

# MS-DAP: Mass Spectrometry Downstream Analysis Pipeline

version: beta 0.1.7.4    <https://github.com/ftwkoopmans/msdap/>

## Contents

<b>1</b>	<b>Quality control</b>	<b>2</b>
1.1	number of peptides and proteins . . . . .	2
1.2	data completeness . . . . .	5
1.3	abundance distributions . . . . .	7
1.4	retention time . . . . .	9
1.5	variation among replicates . . . . .	16
1.6	PCA . . . . .	21
<b>2</b>	<b>Differential abundance analysis</b>	<b>22</b>
2.1	one vs two . . . . .	22
2.2	one vs three . . . . .	29
2.3	two vs three . . . . .	36
<b>3</b>	<b>Summary of differential testing</b>	<b>43</b>
<b>4</b>	<b>log</b>	<b>44</b>
<b>5</b>	<b>R command history</b>	<b>45</b>
<b>6</b>	<b>R session info</b>	<b>46</b>

# 1 Quality control

The quality control figures in this section enable you to investigate reproducibility and global clustering of samples by visualizing:

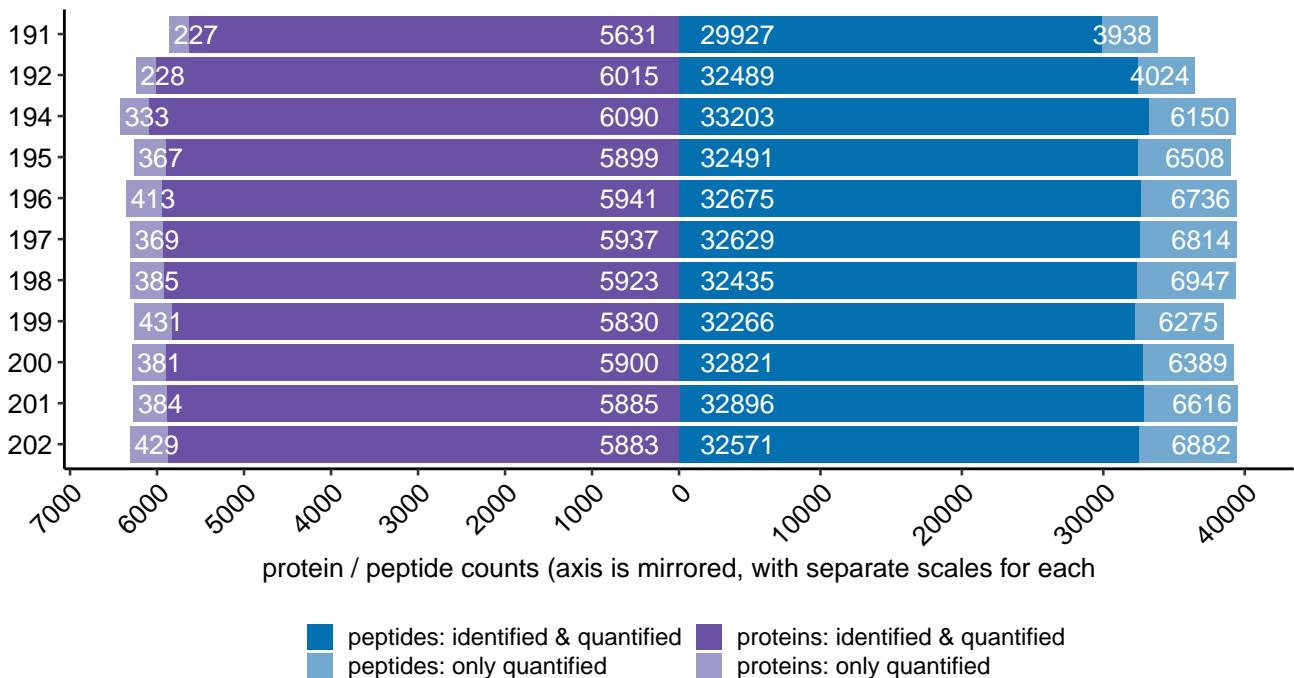
- number of peptides/proteins detected in each sample
- dataset completeness
- local effects in HPLC peptide retention time per sample
- reproducibility of peptide quantification among replicates
- PCA of all samples to visualize clustering

The first set of quality control figures describes individual samples, thereafter group-level quality metrics are described and finally sample clustering is used to highlight structure in the entire dataset.

## 1.1 number of peptides and proteins

These plots show the number of (target) peptides that are ‘detected’ per sample. For DDA, ‘detected’ implies the peptide has a MS/MS identification. Peptides quantified through match-between-runs (MBR) are quantified but not detected/identified. In case of DDA, we also show the number of peptides quantified through MBR. For DIA, we refer to a peptide as ‘detected’ if the confidence score (for identification) is  $\leq 0.01$ .

Samples in this plot are sorted by their experimental group, and then ordered and by their name within each group. This data is also available in the output table ‘samples.xlsx’.

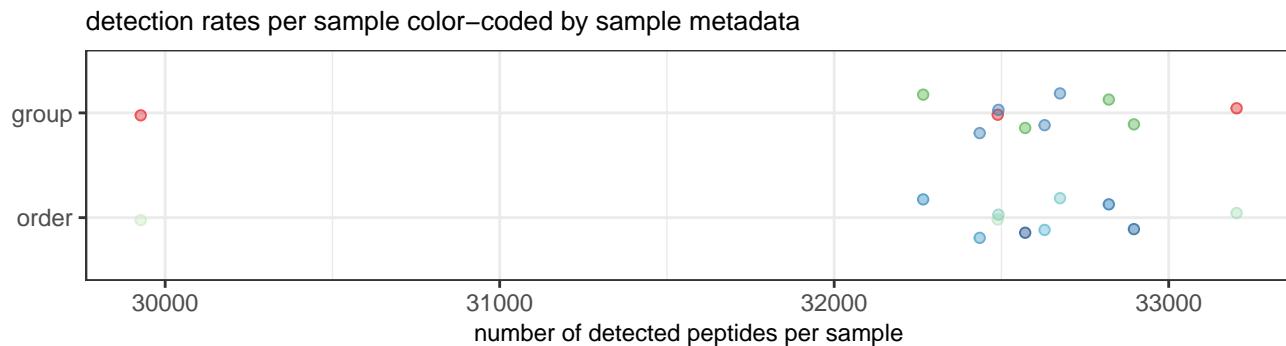


### 1.1.1 color-coding sample metadata

The number of detected peptides in a sample, as compared to other samples within a dataset, can be used as a measure for sample quality. Color-coding individual samples for metadata that you provided as input (e.g. experiment batch, sample handling order, gel lanes, etc.) allows visual inspection as to whether these relate to the rate of successful peptide detection.

The figure below aims to provide an overview of all sample metadata at a first glance. The y-axis shows all sample metadata. On each row all samples in the dataset are shown as a dot, each color-coded by the respective property shown on the y-axis (with minor vertical jitter for visual clarity). If there is a major effect of any experimental condition on the number of detected peptides, this is easily spotted as all outlier samples (extremes in number peptide detection, x-axis) will have the same color. The following figures expand this overview into a detailed figure for each sample property, those visualization are designed to further dive into each experimental condition and provide respective color legends.

Note that the visualization of sample metadata in this report depends on user-provided input; each column in the metadata input table (besides sample names) that contains more than 1 unique value is automatically used as a factor for color-coding all figures in this section. So the data used for color-coding is also available in the output table ‘samples.xlsx’.



### color-coding sample metadata, expanded

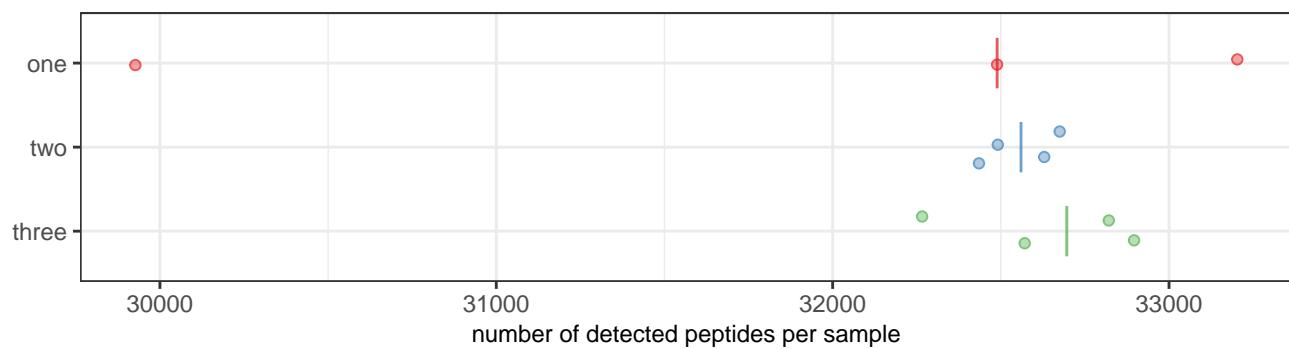
To further detail each sample property, each row in the above figure is now split into separate plots. Thus, a figure is generated for each experimental condition from the user-provided metadata (column in the samples table, it's name shown in the plot title).

The y-axis shows all unique variables, the x-axis the number of detected peptides and each dot is a sample. The vertical line indicates the median value for each row. Colors are consistent with the above plot.

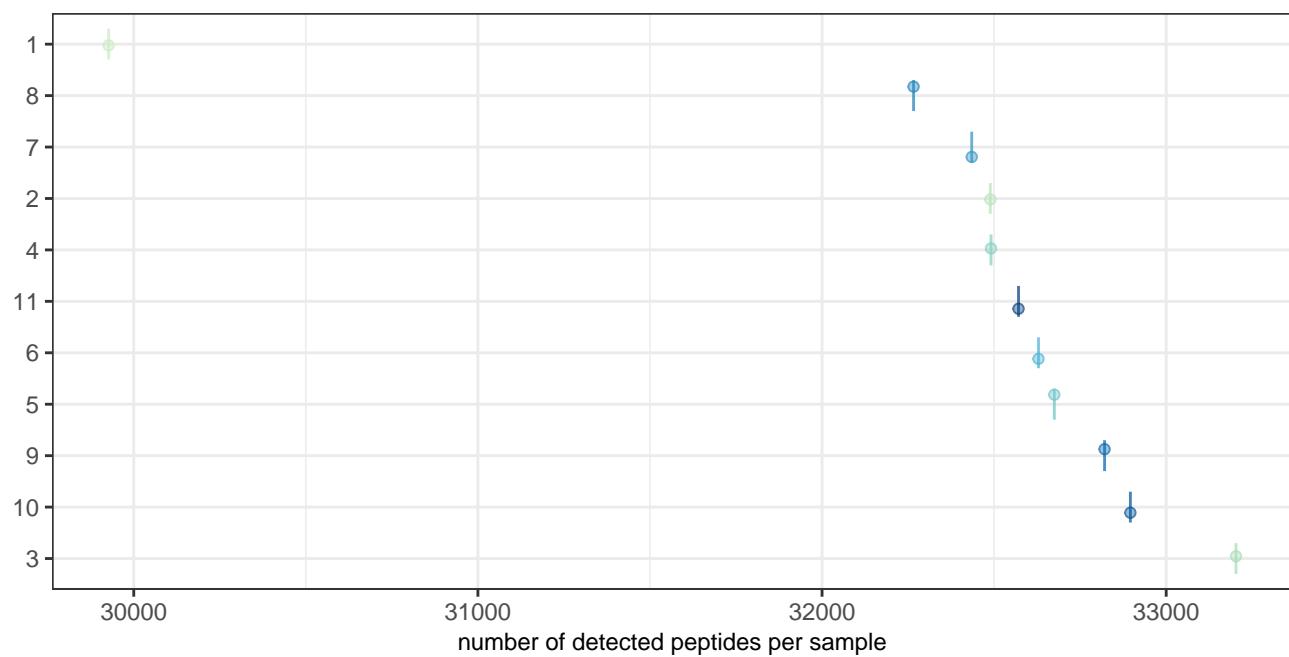
For example: the first plot shows color-coding by the ‘group’ property, so each row represents a sample group. If samples in a particular group systematically yield fewer peptides than another group, a clear pattern will be visible.

Note that the *exclude* metadata is a user-provided indication of ‘outlier samples’ in the dataset that are excluded from downstream statistical analysis.

sample metadata used for color-coding: group



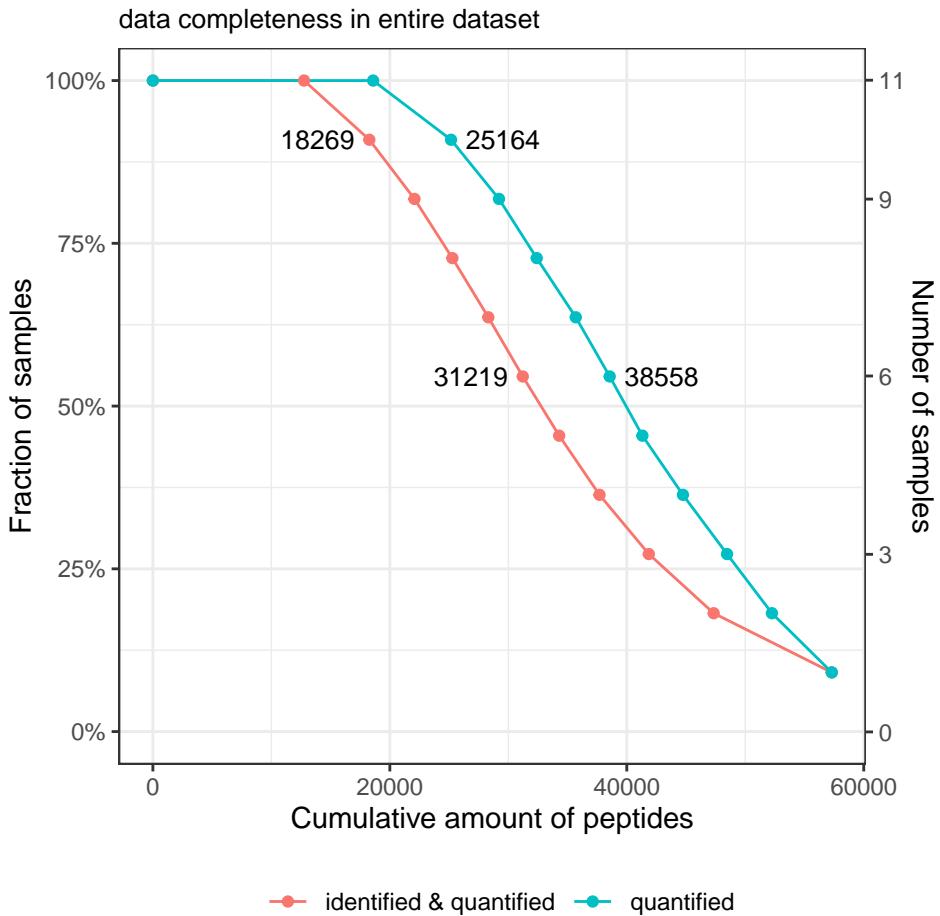
sample metadata used for color-coding: order



## 1.2 data completeness

To visualize how many peptides are consistently identified in multiple samples, the first figure summarizes how common missing values are in the entire dataset. Optimally, most peptides are identified in 100% of samples and this curve slowly falls off. The following figure shows for each sample whether its peptides are also present in other samples in the dataset or whether these are unique to a (minor) subset of samples. You can use this mark of experimental consistency to compare datasets generated by similar protocols and mass-spec acquisition.

### 1.2.1 cumulative distribution



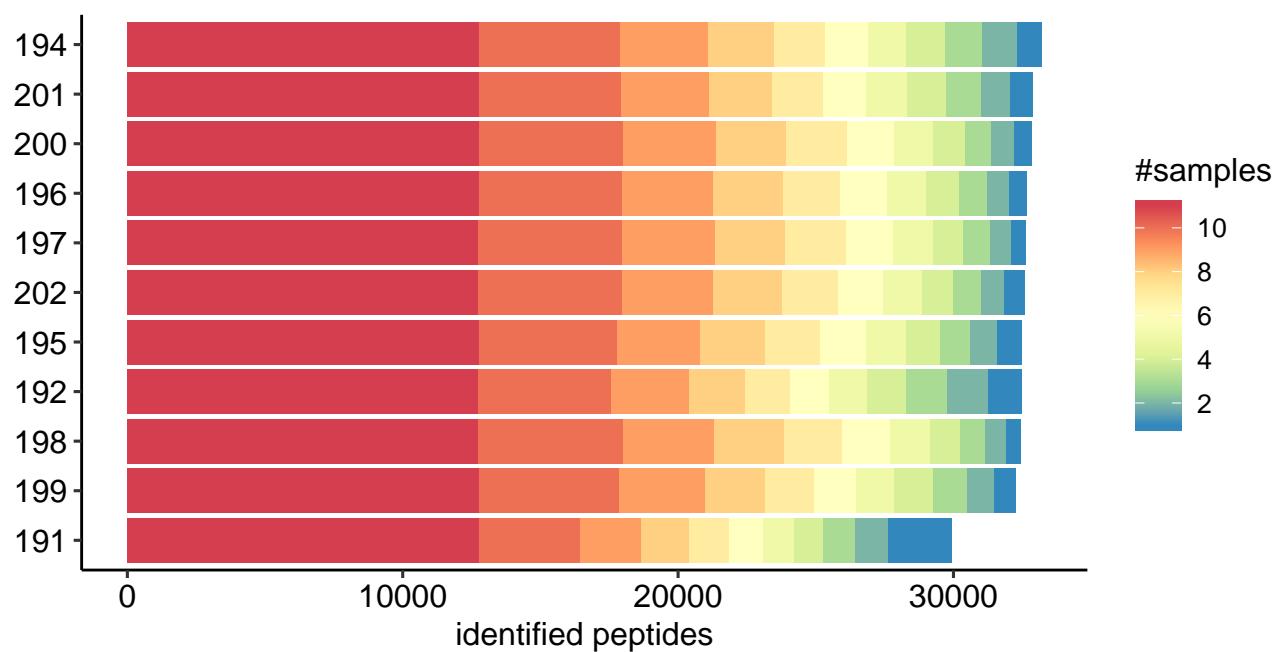
Samples flagged as ‘exclude’ (by user) are not taken into account in this figure. Exact values are shown for data points matching 90% and 50% of samples to convenience comparison between analyses (e.g. before/after configuring ‘exclude’ samples, or comparing between experiments of similar protocol).

### 1.2.2 peptide detection frequency

Each identified peptide in a sample is classified and color-coded by the number of other samples where the same peptide is present. Visualization of the amount of peptides that overlap with other samples in the dataset, from peptides identified in most samples (red) to one-hit-wonders (blue), helps identify uncommon samples (more blue/green than other samples).

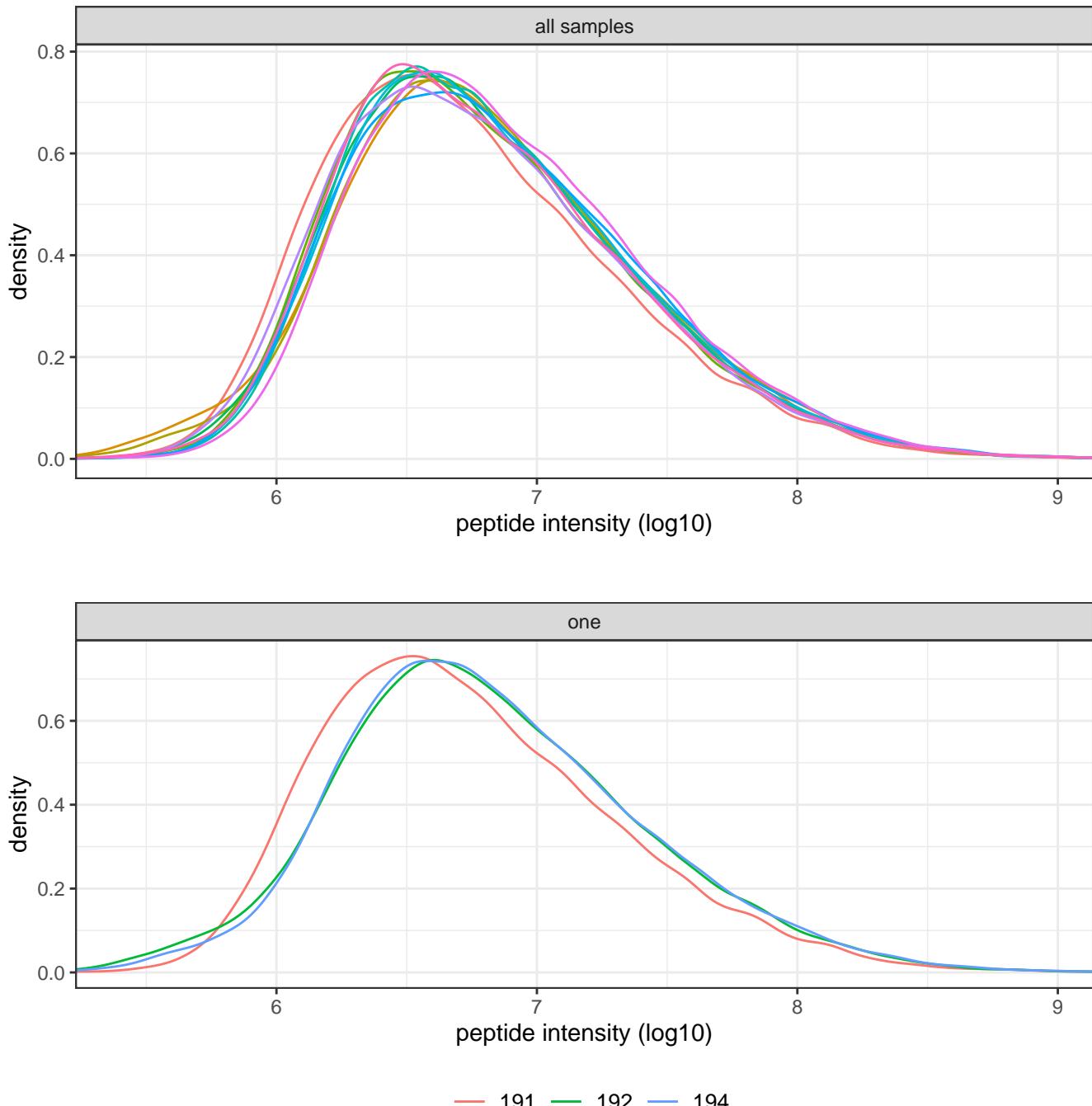
Optimally, the majority of peptides in each sample are red~orange with relatively few uniquely identified peptides (blue~green). Samples are sorted by the total amount of detected peptides.

