MS-DAP: Mass Spectrometry Downstream Analysis Pipeline

version: 1.0.3 https://github.com/ftwkoopmans/msdap/

Contents

1 Quality control

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

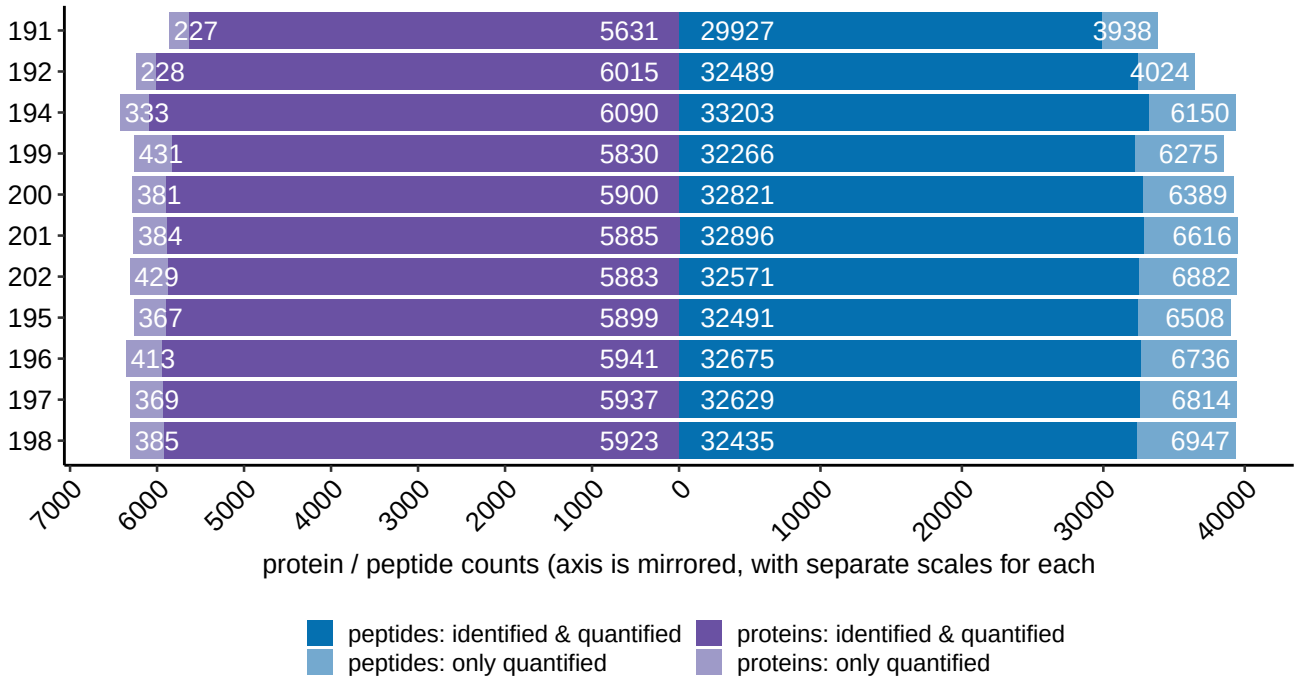
- number of peptides/proteins detected in each sample
- dataset completeness
- local effects in HPLC peptide retention time per sample
- reproducibility of peptide quantification among replicates
- PCA of all samples to visualize clustering

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 <= 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'.

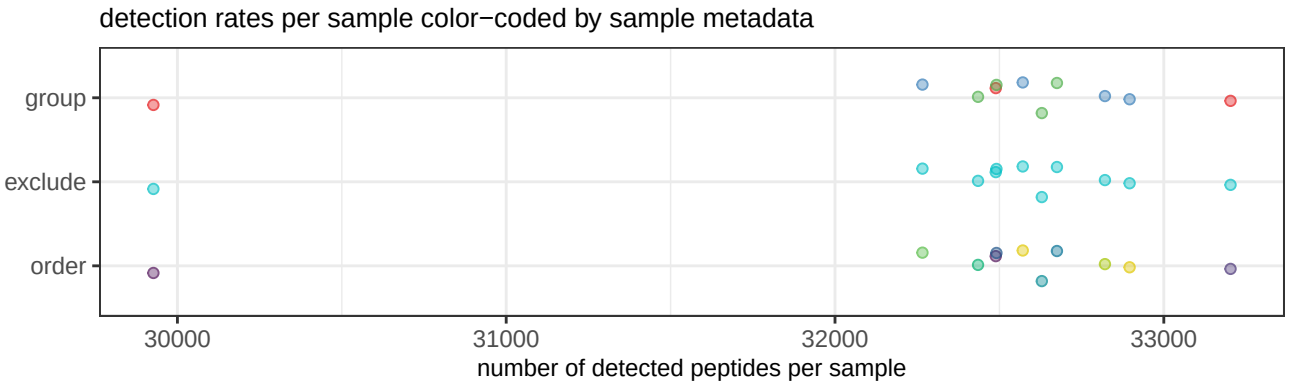


1.1.1 color-coding sample metadata

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 provides an overview of all sample metadata at a first glance. On each row all samples in the dataset are shown as a data point, each color-coded by the respective property shown on the y-axis (with minor vertical jitter for visual clarity). If any of these metadata coincide with a major effect on the number of detected peptides, this should become apparent by a clustering of samples by color-code. Hereafter, an additional set of figures will further expand this overview into detailed figures for each sample property.

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. All information shown in these figures is also available in the output table 'samples.xlsx'.



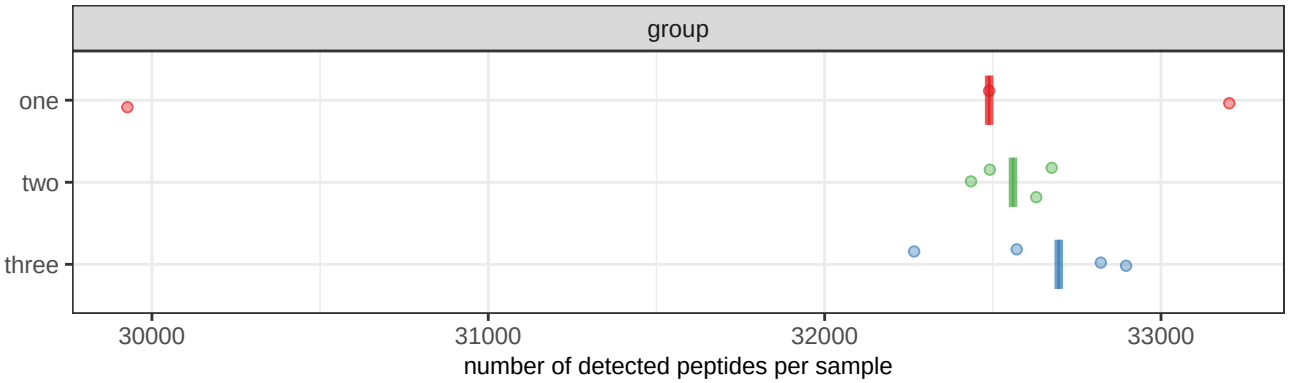
color-coding sample metadata, expanded

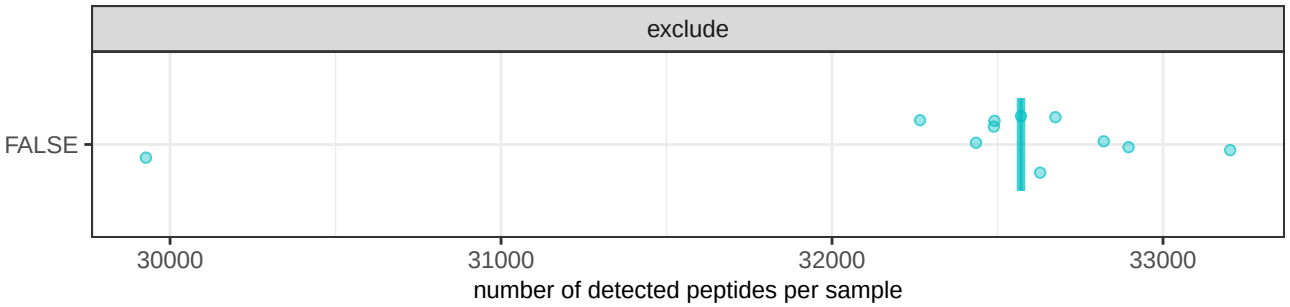
To further detail each sample property, each row in the above figure is now split into separate plots. Thus, a figure is generated for each property in the user-provided metadata (column in the samples table, its name shown in the plot title).

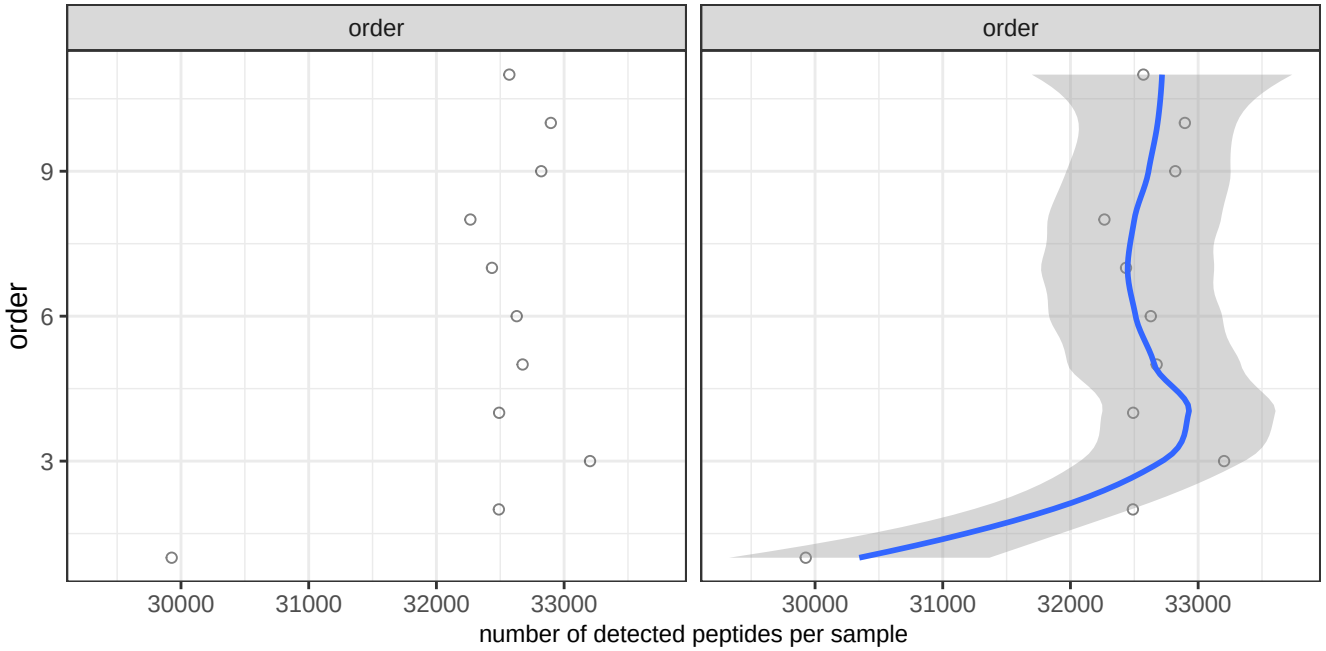
For categorical variables, a scatterplot shows on the y-axis all unique variables while the x-axis depicts the number of detected peptides. Colors are consistent with the above plot. *exclude* samples, if any, are depicted as squares. The median value is shown as a vertical line (thin line = median over all samples, wider line = median while discarding *exclude* samples). For continuous variables, a scatterplot without (left panel) and with Loess fit is shown (right panel, visualized as blue line if data was successfully fitted).

Note that samples flagged as *exclude* are user-provided in the sample metadata table. These are included in data visualizations but excluded from downstream statistical analysis (later part of the report).

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.


group
one
two
three
30000
31000
32000
33000
number of detected peptides per sample

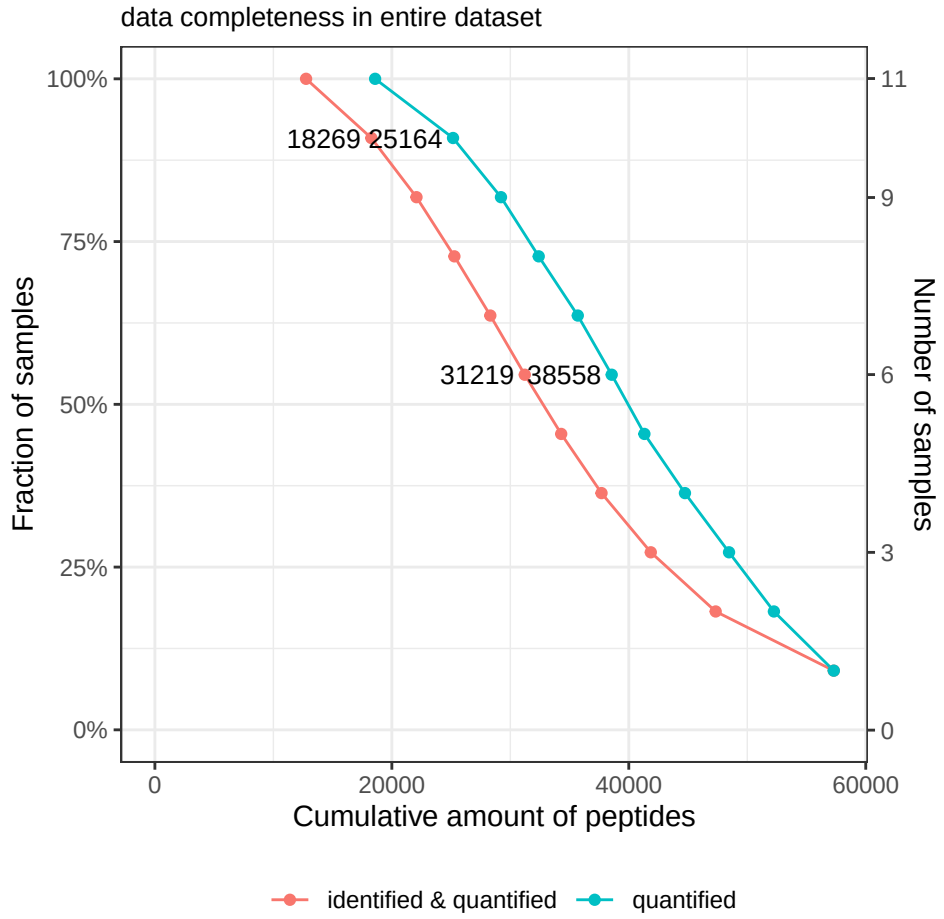

exclude
FALSE
30000
31000
32000
33000
number of detected peptides per sample


order
order
order
9
6
3
30000
31000
32000
33000
number of detected peptides per sample

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 of. 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

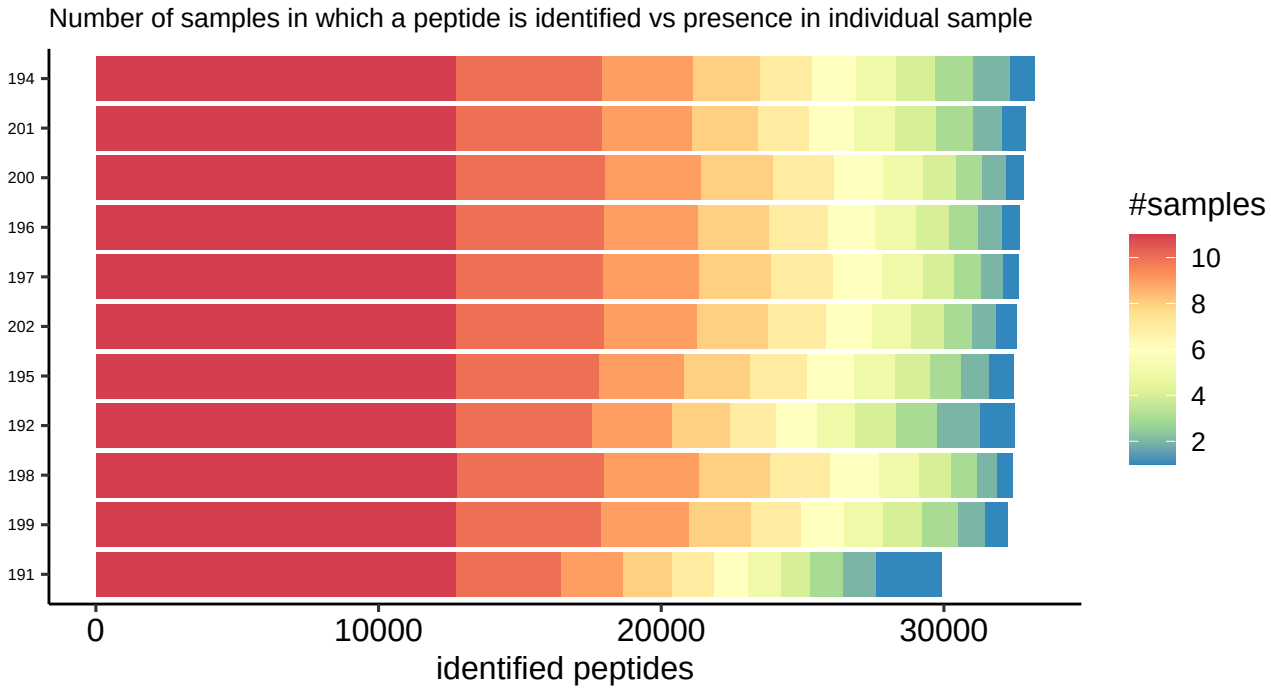


Samples flagged as 'exclude' (by user) are not taken into account in this figure. Exact values are shown for data points matching 90% and 50% of samples to convenience comparison between analyses (e.g. before/after configuring 'exclude' samples, or comparing between experiments of similar protocol).

1.2.2 peptide detection frequency

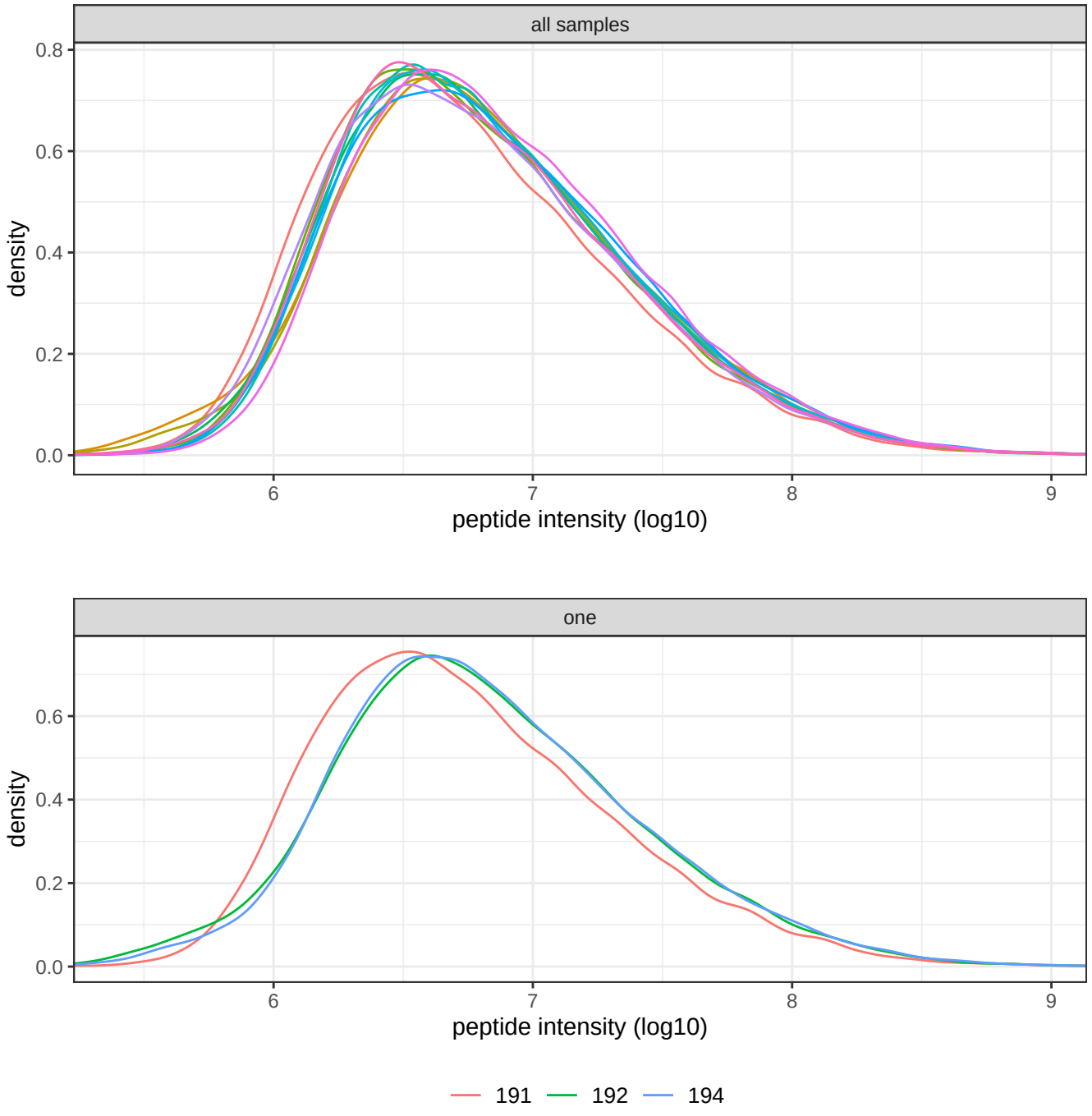
Each identified peptide in a sample is classified and color-coded by the number of other samples where the same peptide is present. Visualization of the amount of peptides that overlap with other samples in the dataset, from peptides identified in most samples (red) to one-hit-wonders (blue), helps identify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

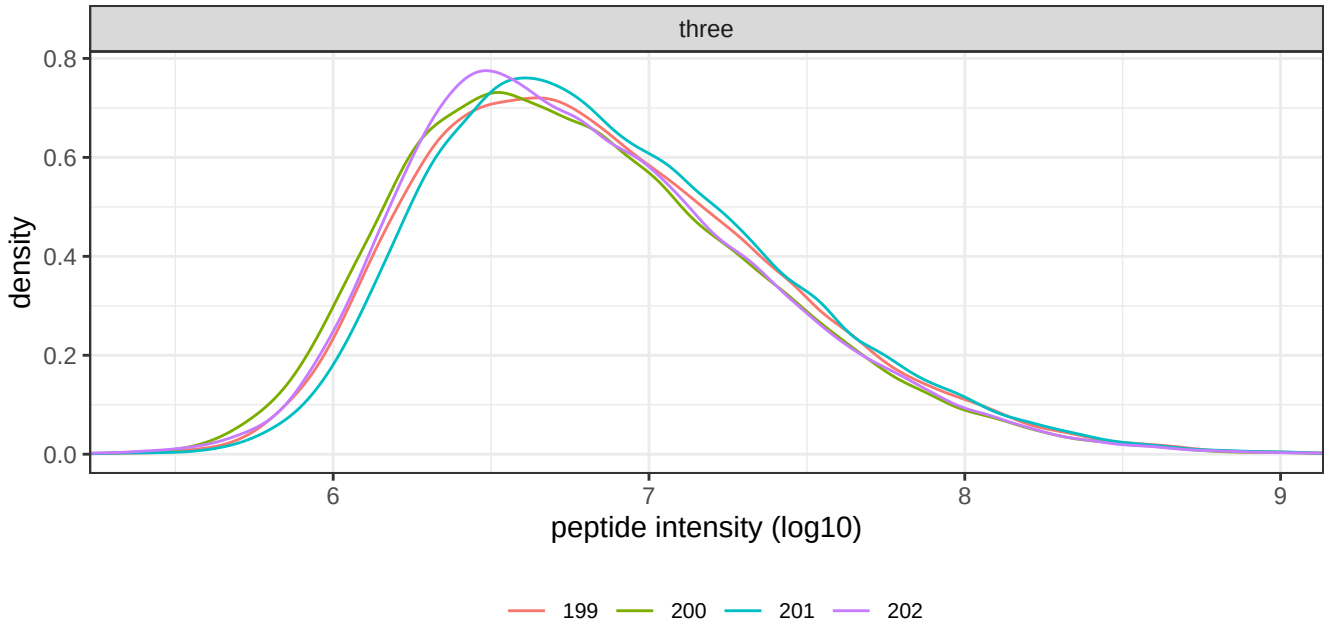
Optimally, the majority of peptides in each sample are red~orange with relatively few uniquely identified peptides (blue~green). Samples are sorted by the total amount of detected peptides.

