156
Condition.AD-Ct	sp|P49006|MRP_HUMAN	1.894262e-04	0.4266677	0.10103153
Condition.AD-Ct	sp|Q9NZH0|GPC5B_HUMAN	2.255137e-04	0.4376124	0.10208458

high uncertainty have more shrinkage applied to their log2 fold changes than those proteins whose log2 fold changes are less uncertain.

3.9 Differential Expression Output

1. Once seaMass-delta has completed processing, a data.frame containing estimates of log2 fold change and quantitative false discovery rate can be obtained using:

```
> data.fdr <- group_quants_fdr(fit.delta)
```

   An example of the results from the Alzheimer's disease study are shown in Table 2.
2. This data frame can be saved as a CSV file e.g. for further downstream processing:

```
> write.csv(data.fdr, file = "MiddleTemporalGyrus-FDR.csv")
```

3.10 seaMass-delta Plots Output

seaMass-delta appends a number of interactive plots into the HTML report by default, and more can be optionally generated after the fact. These include quantitative volcano plots and FDR curves.

1. Volcano plots can be generated from the seaMass-delta results by inputting into the R console:

```
> g <- plot_volcano(fit.delta, output = "ggplot")
> ggplot2::ggsave("volcano.pdf", g,
    width = 7, height = 7)
```

   A volcano plot for the "AD - Ct" comparison of the Alzheimer's disease study is given in Figure 7.

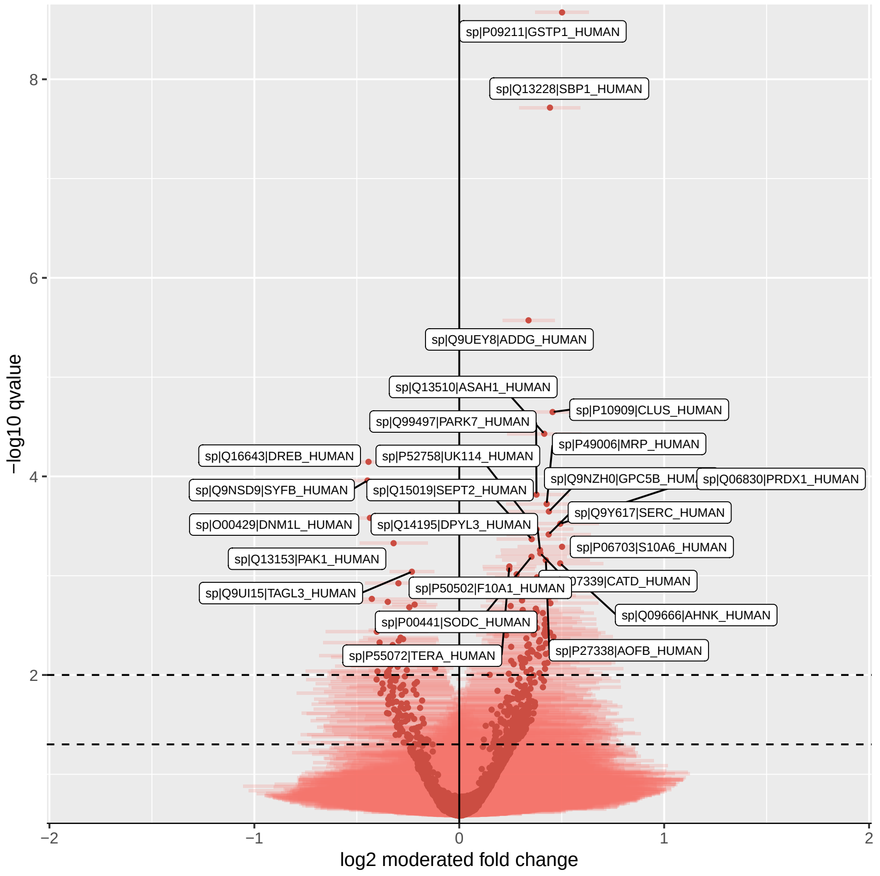

Fig. 7 Volcano plotfor the AD - Ct comparison of the example dataset, with the x-axis denoting the estimated log2 moderated fold-change and the y-axis denoting the FDR as -log10(qvalue). Each point has horizontal error bars denoting the 95% credible interval of the estimated fold-change. The 25 protein groups with the lowest qvalues are labelled and horizontal dashed lines are shown at FDRs of 1% and 5%. 0.05 and 0.01.