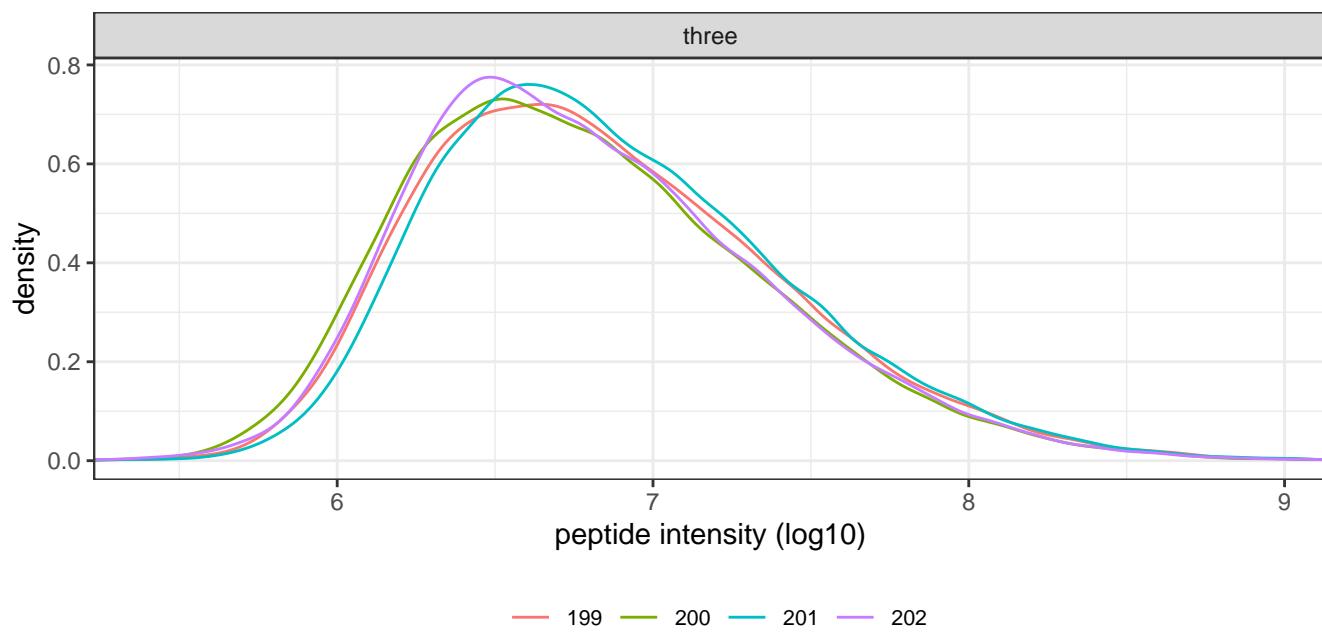
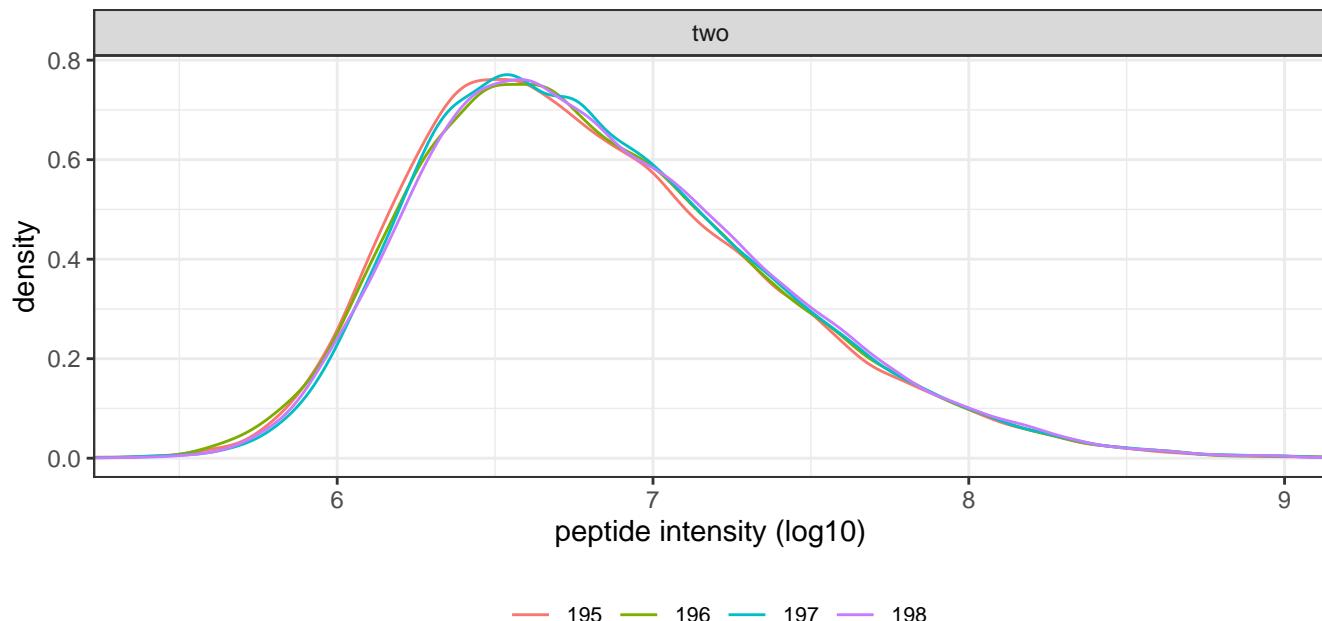
Number of samples in which a peptide is identified vs presence in individual sample



### 1.3 abundance distributions

The figures in this subsection are used to identify unexpected mass-spec sensitivity or sample loading differences. Peptide data is shown as provided in input files, so peptide filtering nor intensity normalization has been applied yet (for proper QC, make sure the software that generated the input data did not apply normalization prior). If the dataset is DDA, match-between-runs (MBR) peptides are included in these distributions whereas for DIA only ‘detected’ peptides (based on confidence score threshold) are included.





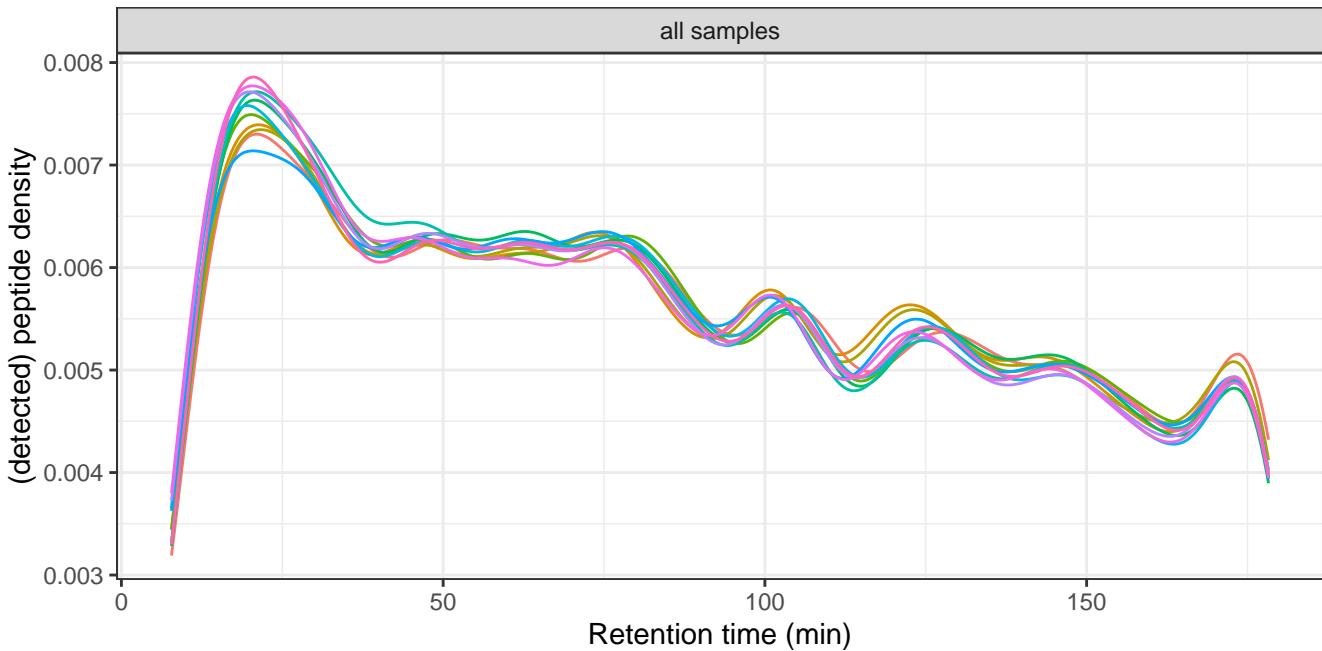
## 1.4 retention time

The figures in this section allow you to identify potential problems during HPLC elution, such as a temporarily blocking column, failing ionization spray or decreasing sensitivity over time. For each sample, all peptides that are also observed in a replicate (such that there is a point of reference available) are visualized.

### 1.4.1 retention time distributions

The density of the number of peptides eluting at each point in time. The figure below presents an overview of all samples that allows for the identification of outlier samples that follow distinct elution patterns. The following section shows details for each sample. Samples marked as ‘exclude’ in the provided sample metadata table are visualized as dashed lines.

progress: RT plots: preparing data took 11 seconds progress: RT plots: creating plots took 2 seconds

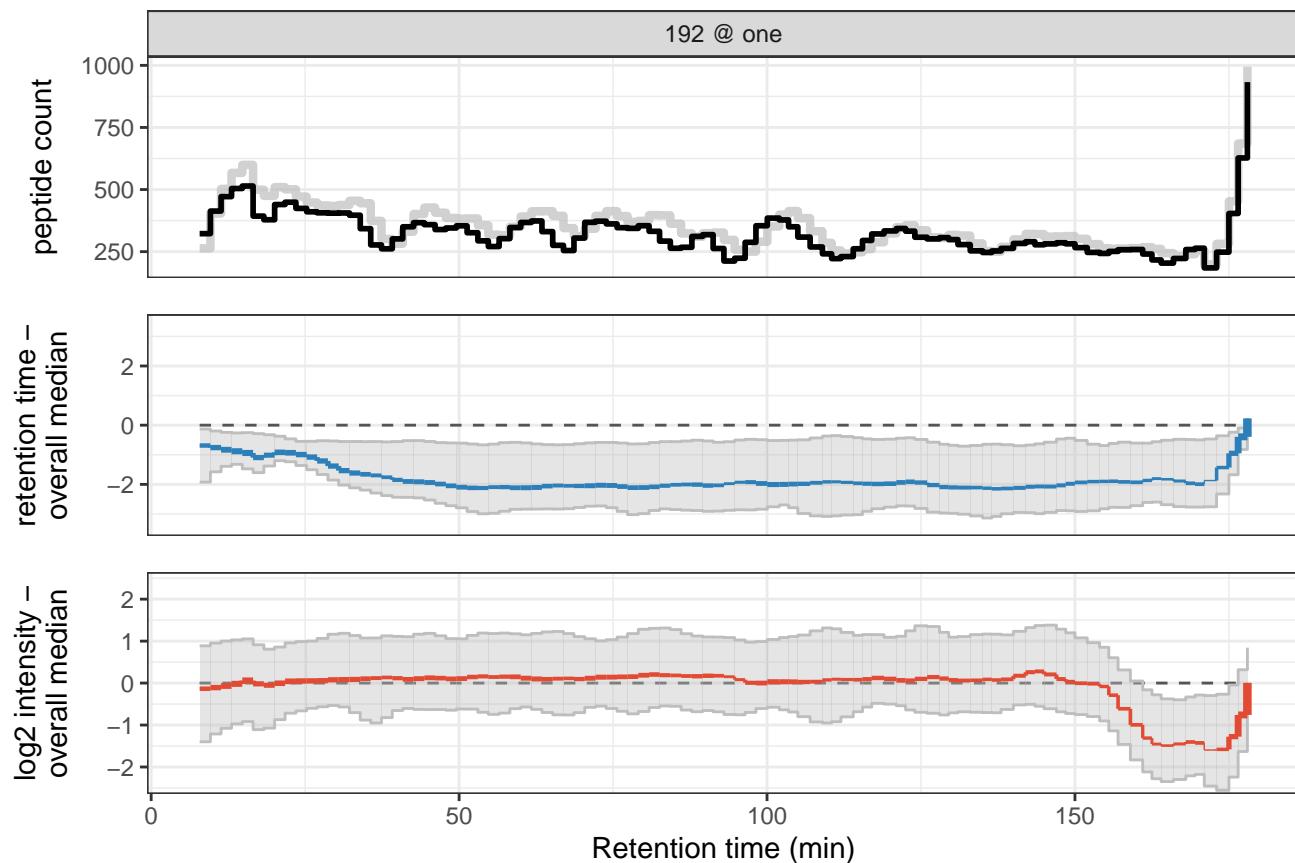
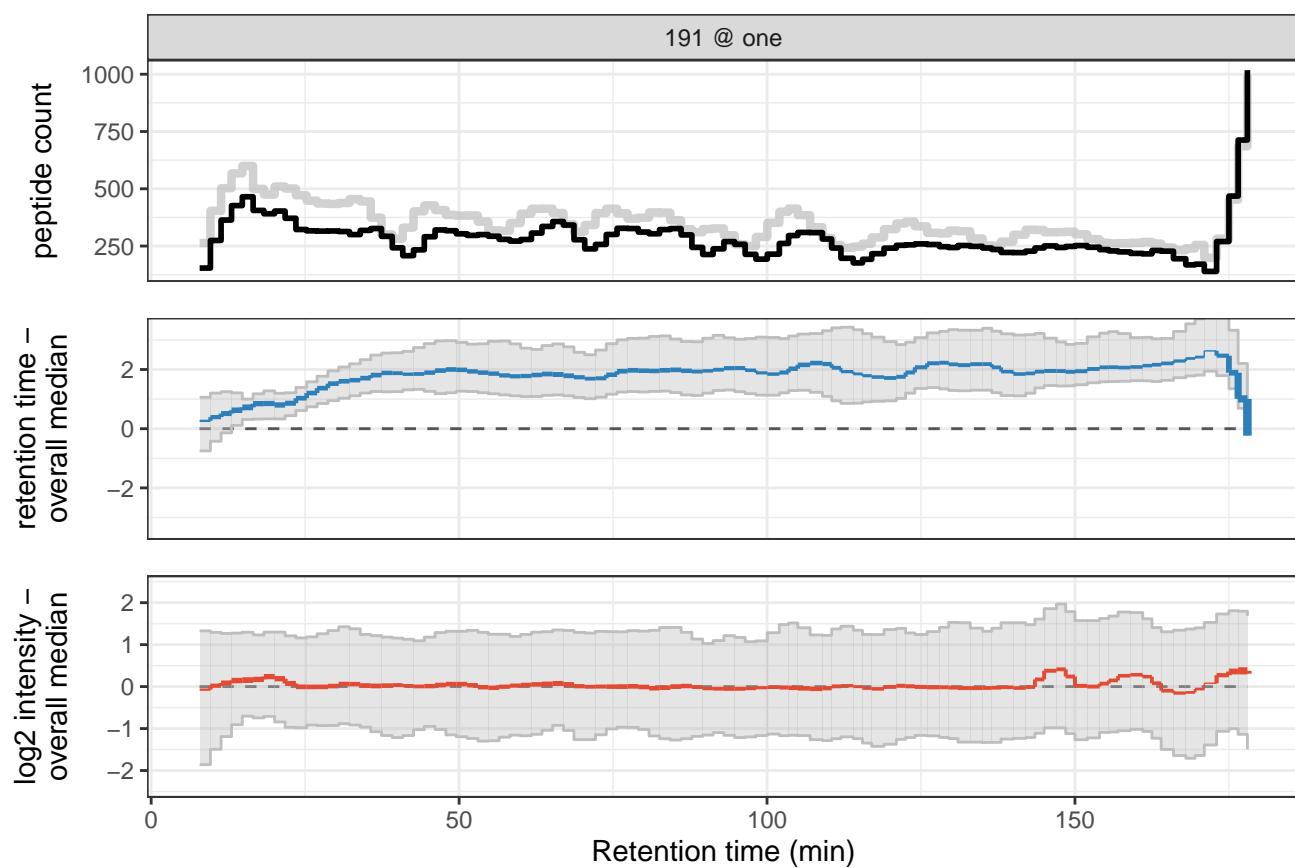


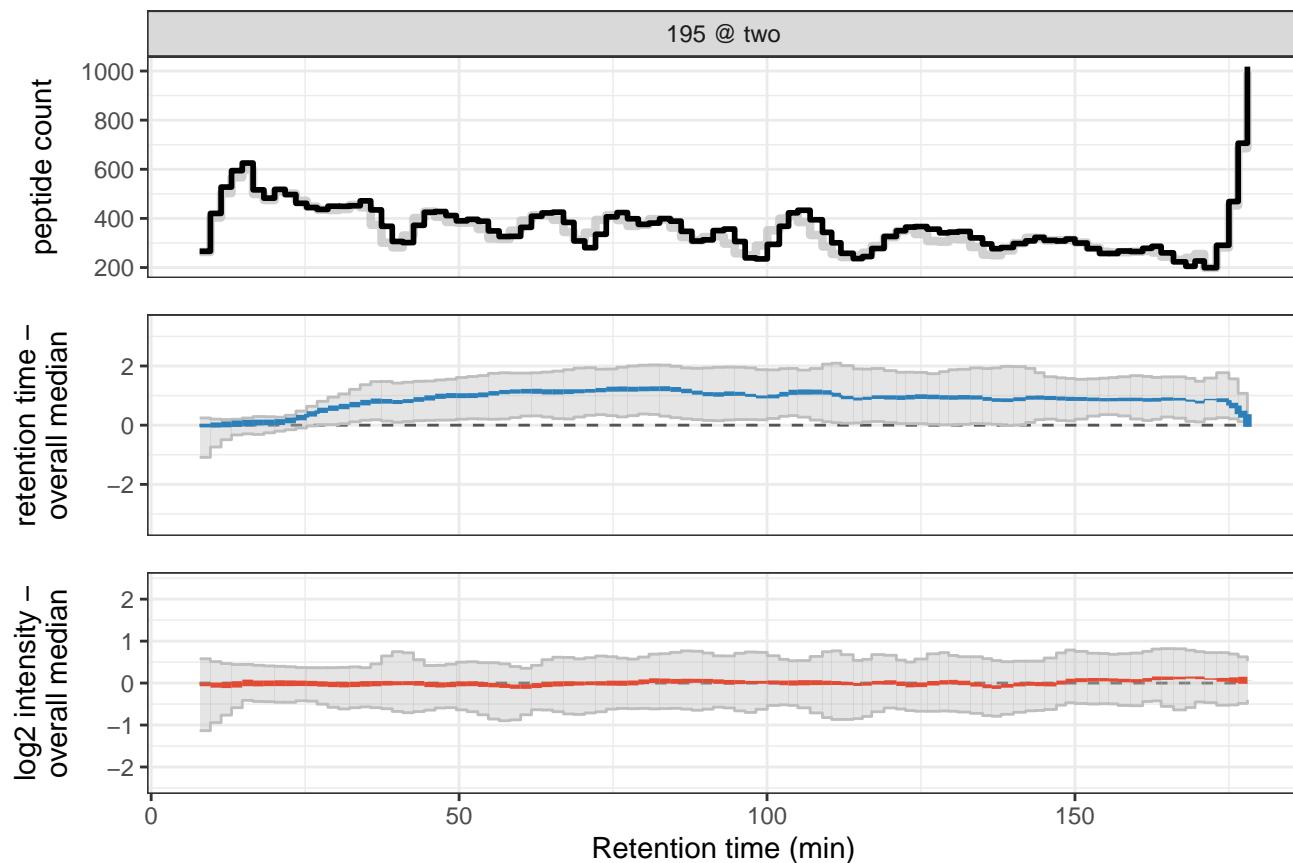
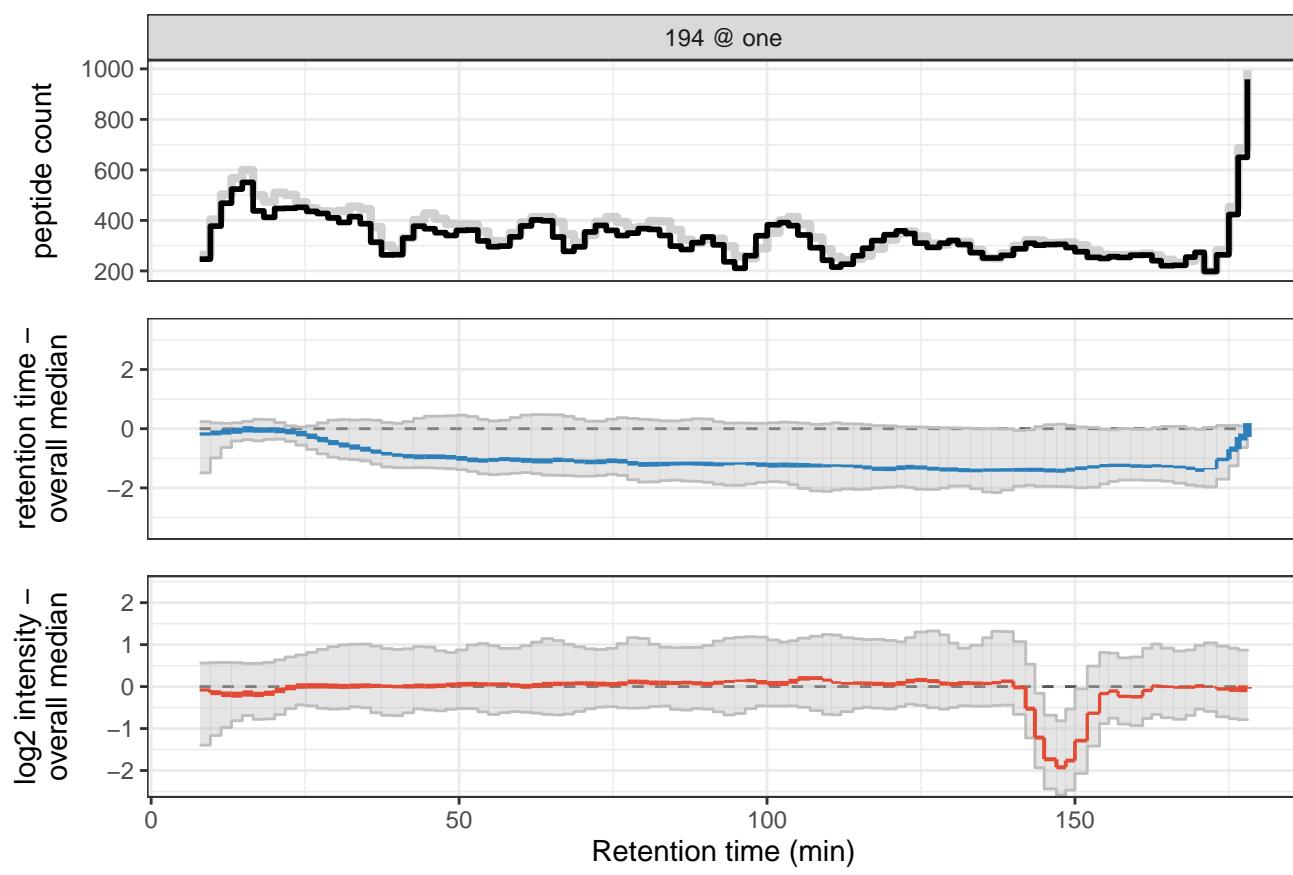
### 1.4.2 retention time local effects