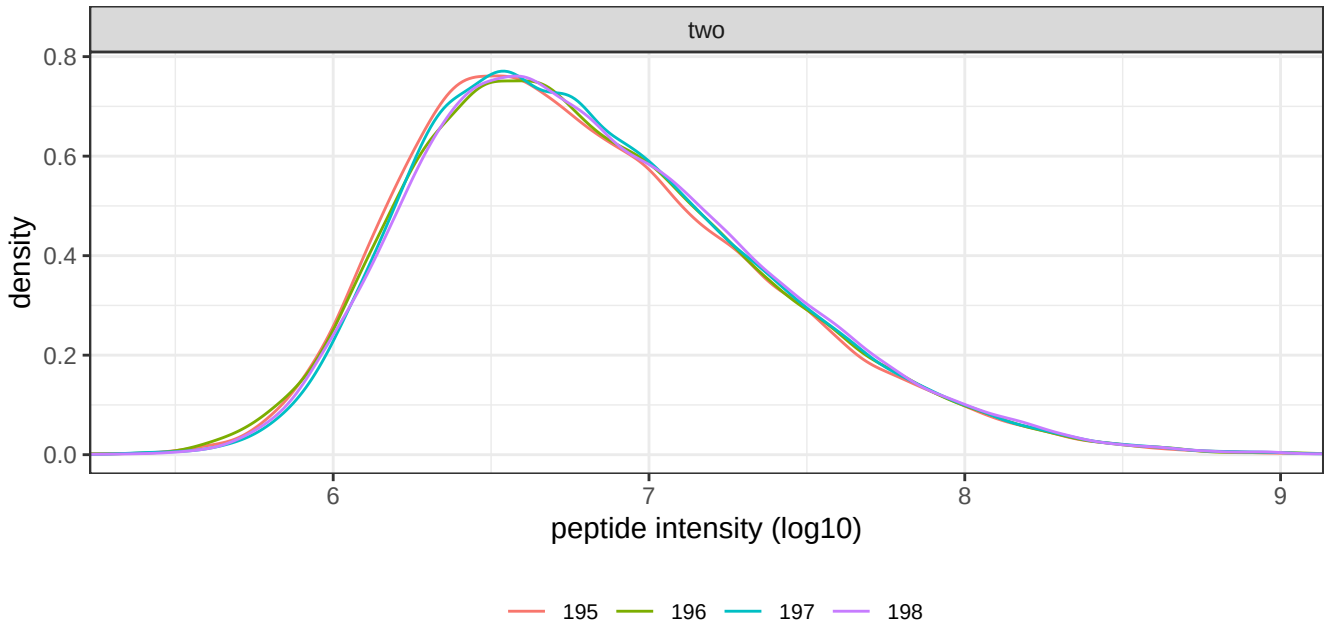

Number of samples in which a peptide is identified vs presence in individual sample
194
201
200
196
197
202
195
192
198
199
191
0
10000
20000
30000
identified peptides
#samples
10
8
6
4
2

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.




three
density
0.0
0.2
0.4
0.6
0.8
6
7
8
9
peptide intensity (log10)
199
200
201
202

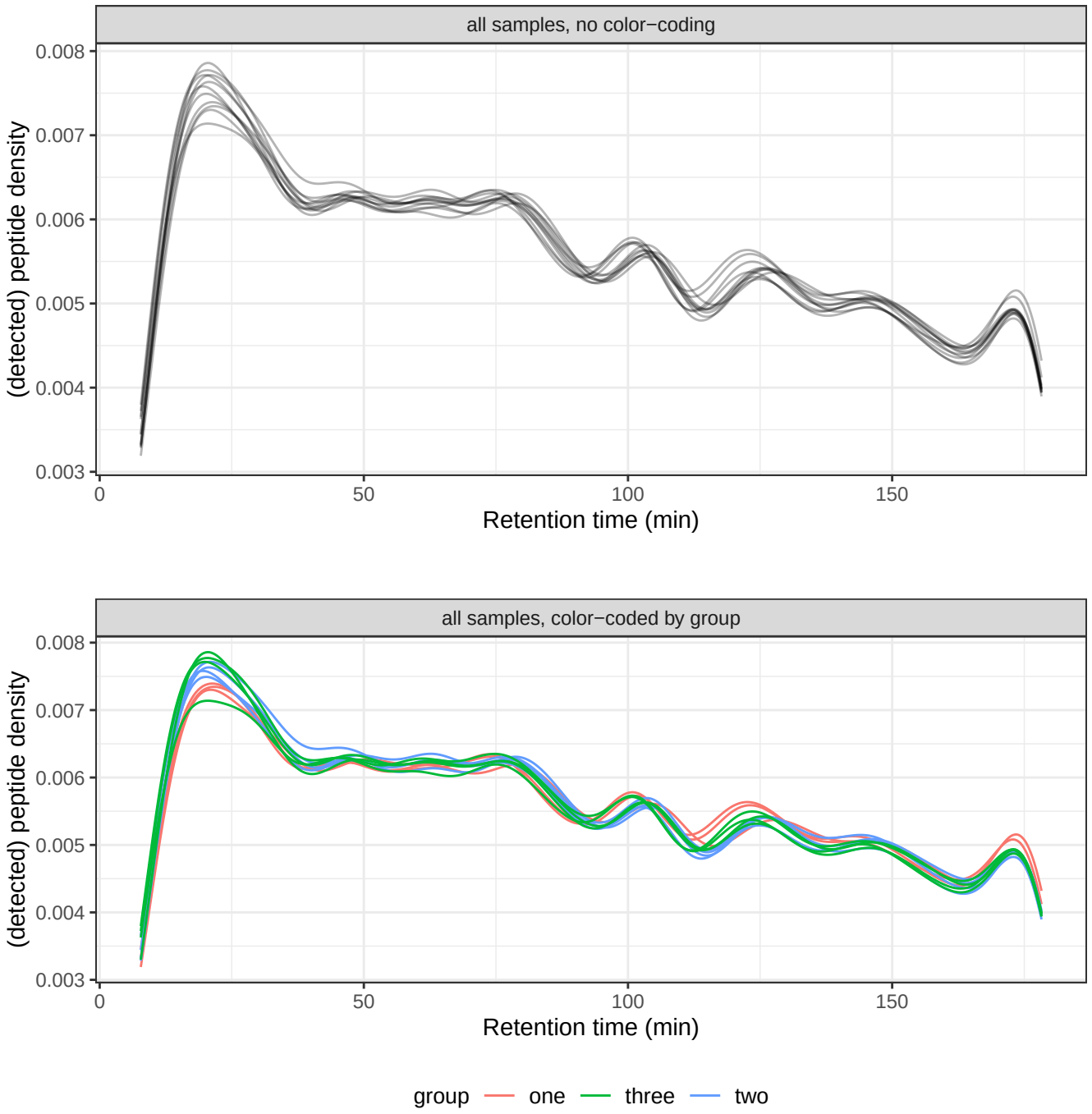

two
density
0.0
0.2
0.4
0.6
0.8
6
7
8
9
peptide intensity (log10)
195
196
197
198

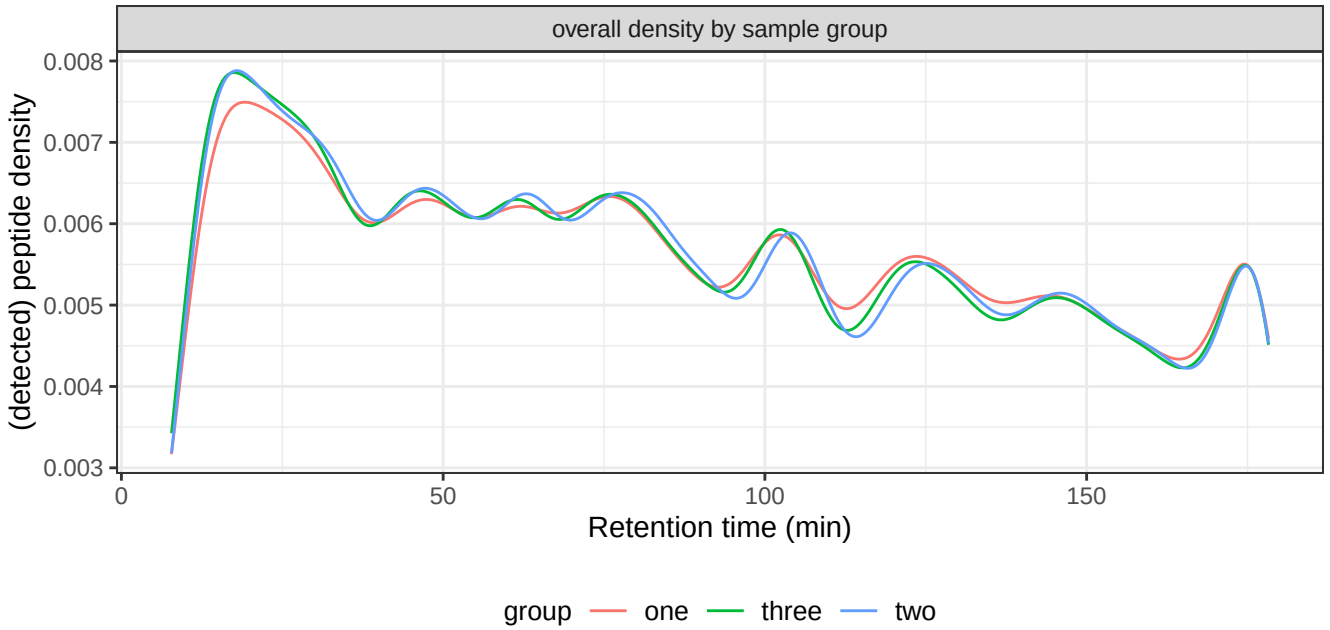
1.4 retention time

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.



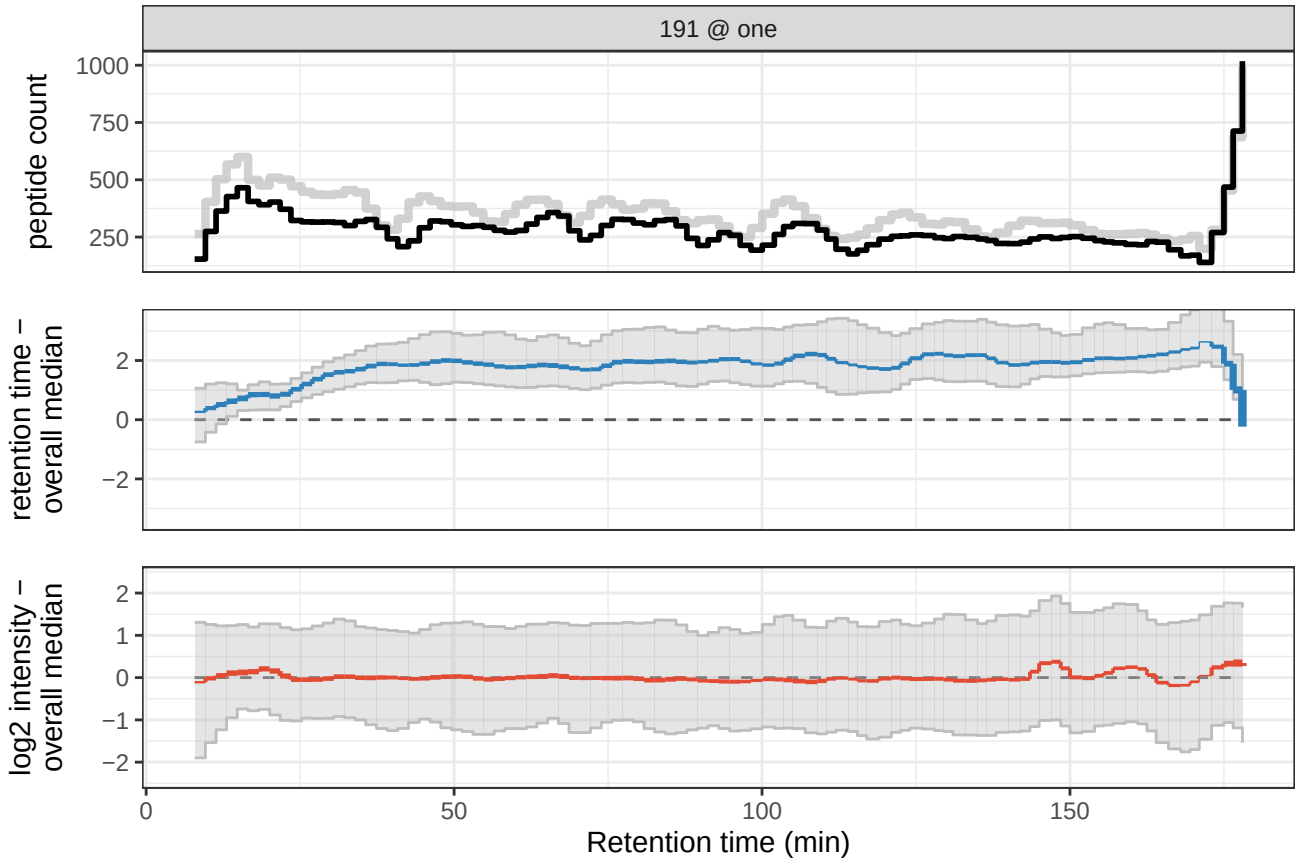


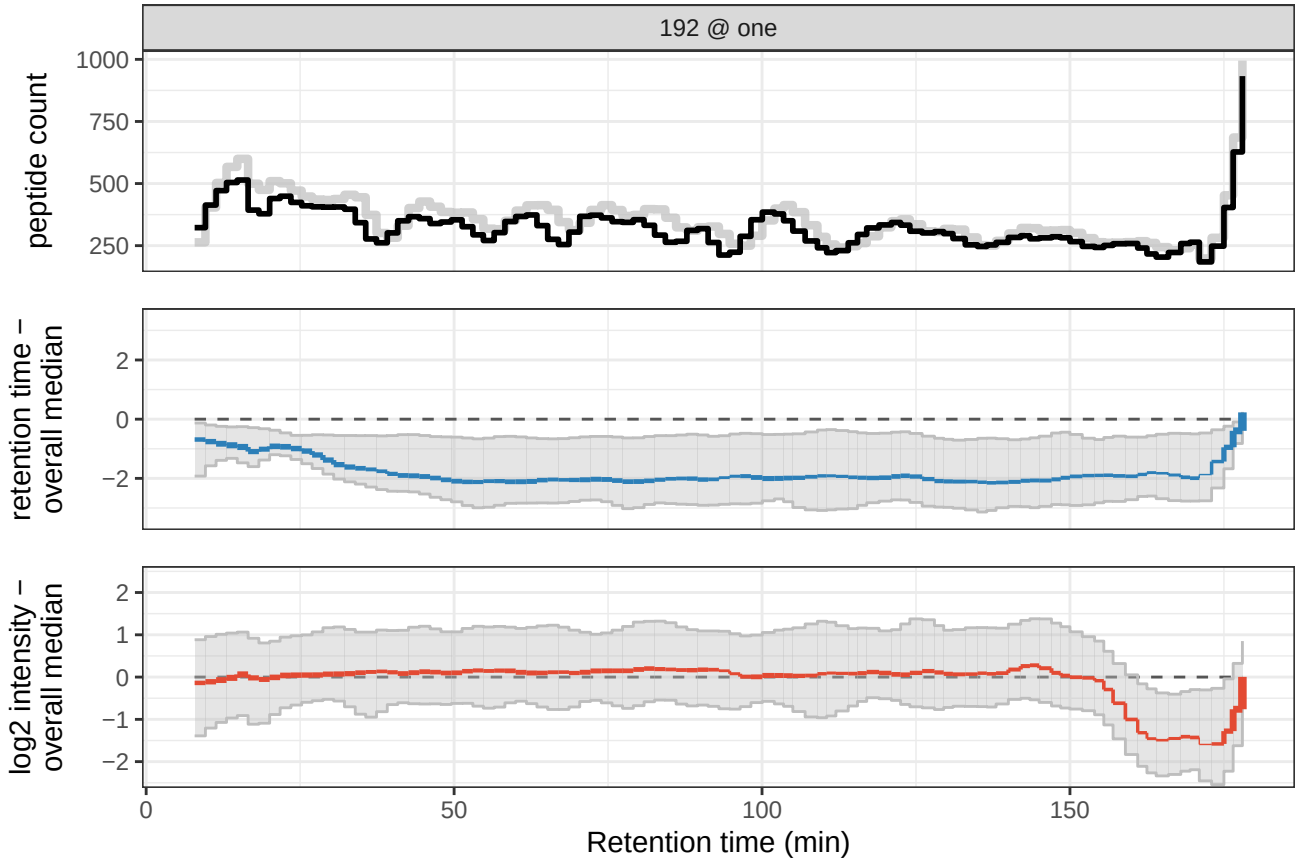
1.4.2 retention time local effects

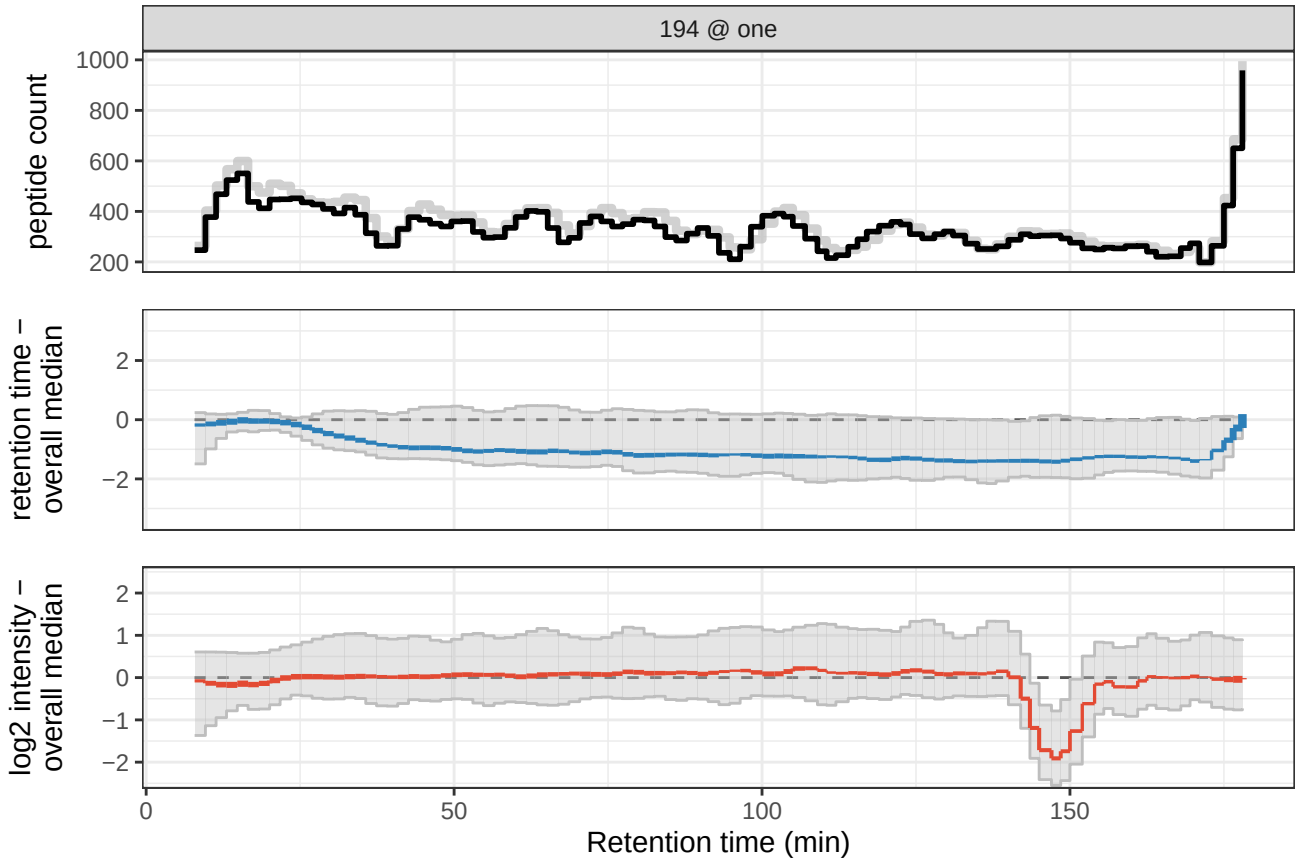
To investigate how each measurement differs from others, we visualize each sample as a 3 panel figure. First, the data is binned across the retention time dimension (x-axis). If a samples was marked as 'exclude' in the provided sample metadata, this is indicated in the plot title.

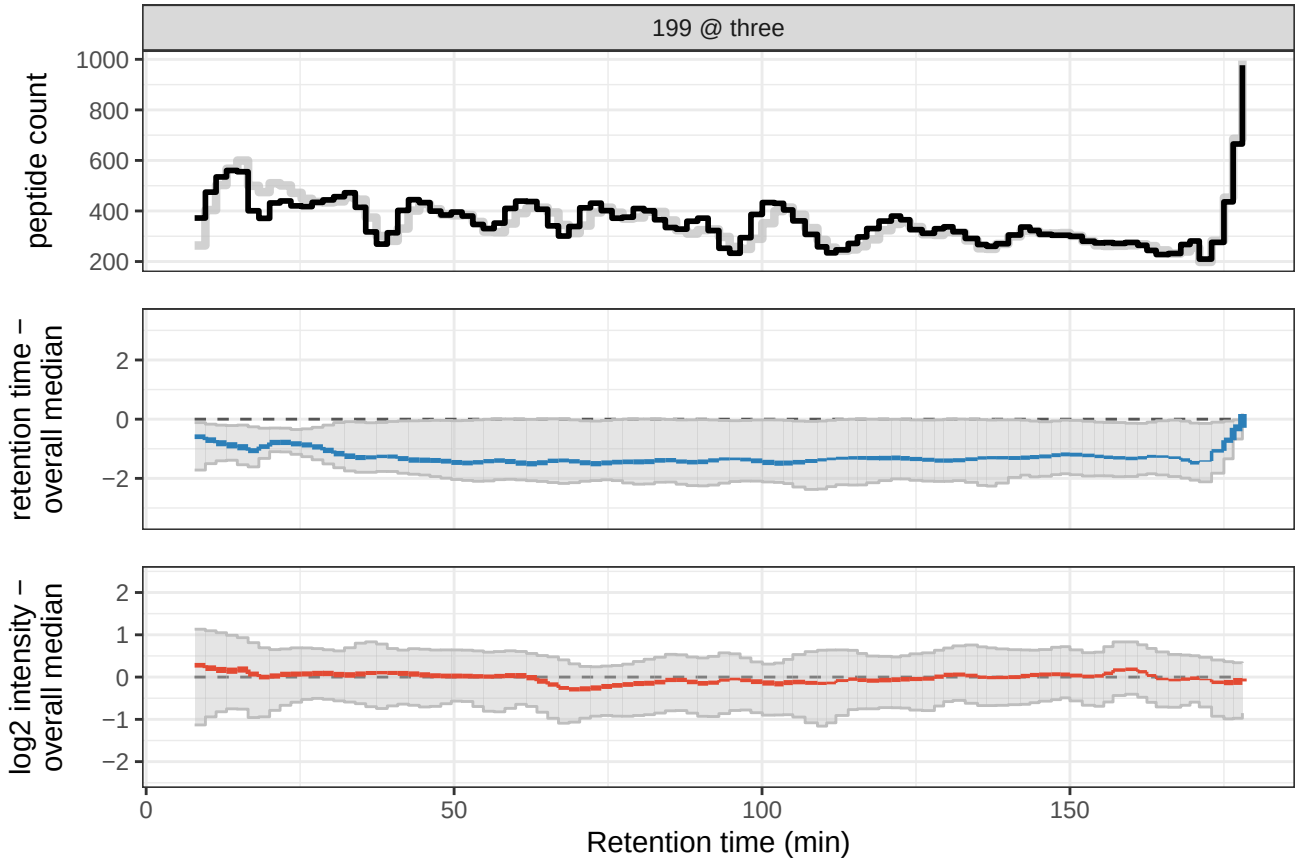
The top panel shows the number of peptides in the input data, e.g. as recognized by the software that generated input for this pipeline, over time (black line). For reference, the grey line shows the median amount over all samples (note; if this is the exact same in all samples, the grey line may not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

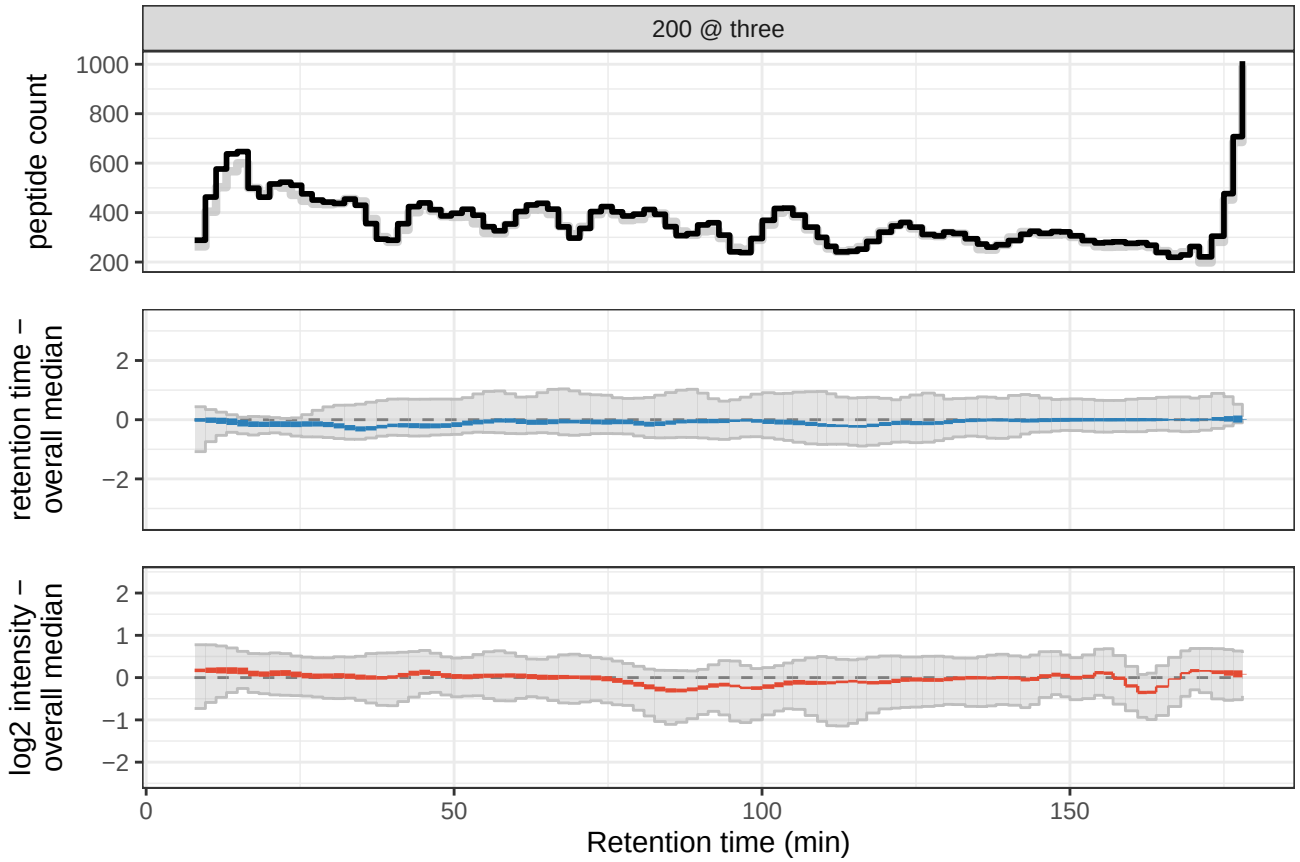
The middle panel indicates whether peptide retention times deviate from their median over all samples (blue line). The grey area depicts the 5% and 95% quantiles, respectively. The line width corresponds to the number of peptides eluting at that time (data from first panel). Analogously, the bottom panel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