2. A plot showing the predicted qvalue FDR against the number of discoveries at that FDR can be plotted with:

```
> g <- plot_fdr(fit.delta, output = "ggplot")
> ggplot2::ggsave("fdr.pdf", g,
    width = 7, height = 4)
```

   The resulting PDF is shown in Figure 8.

3. Finally, local FDR violin plots of the FDR-controlled log2 fold changes can be plotted for any number of protein groups. For example, to plot the 25 most differentially expressed protein groups in the AD - Ct comparison:

```
> g <- plot_group_quants_fdr(fit.delta,
```

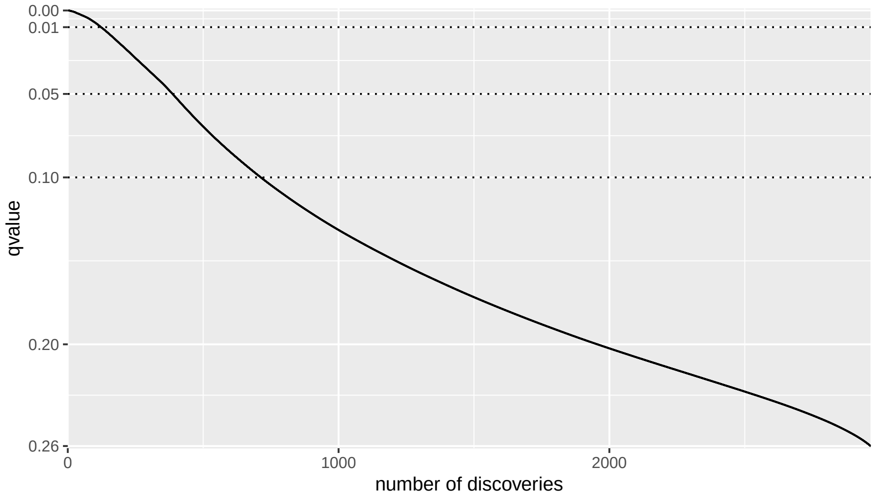

Fig. 8 qvalue FDR vs number of discoveries curve for the example dataset, showing the number of discoveries that would be declared at a given FDR cutoff. Horizontal dashed lines are shown at 1%, 5% and 10% FDR.

```
    group_quants_fdr(fit.delta)$Group[1:25],
    output = "ggplot")
> ggplot2::ggsave("group_quants_fdr.pdf", g,
    width = 7, height = 5)
```

The plot generated is shown in Figure 9.

Notes

1. We are using v1.0.2.0 of seaMass to ensure compatibility with this tutorial. For production use we always recommend you use the latest version of seaMass instead.
2. In [5], the version of the seaMass software used for quantitative analysis was then named Bayesprot.
3. seaMass provides import functions for other input formats which are similarly named, including MaxQuant (`import_MaxQuant`), Progenesis (`import_Progenesis`), ProteomeDiscoverer (`import_ProteomeDiscoverer`), OpenSWATH (`import_OpenSWATH`), and MSstats (`import_MSstats`). The details for the input files required for these import routines can be found by typing `?` and the name of the function in the R console and reading the documentation.
4. As an example, suppose that multiple technical replicates of sample “S1” were included in the experiment. In this scenario, each should be assigned as sample “S1” with different assay names.