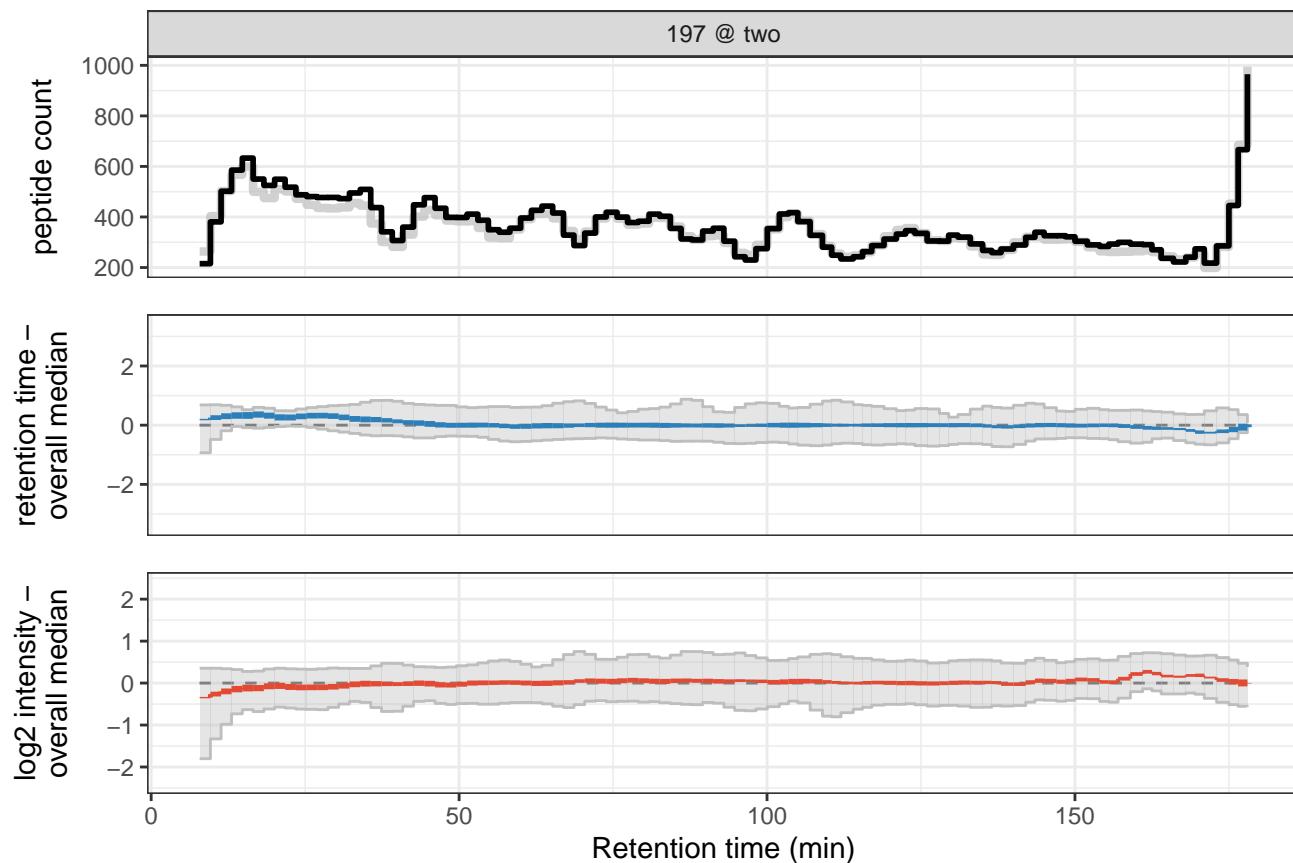
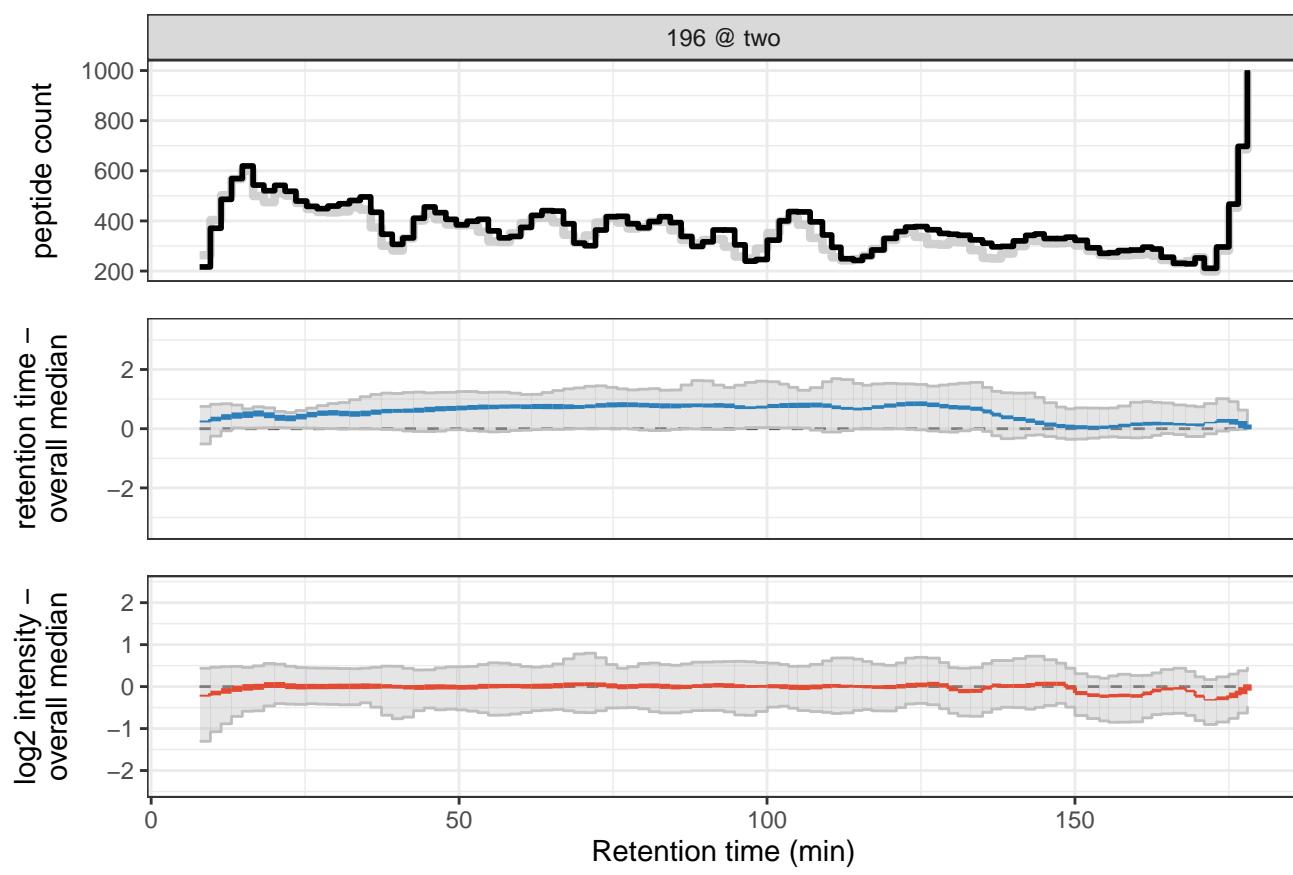
To investigate how each measurement differs from others, we visualize each sample as a 3 panel figure. First, the data is binned across the retention time dimension (x-axis). If a samples was marked as ‘exclude’ in the provided sample metadata, this is indicated in the plot title.

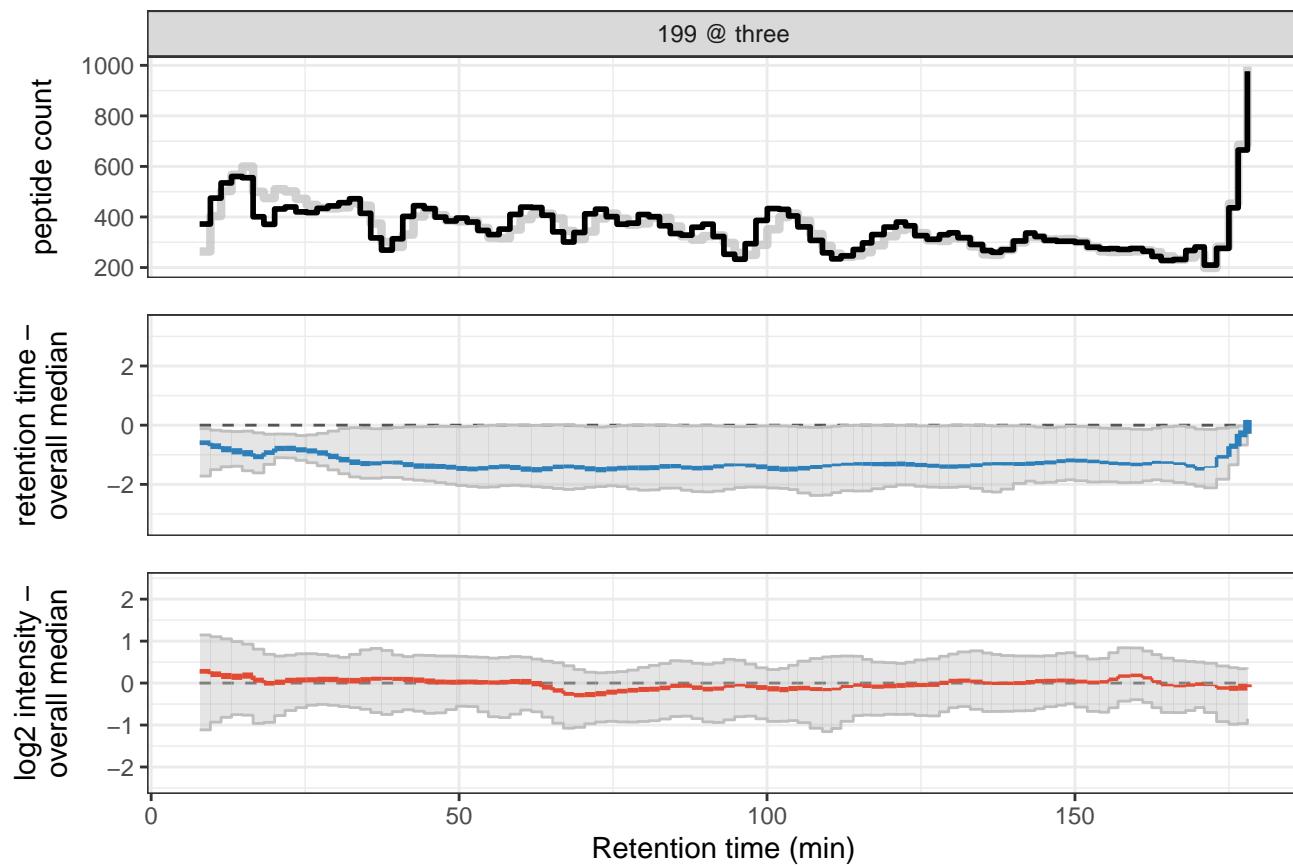
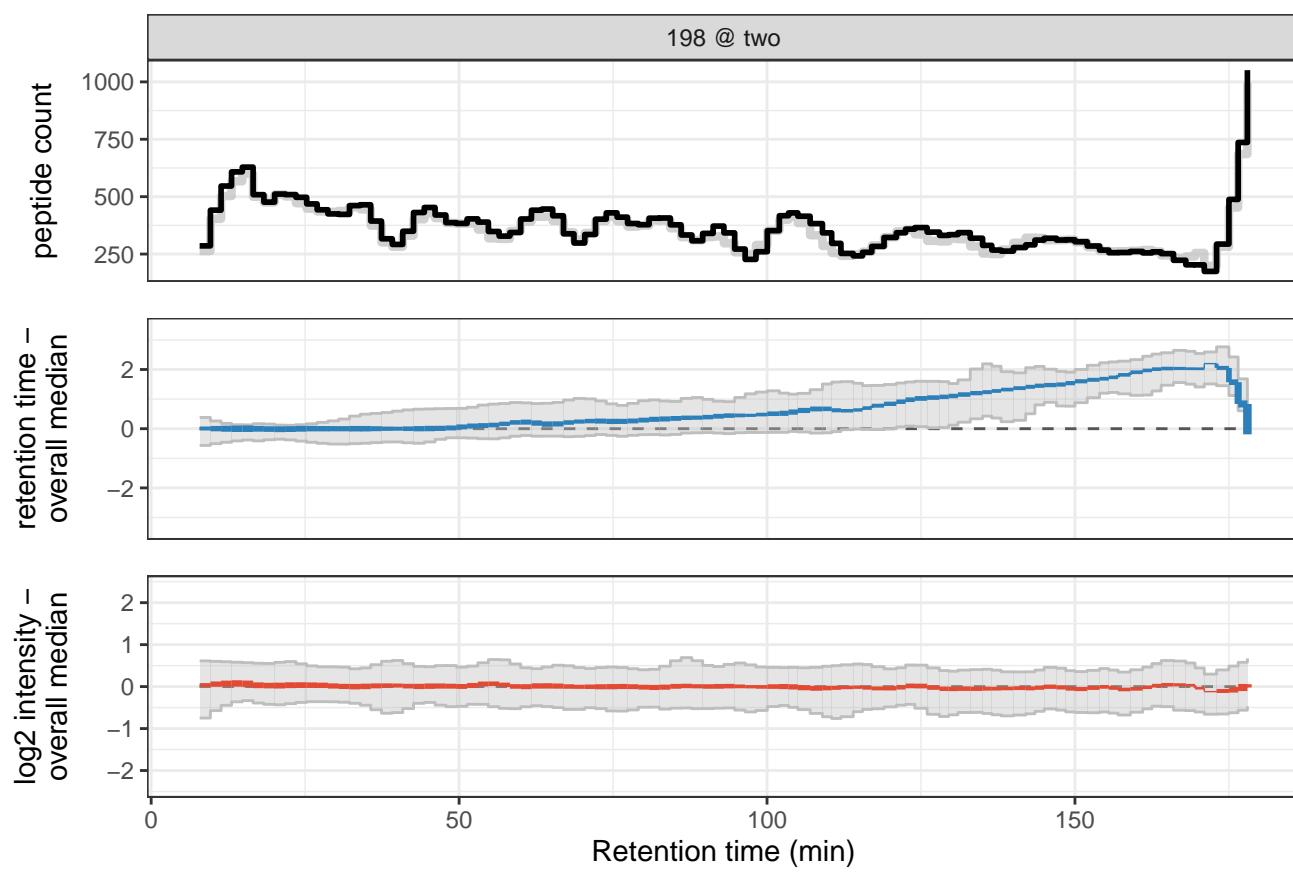
The top panel shows the number of peptides in the input data, e.g. as recognized by the software that generated input for this pipeline, over time (black line). For reference, the grey line shows the median amount over all samples (note; if this is the exact same in all samples, the grey line may not be visible as it falls behind the black line).

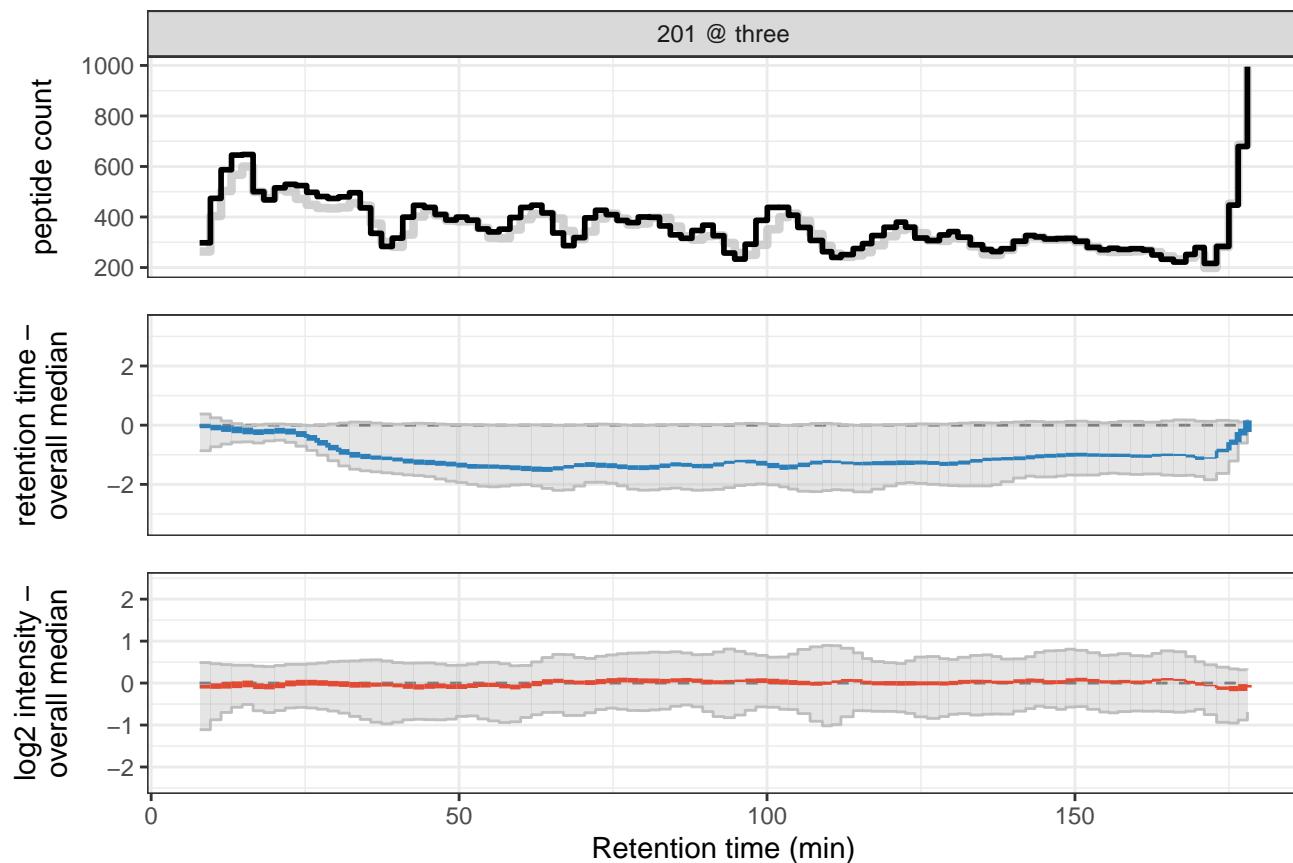
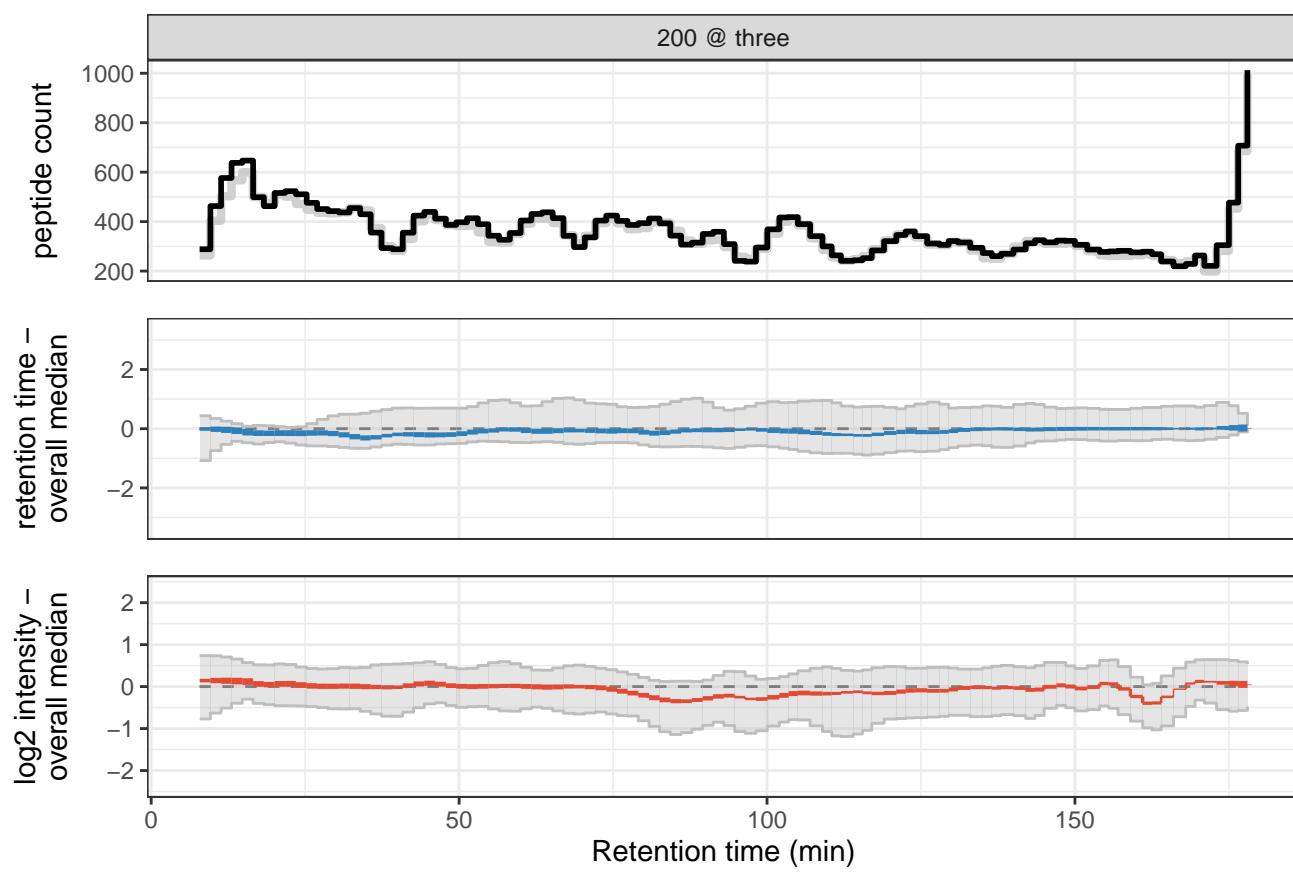
The middle panel indicates whether peptide retention times deviate from their median over all samples (blue line). The grey area depicts the 5% and 95% quantiles, respectively. The line width corresponds to the number of peptides eluting at that time (data from first panel). Analogously, the bottom panel shows the deviation in peptide abundance as compared to the median over all samples (red line).