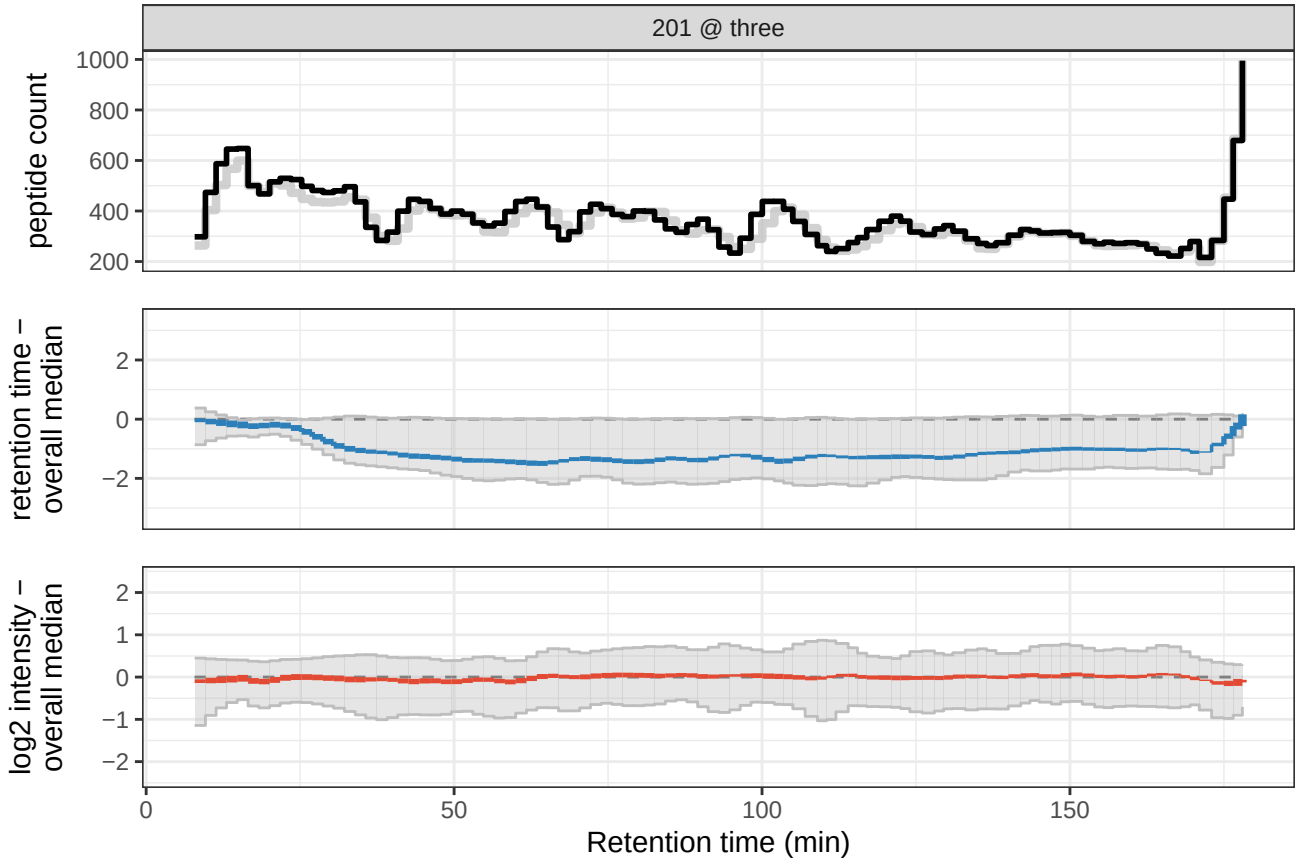

191 @ one
peptide count
1000
750
500
250
retention time –
overall median
2
0
−2
log2 intensity –
overall median
2
1
0
−1
−2
0
50
100
150
Retention time (min)

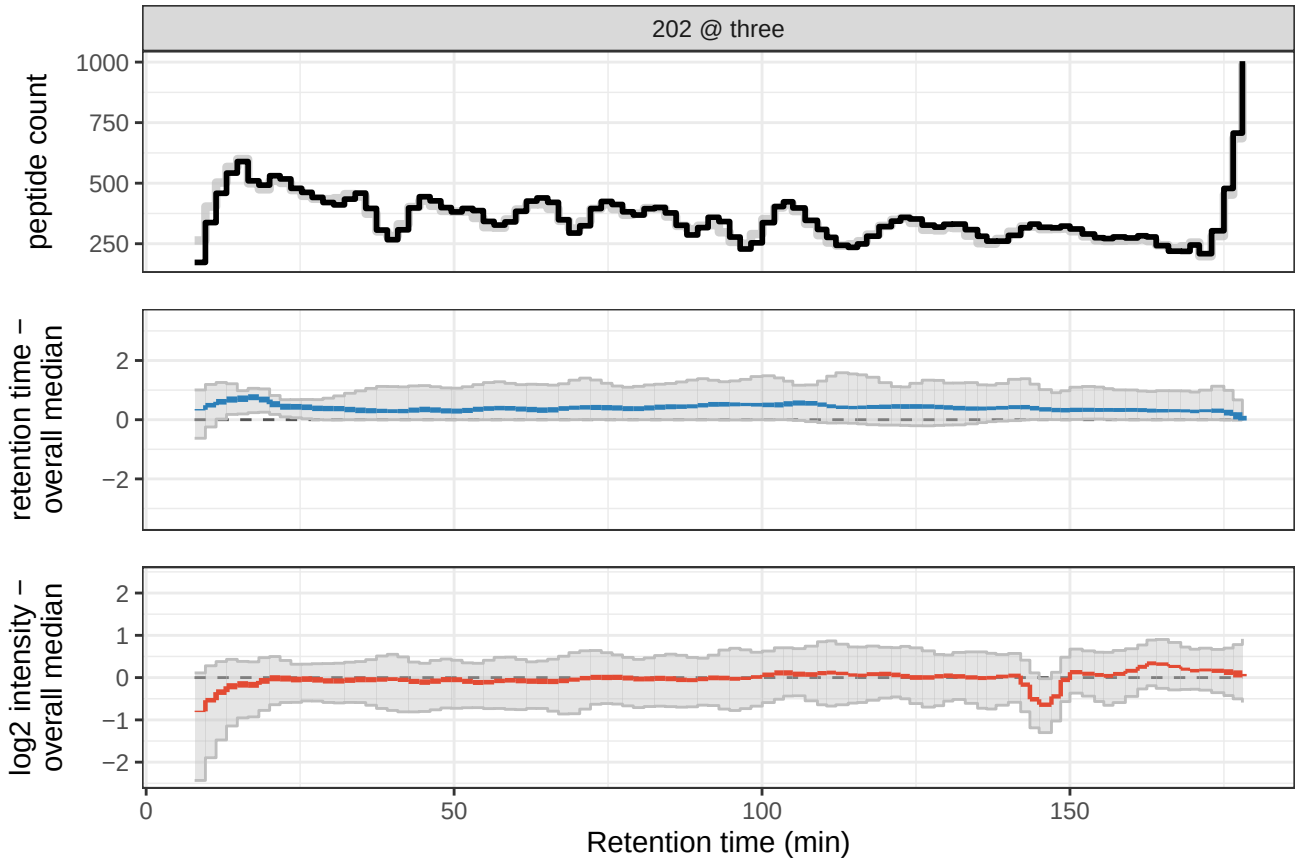

192 @ one
peptide count
1000
750
500
250
retention time –
overall median
2
0
−2
log2 intensity –
overall median
2
1
0
−1
−2
0
50
100
150
Retention time (min)

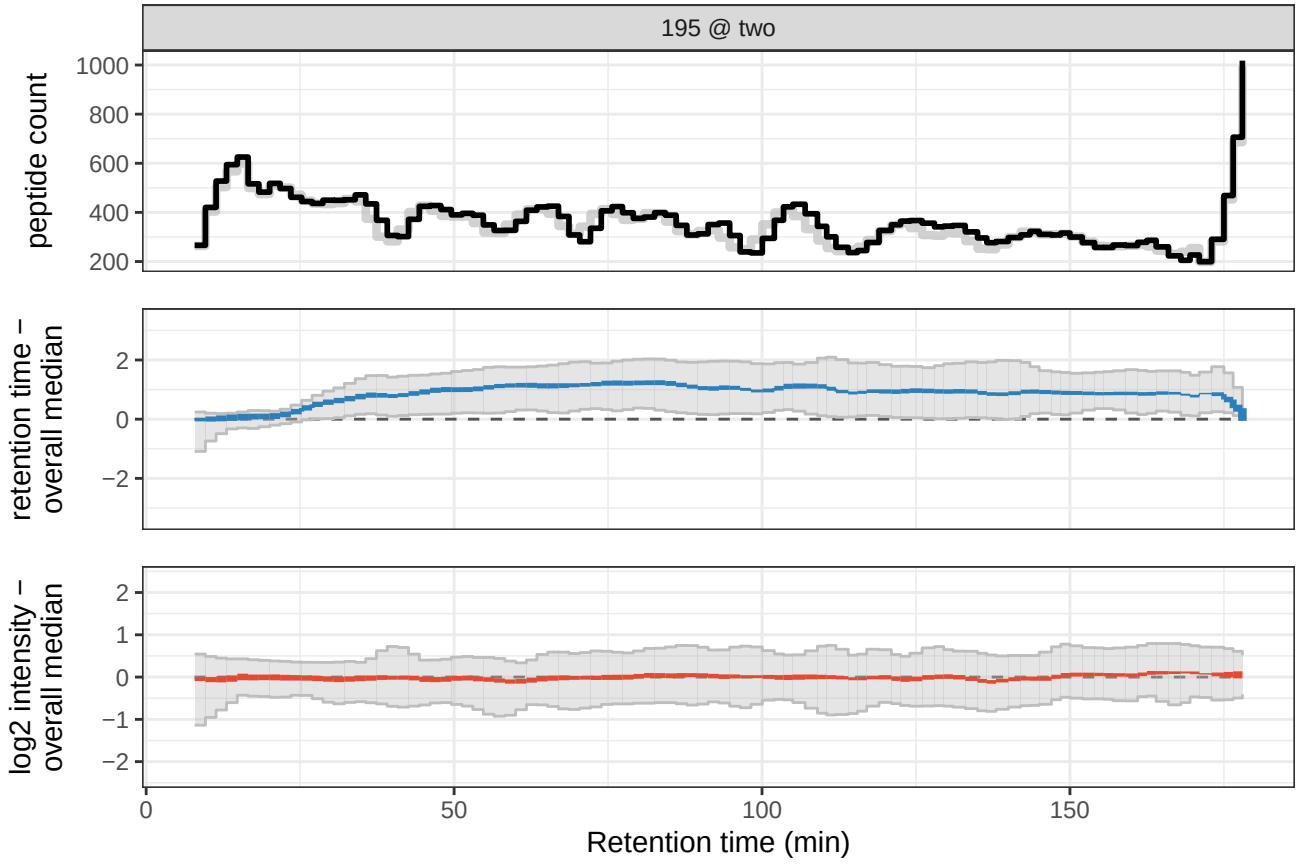

194 @ one
peptide count
1000
800
600
400
200
retention time –
overall median
2
0
−2
log2 intensity –
overall median
2
1
0
−1
−2
0
50
100
150
Retention time (min)

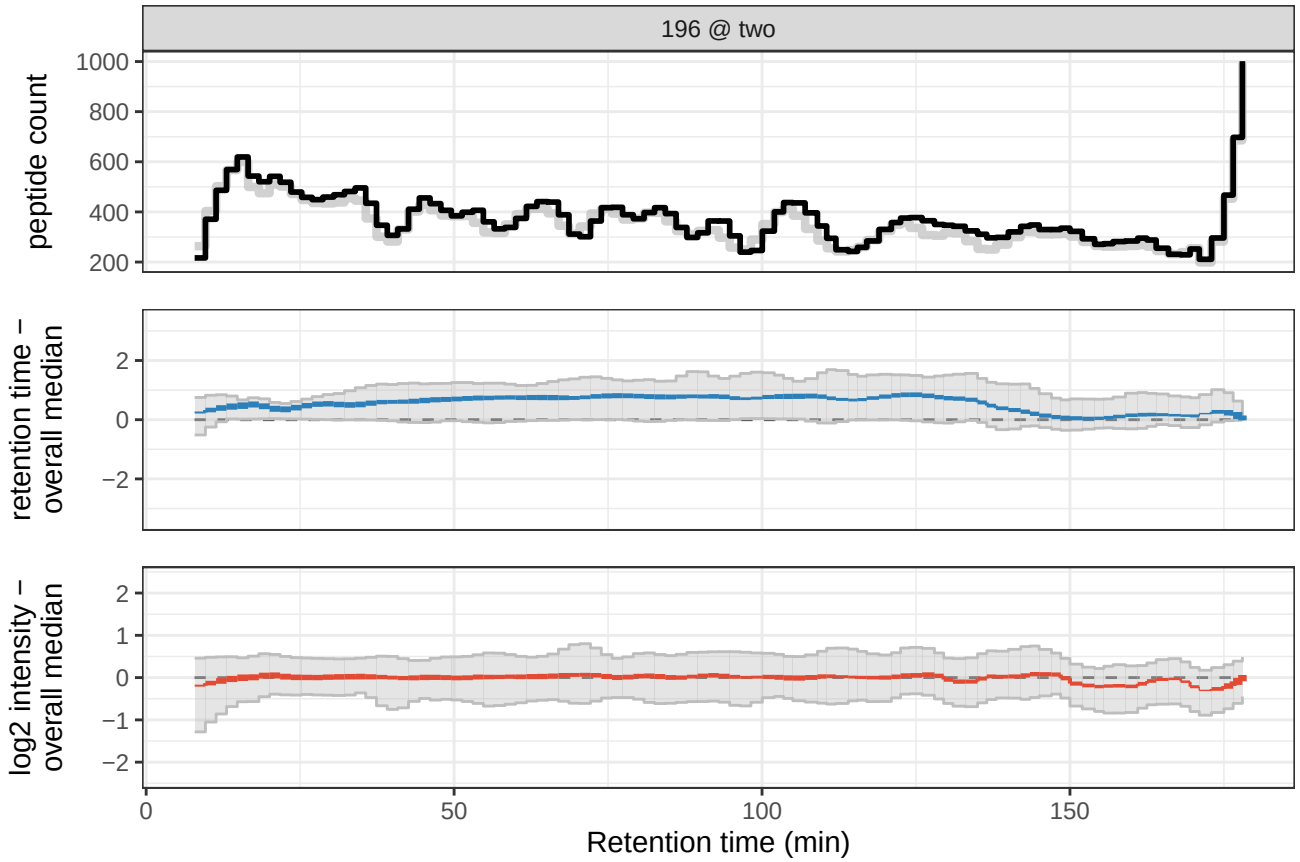

199 @ three
peptide count
1000
800
600
400
200
retention time –
overall median
2
0
−2
log2 intensity –
overall median
2
1
0
−1
−2
0
50
100
150
Retention time (min)

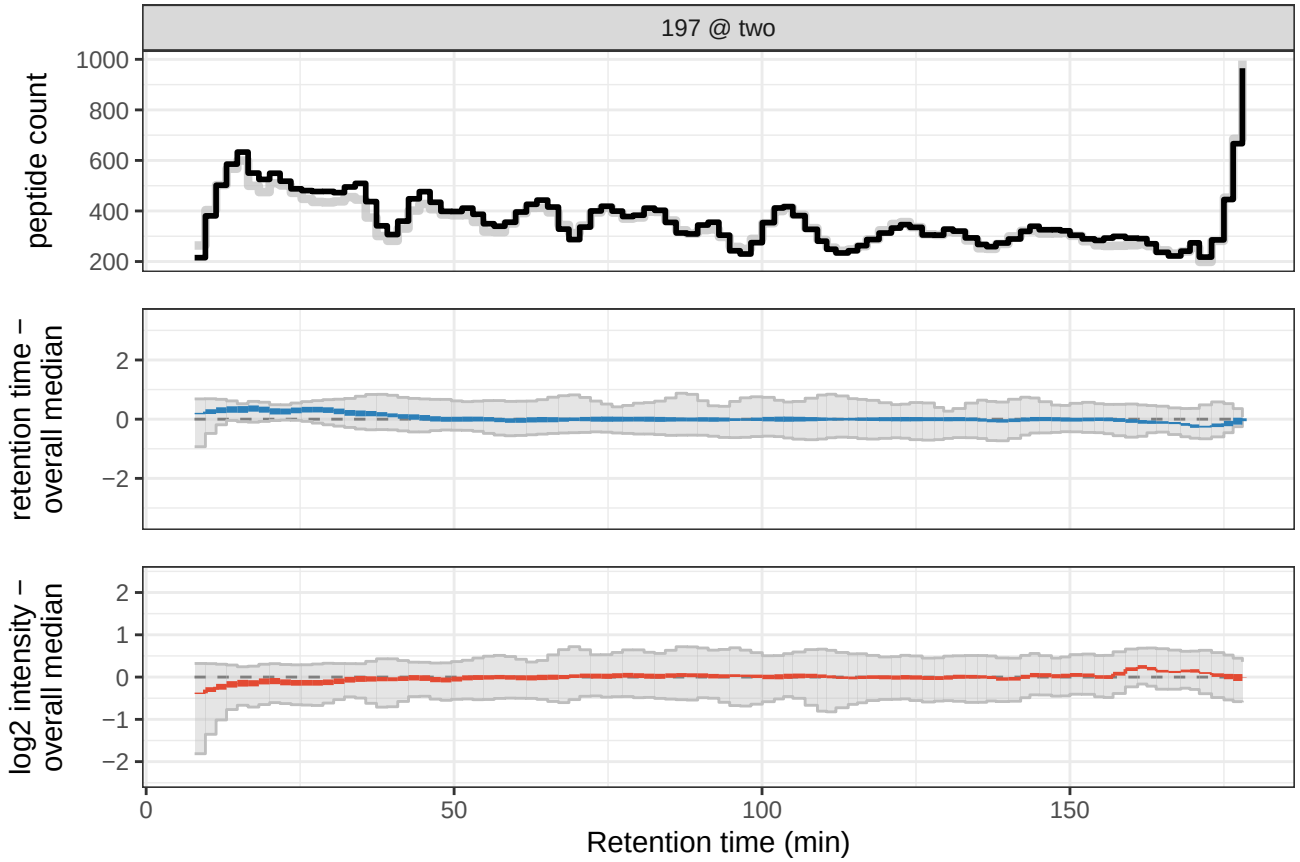

200 @ three
peptide count
1000
800
600
400
200
retention time – overall median
2
0
−2
log2 intensity – overall median
2
1
0
−1
−2
0
50
100
150
Retention time (min)

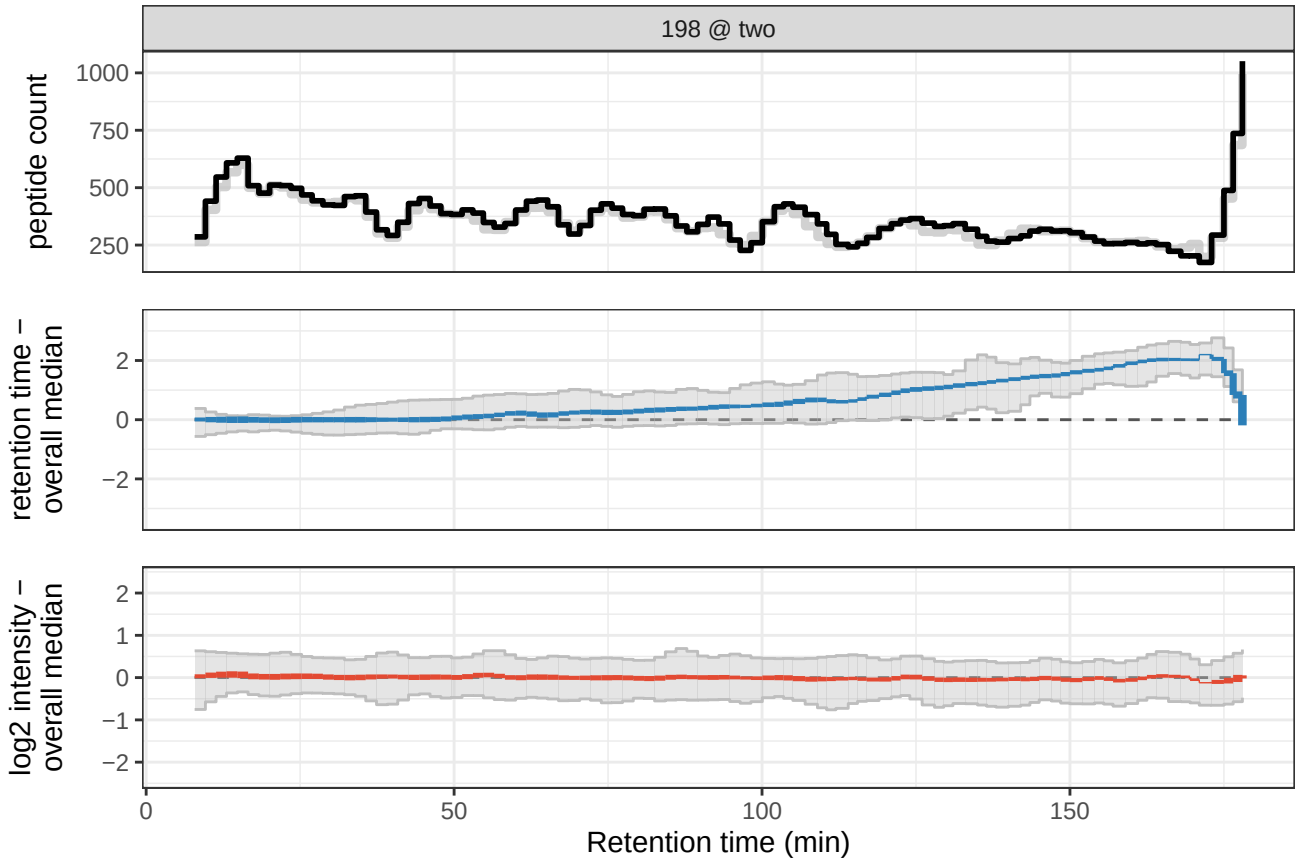

201 @ three
peptide count
1000
800
600
400
200
retention time – overall median
2
0
−2
log2 intensity – overall median
2
1
0
−1
−2
0
50
100
150
Retention time (min)


202 @ three
peptide count
1000
750
500
250
retention time –
overall median
2
0
−2
log2 intensity –
overall median
2
1
0
−1
−2
0
50
100
150
Retention time (min)


195 @ two
peptide count
1000
800
600
400
200
retention time –
overall median
2
0
−2
log2 intensity –
overall median
2
1
0
−1
−2
0
50
100
150
Retention time (min)


196 @ two
peptide count
1000
800
600
400
200
retention time –
overall median
2
0
−2
log2 intensity –
overall median
2
1
0
−1
−2
0
50
100
150
Retention time (min)


197 @ two
peptide count
1000
800
600
400
200
retention time –
overall median
2
0
−2
log2 intensity –
overall median
2
1
0
−1
−2
0
50
100
150
Retention time (min)


198 @ two
peptide count
1000
750
500
250
retention time – overall median
2
0
−2
log2 intensity – overall median
2
1
0
−1
−2
0
50
100
150
Retention time (min)

1.5 variation among replicates

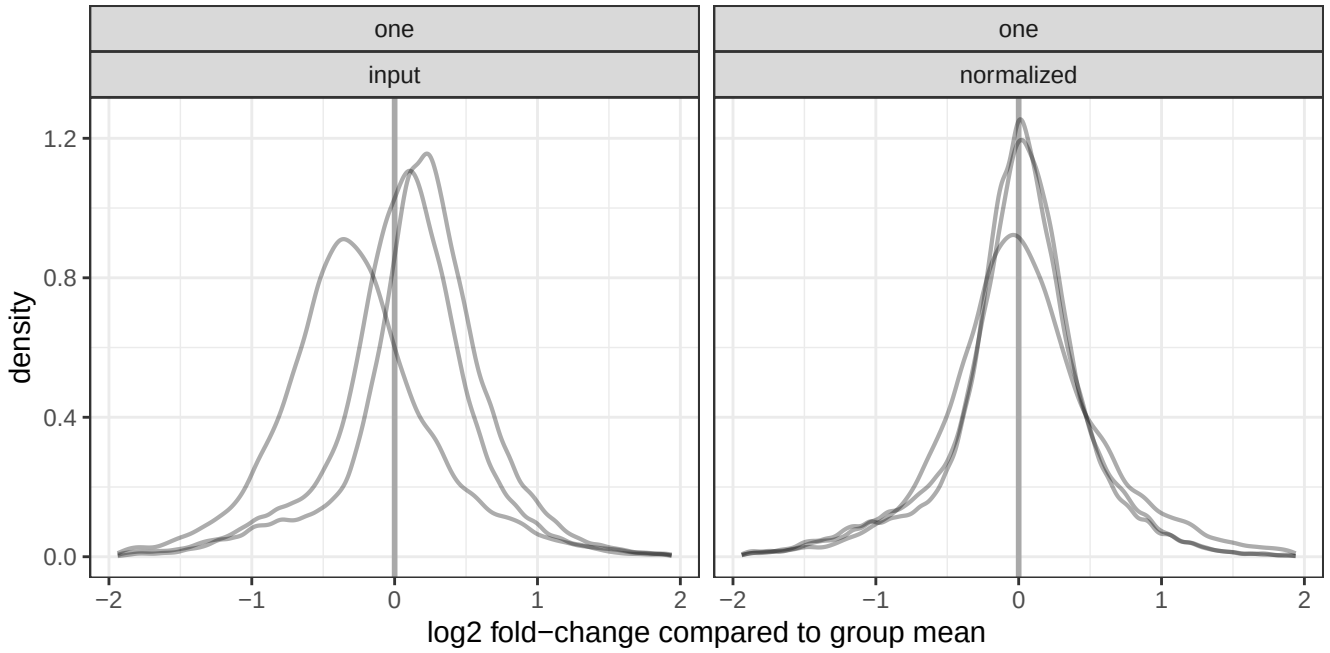
The reproducibility of replicate measurements is expressed in three different analyses. First, the difference between peptide intensities in each sample are compared to the mean value among all replicates (foldchange distributions). Next, the Coefficient of Variation (CoV) is used as a metric for reproducibility 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).