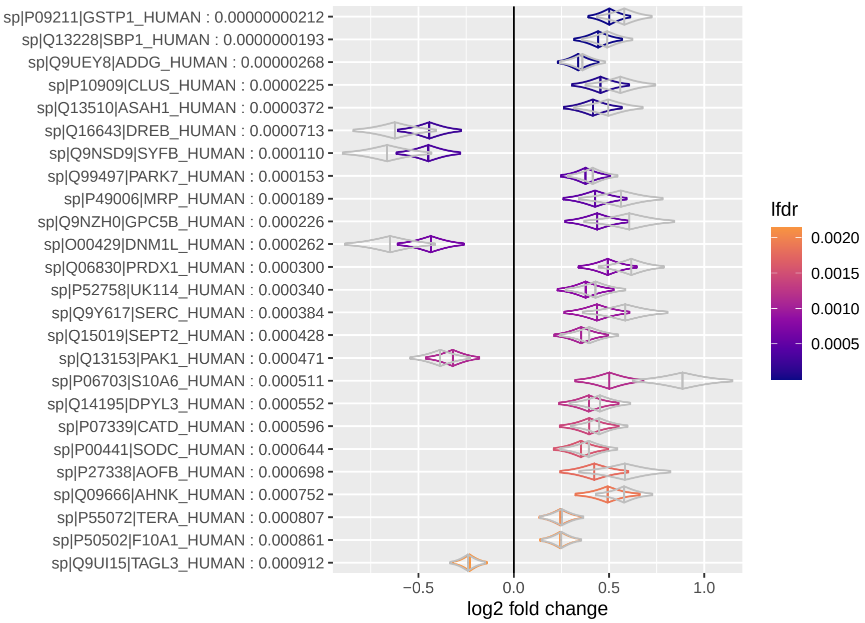

Fig. 9 Local FDR violin plots of the estimated fold-change for the 25 protein groups with lowest qvalue (given in y axis labels) in the AD - Ct comparison. The unmoderated fold changes from the individual MCMCglmm Welch's t-tests are shown in grey, while the fold changes moderated through Ash modelling of the distribution of fold changes in the study are coloured by their local FDR.

5. It is also sometimes desirable to remove an assay from the analysis, say because only some of the iTRAQ reporter channels were filled in a particular run or a sample is identified to have been contaminated. In these scenarios, the assay can be removed by assigning its name as missing with `NA` .

6. Blocking can be specified by adding additional columns containing `TRUE` and `FALSE` values to the the experimental design table with columns names of the form: `Block.1` , `Block.2` etc. Assays may appear in multiple blocks. Also, On a HPC cluster, protein group quantification with seaMass-sigma is able to run in parallel across these blocks.

7. For the purposes of quality control, where e.g. a pooled sample is available, it may be preferable to *not* use the pooled sample assays as reference assays. Then, any deviation between pooled assays in different blocks can be inferred as a measure of quality control. Conversely, when quality control has been assured, the reverse can be done; and the pooled samples can be used as the references such that protein group quantifications are calculated in relation to the reference samples.

8. Running seaMass on high-performance computing (HPC) clusters is also supported. This is achieved by specifying a scheduler in `sigma_control` . In seaMass schedulers for SLURM-managed clusters (`schedule_slurm`), PBS-managed

clusters (`schedule_pbs`) and SGE-managed clusters (`schedule_sge`) are implemented. More details are given at http://github.com/biospi/seamass

9. Due to the large size of unzipped reports, it is preferred to mount the zip as a drive for browsing without uncompressing, as described at https://github.com/biospi/seamass

10. By default, the differential expression model fitted is a Bayesian equivalent to a Welch's t-test, where each condition is assumed to have a separate residual variance. This model can be altered by specifying different formulae and priors for seaMass-delta. These formulae must comply to the syntax used by the MCMCglmm package [1], which is similar to the formula syntax used by the `lme4` R package[6]. For example, a Bayesian model equivalent to a Student's t-test can be fit by specifying `rcov=~units` and `prior=list(R=list(V=1,nu=2e-4))` . If additional covariates were entered into the `data.design` table, these can be included in the model by overriding the `fixed` argument. For example, to include "Age" as a predictor: `fixed=~Condition+Age` . Random effects can be included by specifying a `random` formula argument. Care should be taken here to ensure that the `prior` argument is modified accordingly; details of how the prior should be specified can be found in the documentation for MCMCglmm [1].

Acknowledgements The development of seaMass was supported by BBSRC grants BB/M024954/2 BB/R021430/1 and MRC grant MR/N028457/1.

References

[1] Hadfield JD (2010) MCMC Methods for Multi-Response Generalized Linear Mixed Models: The MCMCglmm R Package. Journal of Statistical Software 33(2):1–22

[2] Stephens M (2017) False discovery rates: A new deal. Biostatistics (Oxford, England) 18(2):275–294, DOI 10.1093/biostatistics/kxw041

[3] Stephens M, Carbonetto P, Gerard D, Lu M, Sun L, Willwerscheid J, Xiao N (2019) Ashr: Methods for Adaptive Shrinkage, Using Empirical Bayes

[4] Tyanova S, Temu T, Cox J (2016) The MaxQuant computational platform for mass spectrometry-based shotgun proteomics. Nature Protocols 11(12):2301–2319, DOI 10.1038/nprot.2016.136

[5] Xu J, Patassini S, Rustogi N, Riba-Garcia I, Hale BD, Phillips AM, Waldvogel H, Haines R, Bradbury P, Stevens A, Faull RLM, Dowsey AW, Cooper GJS, Unwin RD (2019) Regional protein expression in human Alzheimer's brain correlates with disease severity. Communications Biology 2(1):43, DOI 10.1038/s42003-018-0254-9

[6] Bates D, Mächler M, Bolker B, Walker S (2015) Fitting Linear Mixed-Effects Models Using lme4. Journal of Statistical Software 67(1):1–48, DOI 10.18637/jss.v067.i01