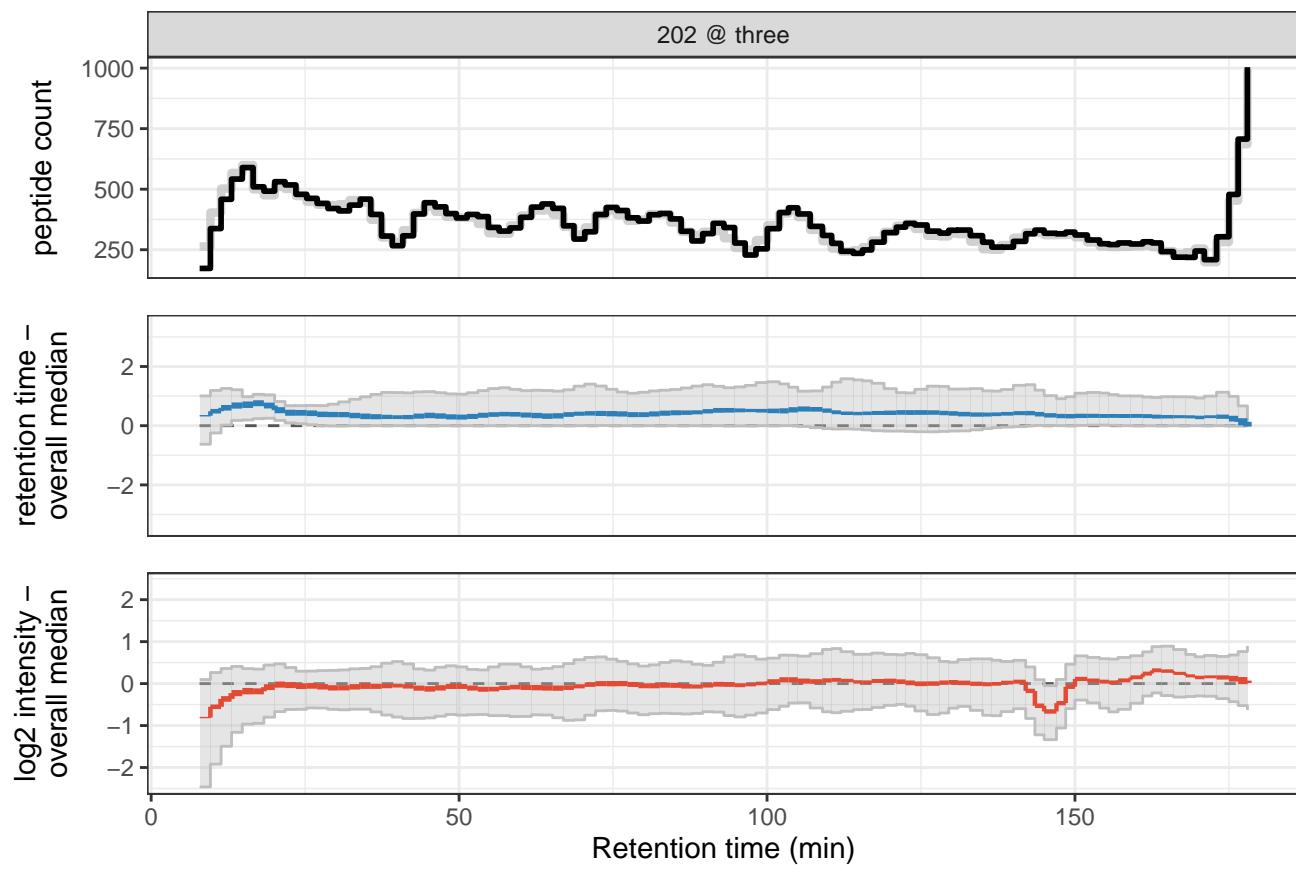










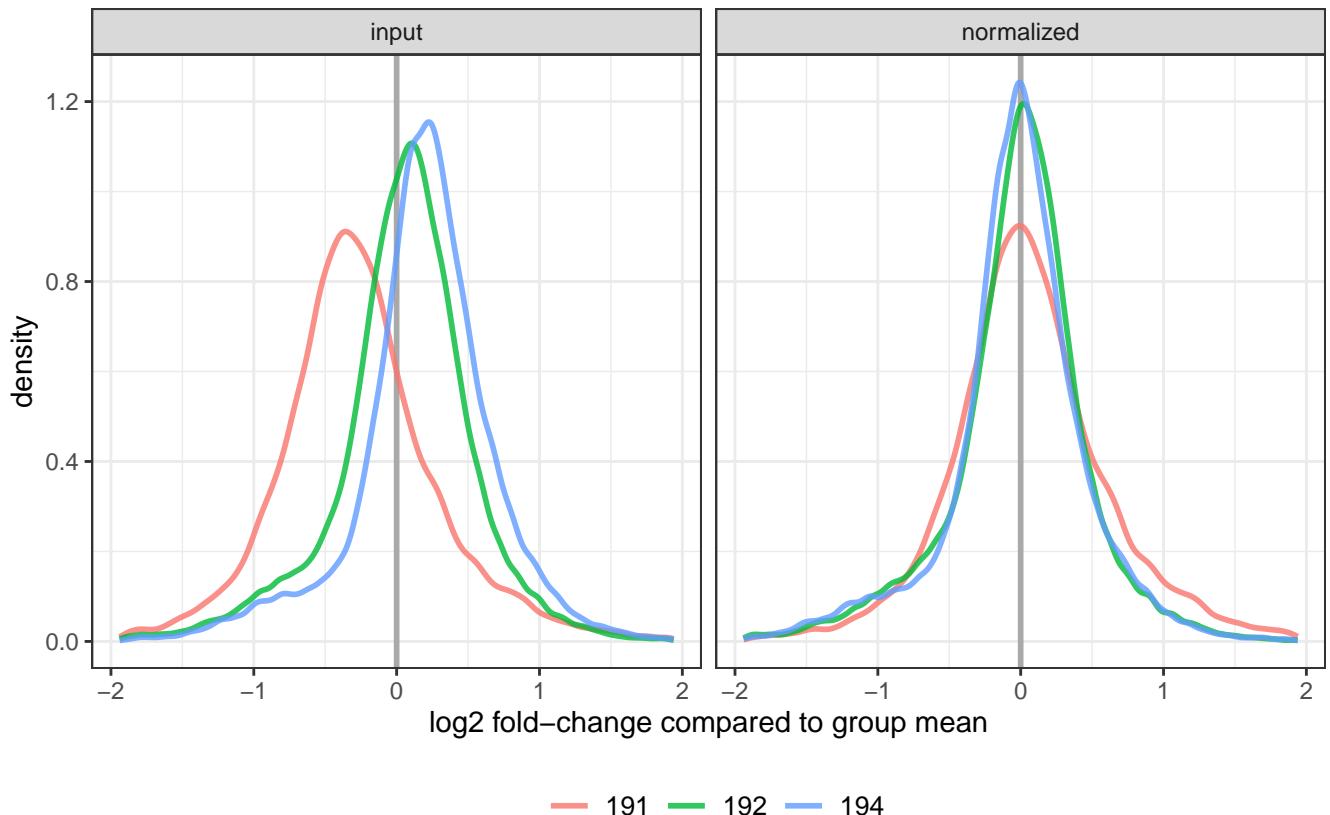


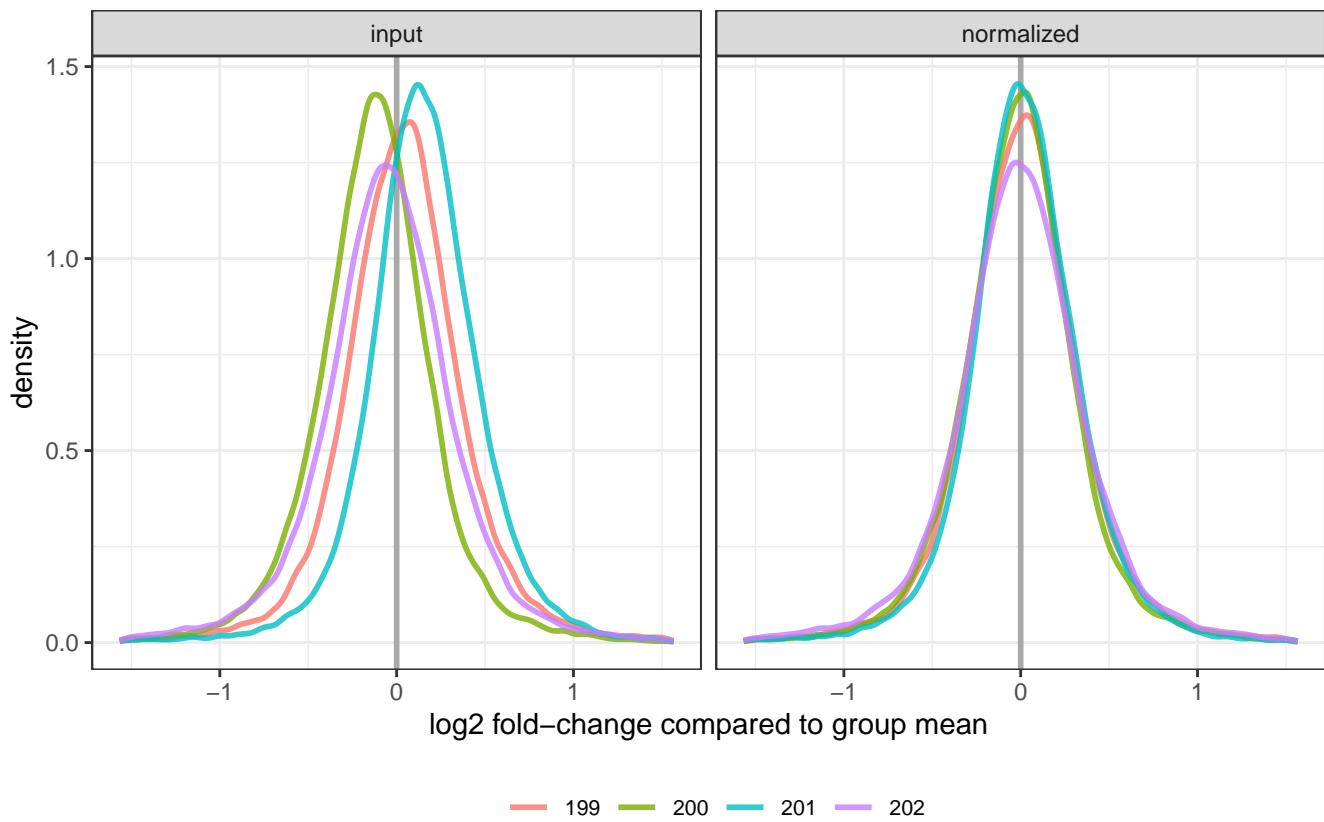
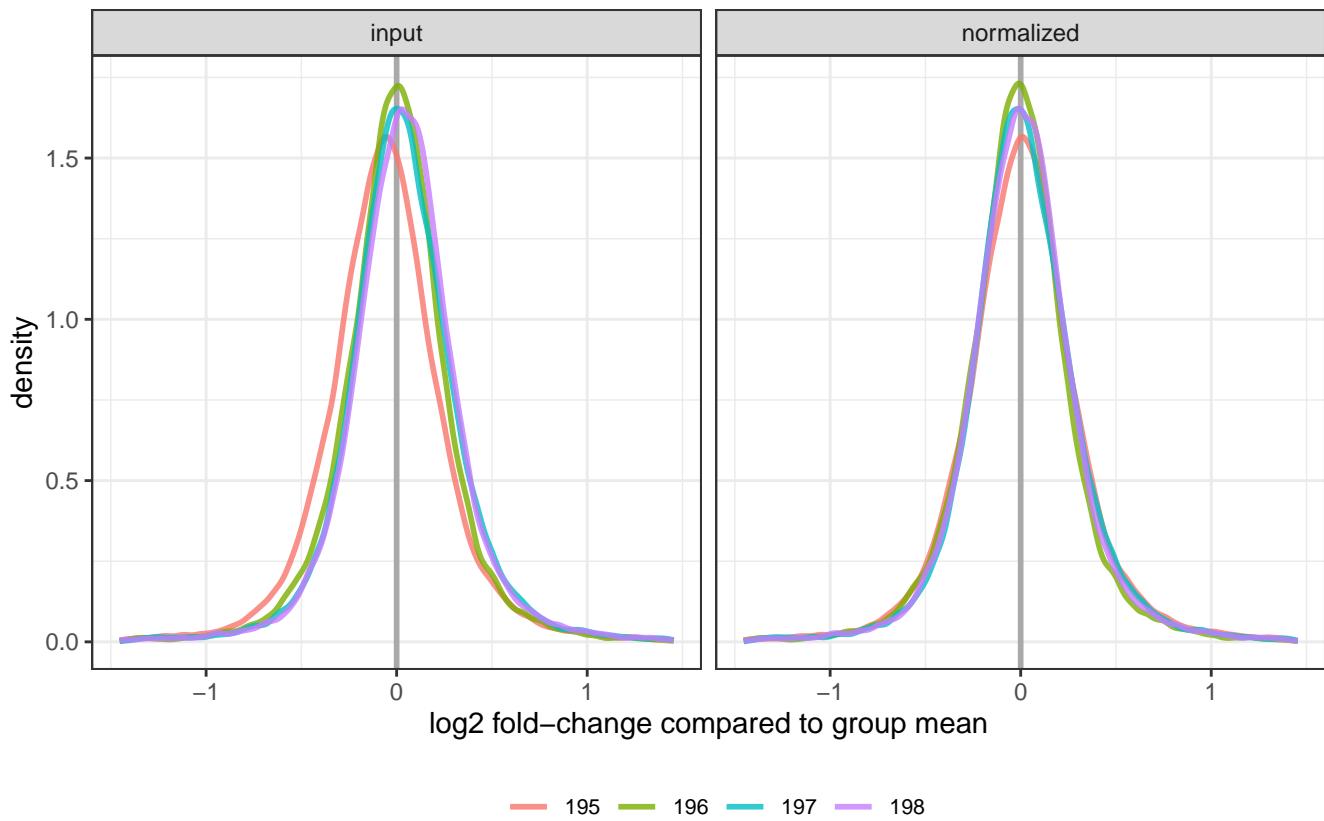
## 1.5 variation among replicates

The reproducibility of replicate measurements is expressed in three different analyses. First, the difference between peptide intensities in each sample are compared to the mean value among all replicates (foldchange distributions). Next, the Coefficient of Variation (CoV) is used as a metric for reproducibility to explore how much the CoV within a sample group can be improved by removing a single sample (eg; if CoV strongly improved after removing sample s, it could be regarded as an outlier). Finally, the CoV within each sample group is visualized as a boxplot and a violin plot, figures commonly seen in proteomics literature and useful for comparing across experiments (of similar protocol).

### 1.5.1 within-group foldchange distributions

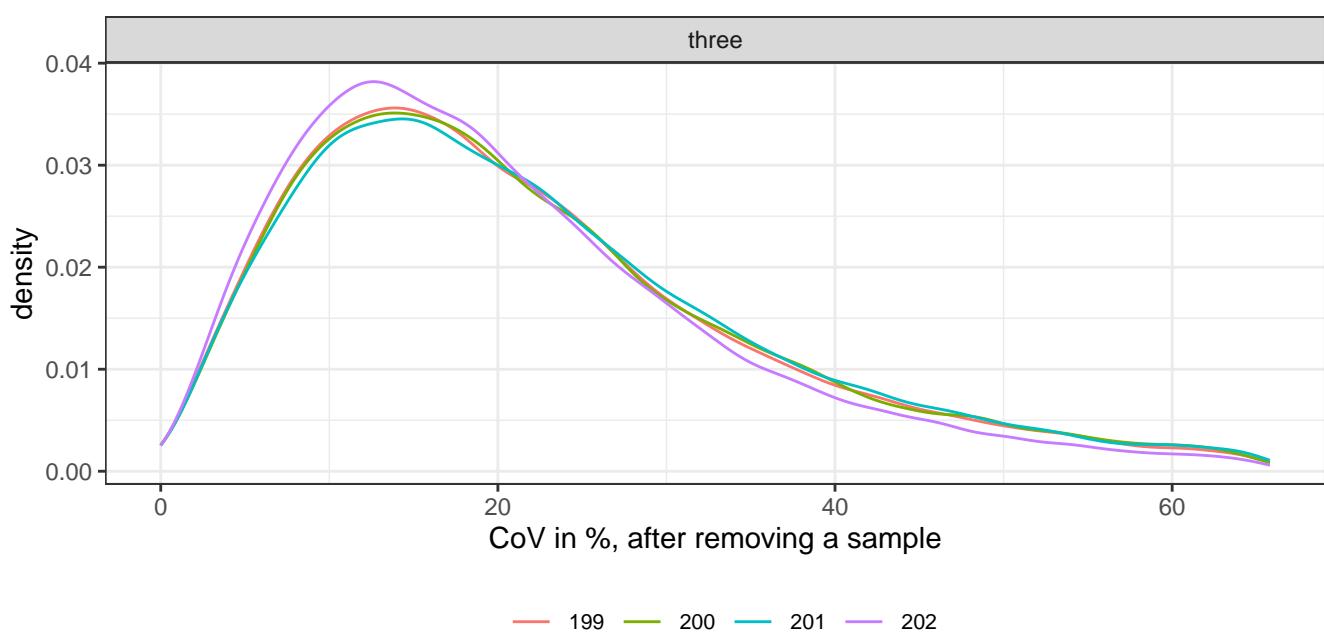
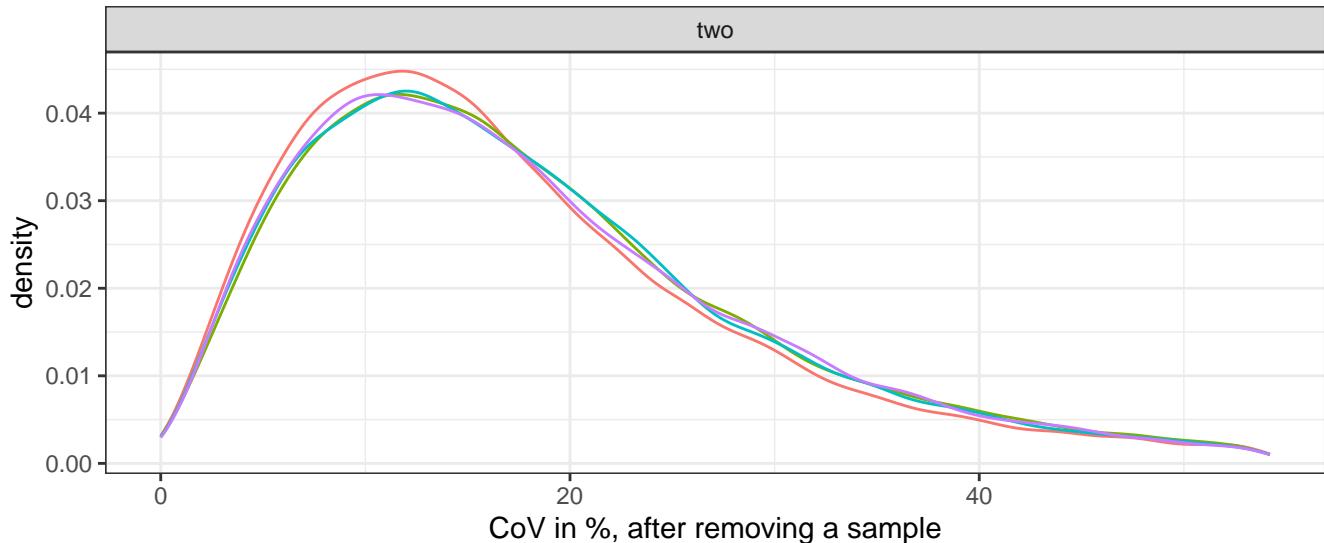
The foldchange of all peptides in a sample is compared to their respective mean value over all samples in the group. This visualizes how strongly each sample deviates from other samples in the same group which helps identify outlier samples. The same data was used as detailed in the “retention time” section above. Samples marked as ‘exclude’ in the provided sample metadata table are visualized as dashed lines.



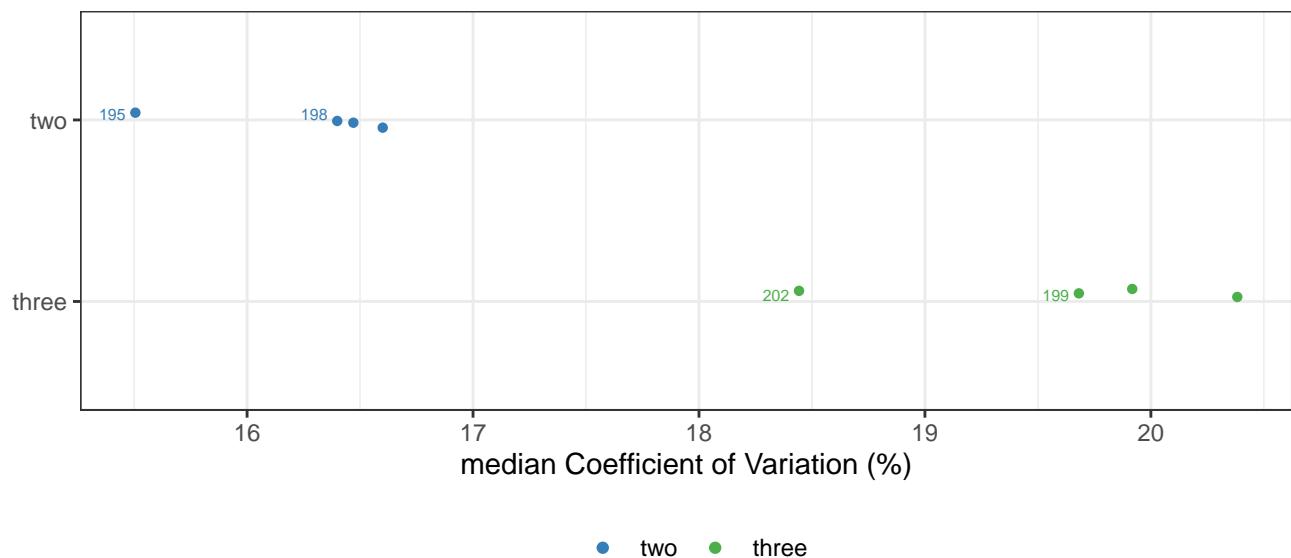


### 1.5.2 CoV, leave-one-out

progress: leave-one-out CoV plot computations took 4 seconds The figures below describe the effect of removing a particular sample prior to within-group Coefficient of Variation (CoV) computation. The lower the CoV distribution is for a sample, the better reproducibility we get by excluding it. Only sample groups with at least 4 replicates can be used for this analysis, so 3 samples remain after leaving one out. Samples marked as ‘exclude’ in the provided sample metadata are included in these analyses (shown as dashed lines), and only peptides with at least 3 data points across replicates samples (after leave-one-out) are used for each CoV computation.



Effect of removing a sample prior to CoV computation on within-group CoV  
 lower value = better CoV after removing sample s



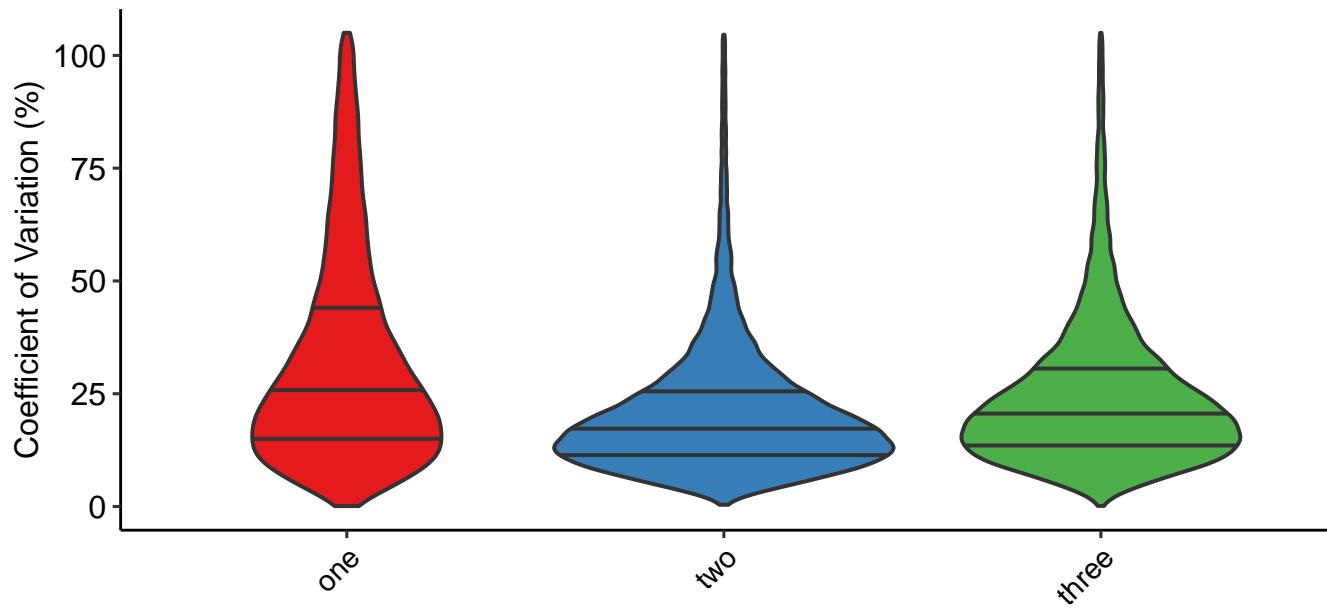
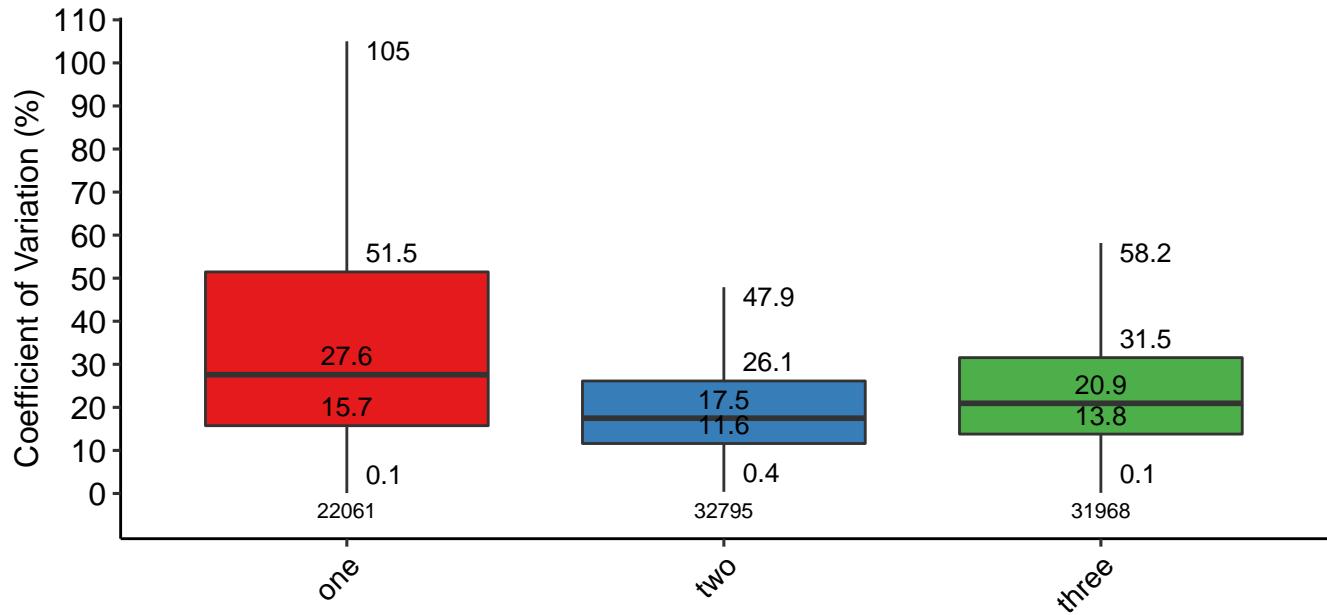
shortname	group	exclude	median CoV
195	two	FALSE	15.5
198	two	FALSE	16.4
197	two	FALSE	16.5
196	two	FALSE	16.6
202	three	FALSE	18.4
199	three	FALSE	19.7
200	three	FALSE	19.9
201	three	FALSE	20.4

*Leave-one-out impact on within-group CoV (%)*

### 1.5.3 Coefficient of Variation

The Coefficient of Variation (CoV) is a quality metric for the reproducibility of replicate measurements, here visualized using box- and violin-plots.

The same data was used as detailed in the “retention time” section above, with only peptides quantified in at least 3 replicates used for CoV computation in a group. Only samples that are NOT marked ‘exclude’ in the provided sample metadata and are in a sample group among at least 3 replicates are used for these figures.



## 1.6 PCA

A visualization of the first three PCA dimensions illustrates sample clustering. The goal of these figures is to detect global effects from a quality control perspective, such as samples from the same experiment batch clustering together, not to be sensitive to a minor subset of differentially abundant proteins (for which specialized statistical models can be applied downstream).

If additional sample metadata was provided, such as experiment batch, sample-prep dates, gel, etc., multiple PCA figures will be generated with respective color-codings. Users are encouraged to provide relevant experiment information as sample metadata and use these figures to search for unexpected batch effects.

The pcaMethods R package is used here to perform the Probabilistic PCA (PPCA). The set of peptides used for this analysis consists of those peptides that pass your filter criteria in every sample group. If any samples are marked as ‘exclude’ in the provided sample metadata, an additional PCA plot is generated with these samples included (depicting the ‘exclude’ samples as square symbols).