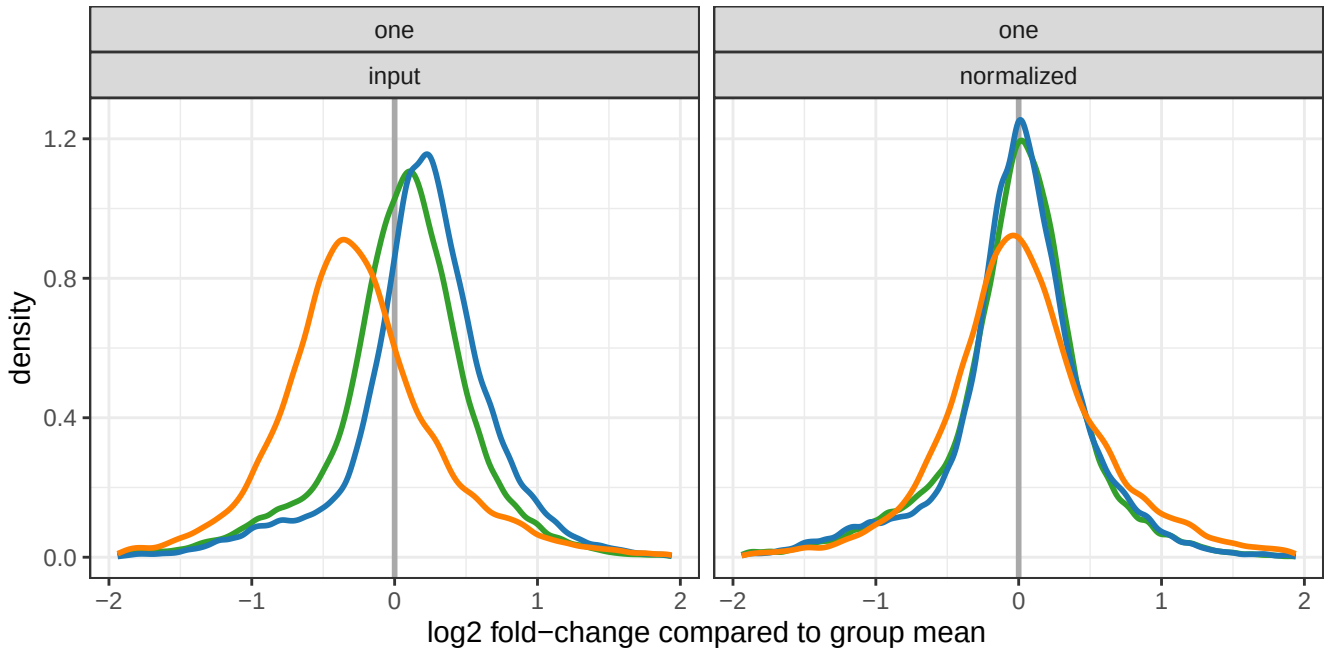
1.5.1 within-group foldchange distributions

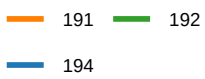
The foldchange of all peptides in a sample is compared to their respective mean value over all samples in the group. This visualizes how strongly each sample deviates from other samples in the same group which helps identify outlier samples. The same data was used as detailed in the "retention time" section above.

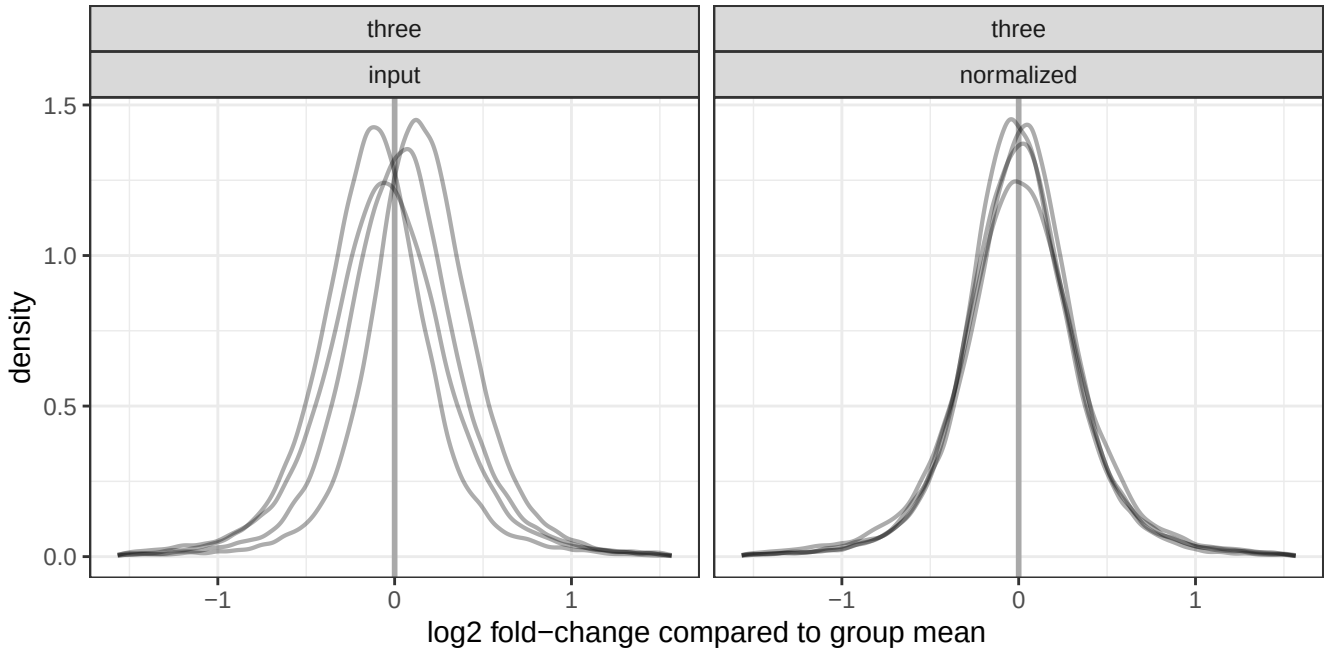
For each sample group, two plots are shown: 1) a basic monochrome plot and 2) a variant that color-codes the top10 'worst' samples based on the standard deviation (sd) of their respective distributions. Note; per sample, the 0.5% quantiles of both top- and bottom-most outliers are disregarded in sd computation. The legend is sorted in column-first descending order (sample with highest sd in column 1 row 1, sample with second highest sd in column 1 row 2, etc.). If there are 10 or fewer samples, all are color-coded. On the pages after the plots, these sd values are shown as tables.

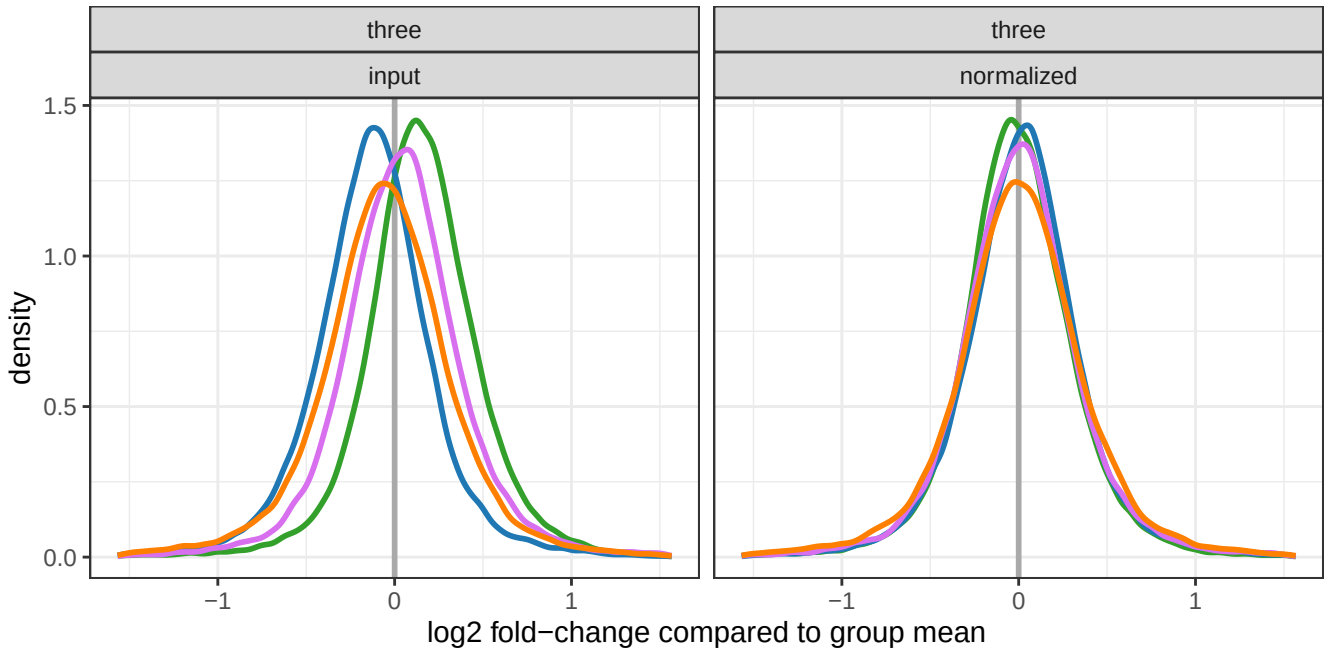
The 'input' panel is based on the peptide intensities as-is (i.e. the user-provided input data from upstream software), the 'normalized' panel shows the exact same samples and peptides after normalization (as specified by user). Samples marked as 'exclude' in the provided sample metadata table are visualized as dashed lines.

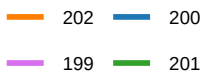

one
input
one
normalized
density
1.2
0.8
0.4
0.0
−2
−1
0
1
2
log2 fold−change compared to group mean

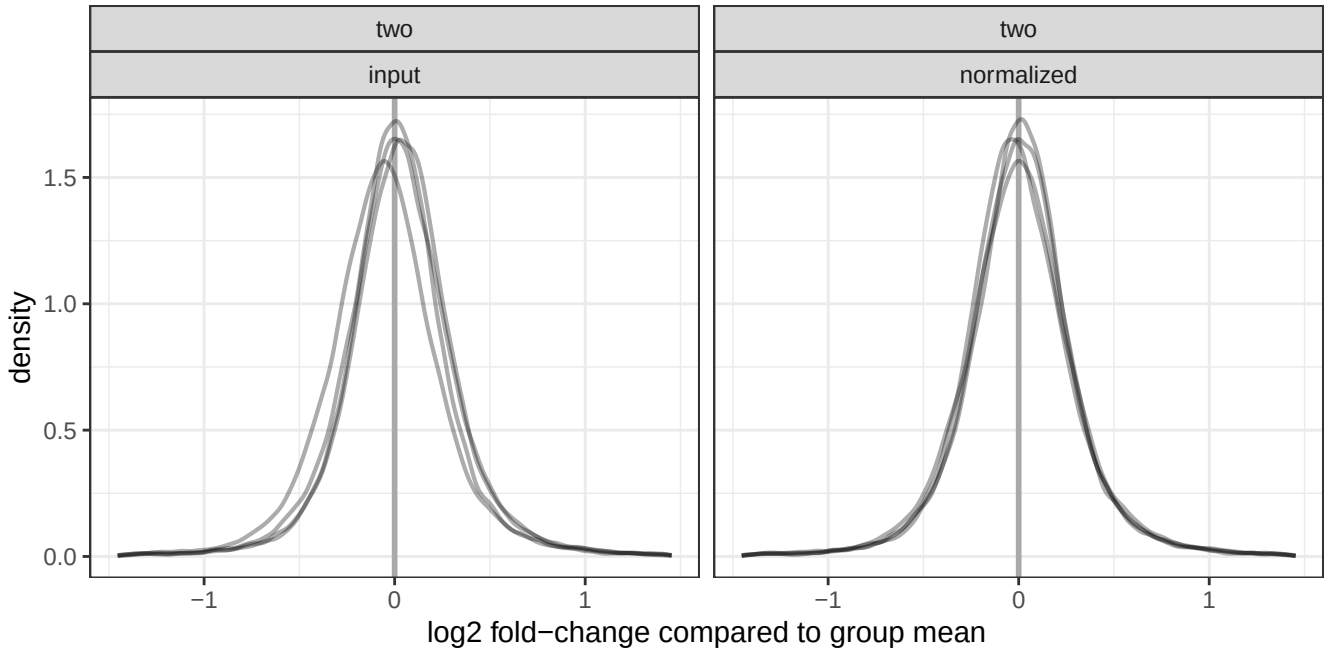

one
input
one
normalized
density
1.2
0.8
0.4
0.0
−2
−1
0
1
2
log2 fold−change compared to group mean

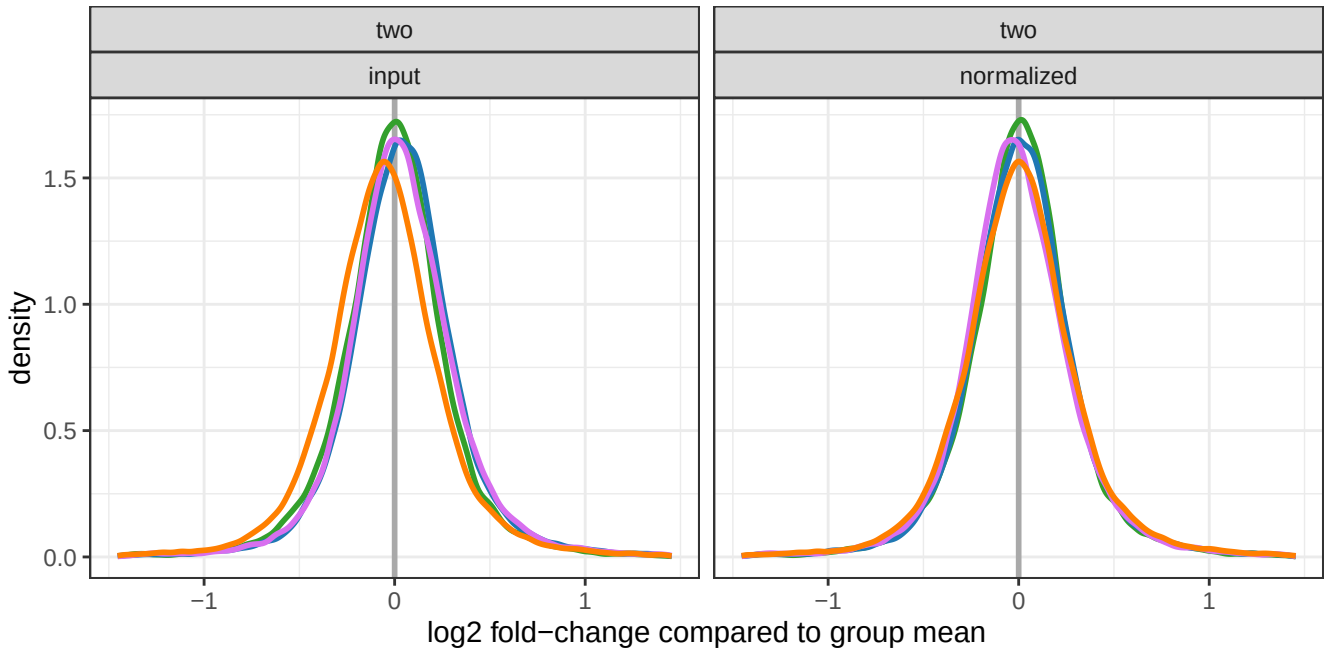

191
192
194

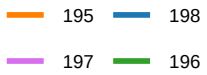

three
input
three
normalized
density
1.5
1.0
0.5
0.0
−1
0
1
log2 fold−change compared to group mean


three
input
three
normalized
density
1.5
1.0
0.5
0.0
−1
0
1
log2 fold−change compared to group mean


202
200
199
201


two
input
two
normalized
density
1.5
1.0
0.5
0.0
−1
0
1
log2 fold−change compared to group mean


two
input
two
normalized
density
1.5
1.0
0.5
0.0
−1
0
1
log2 fold−change compared to group mean


195
198
197
196

top10 outliers per sample group

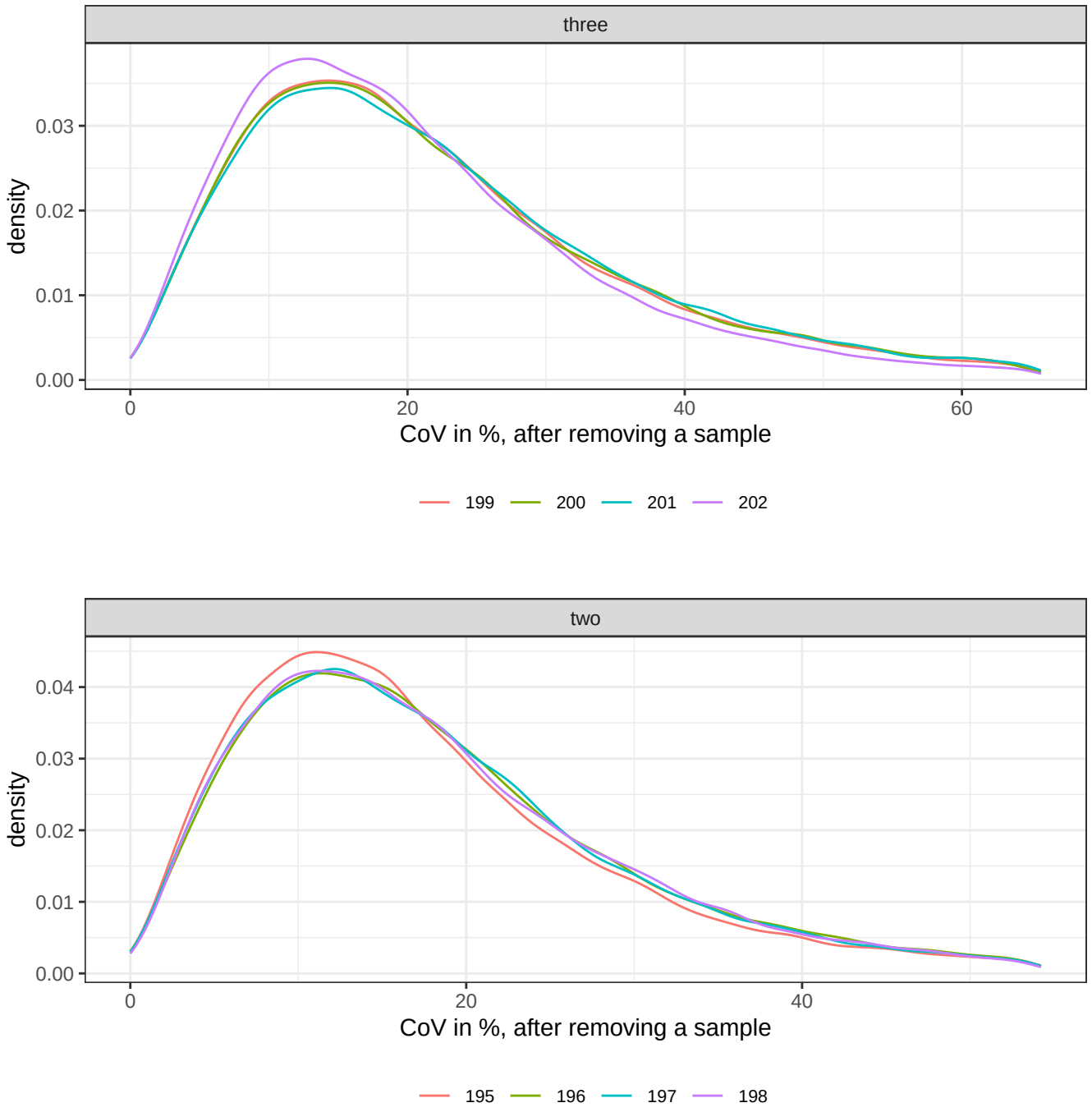
Group	Sample	Standard Deviation
one	191	0.549
one	194	0.471
one	192	0.470

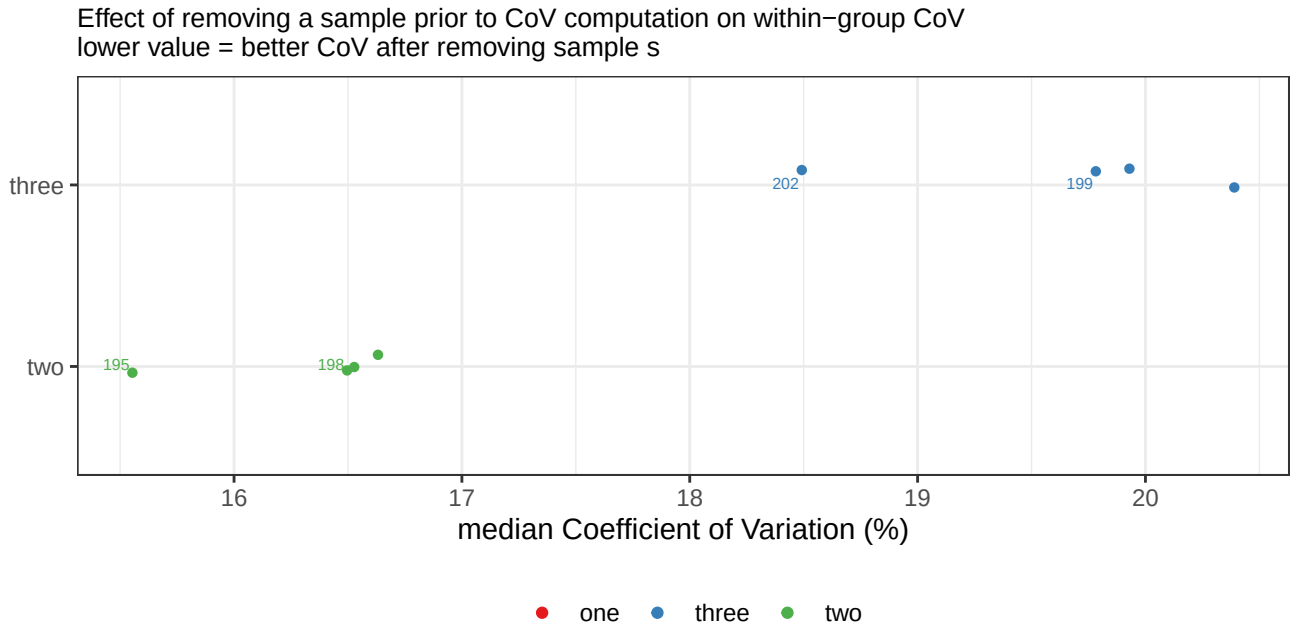
Group	Sample	Standard Deviation
three	202	0.387
three	199	0.346
three	200	0.334
three	201	0.321

Group	Sample	Standard Deviation
two	195	0.318
two	197	0.301
two	198	0.293
two	196	0.290

1.5.2 CoV, leave-one-out

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 replicate samples (after leave-one-out) are used for each CoV computation.





shortname	group	exclude	median CoV
202	three	FALSE	18.5
199	three	FALSE	19.8
200	three	FALSE	19.9
201	three	FALSE	20.4
195	two	FALSE	15.6
198	two	FALSE	16.5
197	two	FALSE	16.5
196	two	FALSE	16.6

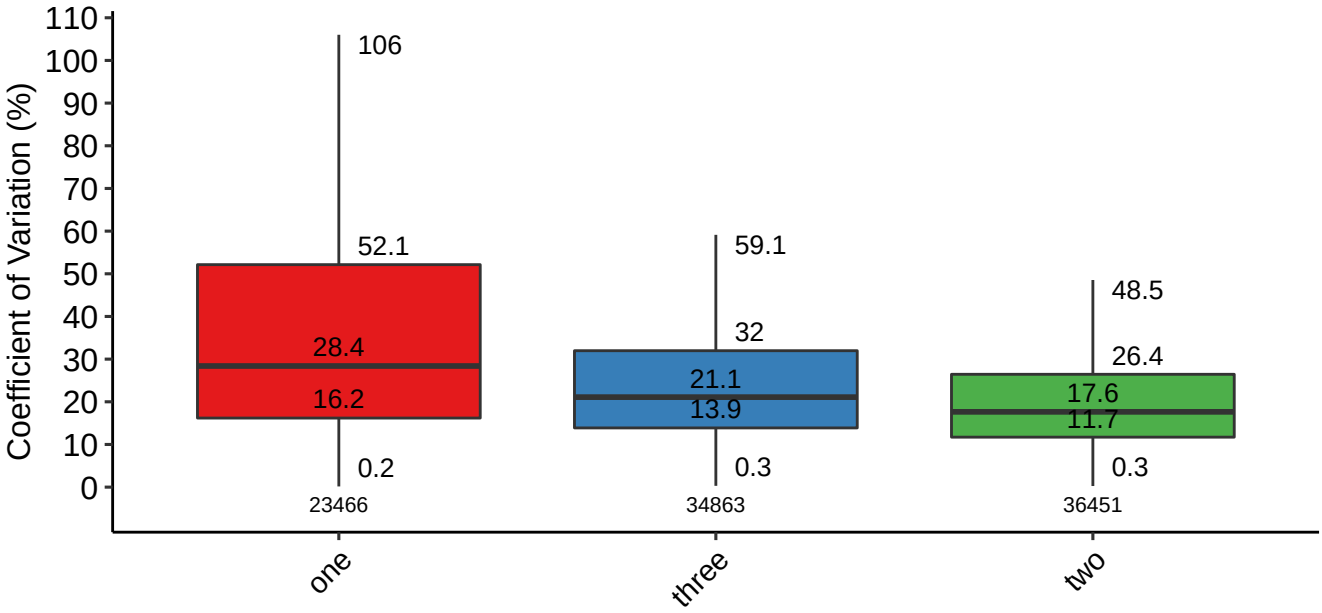
Leave-one-out impact on within-group CoV (%)

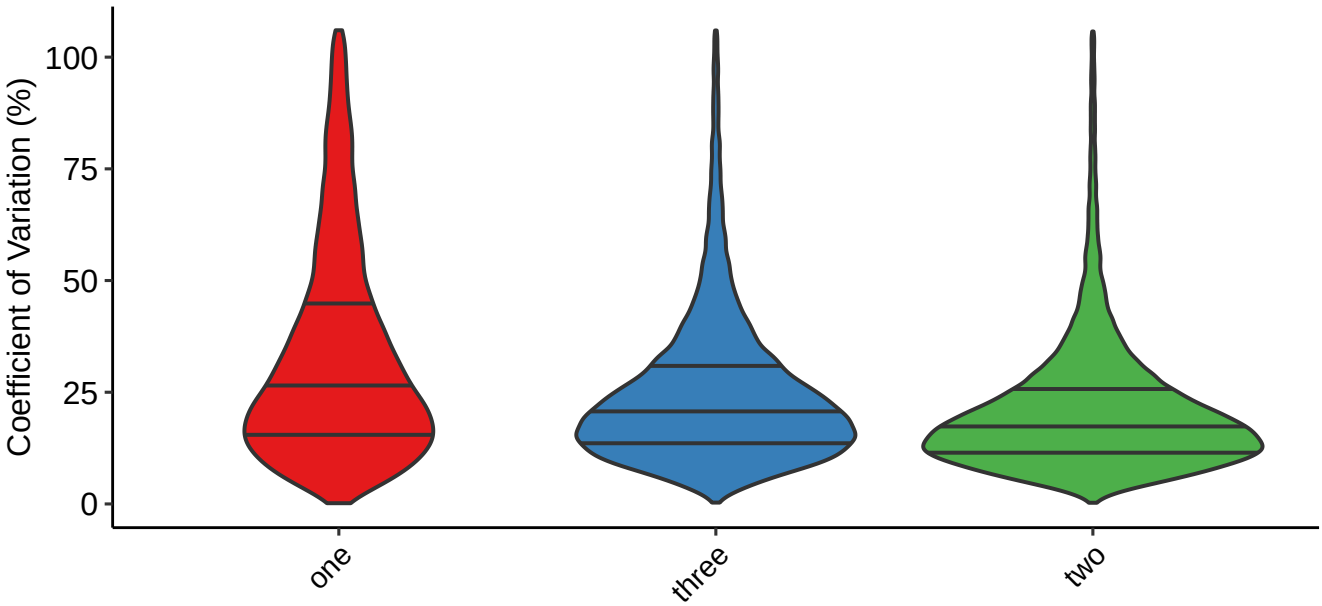
1.5.3 Coefficient of Variation

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

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. The user-specified filtering rules (eg; filter_min_detect, filter_min_peptide_per_prot, etc.) were applied within each sample group independently and remaining peptides were subsequently normalized using the " vsn&modebetween_protein " algorithm (an important parameter to keep in mind when comparing across datasets/analyses). Only peptides with at least 3 data points across replicate samples are used for each CoV computation.