### Rationale behind data filter

As mentioned above, the aim of the PCA figures is to identify global effects. To achieve this, we compute sample distances on the subset of peptides identified in each group which prevents rarely detected peptides/proteins from having a disproportionate effect on sample clustering. This pertains not only to ‘randomly detected contaminant proteins’ but also to proteins with abundance levels near the detection limit, which may be detected in only a subset of samples (eg; some measurements will be more successful/sensitive than others).

### Figure legends

The first 3 principle components compared visually (*1 vs 2*, *1 vs 3*, *2 vs 3*) on the rows. Left- and right-side panels on each row represent the same figure without and with sample labels. The principle components are shown on the axis labels together with their respective percentage of variance explained. Samples marked as ‘exclude’ in the provided sample metadata, if any, are visualized as square shapes.

## 2 Differential abundance analysis

### goal: maximize reliable features for quantification

In a pairwise analysis of two groups of samples, only peptides with N data-points in both groups are used for quantitative analysis (where N = defined by user settings). For example; if peptide  $p$  is consistently quantified in sample groups A and B but not in C/D/E, it can be used when comparing group A *versus* group B but should not be used in any other group comparisons. This approach is particularly suited to maximize the number of peptides used for statistical analysis in experimental designs with many sample groups.

A common alternative strategy is a global filtering approach where peptides are selected based on their properties in the overall dataset (eg; present in x% of samples or x% of replicates in all groups) and subsequently the resulting data matrix is used for all downstream statistical analyses. In the example above where peptide  $p$  is present in a subset of sample groups,  $p$  would either be left out (not present in majority of samples in entire dataset) or erroneously used when applying t-statistics to groups B and C (since  $p$  is not present in group C, it may differentially detected but there are no features available for quantitative analysis)

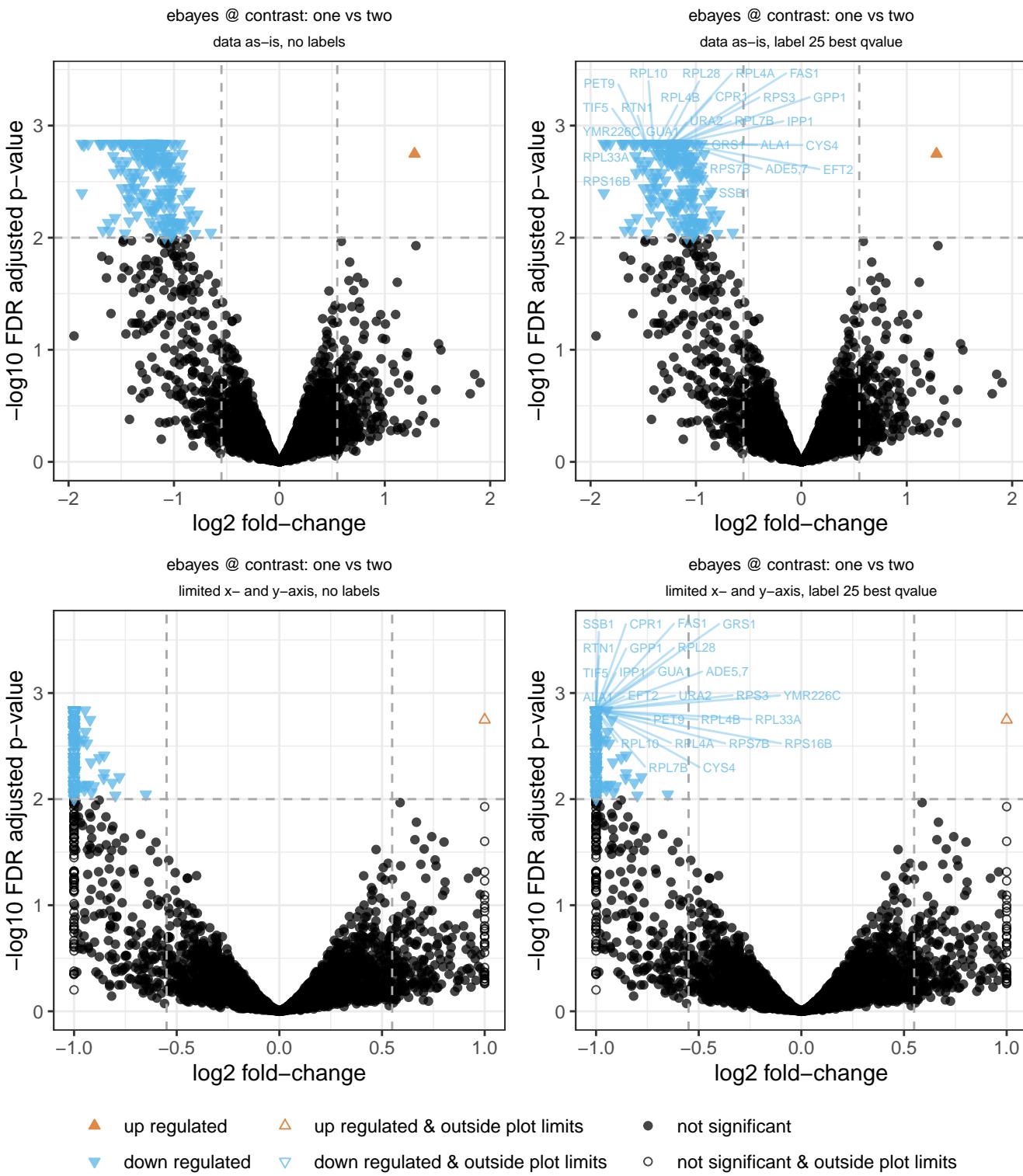
### 2.1 one vs two

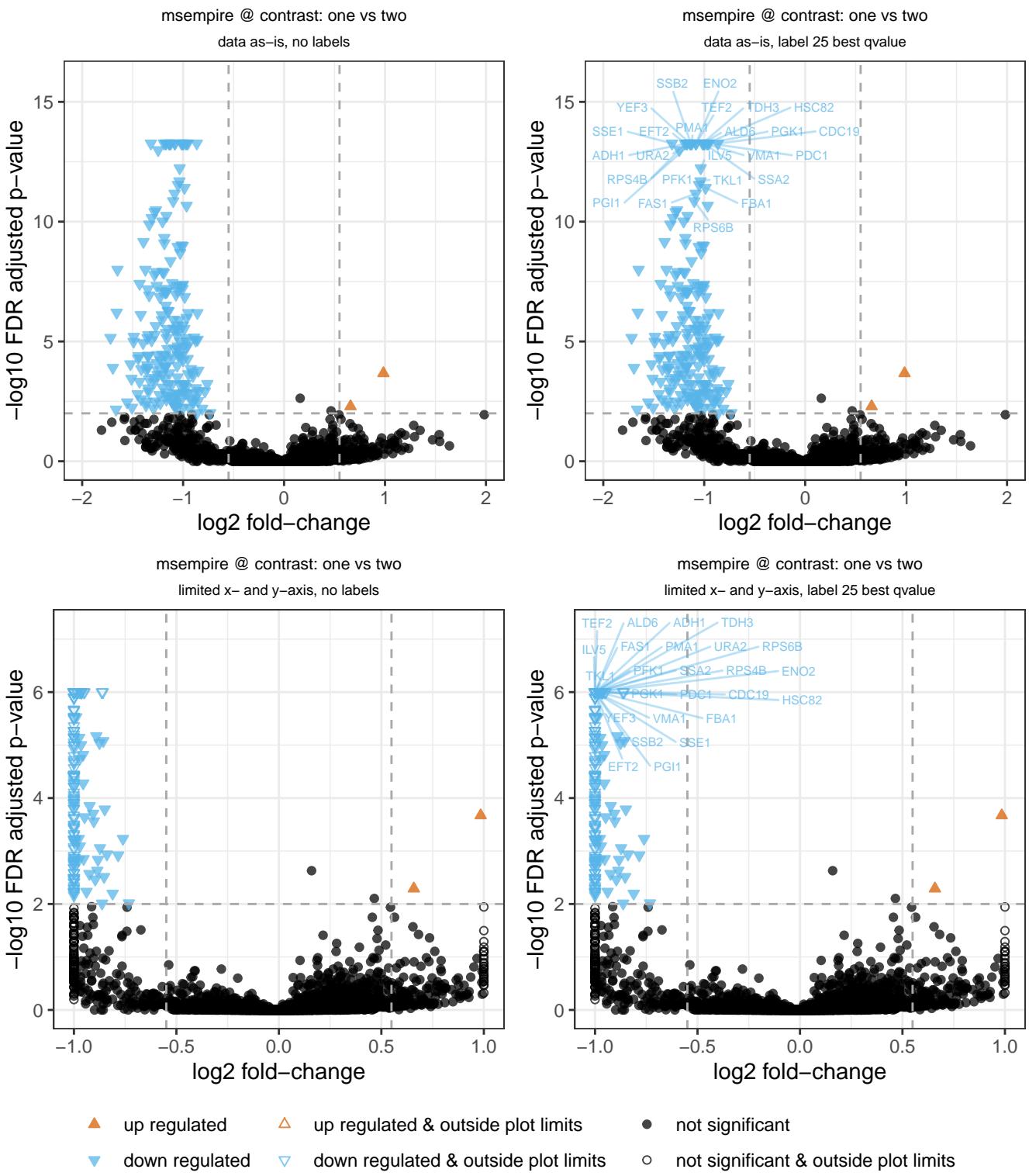
- **user setting:** using ‘filter by contrast’ peptide filtering approach
- 57302 peptides in 7320 proteins remain in the current contrast after peptide filters and are used for the statistical analysis in this section
- qvalue threshold: 0.01
- log2 foldchange threshold: 0.549375

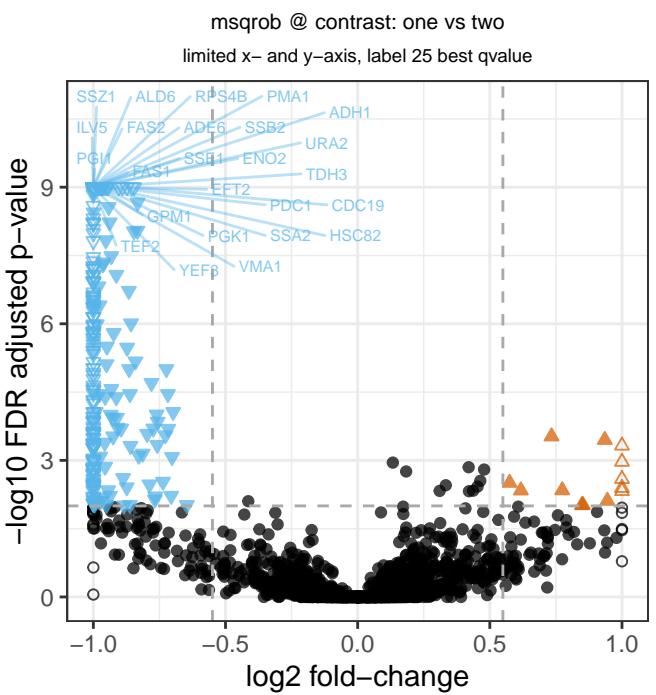
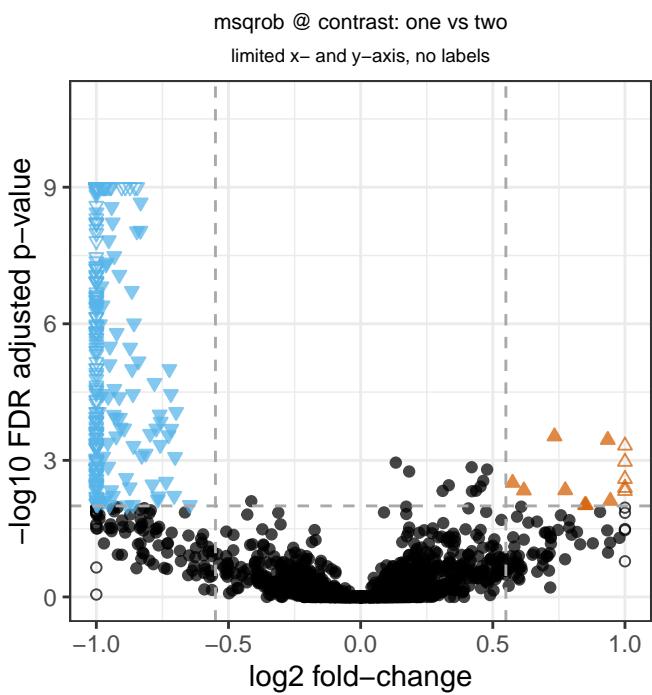
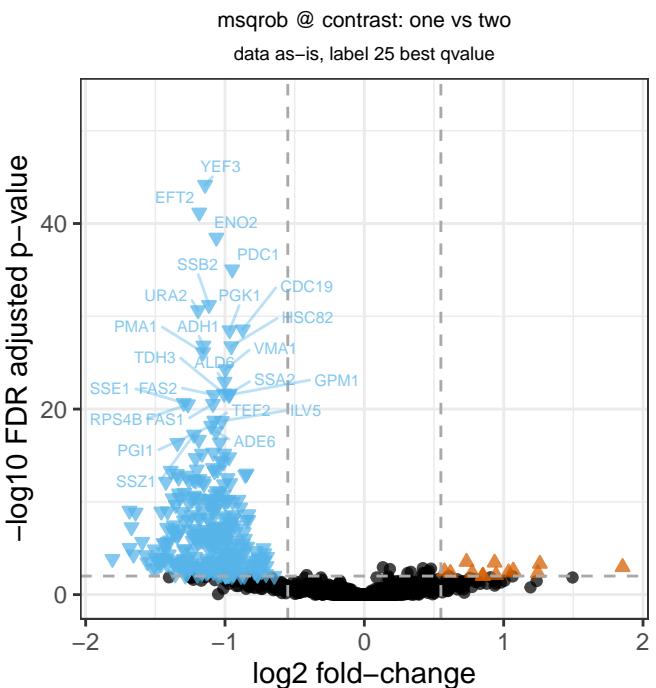
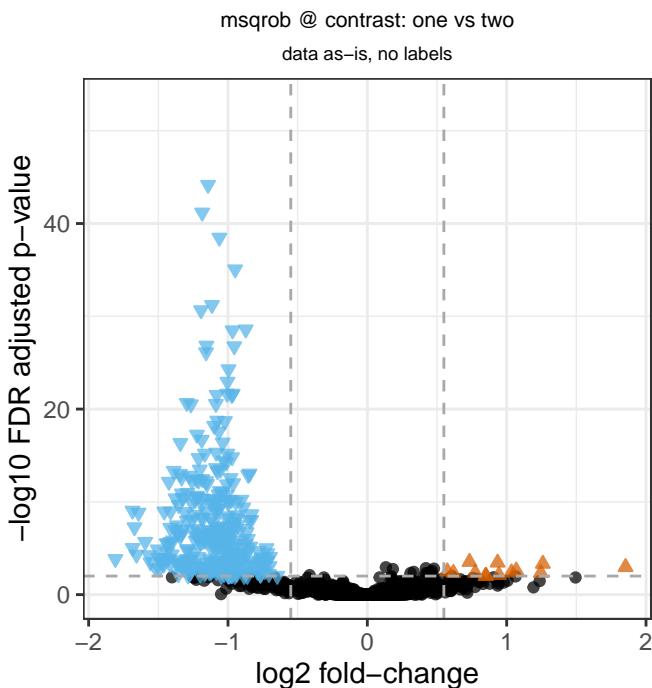
#### 2.1.1 volcano

The plot title shows the statistical model and contrast (sample groups in the comparison). Left- and right-side figure panels on each row represent the same figure without and with labels for the 25 proteins with lowest p-value.

Bottom figure panels have limited x- and y-axis. For datasets with a small number of strong outliers in p-value or fold-change, which may have a profound effect on the plot scales, this allows inspection of the remainder of the volcano plot without disproportionate influence by ‘extreme’ values.





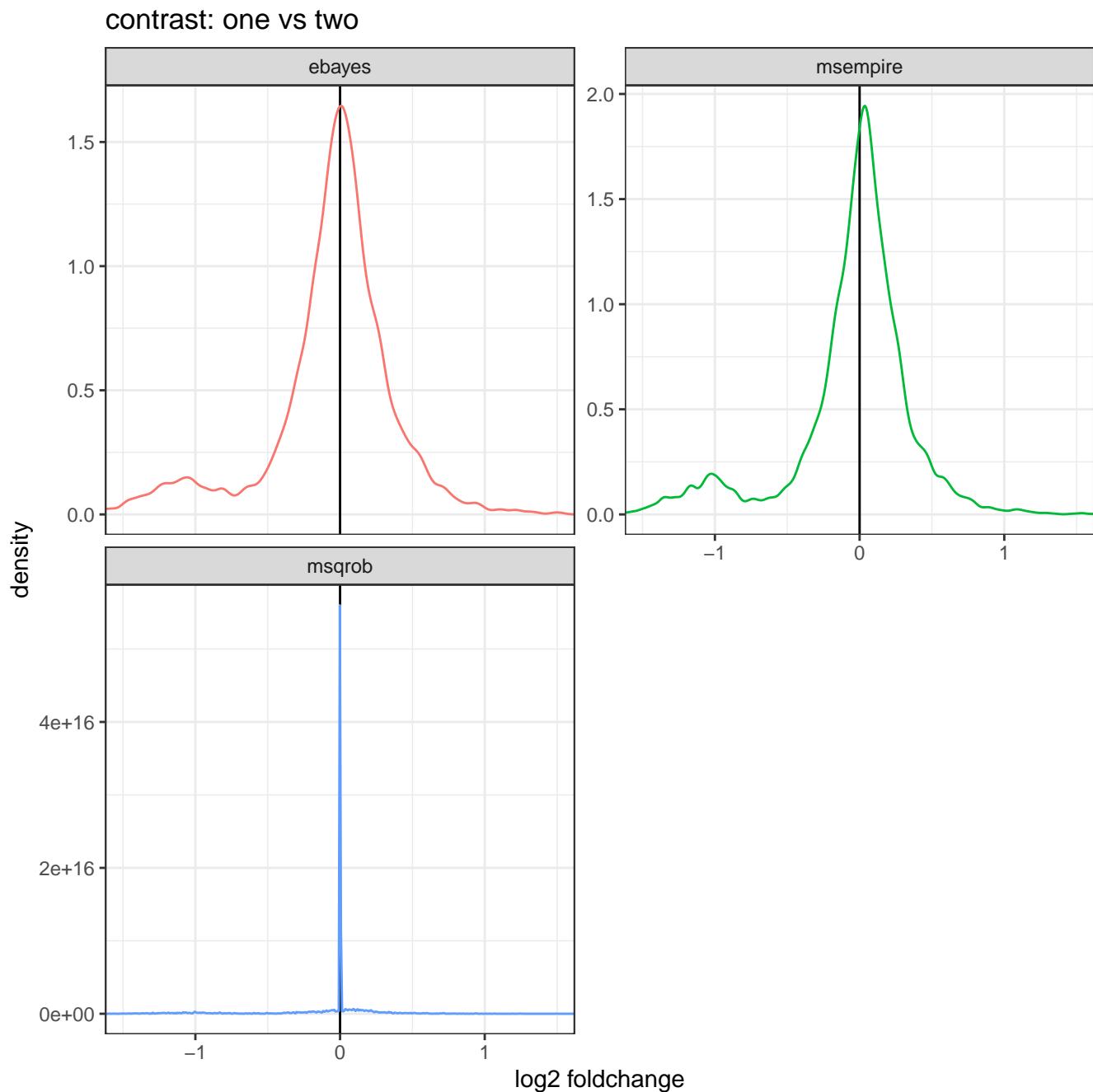


- ▲ up regulated ▲ up regulated & outside plot limits ● not significant
- ▼ down regulated ▼ down regulated & outside plot limits ○ not significant & outside plot limits

### 2.1.2 foldchange distribution

Distributions of estimated foldchanges produced by the statistical models. If the mode is far from 0, consider alternative normalization strategies. Do note the scale on the x-axis, for some experiments the foldchanges are very low which in turn may exaggerate this figure.

*note; the MSqRob model tends to assign zero (log)foldchange for proteins with minor difference between conditions where the model is very sure the null hypothesis cannot be rejected (shrinkage by the ridge regression model). As a result, many foldchanges will be zero and the density plot for MSqRob may look like a spike instead of the expected Gaussian shape observed in other models*



### 2.1.3 p-value distribution

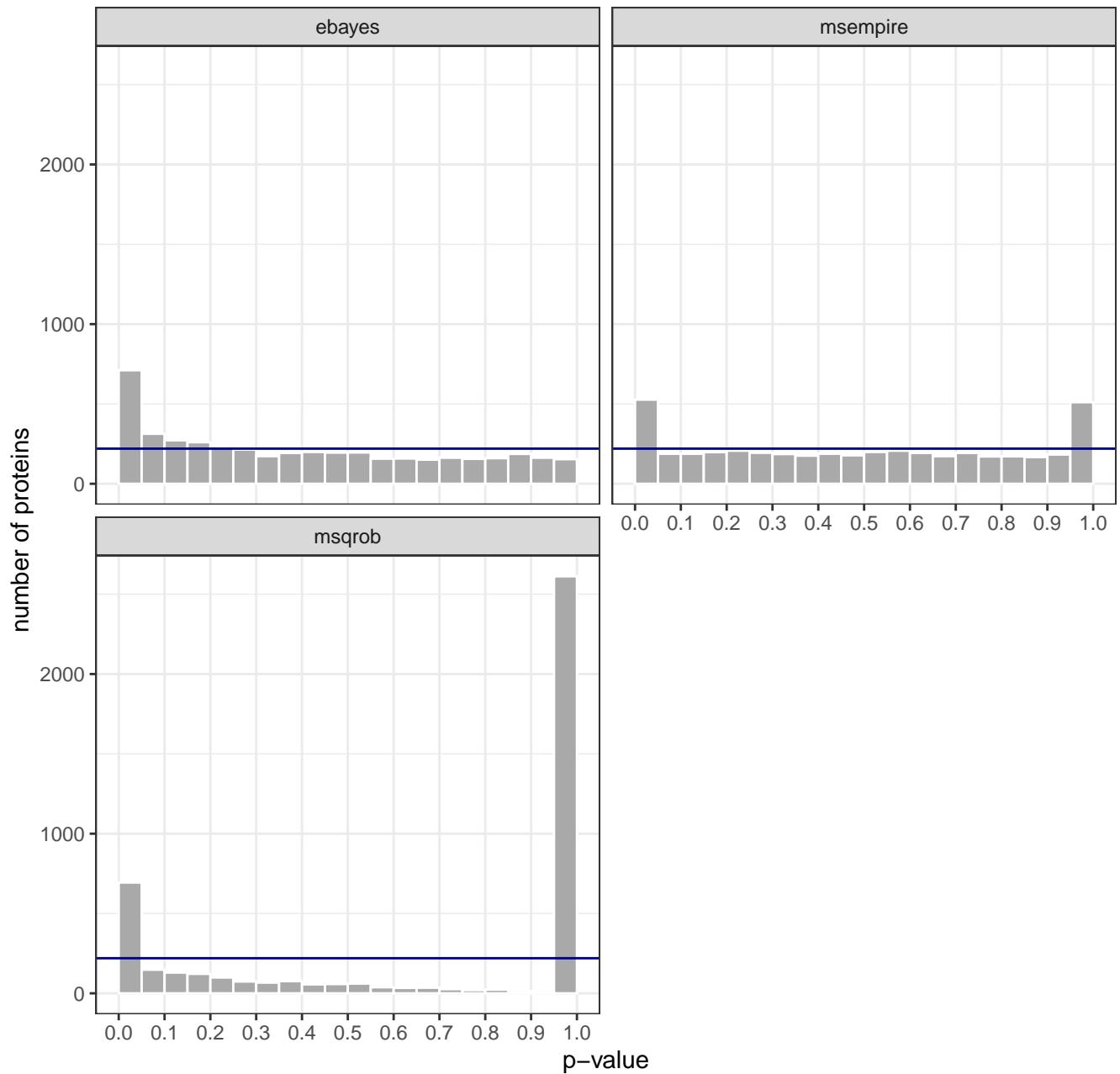
Histogram of p-values computed by differential expression analysis algorithms, as-is, for quality-control inspection. The horizontal line indicates the expected counts assuming a uniform distribution (total number of p-values divided by number of histogram bins)

See further: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6164648/>

See further: <http://varianceexplained.org/statistics/interpreting-pvalue-histogram/>

*note; the MSqRob and MS-Empire models often yield p-value distributions that show a large peak at p-value 1, these are typically proteins with estimated log foldchanges at/near zero where these models are very sure the null hypothesis cannot be rejected*

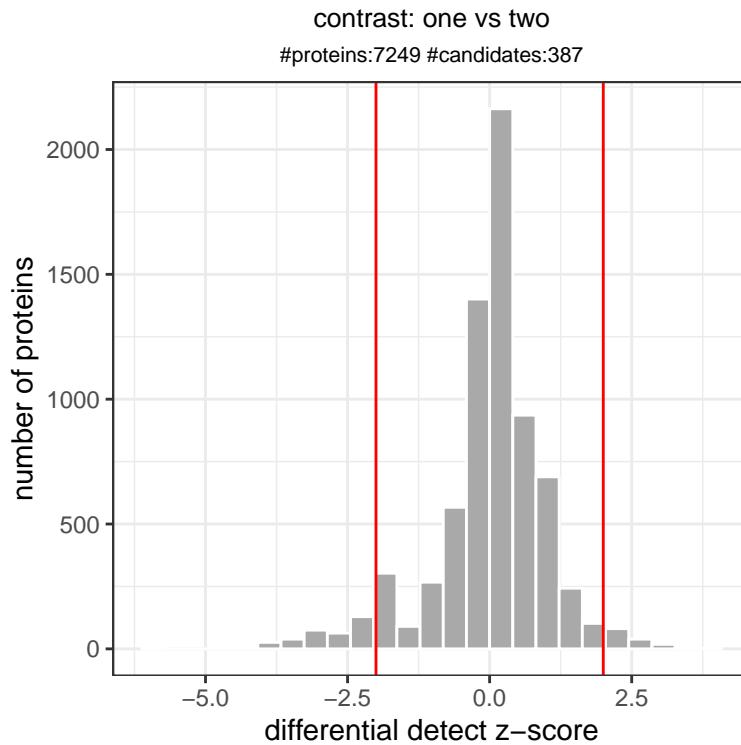
contrast: one vs two



#### 2.1.4 differential detect

Some proteins may not have peptides with sufficient abundance values for differential expression analysis, but do show strong differences in the number of detected peptides between sample groups. A simple z-score is computed for each protein based on the number of detected peptides per sample group. DEA is much more powerful for differential testing, but these z-scores can offer complementary information in some experimental designs (e.g. a wildtype-knockout APMS study).

Below figure shows the distribution of these scores with thresholds at 2 std. Both the z-scores and the counts these are based upon are available in the statistical result Excel table.



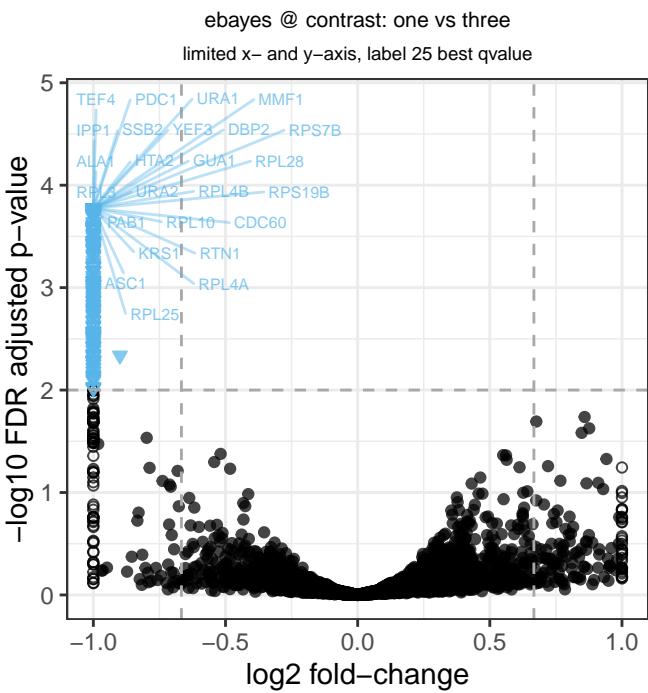
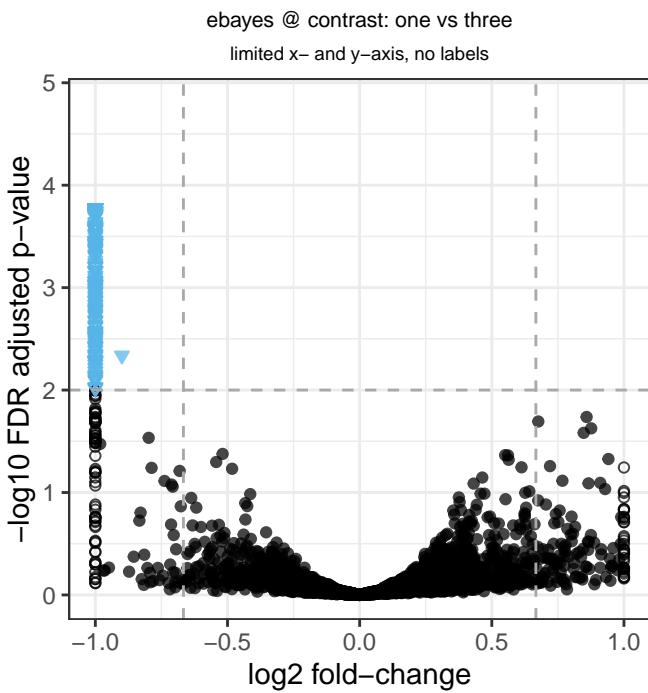
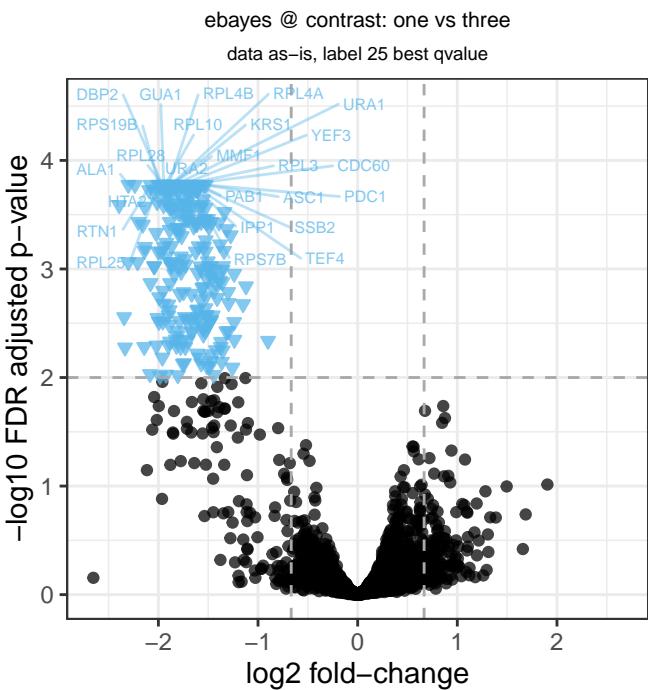
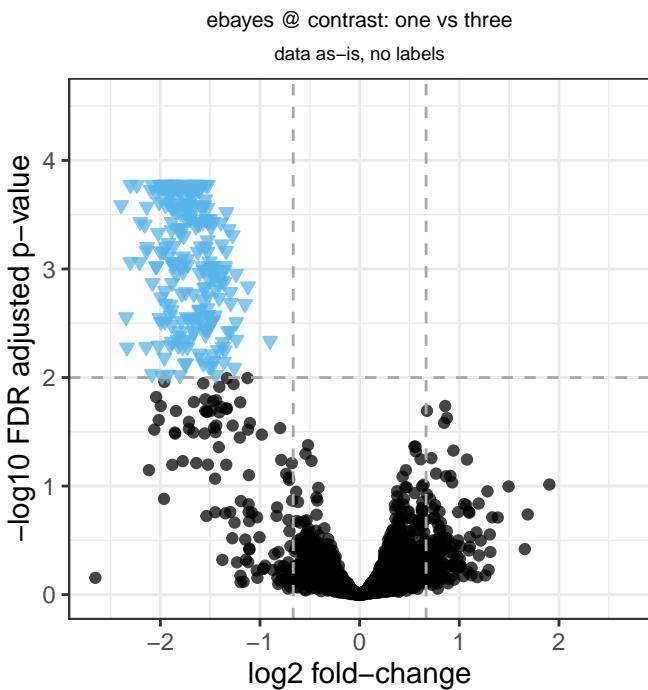
## 2.2 one vs three

- **user setting:** using ‘filter by contrast’ peptide filtering approach
- 57302 peptides in 7320 proteins remain in the current contrast after peptide filters and are used for the statistical analysis in this section
- qvalue threshold: 0.01
- log2 foldchange threshold: 0.6667772

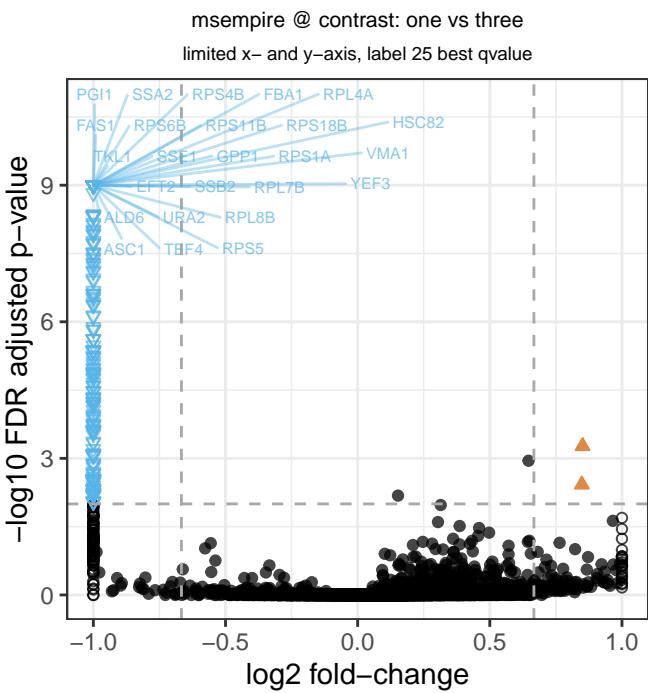
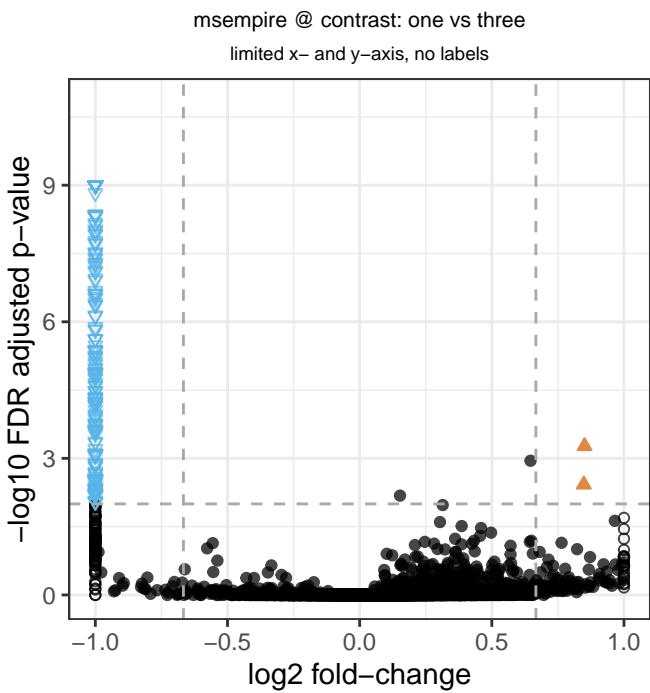
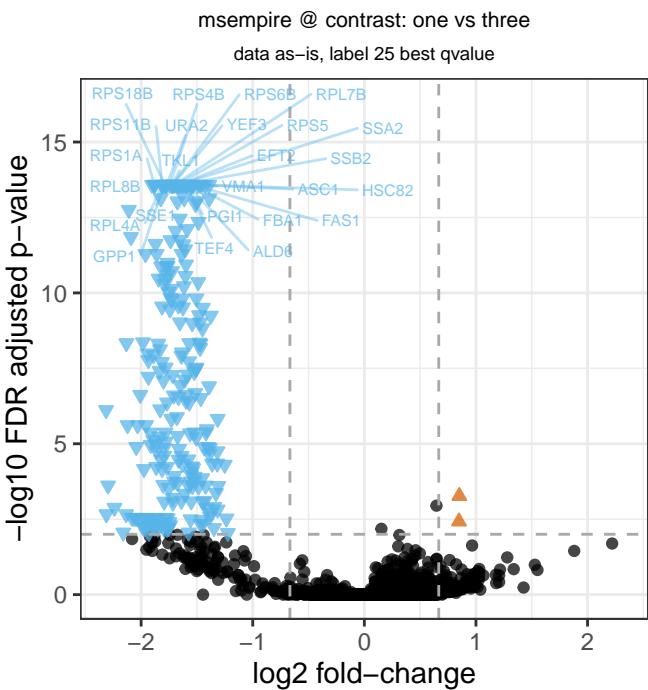
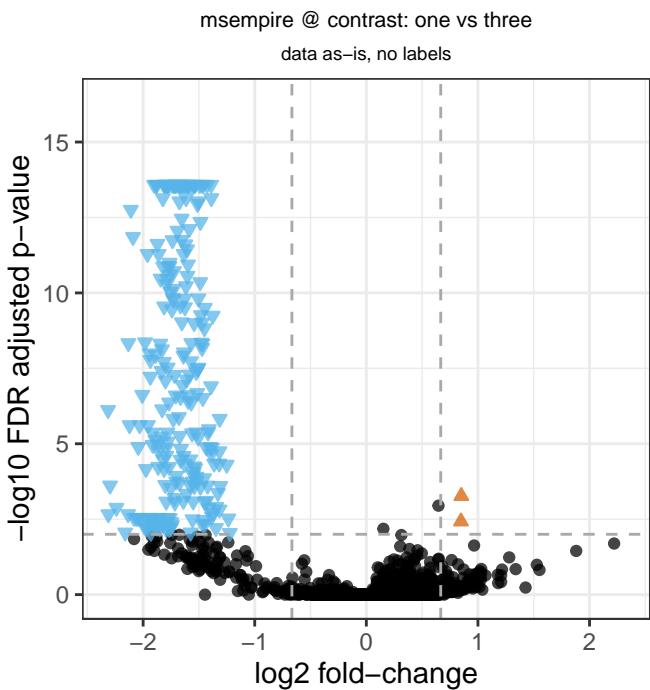
### 2.2.1 volcano

The plot title shows the statistical model and contrast (sample groups in the comparison). Left- and right-side figure panels on each row represent the same figure without and with labels for the 25 proteins with lowest p-value.

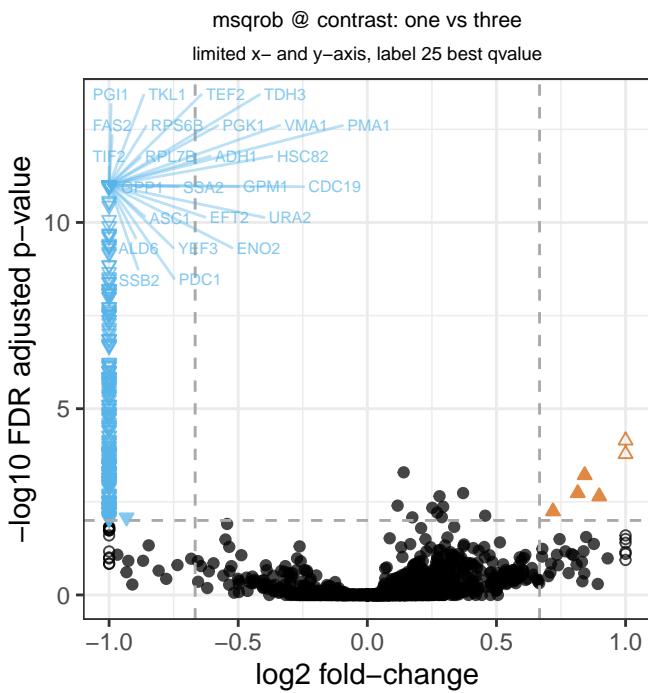
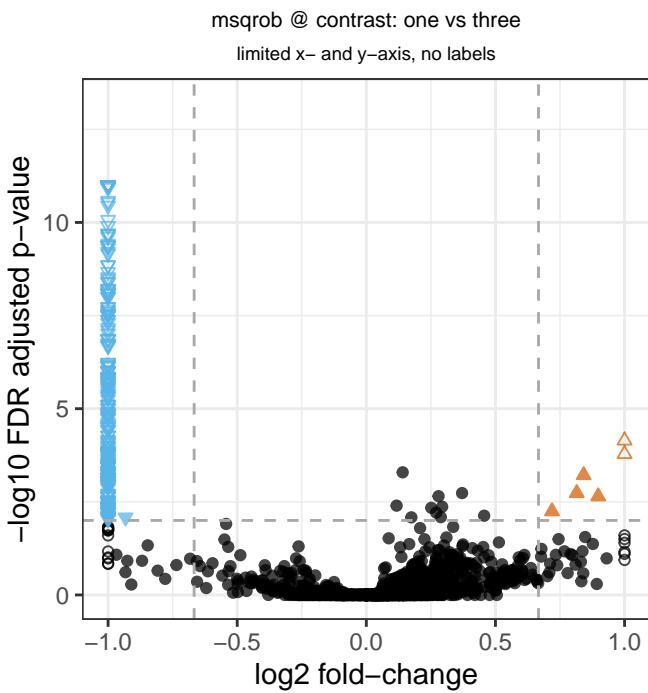
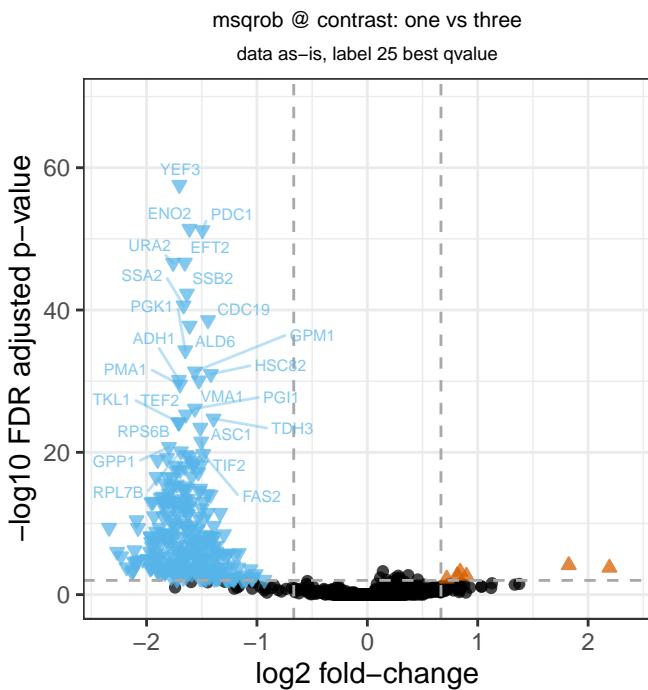
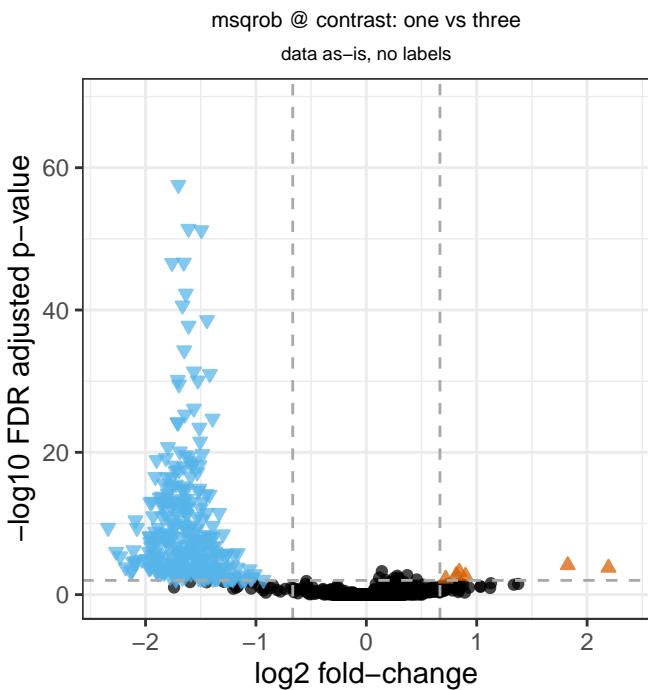
Bottom figure panels have limited x- and y-axis. For datasets with a small number of strong outliers in p-value or fold-change, which may have a profound effect on the plot scales, this allows inspection of the remainder of the volcano plot without disproportionate influence by ‘extreme’ values.



- ▲ up regulated      ▲ up regulated & outside plot limits
- ▼ down regulated    ▽ down regulated & outside plot limits
- not significant
- not significant & outside plot limits



- ▲ up regulated ▲ up regulated & outside plot limits ● not significant
- ▼ down regulated ▼ down regulated & outside plot limits ○ not significant & outside plot limits

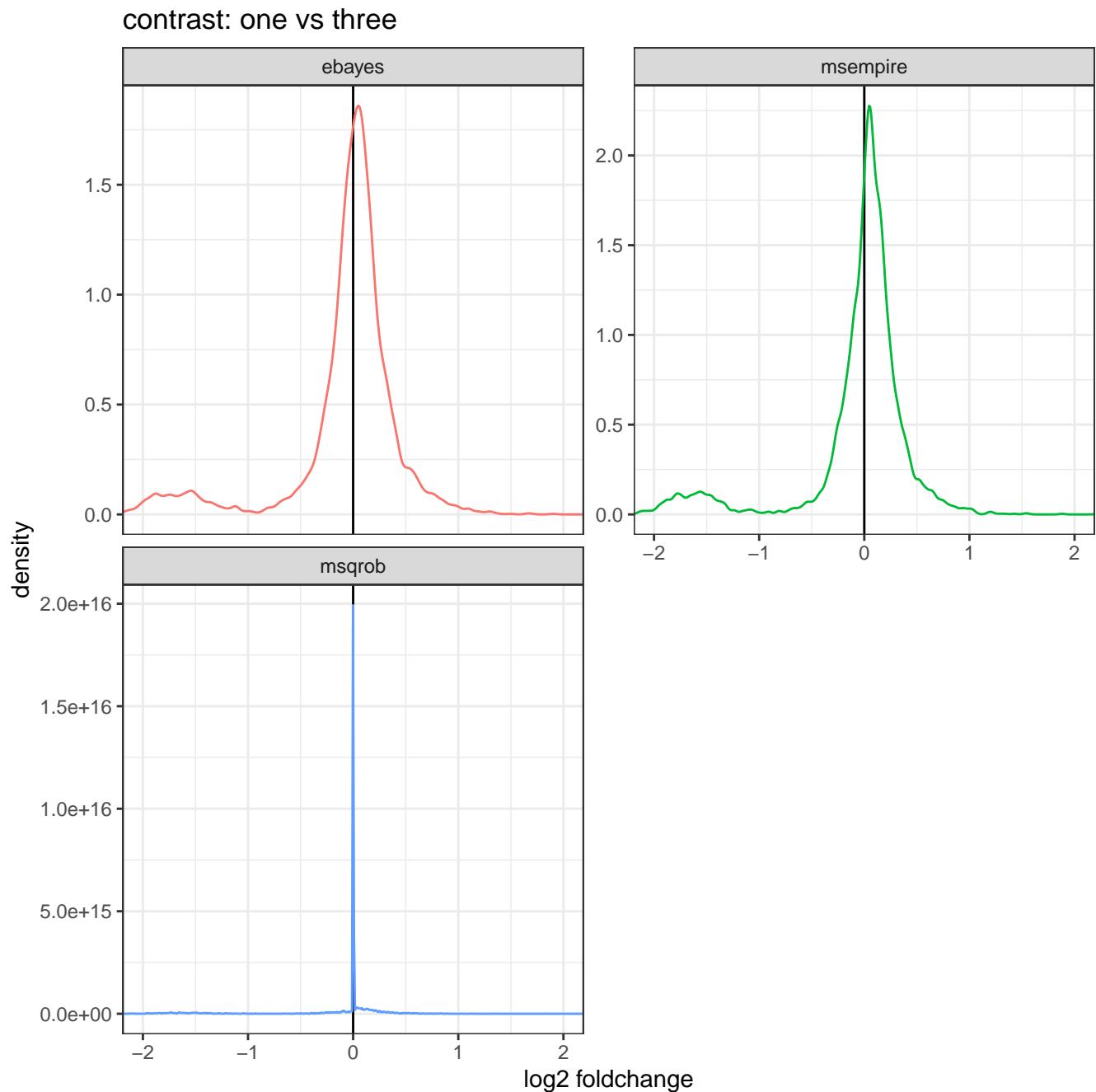


- ▲ up regulated ▲ up regulated & outside plot limits ● not significant
- ▼ down regulated ▼ down regulated & outside plot limits ○ not significant & outside plot limits

## 2.2.2 foldchange distribution

Distributions of estimated foldchanges produced by the statistical models. If the mode is far from 0, consider alternative normalization strategies. Do note the scale on the x-axis, for some experiments the foldchanges are very low which in turn may exaggerate this figure.