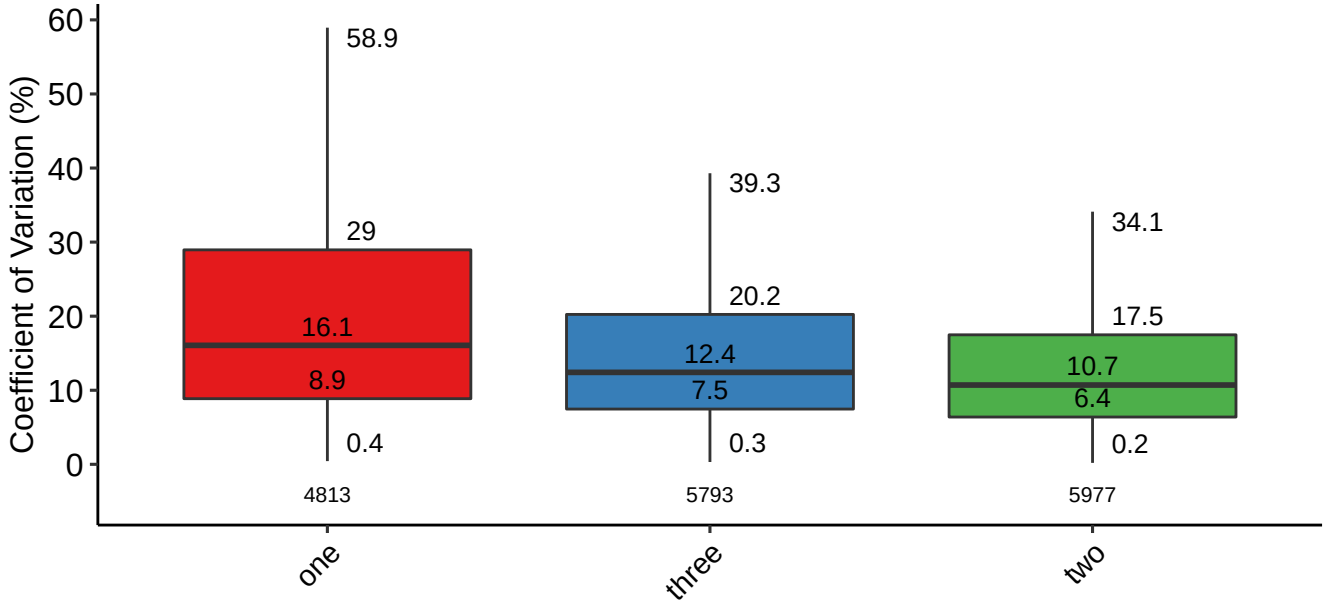
Peptide-level CoV:

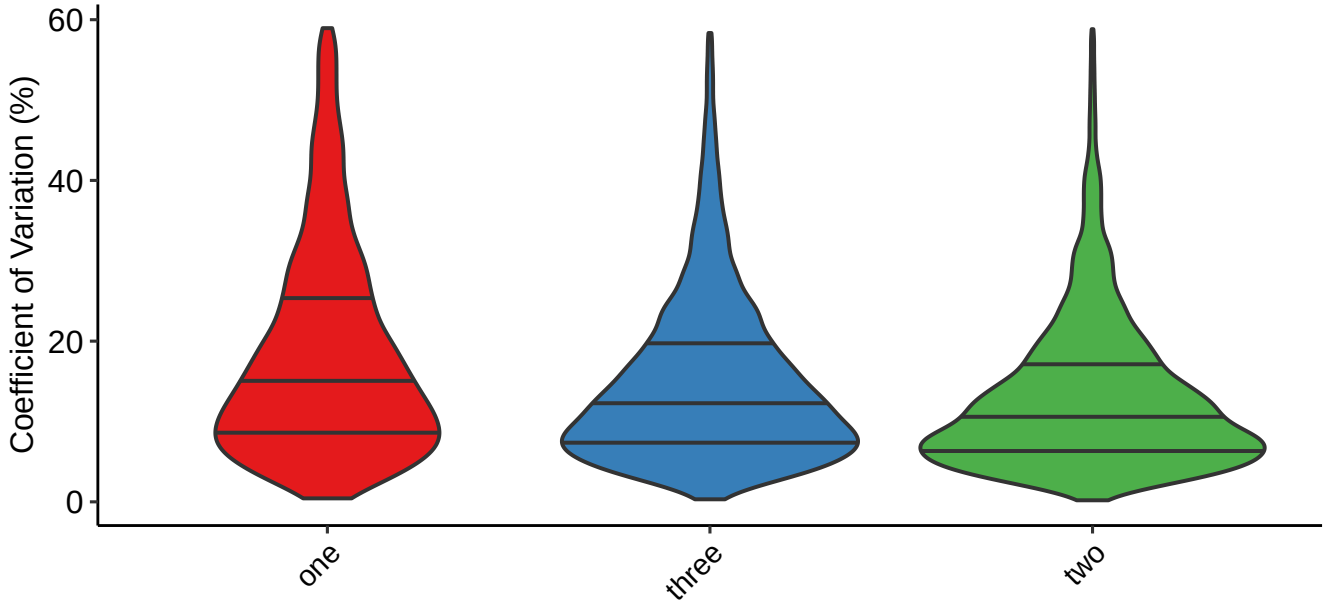




Protein-level CoV:

(analogous to peptide CoV's, but with additional rollup to protein abundances using 'maxlfq' algorithm)





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).

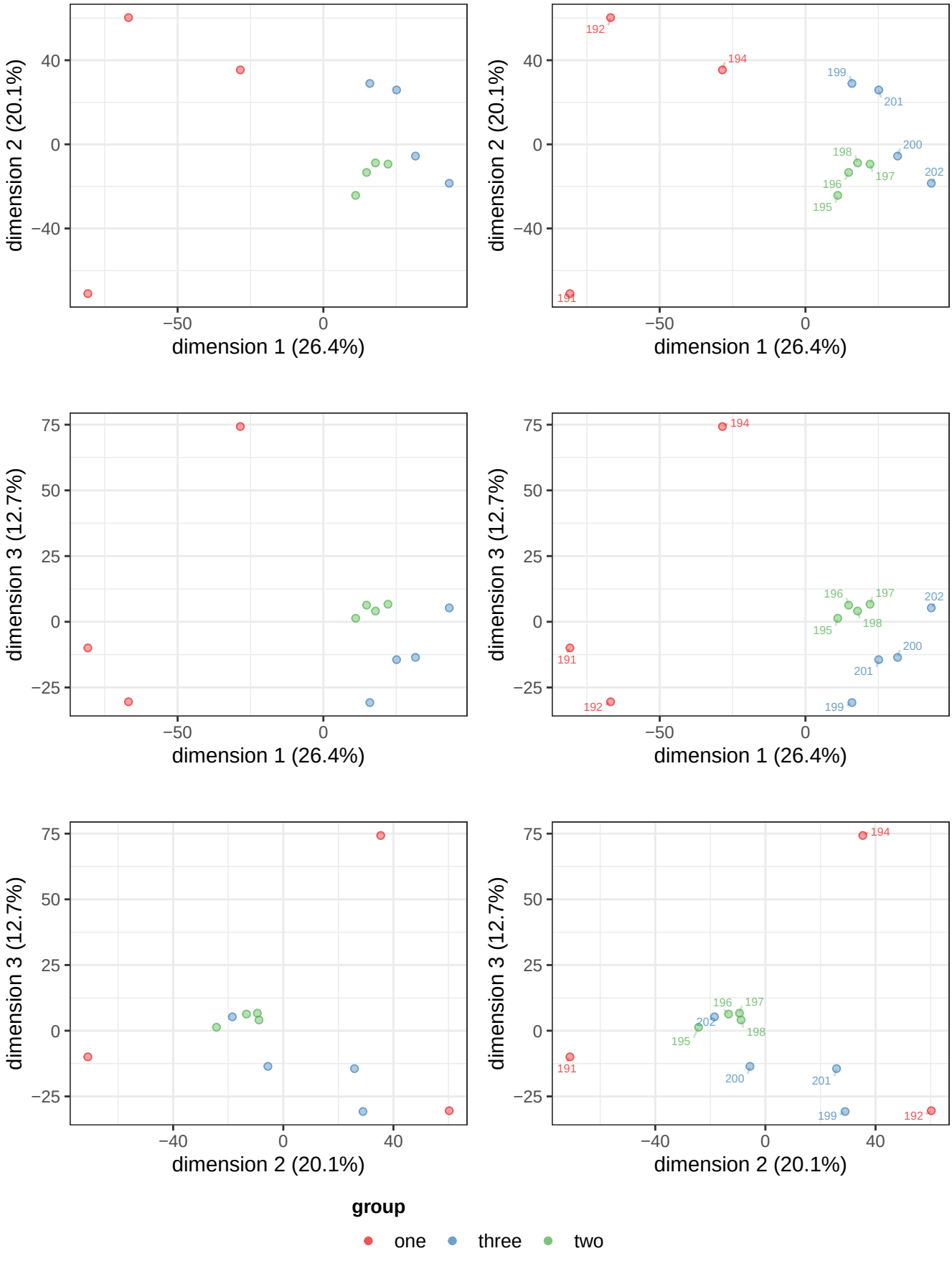
If additional sample metadata was provided, such as experiment batch, sample-prep dates, gel, etc., multiple PCA figures will be generated with respective color-codings. Users are encouraged to provide relevant experiment information as sample metadata and use these figures to search for unexpected batch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

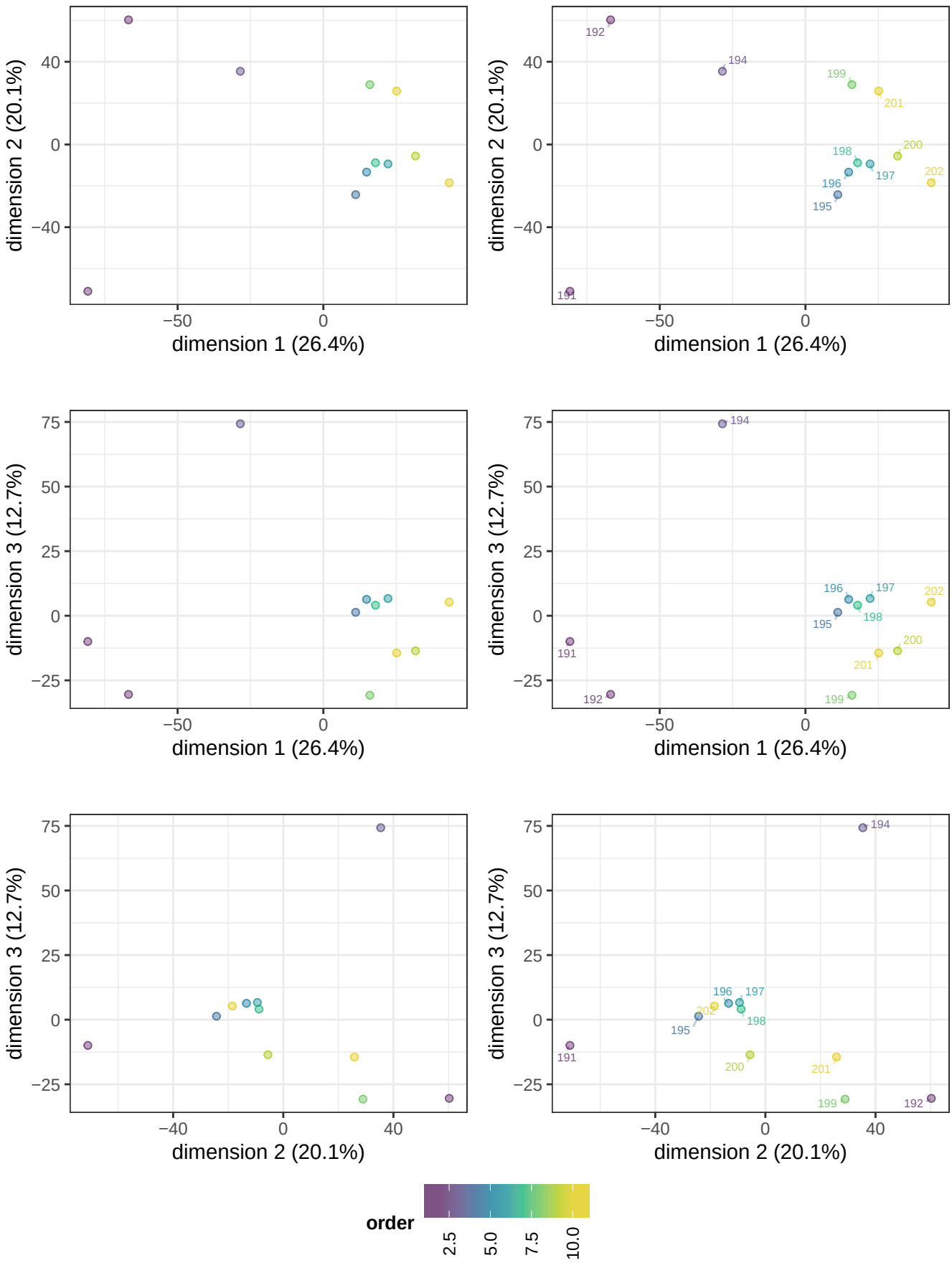
The pcaMethods R package is used here to perform the Probabilistic PCA (PPCA). The set of peptides used for this analysis consists of those peptides that pass your filter criteria in every sample group. If any samples are marked as 'exclude' in the provided sample metadata, an additional PCA plot is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

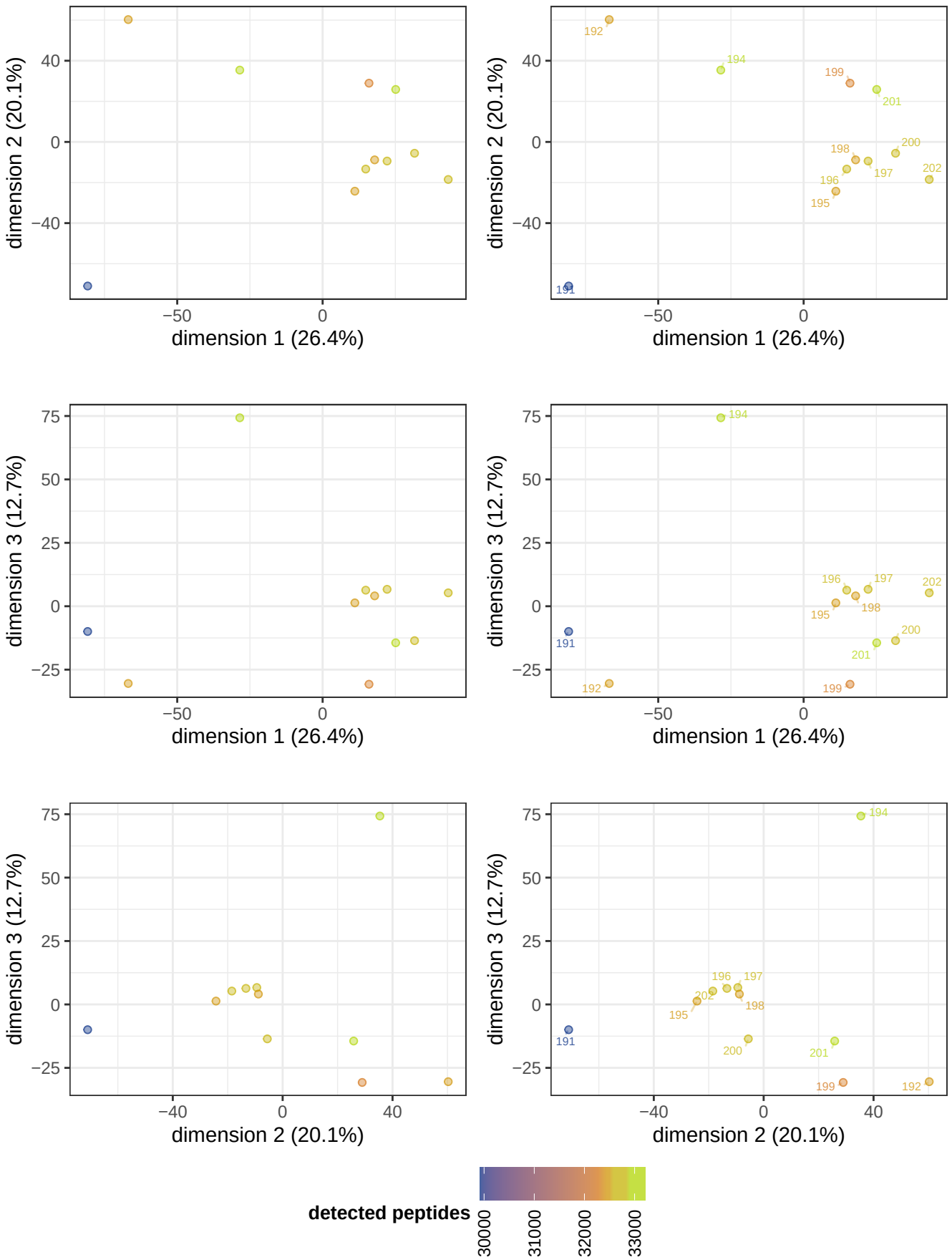
Rationale behind data filter
As mentioned above, the aim of the PCA figures is to identify global effects. To achieve this, we compute sample distances on the subset of peptides identified in each group which prevents rarely detected peptides/proteins from having a disproportionate effect on sample clustering. 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).

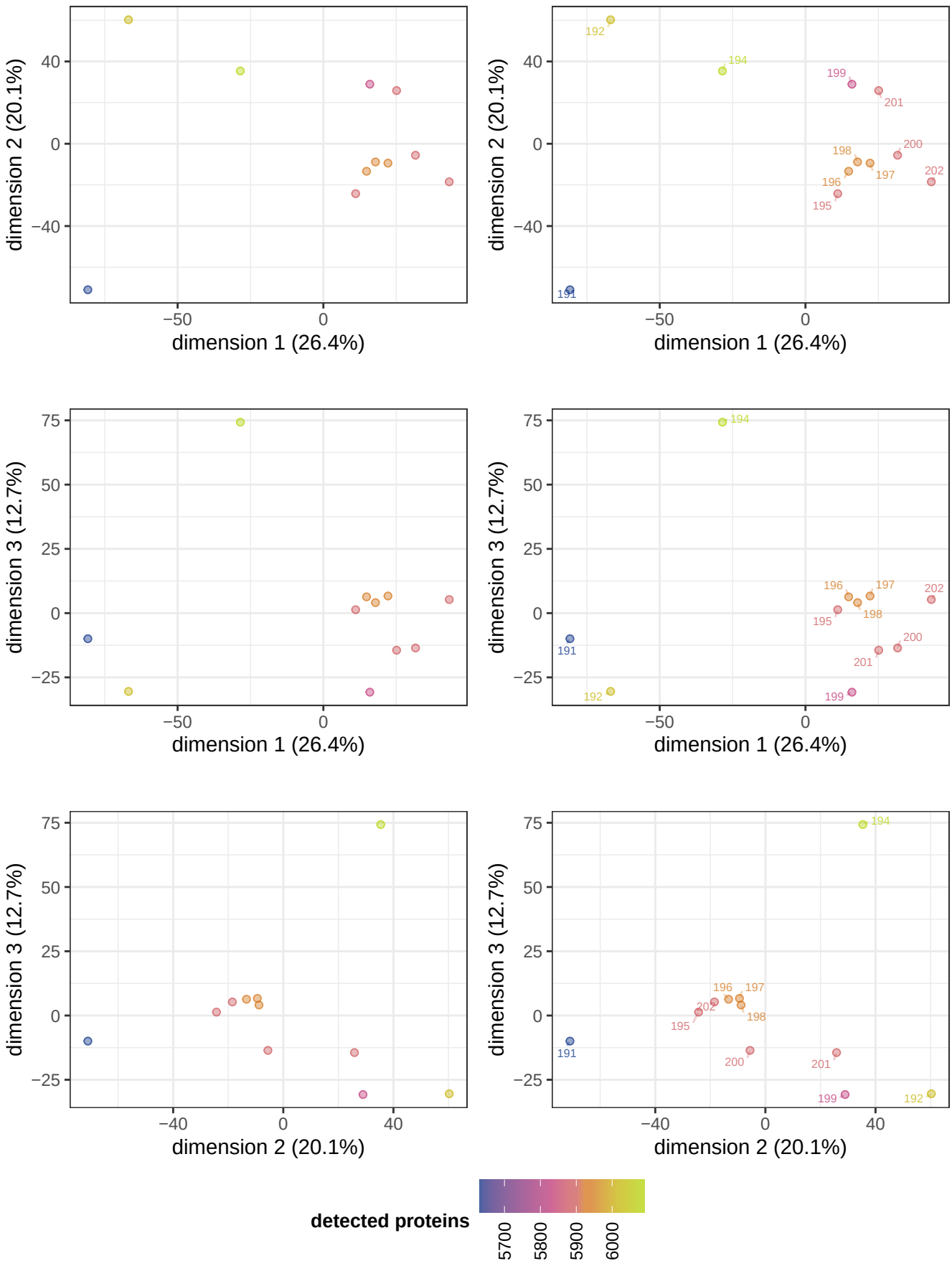
Figure legends
The first 3 principle components compared visually (1 *vs* 2, 1 *vs* 3, 2 *vs* 3) on the rows. Left- and right-side panels on each row represent the same figure without and with sample labels. The principle components are shown on the axis labels together with their respective percentage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