*note; the MSqRob model tends to assign zero (log)foldchange for proteins with minor difference between conditions where the model is very sure the null hypothesis cannot be rejected (shrinkage by the ridge regression model). As a result, many foldchanges will be zero and the density plot for MSqRob may look like a spike instead of the expected Gaussian shape observed in other models*



### 2.2.3 p-value distribution

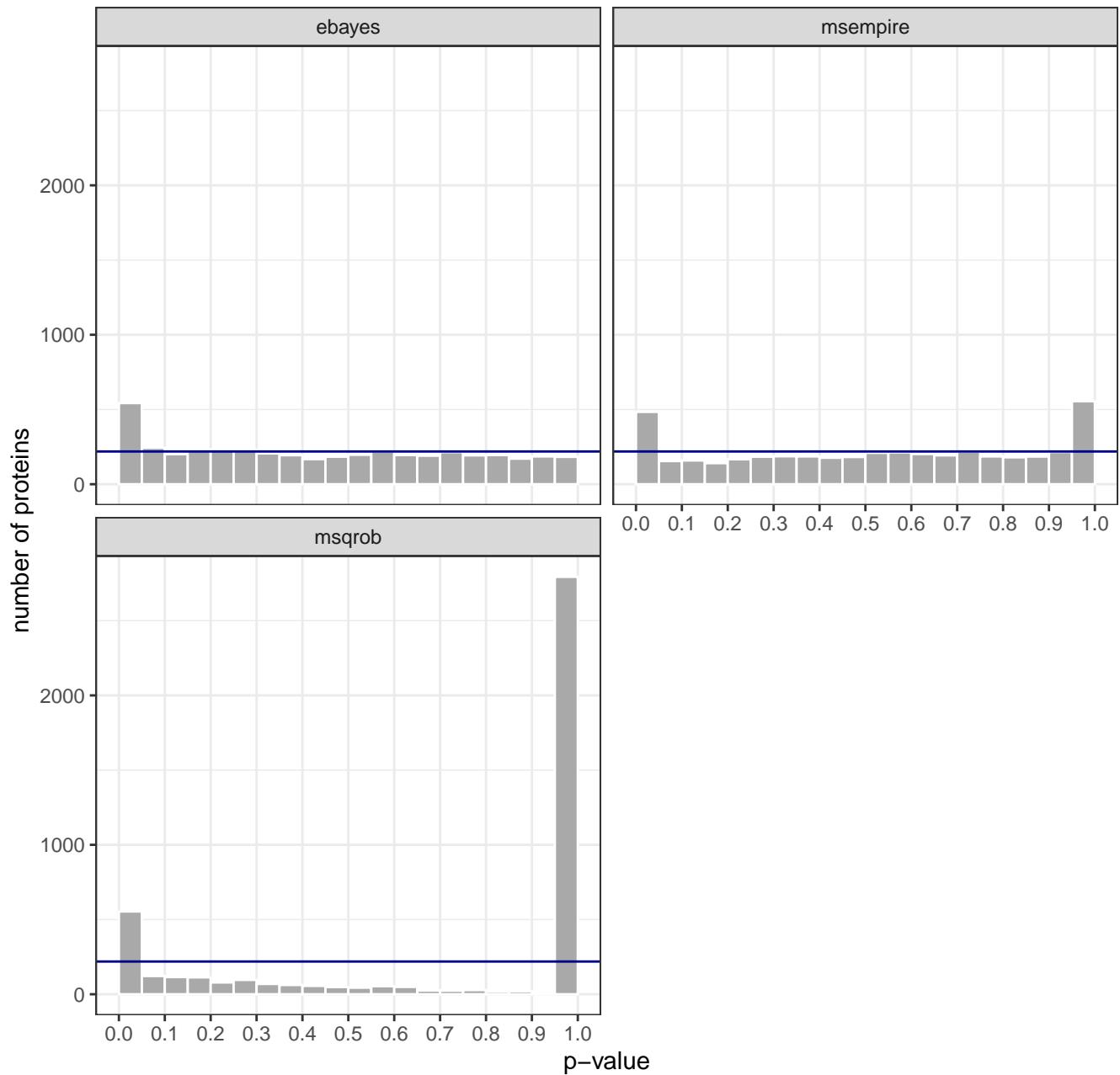
Histogram of p-values computed by differential expression analysis algorithms, as-is, for quality-control inspection. The horizontal line indicates the expected counts assuming a uniform distribution (total number of p-values divided by number of histogram bins)

See further: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6164648/>

See further: <http://varianceexplained.org/statistics/interpreting-pvalue-histogram/>

*note; the MSqRob and MS-Empire models often yield p-value distributions that show a large peak at p-value 1, these are typically proteins with estimated log foldchanges at/near zero where these models are very sure the null hypothesis cannot be rejected*

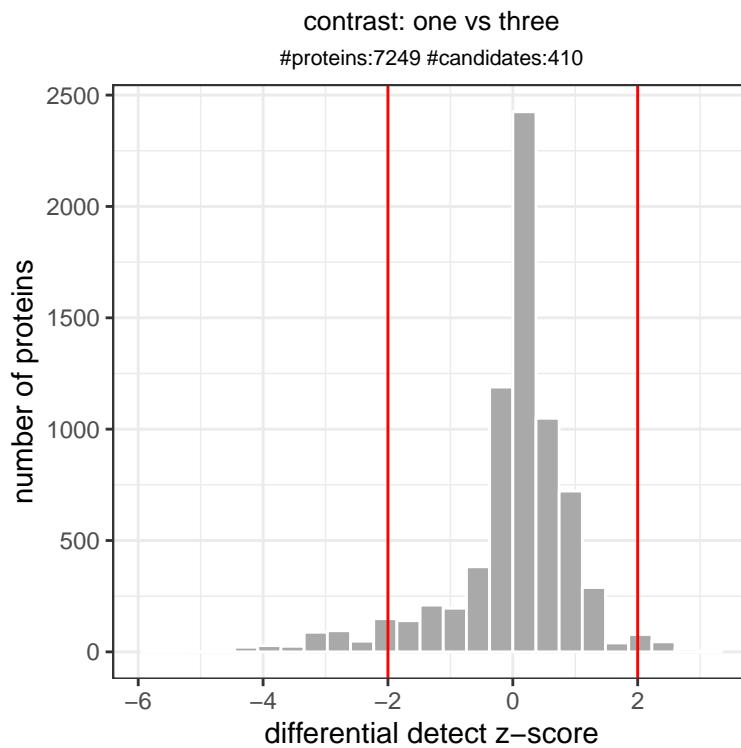
contrast: one vs three



## 2.2.4 differential detect

Some proteins may not have peptides with sufficient abundance values for differential expression analysis, but do show strong differences in the number of detected peptides between sample groups. A simple z-score is computed for each protein based on the number of detected peptides per sample group. DEA is much more powerful for differential testing, but these z-scores can offer complementary information in some experimental designs (e.g. a wildtype-knockout APMS study).

Below figure shows the distribution of these scores with thresholds at 2 std. Both the z-scores and the counts these are based upon are available in the statistical result Excel table.



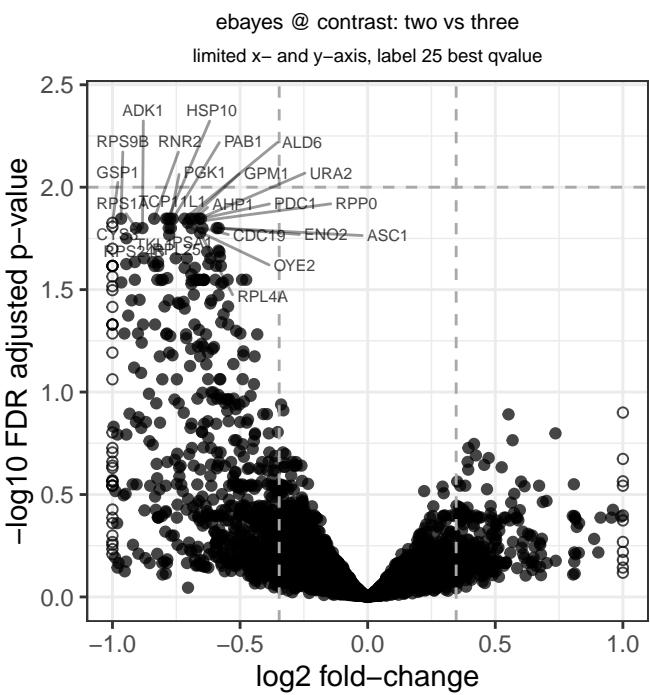
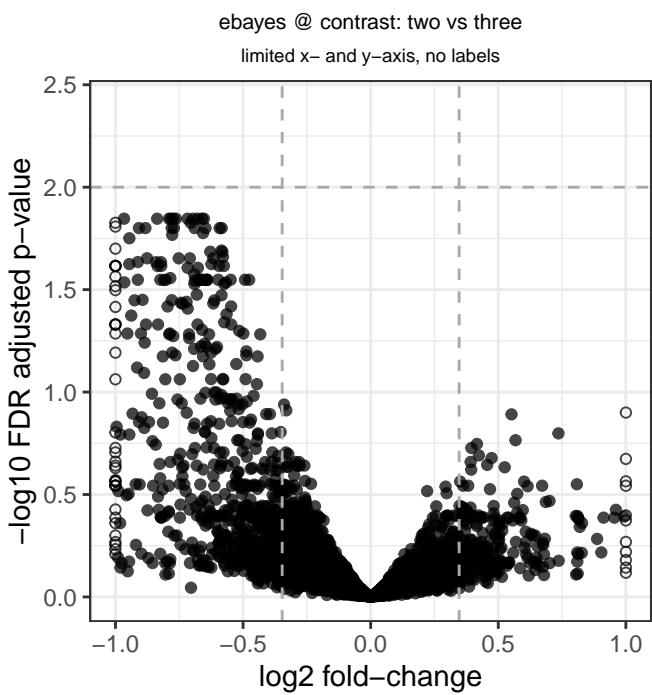
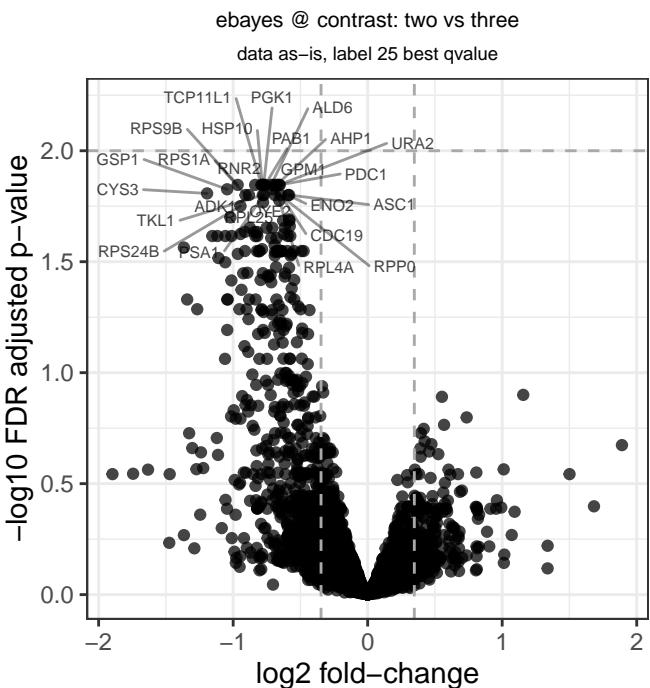
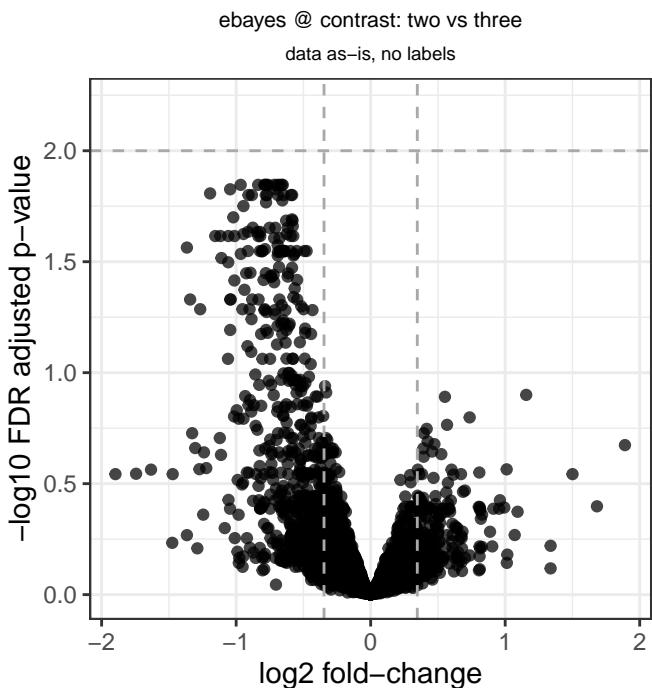
## 2.3 two vs three

- **user setting:** using ‘filter by contrast’ peptide filtering approach
- 57302 peptides in 7320 proteins remain in the current contrast after peptide filters and are used for the statistical analysis in this section
- qvalue threshold: 0.01
- log2 foldchange threshold: 0.3466925

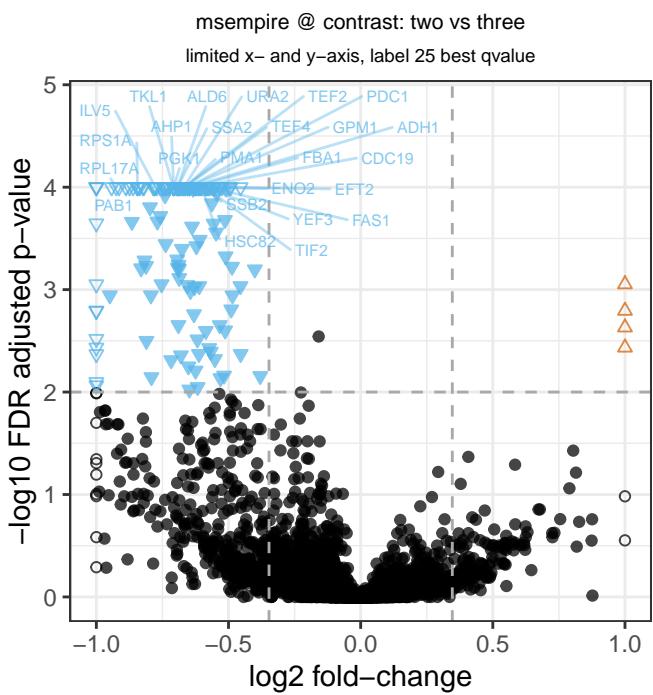
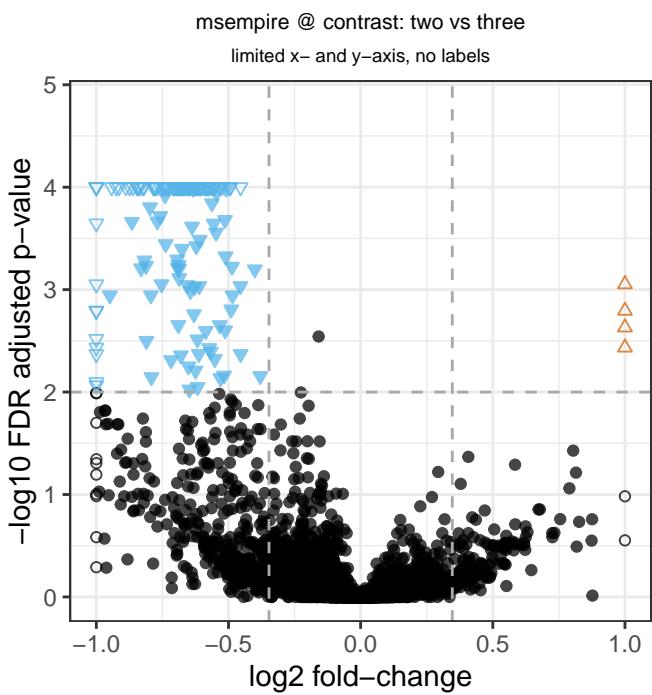
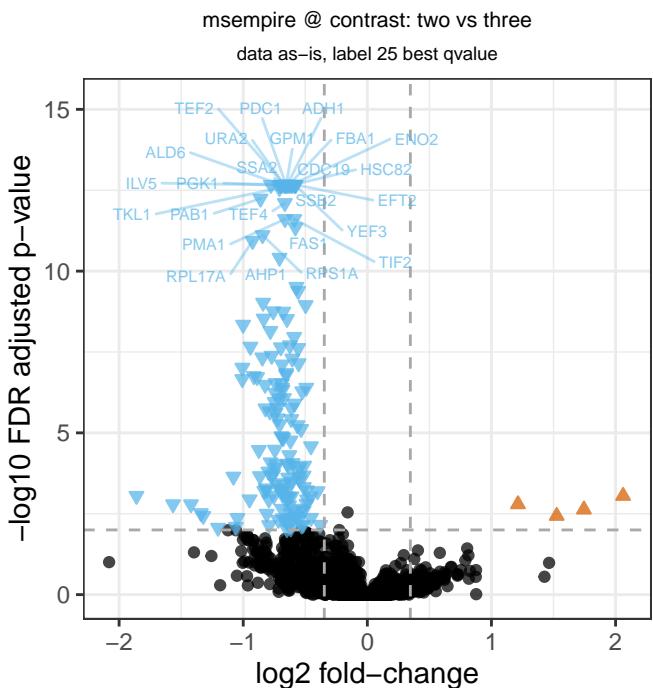
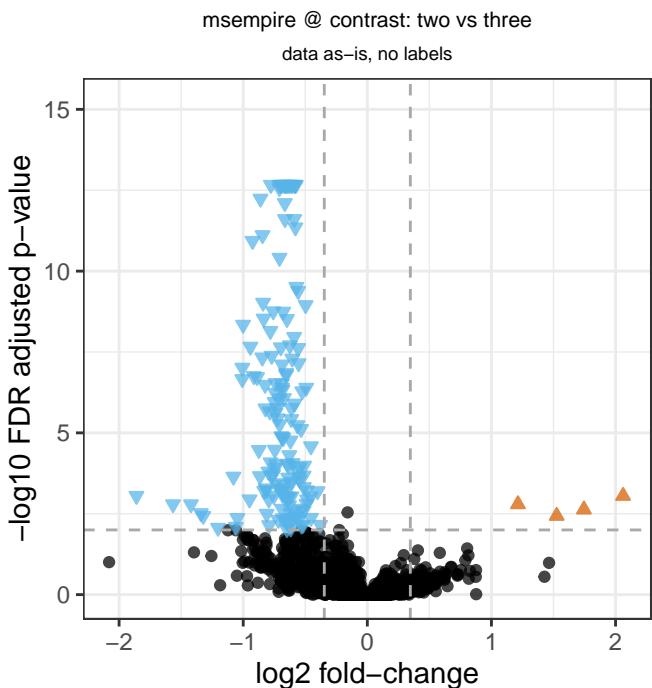
### 2.3.1 volcano

The plot title shows the statistical model and contrast (sample groups in the comparison). Left- and right-side figure panels on each row represent the same figure without and with labels for the 25 proteins with lowest p-value.

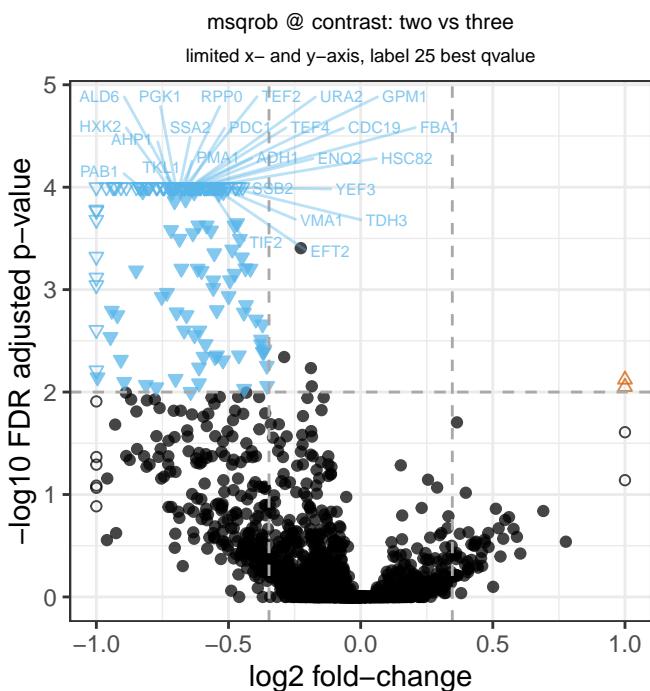
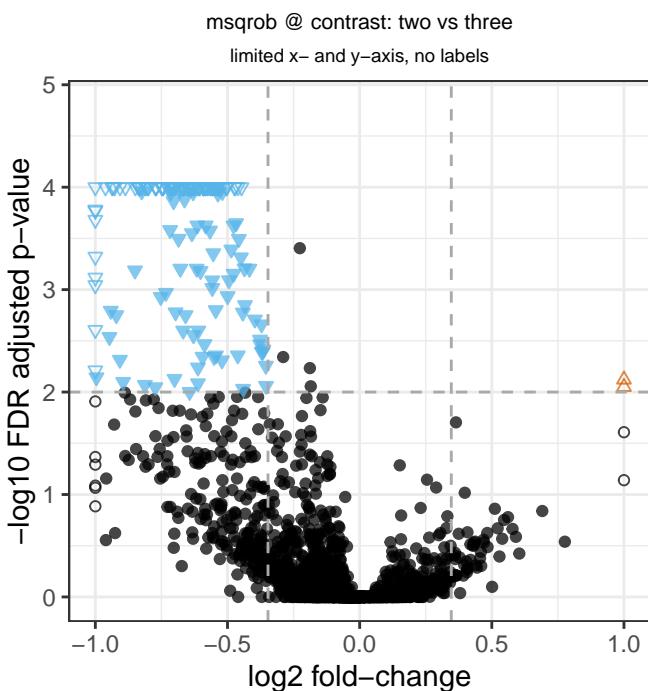
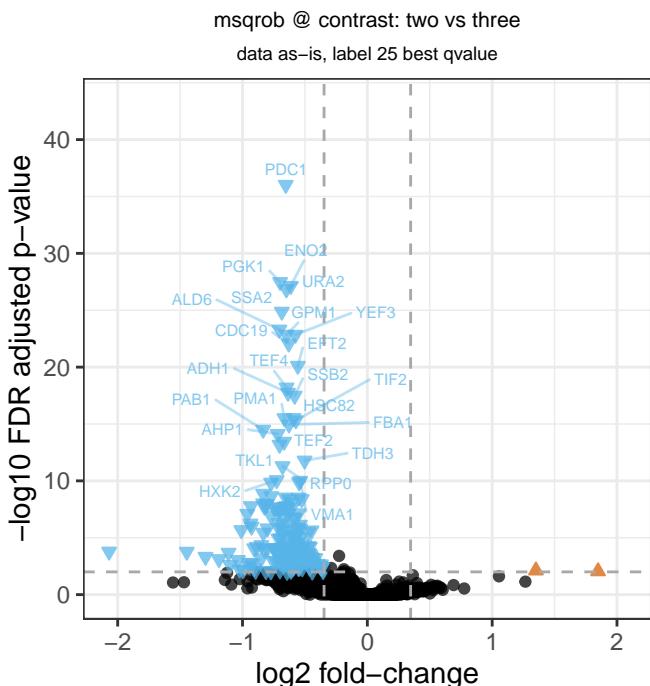
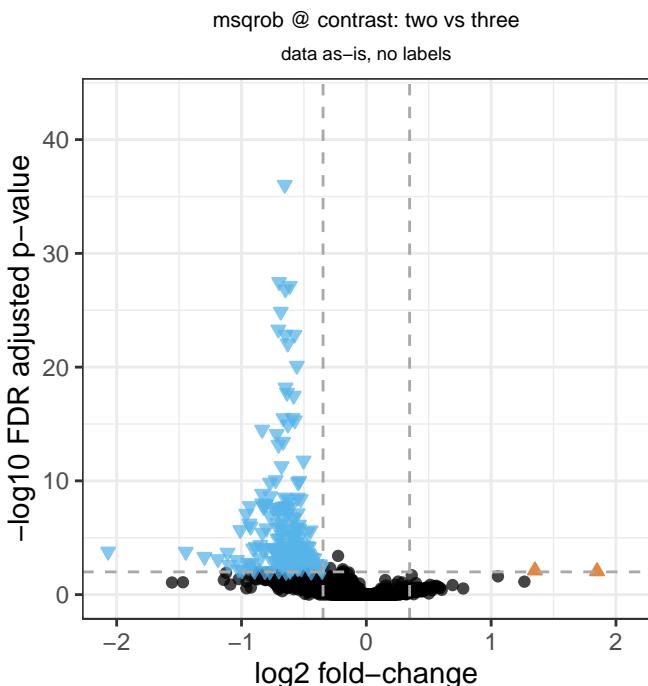
Bottom figure panels have limited x- and y-axis. For datasets with a small number of strong outliers in p-value or fold-change, which may have a profound effect on the plot scales, this allows inspection of the remainder of the volcano plot without disproportionate influence by ‘extreme’ values.