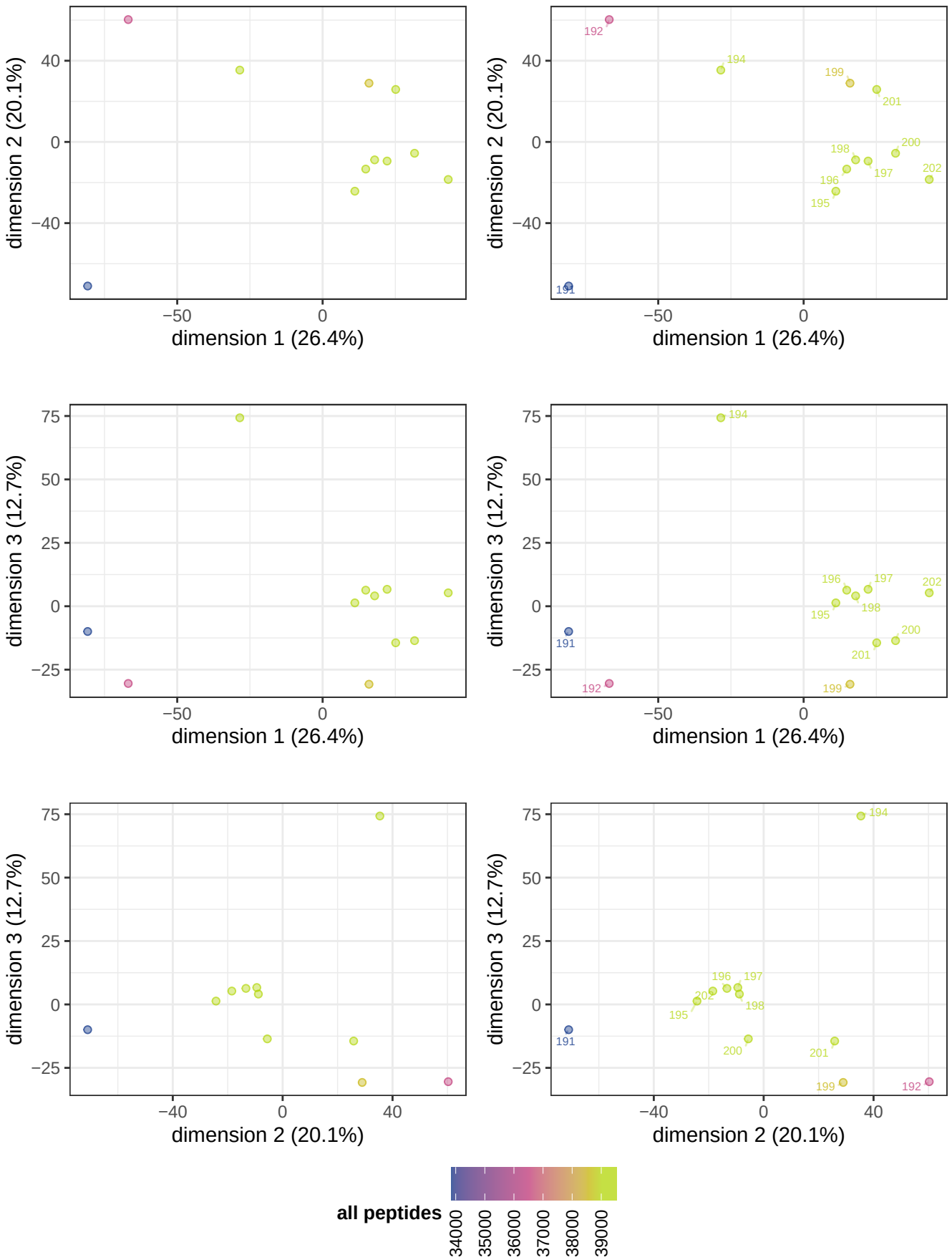
PCA only on samples not flagged as 'exclude', using 21164 peptides

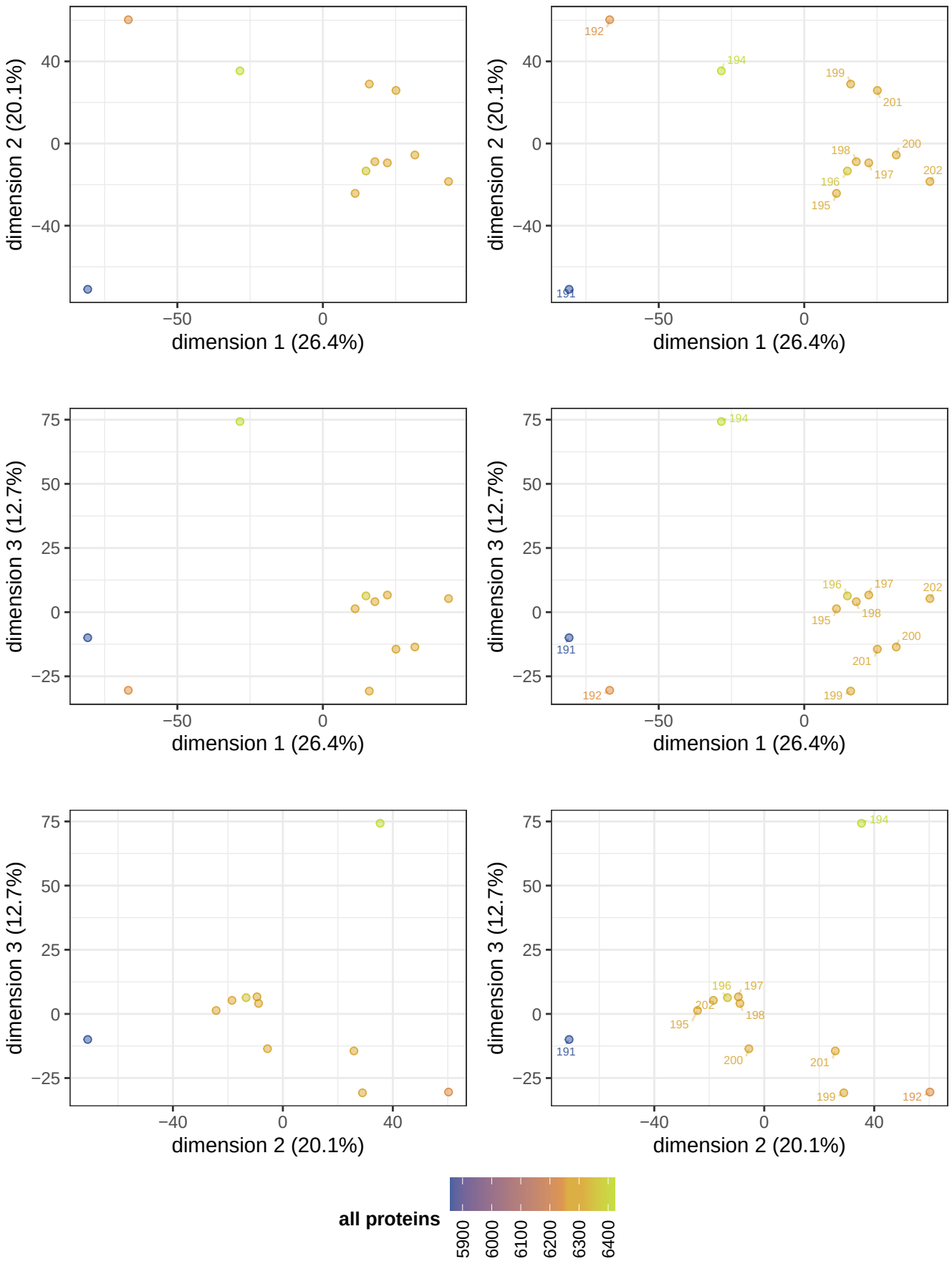



dimension 2 (20.1%)
dimension 1 (26.4%)
dimension 3 (12.7%)
order
2.5
5.0
7.5
10.0


dimension 2 (20.1%)
dimension 1 (26.4%)
dimension 3 (12.7%)
dimension 1 (26.4%)
dimension 3 (12.7%)
dimension 2 (20.1%)
detected peptides
30000
31000
32000
33000


dimension 2 (20.1%)
dimension 1 (26.4%)
dimension 3 (12.7%)
dimension 1 (26.4%)
dimension 3 (12.7%)
dimension 2 (20.1%)
191
192
194
195
196
197
198
199
200
201
202
detected proteins
5700
5800
5900
6000


dimension 2 (20.1%)
dimension 1 (26.4%)
dimension 3 (12.7%)
dimension 1 (26.4%)
dimension 3 (12.7%)
dimension 2 (20.1%)
all peptides
34000
35000
36000
37000
38000
39000


dimension 2 (20.1%)
dimension 1 (26.4%)
dimension 3 (12.7%)
dimension 1 (26.4%)
dimension 3 (12.7%)
dimension 2 (20.1%)
all proteins
5900
6000
6100
6200
6300
6400

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. In MS-DAP this is referred to as "by contrast" filtering.

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 subsequentially 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 erronously 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)

contrasts and foldchanges

Note that a MS-DAP contrast for "A vs B" returns foldchanges for B/A. For example, for the contrast "control vs disease" a positive log2 foldchange implies protein abundances are higher in the "disease" sample group.

2.1 one vs two

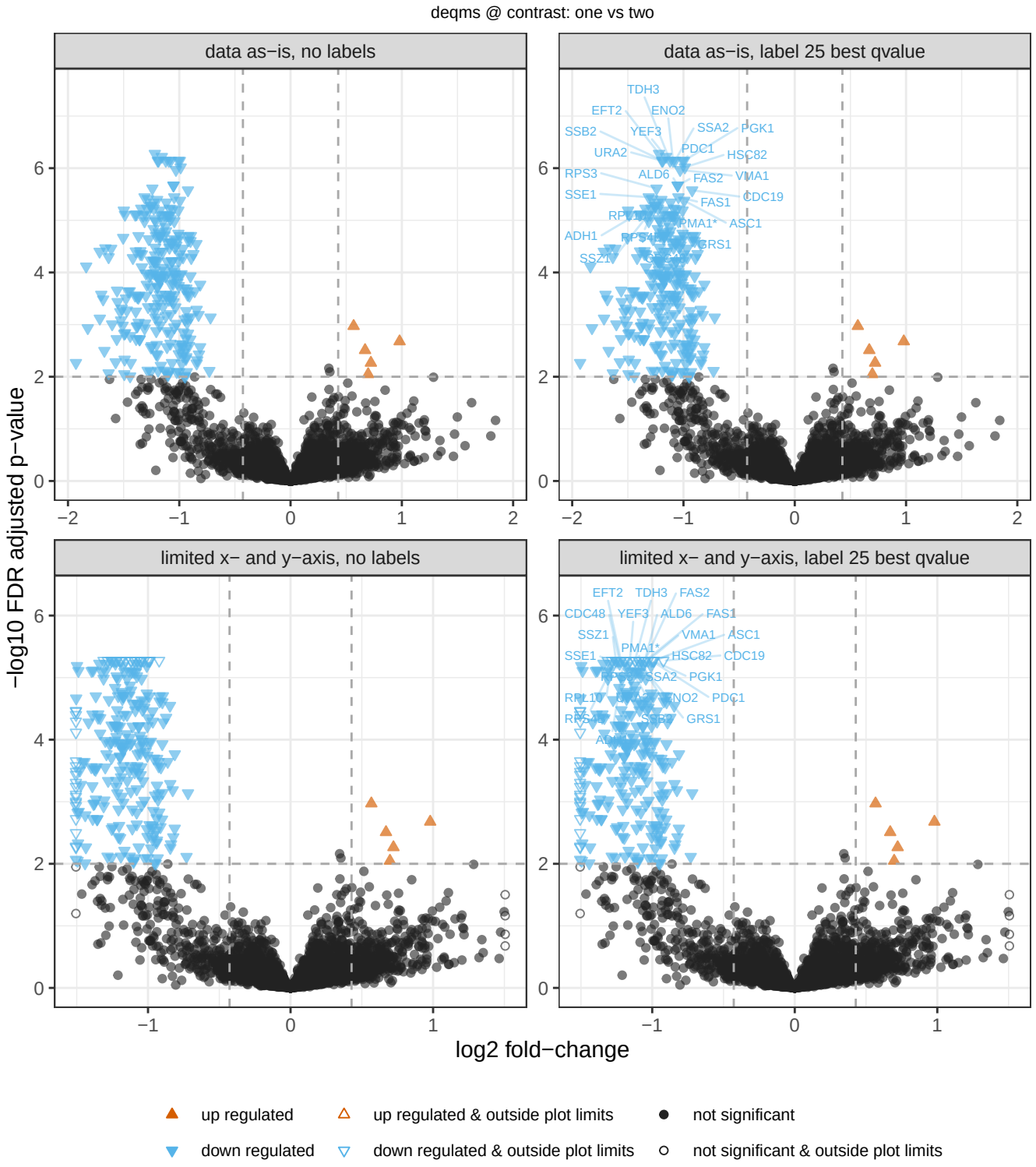
- **user setting:** using 'filter by contrast' peptide filtering approach
- 21798 peptides in 4537 proteins remain in the current contrast after peptide filters and are used for the statistical analysis in this section
- qvalue threshold: 0.01
- log2 foldchange threshold: 0.428

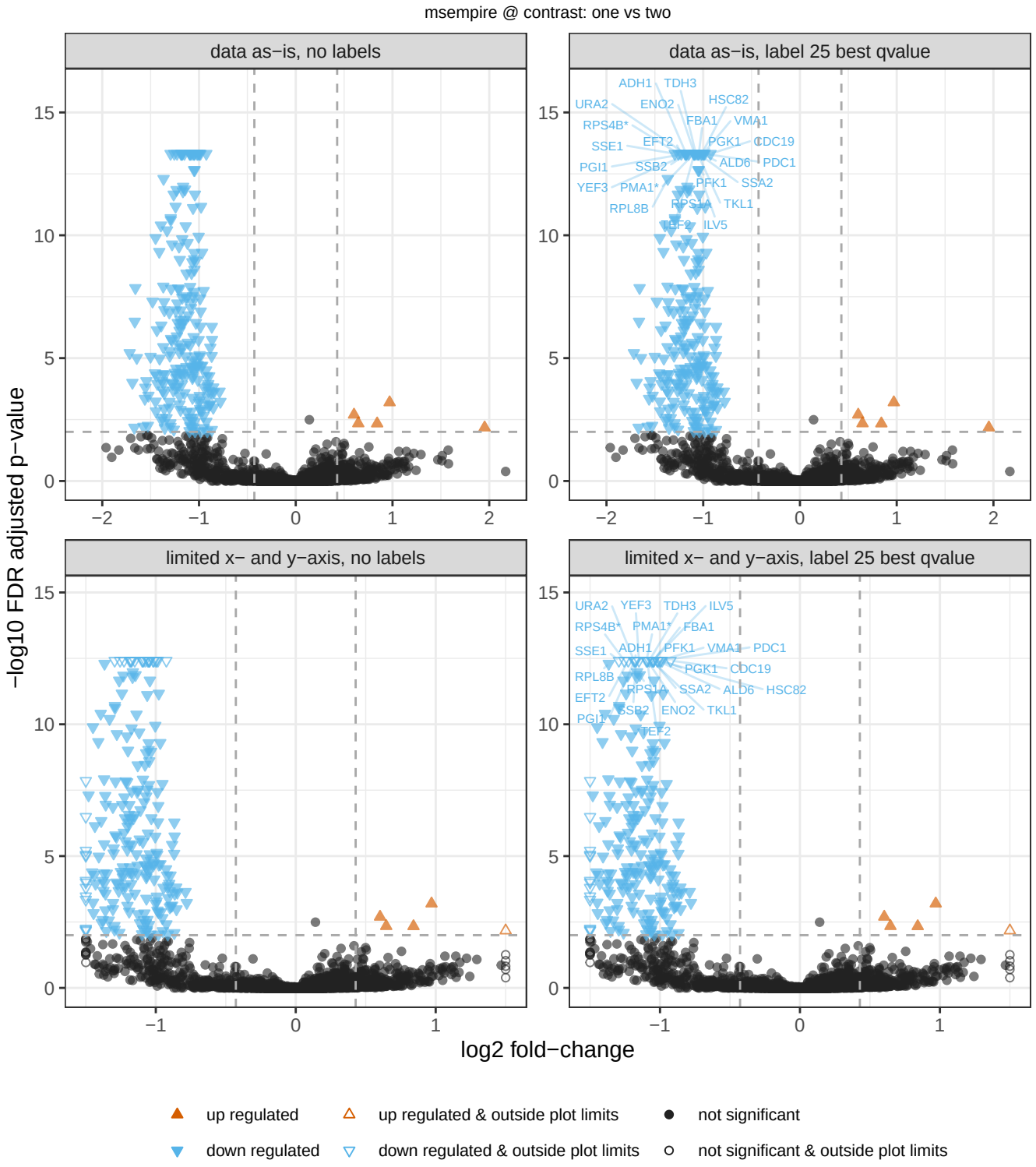
2.1.1 volcano

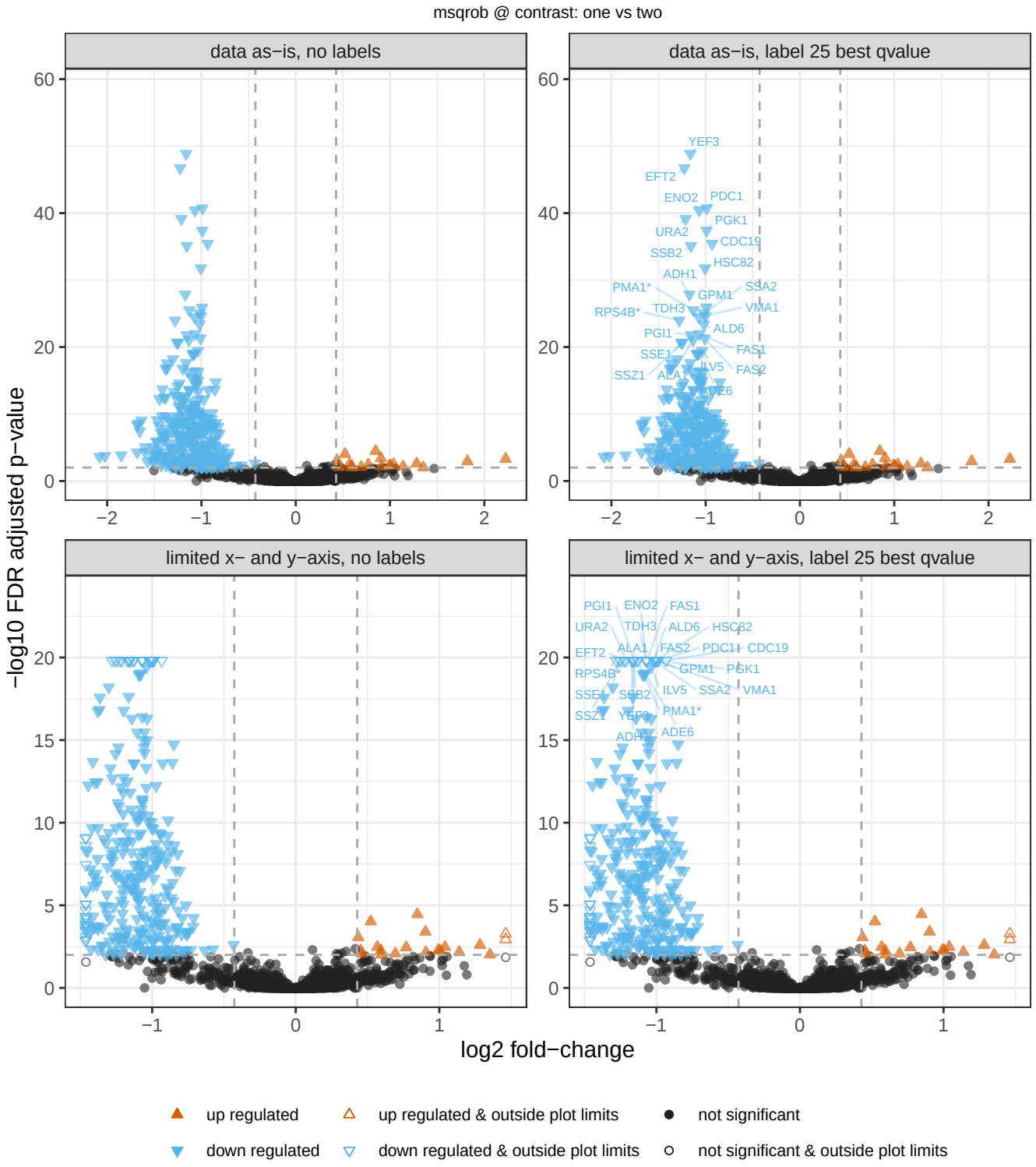
The plot title shows the statistical model and contrast (sample groups in the comparison). Left- and right-side figure panels on each row represent the same figure without and with labels for the 25 proteins with lowest p-value.

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.

Labels for proteins that are more than 12 characters long are truncated for visual clarity (indicated by trailing …). For protein identifiers that are ambiguous, e.g. a protein-group with assigned genes "gene1a;gene1b", only the first label/ID is shown for visual clarity (indicated by trailing *).


deqms @ contrast: one vs two
data as−is, no labels
data as−is, label 25 best qvalue
limited x− and y−axis, no labels
limited x− and y−axis, label 25 best qvalue
−log10 FDR adjusted p−value
log2 fold−change
TDH3
EFT2
ENO2
SSB2
YEF3
SSA2
PGK1
URA2
PDC1
HSC82
RPS3
ALD6
FAS2
VMA1
SSE1
FAS1
CDC19
PMA1*
ASC1
ADH1
GRS1
SSZ1
CDC48
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits

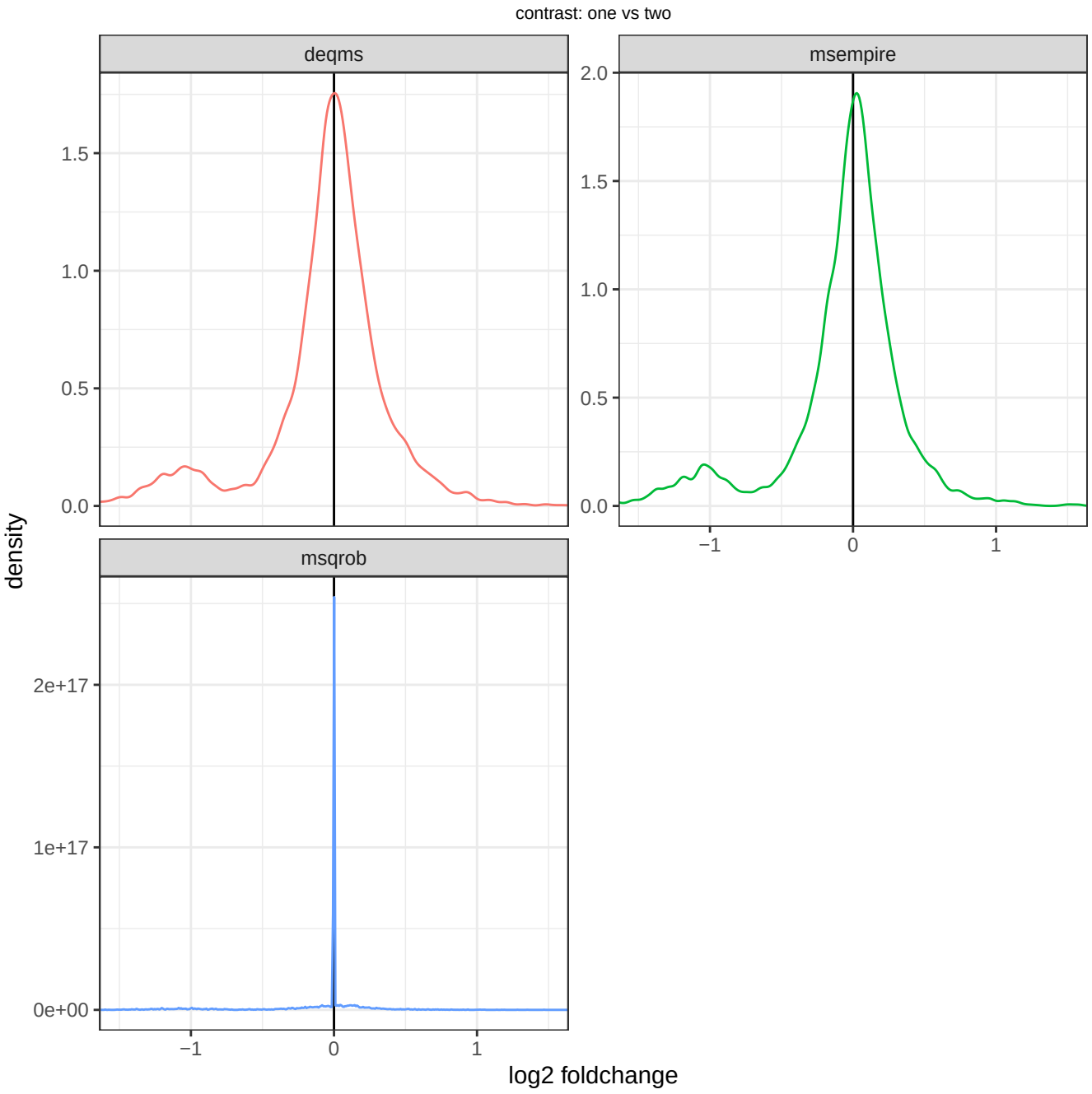

msempire @ contrast: one vs two
data as−is, no labels
data as−is, label 25 best qvalue
limited x− and y−axis, no labels
limited x− and y−axis, label 25 best qvalue
−log10 FDR adjusted p−value
log2 fold−change
ADH1
TDH3
HSC82
URA2
ENO2
FBA1
VMA1
RPS4B*
SSE1
EFT2
PGK1
CDC19
PGI1
SSB2
ALD6
PDC1
YEF3
PMA1*
PFK1
SSA2
RPL8B
RPS1A
TKL1
TEF2
ILV5
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits


msqrob @ contrast: one vs two
data as−is, no labels
data as−is, label 25 best qvalue
limited x− and y−axis, no labels
limited x− and y−axis, label 25 best qvalue
−log10 FDR adjusted p−value
log2 fold−change
YEF3
EFT2
ENO2
PDC1
PGK1
URA2
CDC19
SSB2
HSC82
ADH1
PMA1*
SSA2
GPM1
RPS4B*
TDH3
VMA1
PGI1
ALD6
SSE1
FAS1
SSZ1
ILV5
FAS2
ALA1
ADE6
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