- ▲ up regulated      ▲ up regulated & outside plot limits      ● not significant
- ▼ down regulated      ▽ down regulated & outside plot limits      ○ not significant & outside plot limits



- ▲ up regulated ▲ up regulated & outside plot limits ● not significant
- ▼ down regulated ▼ down regulated & outside plot limits ○ not significant & outside plot limits



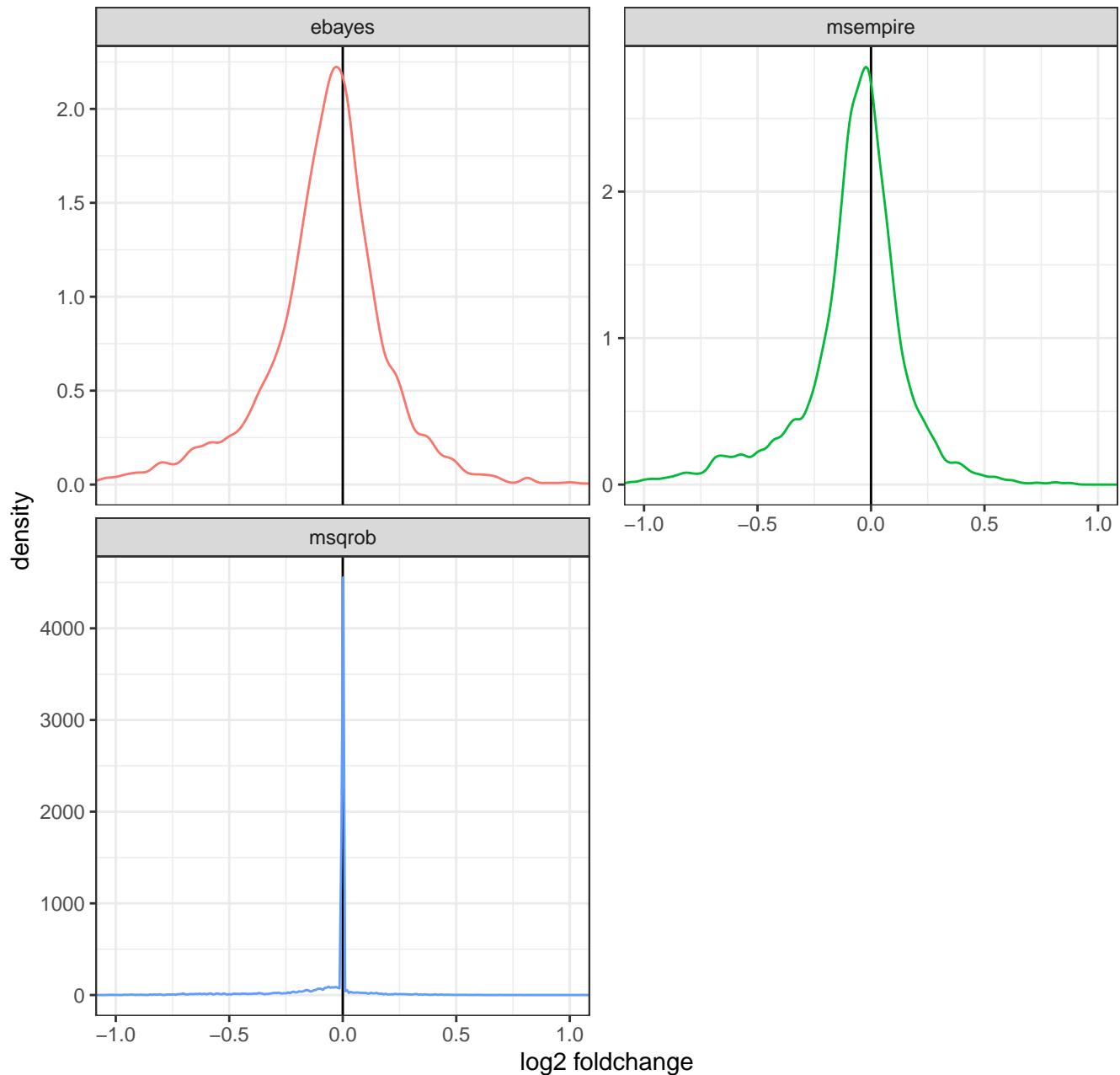
- ▲ up regulated ▲ up regulated & outside plot limits ● not significant
- ▼ down regulated ▼ down regulated & outside plot limits ○ not significant & outside plot limits

### 2.3.2 foldchange distribution

Distributions of estimated foldchanges produced by the statistical models. If the mode is far from 0, consider alternative normalization strategies. Do note the scale on the x-axis, for some experiments the foldchanges are very low which in turn may exaggerate this figure.

*note; the MSqRob model tends to assign zero (log)foldchange for proteins with minor difference between conditions where the model is very sure the null hypothesis cannot be rejected (shrinkage by the ridge regression model). As a result, many foldchanges will be zero and the density plot for MSqRob may look like a spike instead of the expected Gaussian shape observed in other models*

contrast: two vs three



### 2.3.3 p-value distribution

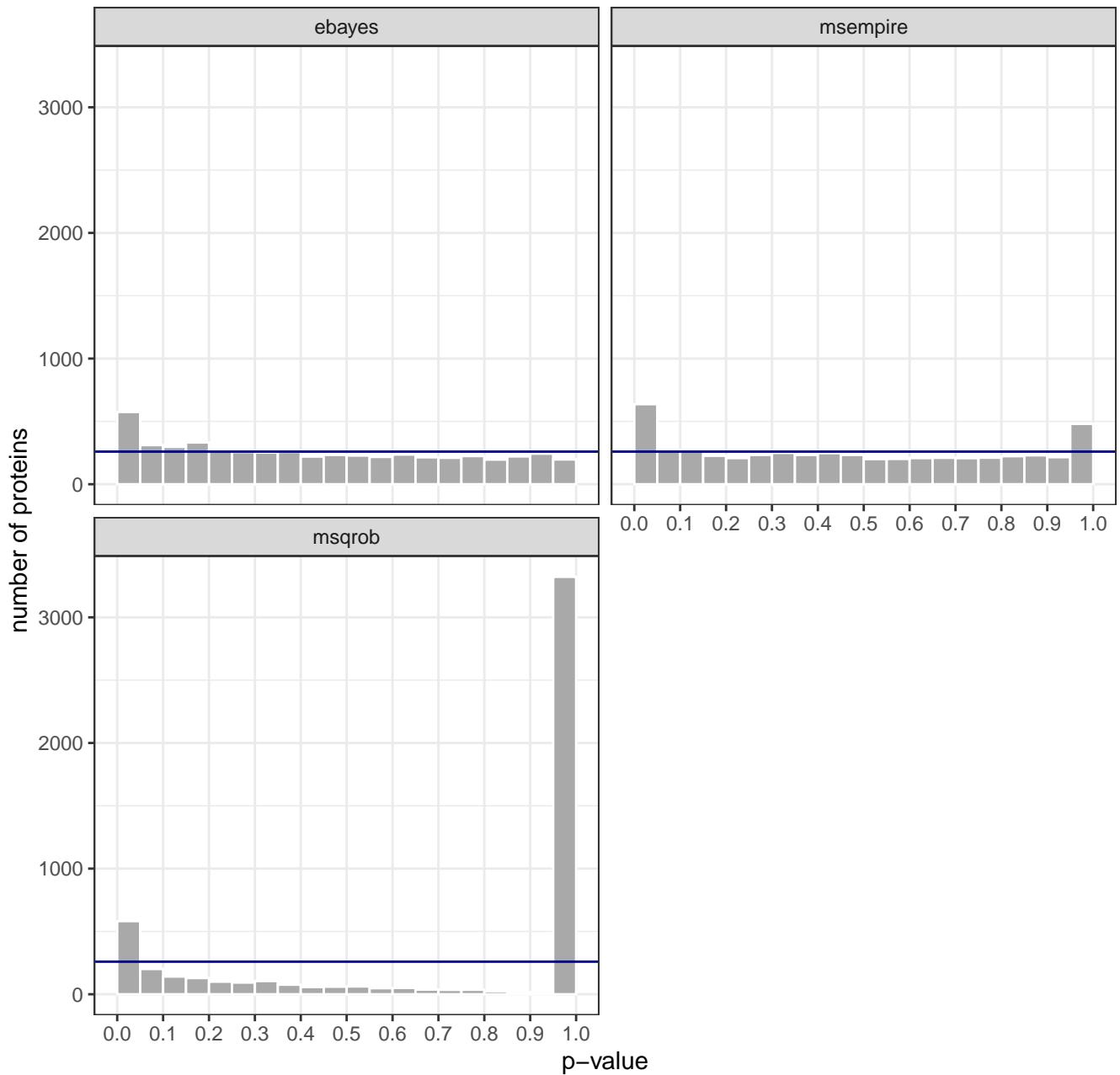
Histogram of p-values computed by differential expression analysis algorithms, as-is, for quality-control inspection. The horizontal line indicates the expected counts assuming a uniform distribution (total number of p-values divided by number of histogram bins)

See further: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6164648/>

See further: <http://varianceexplained.org/statistics/interpreting-pvalue-histogram/>

*note; the MSqRob and MS-Empire models often yield p-value distributions that show a large peak at p-value 1, these are typically proteins with estimated log foldchanges at/near zero where these models are very sure the null hypothesis cannot be rejected*

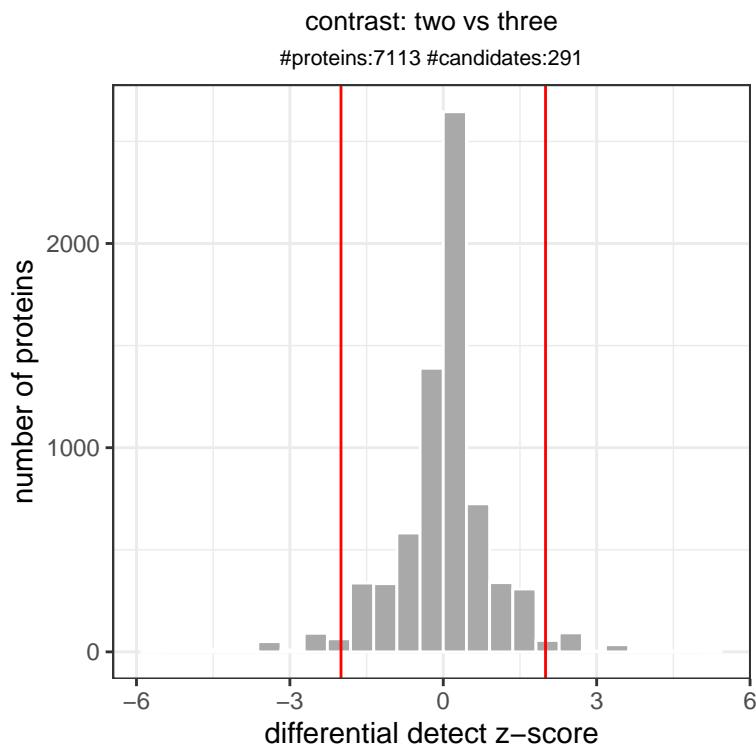
contrast: two vs three



#### 2.3.4 differential detect

Some proteins may not have peptides with sufficient abundance values for differential expression analysis, but do show strong differences in the number of detected peptides between sample groups. A simple z-score is computed for each protein based on the number of detected peptides per sample group. DEA is much more powerful for differential testing, but these z-scores can offer complementary information in some experimental designs (e.g. a wildtype-knockout APMS study).

Below figure shows the distribution of these scores with thresholds at 2 std. Both the z-scores and the counts these are based upon are available in the statistical result Excel table.



### 3 Summary of differential testing

Differential Expression Analysis: number of proteins found statistically significant.

contrast	algorithm	#test	#hits	top10 significant
one vs three	ebayes	4380	247	rpl10, rps19b;..., ura2, hta2;hta1, ala1, rpl4b, gua1, rpl25, ipp1, ura1
one vs three	msempire	4380	240	adh1, tdh3, cdc19, pgk1, eno2, gpm1, tef2, pab1, pma1;pma2, ilv5
one vs three	msqrob	4380	306	yef3, eno2, pdc1, eft2, ura2, ssb2, ssa2, cdc19, ald6, pgk1
one vs two	ebayes	4399	208	rpl4b, gua1, rpl33a, rpl4a, grs1, rps16b;..., fas1, rpl10, rpl7b;r..., eft2
one vs two	msempire	4399	205	adh1, tdh3, cdc19, pgk1, eno2, pdc1, ura2, ssa2, pg1, hsc82
one vs two	msqrob	4399	305	yef3, eft2, eno2, pdc1, ssb2, ura2, cdc19, pgk1, adh1, hsc82
two vs three	ebayes	5187	0	
two vs three	msempire	5187	158	adh1, cdc19, pgk1, eno2, gpm1, tef2, ilv5, pdc1, ura2, ssa2
two vs three	msqrob	5187	174	pdc1, pgk1, eno2, ura2, ssa2, ald6, gpm1, yef3, cdc19, eft2

Differential Detection: prioritize proteins with more peptide detections in some group. A simple metric to complement results from DEA, which is the main result (eg; consider proteins with too few data points for DEA).

contrast	#test	#hits	top10 candidates
one vs three	7249	410	gcn1, rnr1, cpa2, glt1, paa1, ncp1, spt16, wtm1, rpg1, spf1
one vs two	7249	387	cpa2, hem2, aah1, rvb2, gcn20, sec24, rlp7, mug1, wrs1, lap3
two vs three	7113	291	gcn1, rasal2, ncl1, dnx35, rpa190, gmnn, tjap1, cbf5, sirt2, sqor

## 4 log

```
reading MaxQuant 'txt' folder
Parsing MaxQuant proteinGroups.txt, 5/58442 peptides are not a razor peptide in any protein-group
and therefore removed
57302 target precursors, 54504 (plain)sequences, 7320 proteins
7641/7641 protein accessions and 7320/7320 protein groups were mapped to provided fasta file(s)
using 7 threads for multiprocessing
differential abundance analysis for contrast: one vs two
using data from peptide filter: filter by contrast
log2 foldchange threshold estimated by bootstrap analysis: 0.549
differential abundance analysis for contrast: one vs three
using data from peptide filter: filter by contrast
log2 foldchange threshold estimated by bootstrap analysis: 0.667
differential abundance analysis for contrast: two vs three
using data from peptide filter: filter by contrast
log2 foldchange threshold estimated by bootstrap analysis: 0.347
```

## 5 R command history

This shows the history commands from your R script that starts this pipeline, thereby automatically documenting the parameters/settings used. All lines of executed code since (last) importing data using this R package are shown.

### Using this feature

Do not use RStudio's `source` option to execute our pipeline since it will only write `source(...yourscript.R)` to the session history, and consequentially that is all you see in this 'code log'. Instead, select all lines in your script (`control + A`) and then "run" the selected code (either click the run button in RStudio, or use `control + enter`). All lines shown in this section are the same as shown in the RStudio 'History' pane (a tab on the top-right of its UI).

```
dataset = import_dataset_maxquant_evidencetxt(path = "C:/DATA/PXD007683/tx_mbr")
dataset = import_fasta(dataset,
  files = c(
    "C:/DATA/PXD007683/fasta/UP000002311_559292.fasta",
    "C:/DATA/PXD007683/fasta/UP000005640_9606.fasta"
  )
)
dataset = import_sample_metadata(
  dataset,
  "C:/DATA/PXD007683/msdap/oconnell_samples.xlsx"
)
dataset = setup_contrasts(dataset,
  contrast_list = list(c(
    "one",
    "two"
  ), c(
    "one",
    "three"
  ), c(
    "two",
    "three"
  )))
)
dataset = analysis_quickstart(dataset,
  filter_min_detect = 2,
  filter_min_quant = 3,
  filter_by_contrast = TRUE,
  filter_topn_peptides = 0,
  filter_min_peptide_per_prot = 1,
  norm_algorithm = "vwmb",
  dea_algorithm = c(
    "ebayes",
    "msemprise", "msqrob"
  ),
  dea_value_threshold = 0.01,
  dea_log2foldchange_threshold = NA,
  output_qc_report = TRUE,
  output_peptide_plots = "none",
  output_abundance_tables = FALSE,
  output_dir = "C:/DATA/PXD007683/msdap",
  output_within_timestamped_subdirectory = TRUE
)
```

## 6 R session info

The computer system and versioning of all R packages used to run this analysis are shown below to facilitate, in combination with the previous section, reproducibility.

setting	value
version	R version 3.6.1 (2019-07-05)
os	Windows 10 x64
system	x86_64, mingw32
ui	RStudio
language	(EN)
collate	English_United States.1252
ctype	English_United States.1252
tz	Europe/Berlin
date	2020-06-01

*System*

package	loadedversion	source
dplyr	0.8.4	CRAN (R 3.6.2)
ggplot2	3.2.1	CRAN (R 3.6.1)
msdap	0.1.7.4	
rlang	0.4.6	CRAN (R 3.6.3)
testthat	2.3.2	CRAN (R 3.6.3)
tibble	2.1.3	CRAN (R 3.6.1)
tidyverse	1.0.2	CRAN (R 3.6.2)

*Attached packages*

package	loadedversion	source
affy	1.64.0	Bioconductor
affyio	1.56.0	Bioconductor
askpass	1.1	CRAN (R 3.6.1)
assertthat	0.2.1	CRAN (R 3.6.1)
backports	1.1.5	CRAN (R 3.6.1)
Biobase	2.46.0	Bioconductor
BiocGenerics	0.32.0	Bioconductor
BiocManager	1.30.10	CRAN (R 3.6.3)
BiocParallel	1.20.1	Bioconductor
bit	1.1-14	CRAN (R 3.6.0)
bit64	0.9-7	CRAN (R 3.6.0)
boot	1.3-24	CRAN (R 3.6.2)
callr	3.4.3	CRAN (R 3.6.3)
cli	2.0.2	CRAN (R 3.6.3)
codetools	0.2-16	CRAN (R 3.6.1)
colorspace	1.4-1	CRAN (R 3.6.1)
cowplot	1.0.0	CRAN (R 3.6.1)
crayon	1.3.4	CRAN (R 3.6.1)
data.table	1.12.8	CRAN (R 3.6.2)
desc	1.2.0	CRAN (R 3.6.1)
devtools	2.3.0	CRAN (R 3.6.3)
digest	0.6.25	CRAN (R 3.6.1)
doParallel	1.0.15	CRAN (R 3.6.1)
ellipsis	0.3.0	CRAN (R 3.6.1)
evaluate	0.14	CRAN (R 3.6.1)
fansi	0.4.1	CRAN (R 3.6.2)
farver	2.0.3	CRAN (R 3.6.2)
forcats	0.4.0	CRAN (R 3.6.1)
foreach	1.4.8	CRAN (R 3.6.2)
formatR	1.7	CRAN (R 3.6.1)
fs	1.3.1	CRAN (R 3.6.1)
ggpubr	0.2.5	CRAN (R 3.6.2)
ggrepel	0.8.1	CRAN (R 3.6.1)
ggsignif	0.6.0	CRAN (R 3.6.1)
glue	1.4.1	CRAN (R 3.6.3)
gridExtra	2.3	CRAN (R 3.6.1)
gttable	0.3.0	CRAN (R 3.6.1)
gtools	3.8.1	CRAN (R 3.6.0)
hms	0.5.3	CRAN (R 3.6.2)
htmltools	0.4.0	CRAN (R 3.6.1)
impute	1.60.0	Bioconductor
IRanges	2.20.2	Bioconductor
iterators	1.0.12	CRAN (R 3.6.1)
knitr	1.28	CRAN (R 3.6.2)
labeling	0.3	CRAN (R 3.6.0)
lattice	0.20-40	CRAN (R 3.6.1)
lazyeval	0.2.2	CRAN (R 3.6.1)
lifecycle	0.1.0	CRAN (R 3.6.1)
limma	3.42.2	Bioconductor
lme4	1.1-21	CRAN (R 3.6.1)

package	loadedversion	source
magrittr	1.5	CRAN (R 3.6.1)
MALDIquant	1.19.3	CRAN (R 3.6.1)
MASS	7.3-51.5	CRAN (R 3.6.2)
Matrix	1.2-18	CRAN (R 3.6.2)
matrixStats	0.55.0	CRAN (R 3.6.1)
memoise	1.1.0	CRAN (R 3.6.1)
minqa	1.2.4	CRAN (R 3.6.1)
msEmpiRe	0.1.0	Github (zimmerlab/MS-EmpiRe@ae985a9)
MSnbase	2.12.0	Bioconductor
munsell	0.5.0	CRAN (R 3.6.1)
mzID	1.24.0	Bioconductor
mzR	2.20.0	Bioconductor
ncdf4	1.17	CRAN (R 3.6.1)
nlme	3.1-144	CRAN (R 3.6.2)
nloptr	1.2.1	CRAN (R 3.6.1)
openxlsx	4.1.4	CRAN (R 3.6.2)
packrat	0.5.0	CRAN (R 3.6.3)
patchwork	1.0.0	CRAN (R 3.6.3)
pcaMethods	1.78.0	Bioconductor
pdfTools	2.3	CRAN (R 3.6.1)
pillar	1.4.3	CRAN (R 3.6.2)
pkgbuild	1.0.6	CRAN (R 3.6.1)
pkgconfig	2.0.3	CRAN (R 3.6.1)
pkgload	1.0.2	CRAN (R 3.6.1)
plyr	1.8.5	CRAN (R 3.6.2)
preprocessCore	1.48.0	Bioconductor
prettyunits	1.1.1	CRAN (R 3.6.2)
pROC	1.16.1	CRAN (R 3.6.2)
processx	3.4.2	CRAN (R 3.6.2)
ProtGenerics	1.18.0	Bioconductor
ps	1.3.2	CRAN (R 3.6.2)
purrr	0.3.3	CRAN (R 3.6.1)
qpdf	1.1	CRAN (R 3.6.1)
R6	2.4.1	CRAN (R 3.6.1)
RColorBrewer	1.1-2	CRAN (R 3.6.0)
Rcpp	1.0.3	CRAN (R 3.6.1)
readr	1.3.1	CRAN (R 3.6.1)
remotes	2.1.1	CRAN (R 3.6.2)
reshape2	1.4.3	CRAN (R 3.6.1)
rmarkdown	2.1	CRAN (R 3.6.2)
rprojroot	1.3-2	CRAN (R 3.6.1)
rstudioapi	0.11	CRAN (R 3.6.2)
S4Vectors	0.24.3	Bioconductor
scales	1.1.0	CRAN (R 3.6.1)
sessioninfo	1.1.1	CRAN (R 3.6.1)
stringi	1.4.6	CRAN (R 3.6.2)
stringr	1.4.0	CRAN (R 3.6.1)
styler	1.3.2	CRAN (R 3.6.2)
tidyselect	1.0.0	CRAN (R 3.6.2)
tinytex	0.20	CRAN (R 3.6.2)

package	loadedversion	source
usethis	1.6.1	CRAN (R 3.6.3)
vctrs	0.2.3	CRAN (R 3.6.2)
viridis	0.5.1	CRAN (R 3.6.1)
viridisLite	0.3.0	CRAN (R 3.6.1)
vsn	3.54.0	Bioconductor
withr	2.1.2	CRAN (R 3.6.1)
xfun	0.12	CRAN (R 3.6.2)
XML	3.98-1.20	CRAN (R 3.6.0)
xtable	1.8-4	CRAN (R 3.6.1)
yaml	2.2.1	CRAN (R 3.6.2)
zip	2.0.4	CRAN (R 3.6.1)
zlibbioc	1.32.0	Bioconductor

*Packages that are not attached*