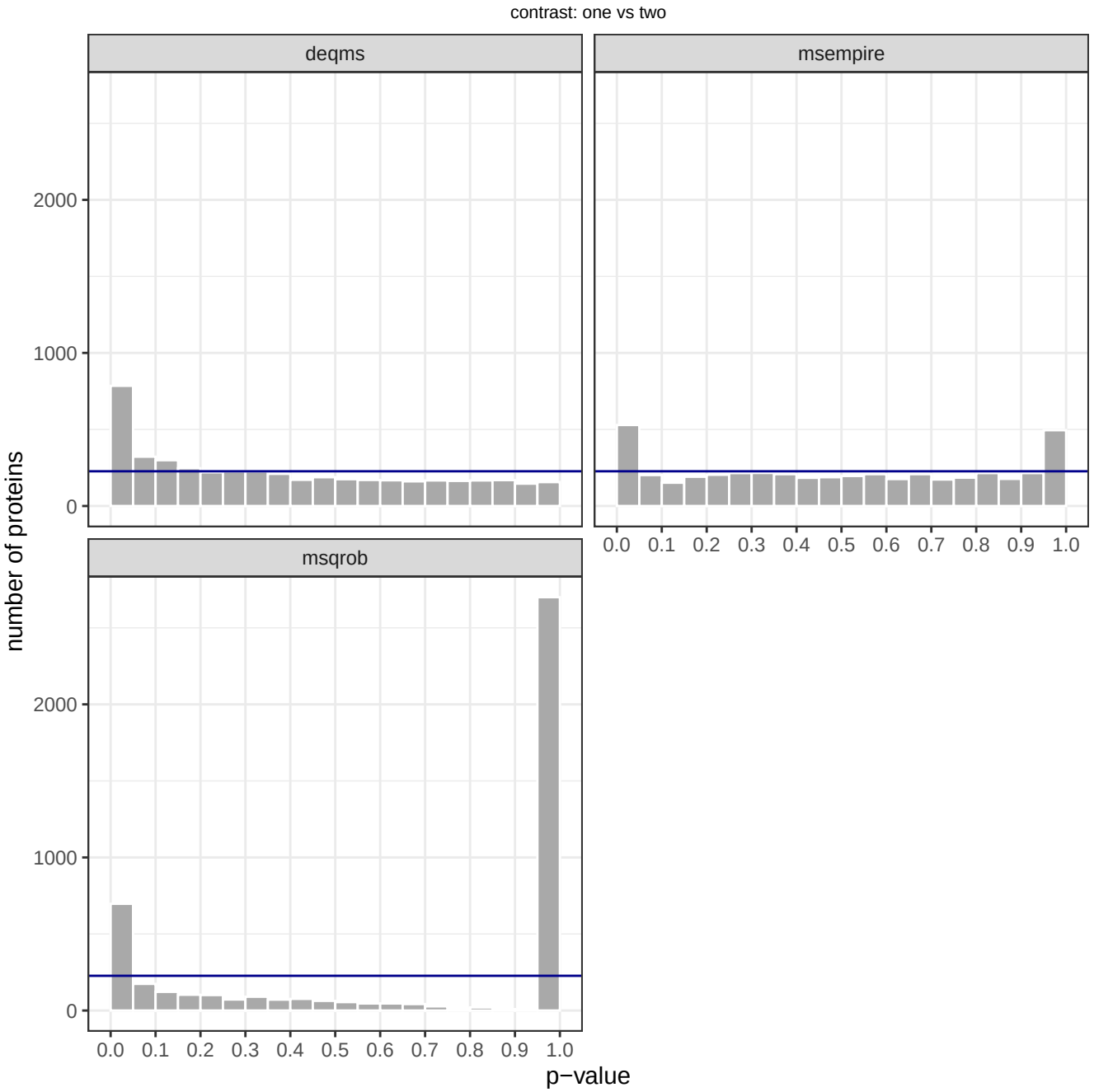
2.1.3 p-value distribution

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/

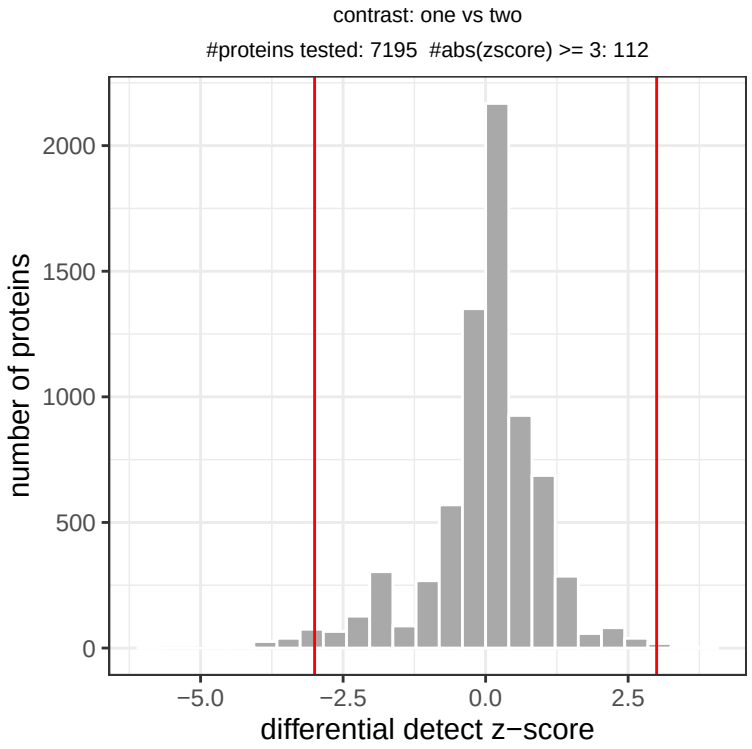
note; the MSqRob and MS-EmpiRe models often yield p-value distributions that show a large peak at p-value 1, these are typically proteins with estimated log foldchanges at/near zero where these models are very sure the null hypothesis cannot be rejected



2.1.4 differential detect

Some proteins may not have peptides with sufficient data points over samples to be used for differential expression analysis (depending on the user-defined filtering criteria in how many replicates peptides should be observed), but do show a strong difference in the number of detected peptides between sample groups. In some proteomics experimental designs, for example a wildtype-knockout APMS study, those are interesting proteins. The DEA based on peptide abundance values (volcano plots above) are the main result for differential testing in MS-DAP but as a situationally useful tool MS-DAP also includes a 'protein detection' z-score, based on the number of times a peptide for each protein was detected per sample group (/experimental condition), as an alternative means of differential testing.

Below figure shows the distribution of these scores with thresholds at 3 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

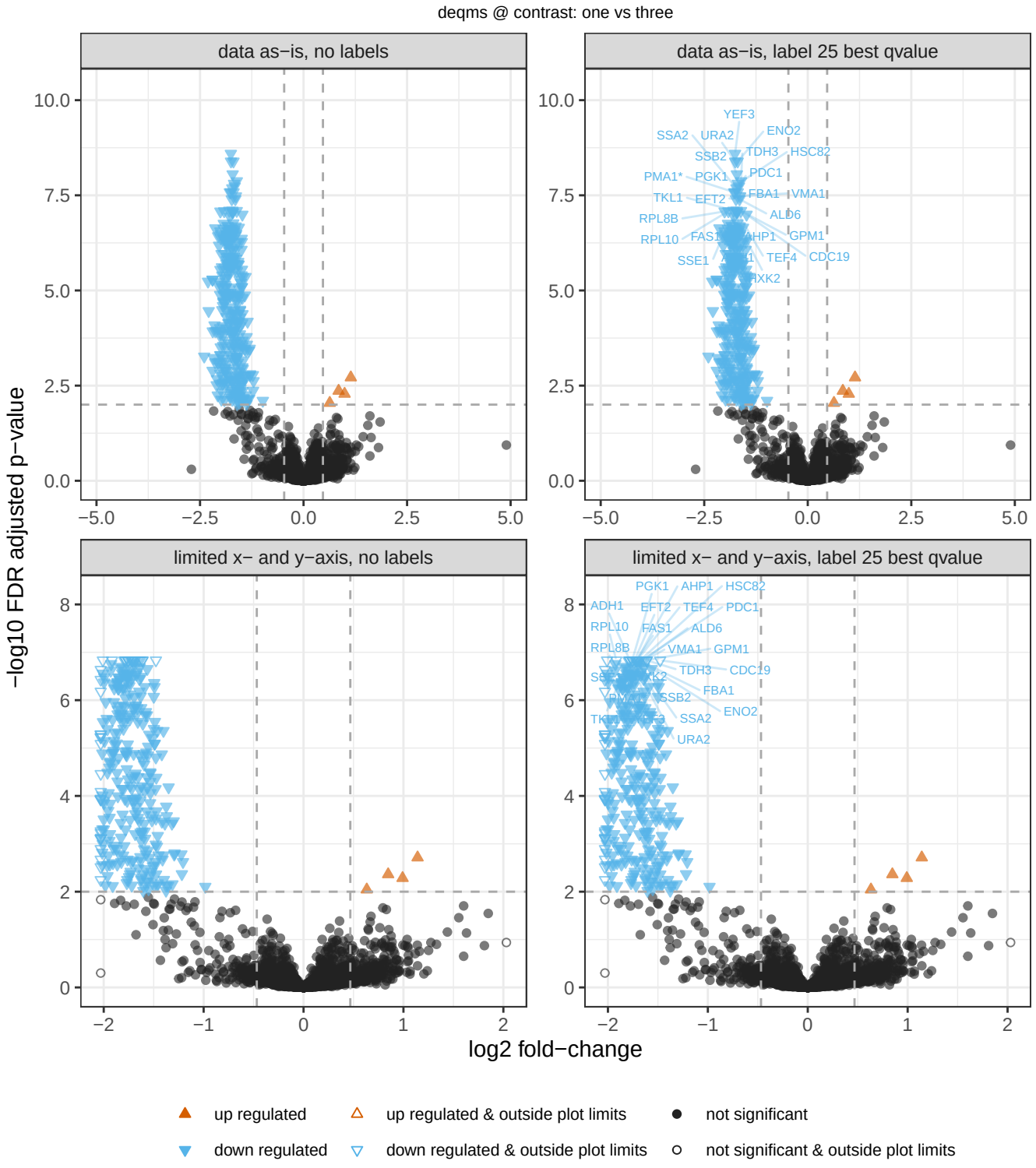
- **user setting:** using 'filter by contrast' peptide filtering approach
- 21849 peptides in 4527 proteins remain in the current contrast after peptide filters and are used for the statistical analysis in this section
- qvalue threshold: 0.01
- log2 foldchange threshold: 0.468

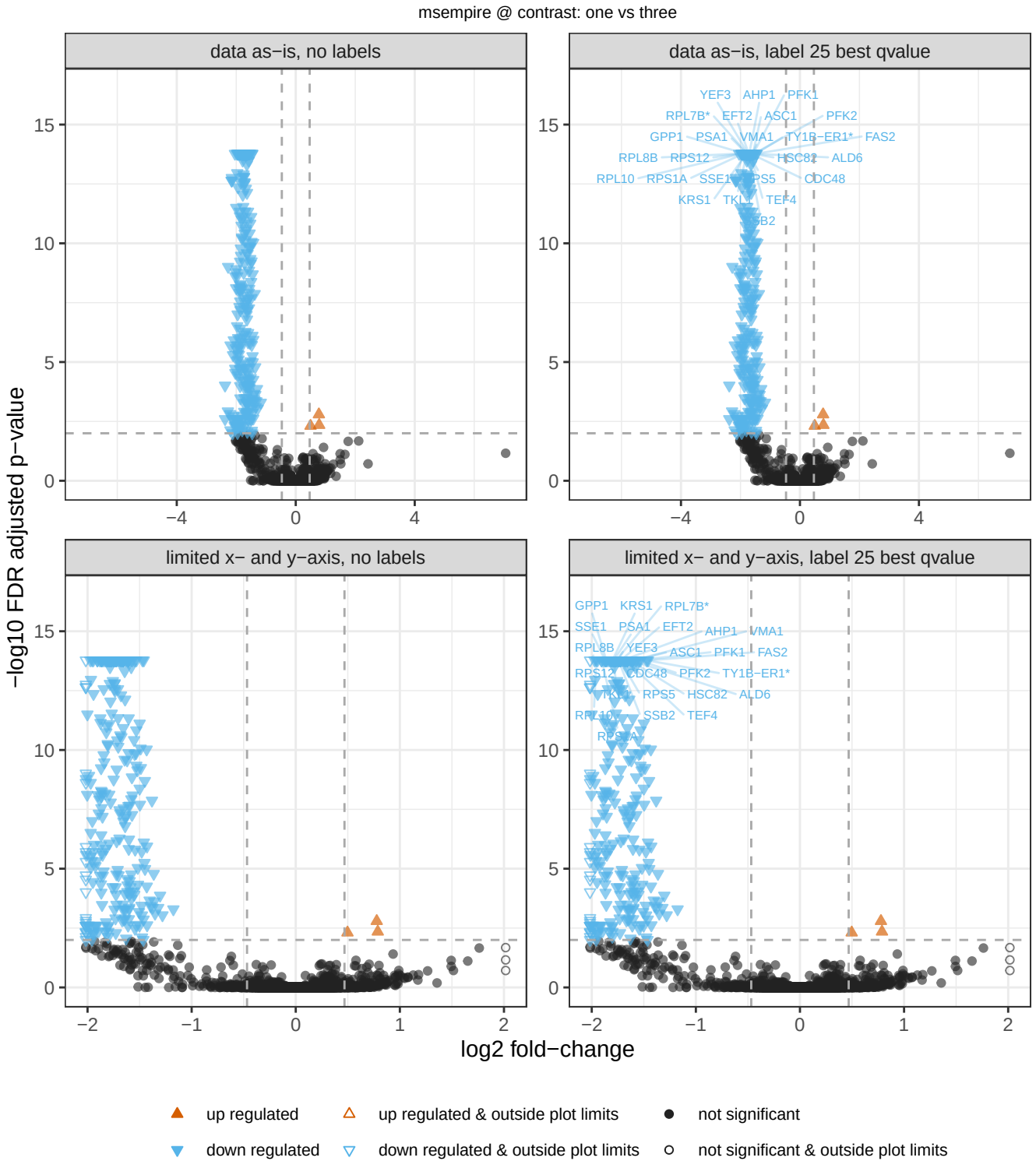
2.2.1 volcano

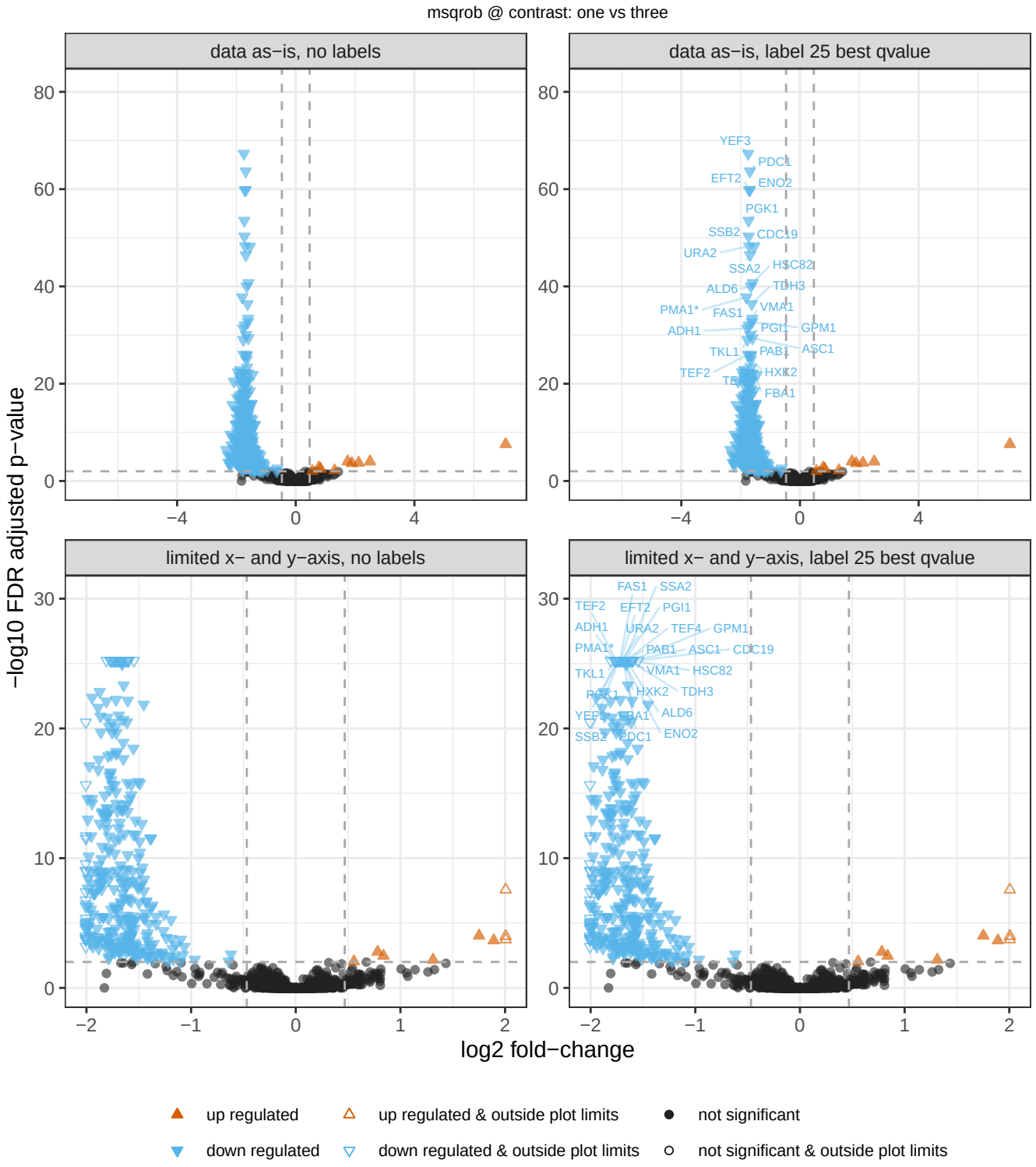
The plot title shows the statistical model and contrast (sample groups in the comparison). Left- and right-side figure panels on each row represent the same figure without and with labels for the 25 proteins with lowest p-value.

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.

Labels for proteins that are more than 12 characters long are truncated for visual clarity (indicated by trailing …). For protein identifiers that are ambiguous, e.g. a protein-group with assigned genes "gene1a;gene1b", only the first label/ID is shown for visual clarity (indicated by trailing *).


deqms @ contrast: one vs three
data as−is, no labels
data as−is, label 25 best qvalue
limited x− and y−axis, no labels
limited x− and y−axis, label 25 best qvalue
−log10 FDR adjusted p−value
log2 fold−change
YEF3
SSA2
URA2
ENO2
SSB2
TDH3
HSC82
PMA1*
PGK1
PDC1
TKL1
EFT2
FBA1
VMA1
RPL8B
ALD6
RPL10
FAS1
AHP1
GPM1
SSE1
TEF4
CDC19
HXK2
ADH1
PDC1
ALD6
VMA1
GPM1
TDH3
CDC19
FBA1
ENO2
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits

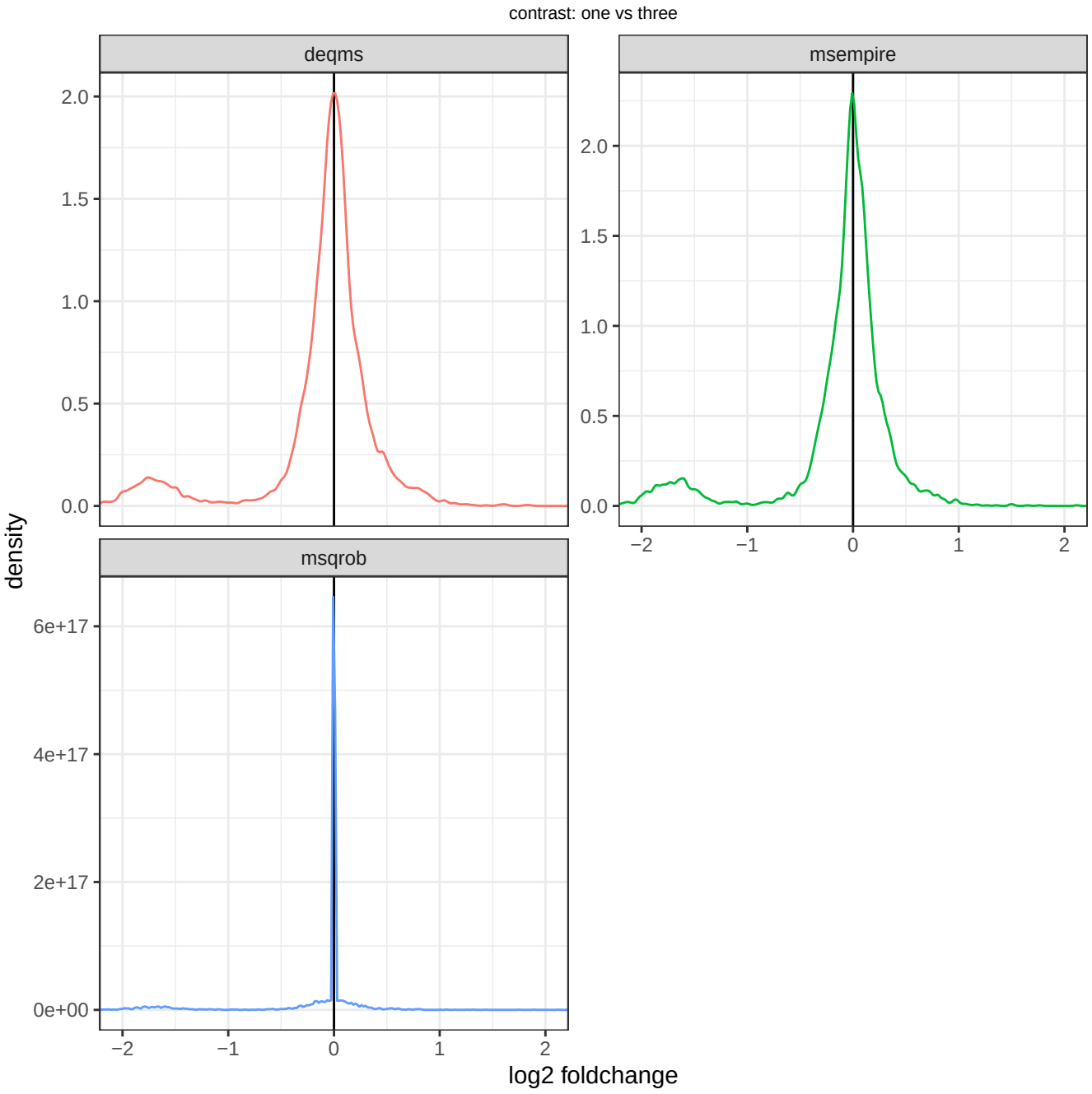

msempire @ contrast: one vs three
data as−is, no labels
data as−is, label 25 best qvalue
limited x− and y−axis, no labels
limited x− and y−axis, label 25 best qvalue
−log10 FDR adjusted p−value
log2 fold−change
YEF3 AHP1 PFK1
RPL7B* EFT2 ASC1 PFK2
GPP1 PSA1 VMA1 TY1B−ER1* FAS2
RPL8B RPS12 HSC82 ALD6
RPL10 RPS1A SSE1 RPS5 CDC48
KRS1 TKL1 TEF4
SSB2
GPP1 KRS1 RPL7B*
SSE1 PSA1 EFT2 AHP1 VMA1
RPL8B YEF3 ASC1 PFK1 FAS2
RPS12 CDC48 PFK2 TY1B−ER1*
TKL1 RPS5 HSC82 ALD6
RPL10 SSB2 TEF4
RPS1A
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits


msqrob @ contrast: one vs three
data as−is, no labels
data as−is, label 25 best qvalue
limited x− and y−axis, no labels
limited x− and y−axis, label 25 best qvalue
−log10 FDR adjusted p−value
log2 fold−change
YEF3
PDC1
EFT2
ENO2
PGK1
SSB2
CDC19
URA2
SSA2
HSC82
ALD6
TDH3
PMA1*
FAS1
VMA1
ADH1
PGI1
GPM1
TKL1
PAB1
ASC1
TEF2
HXK2
FBA1
TEF4
PGK1
ALD6
SSB2
ENO2
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



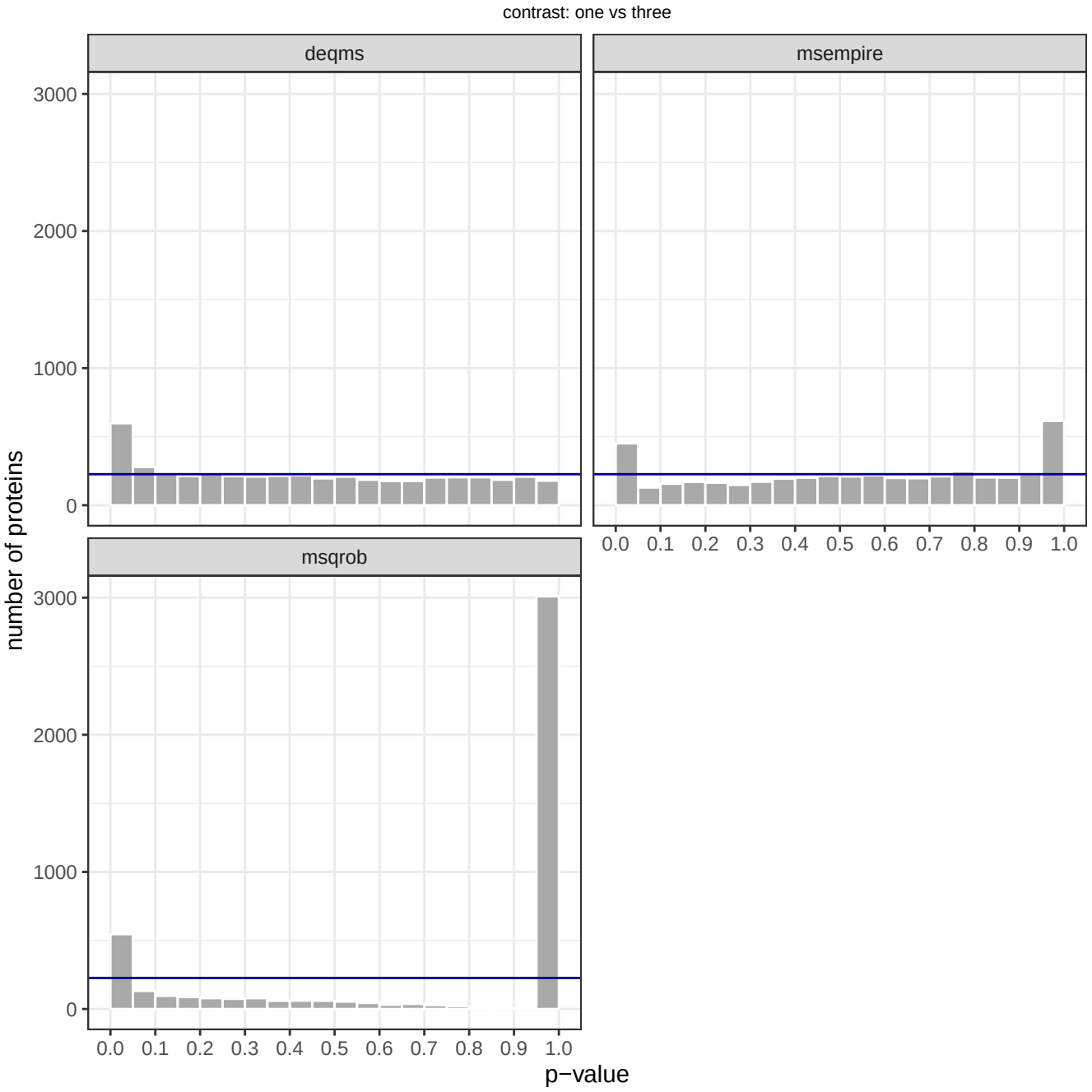
2.2.3 p-value distribution

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/

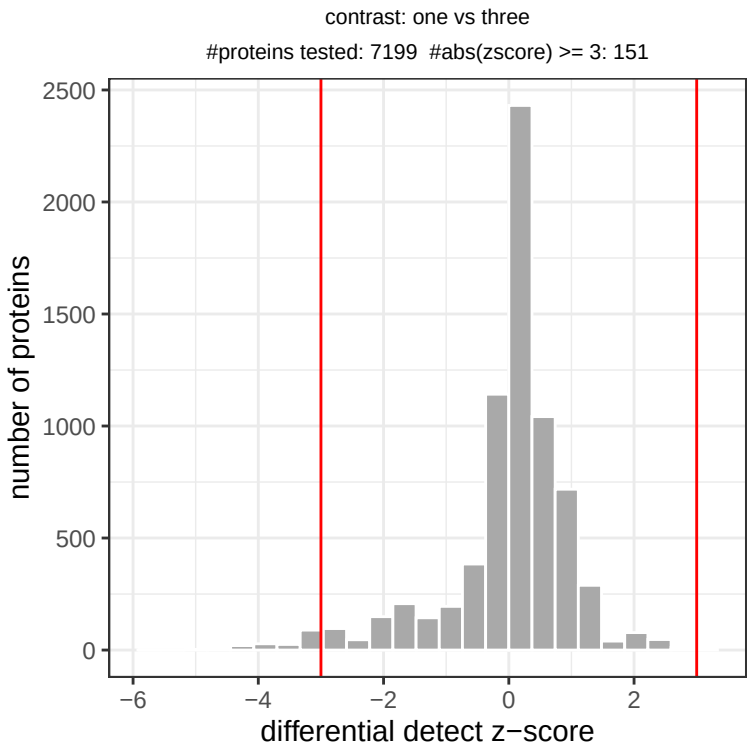
note; the MSqRob and MS-EmpiRe models often yield p-value distributions that show a large peak at p-value 1, these are typically proteins with estimated log foldchanges at/near zero where these models are very sure the null hypothesis cannot be rejected



2.2.4 differential detect

Some proteins may not have peptides with sufficient data points over samples to be used for differential expression analysis (depending on the user-defined filtering criteria in how many replicates peptides should be observed), but do show a strong difference in the number of detected peptides between sample groups. In some proteomics experimental designs, for example a wildtype-knockout APMS study, those are interesting proteins. The DEA based on peptide abundance values (volcano plots above) are the main result for differential testing in MS-DAP but as a situationally useful tool MS-DAP also includes a 'protein detection' z-score, based on the number of times a peptide for each protein was detected per sample group (/experimental condition), as an alternative means of differential testing.

Below figure shows the distribution of these scores with thresholds at 3 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

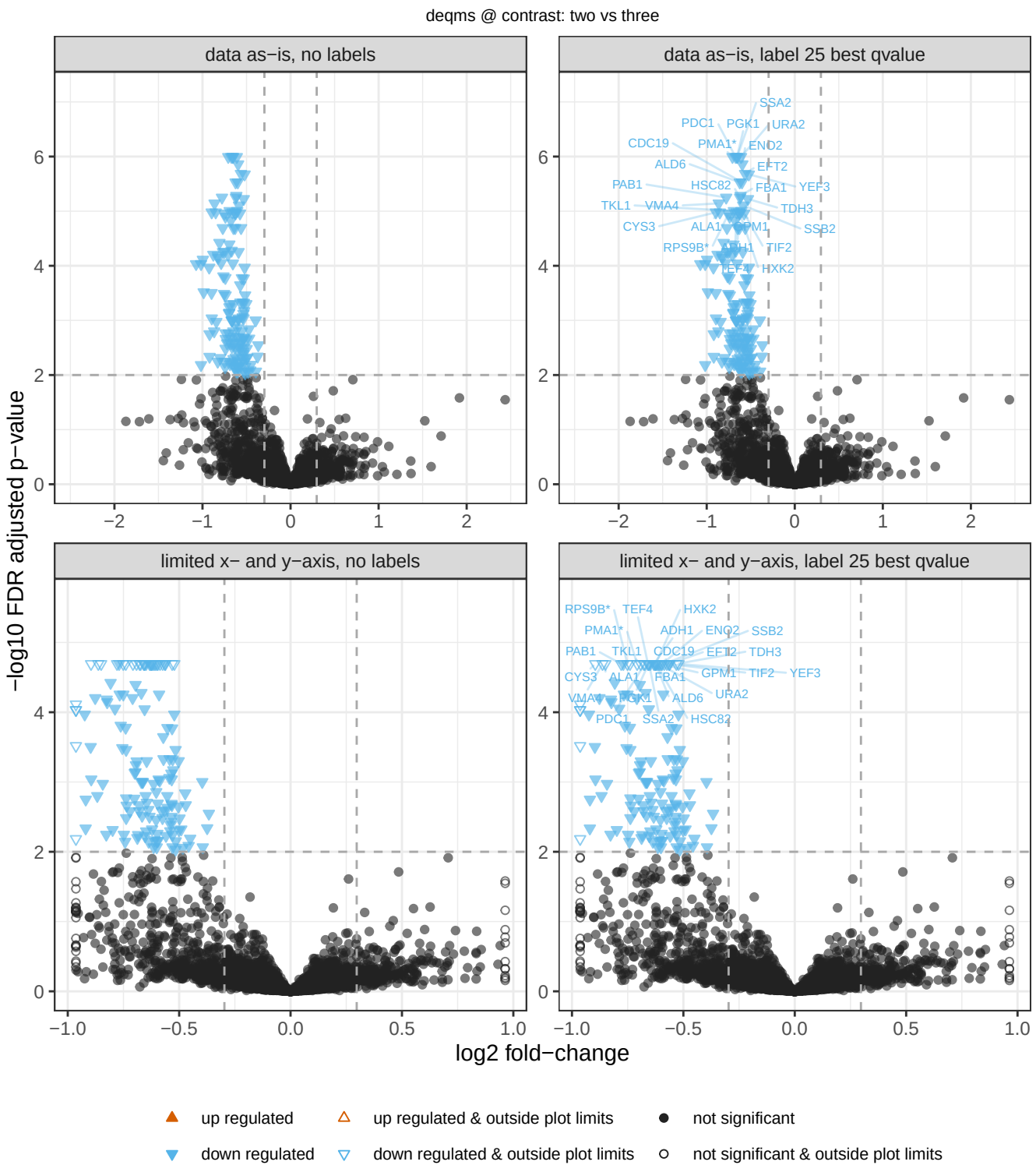
- **user setting:** using 'filter by contrast' peptide filtering approach
- 31125 peptides in 5420 proteins remain in the current contrast after peptide filters and are used for the statistical analysis in this section
- qvalue threshold: 0.01
- log2 foldchange threshold: 0.297

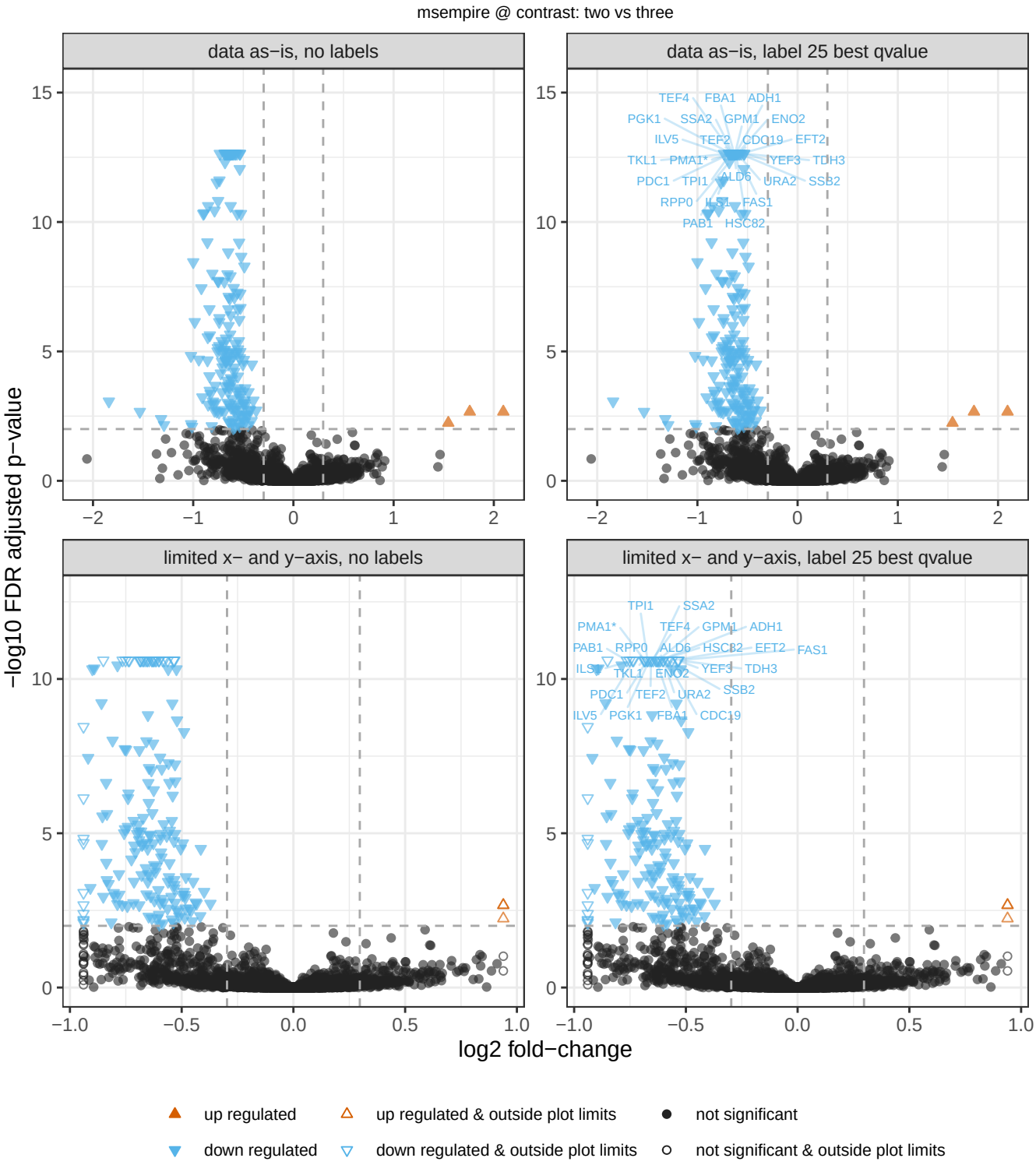
2.3.1 volcano

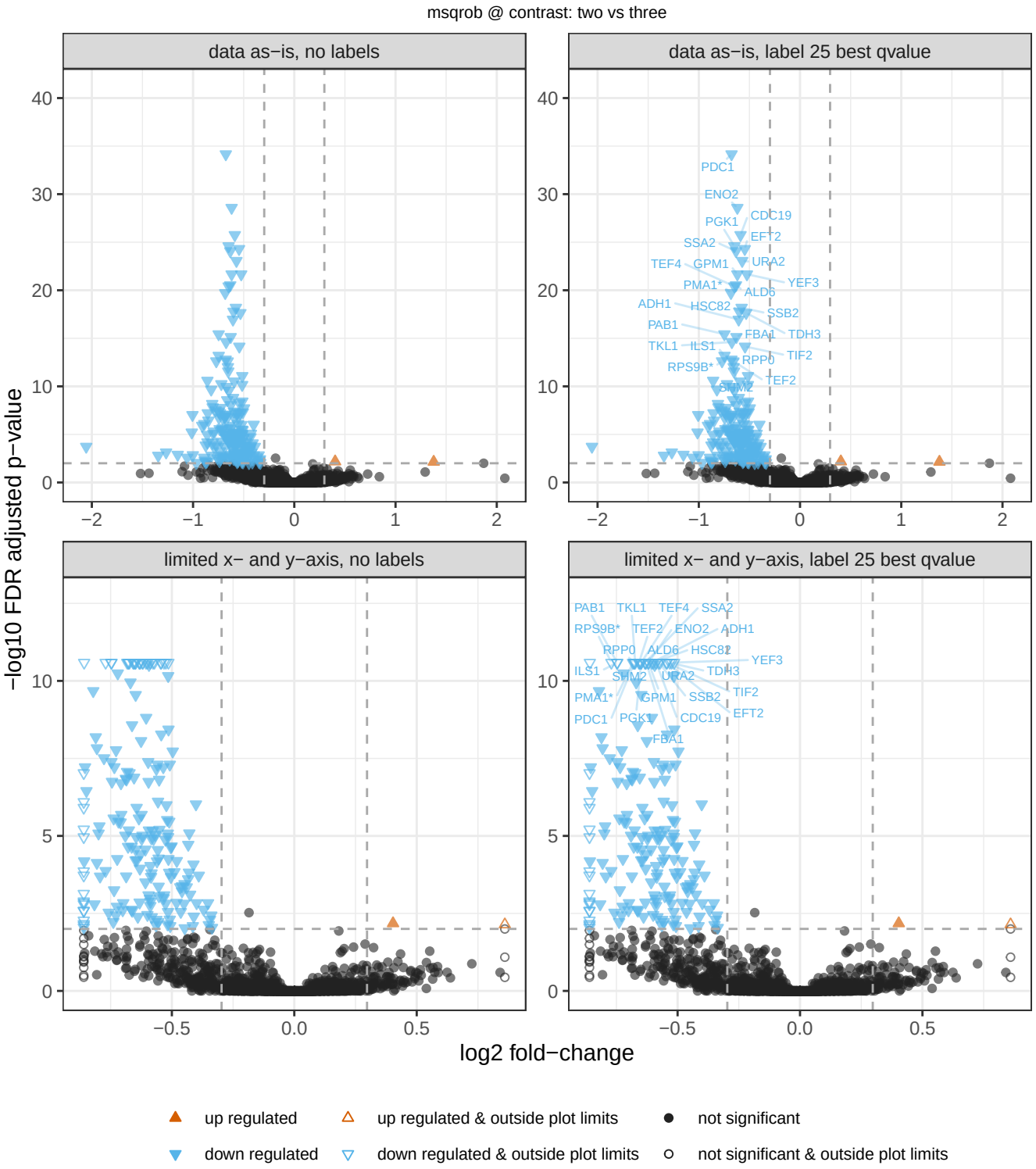
The plot title shows the statistical model and contrast (sample groups in the comparison). Left- and right-side figure panels on each row represent the same figure without and with labels for the 25 proteins with lowest p-value.

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.

Labels for proteins that are more than 12 characters long are truncated for visual clarity (indicated by trailing …). For protein identifiers that are ambiguous, e.g. a protein-group with assigned genes "gene1a;gene1b", only the first label/ID is shown for visual clarity (indicated by trailing *).


deqms @ contrast: two vs three
data as−is, no labels
data as−is, label 25 best qvalue
limited x− and y−axis, no labels
limited x− and y−axis, label 25 best qvalue
−log10 FDR adjusted p−value
log2 fold−change
SSA2
PDC1
PGK1
URA2
CDC19
PMA1*
ENO2
ALD6
EFT2
PAB1
HSC82
FBA1
YEF3
TKL1
VMA4
TDH3
CYS3
ALA1
GPM1
SSB2
RPS9B*
ADH1
TIF2
TEF4
HXK2
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits

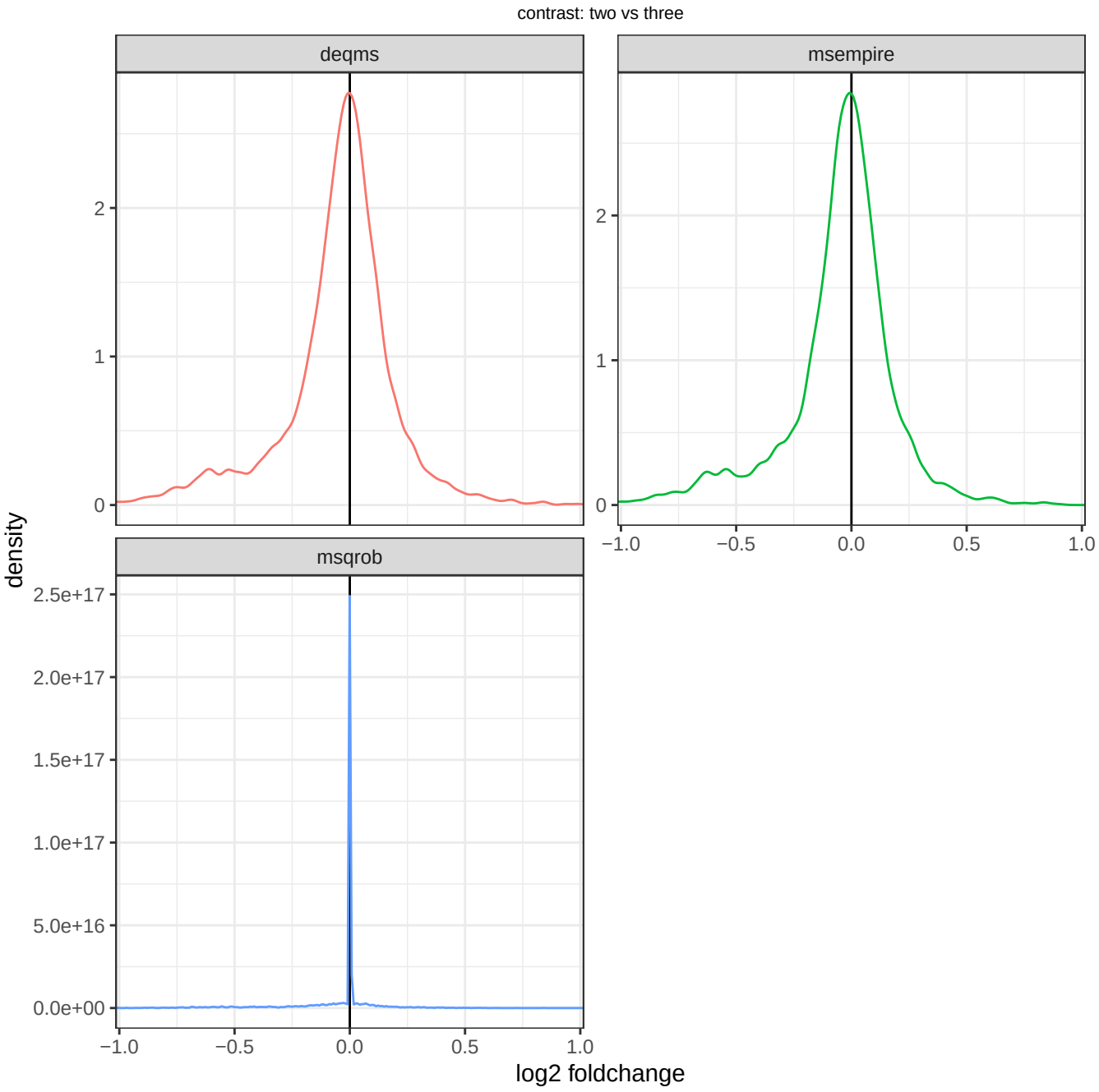

msempire @ contrast: two vs three
data as−is, no labels
data as−is, label 25 best qvalue
limited x− and y−axis, no labels
limited x− and y−axis, label 25 best qvalue
−log10 FDR adjusted p−value
log2 fold−change
TEF4 FBA1 ADH1
PGK1 SSA2 GPM1 ENO2
ILV5 TEF2 CDC19 EFT2
TKL1 PMA1* YEF3 TDH3
PDC1 TPI1 ALD6 URA2 SSB2
RPP0 HSC82 FAS1
PAB1
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits


msqrob @ contrast: two vs three
data as−is, no labels
data as−is, label 25 best qvalue
limited x− and y−axis, no labels
limited x− and y−axis, label 25 best qvalue
−log10 FDR adjusted p−value
log2 fold−change
PDC1
ENO2
PGK1
CDC19
EFT2
SSA2
TEF4
GPM1
URA2
PMA1*
YEF3
ALD6
ADH1
HSC82
SSB2
PAB1
FBA1
TDH3
TKL1
ILS1
TIF2
RPS9B*
RPP0
TEF2
SHM2
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



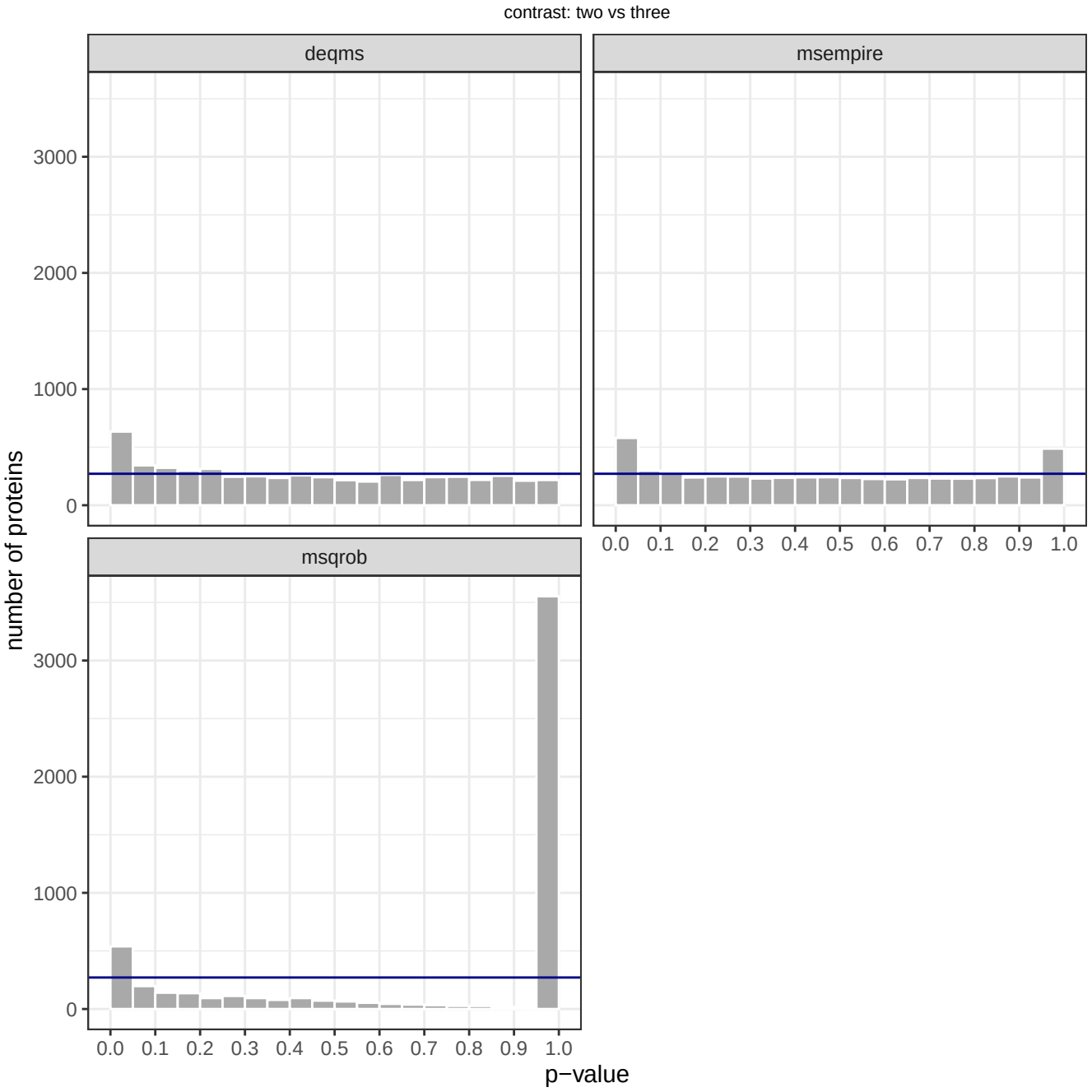
2.3.3 p-value distribution

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/

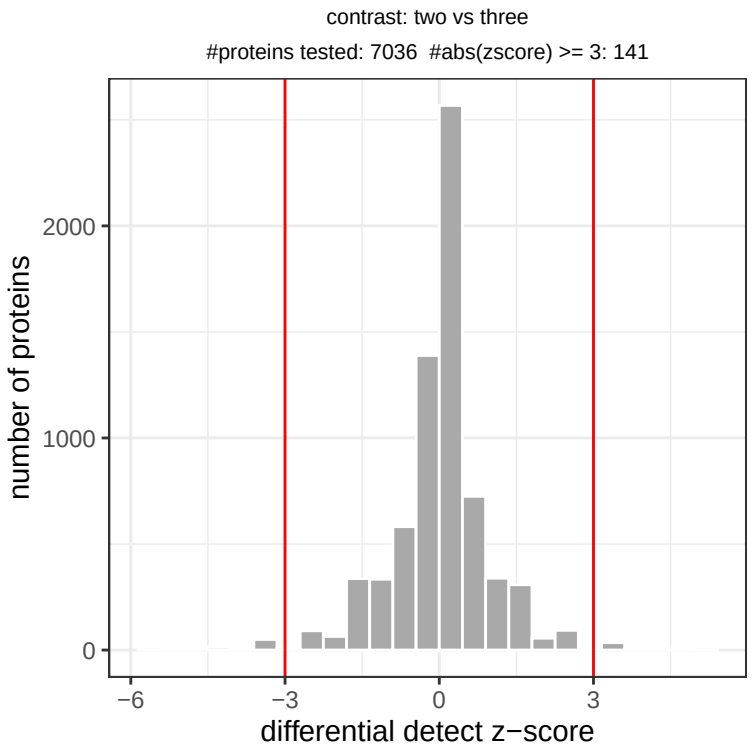
note; the MSqRob and MS-EmpiRe models often yield p-value distributions that show a large peak at p-value 1, these are typically proteins with estimated log foldchanges at/near zero where these models are very sure the null hypothesis cannot be rejected



2.3.4 differential detect

Some proteins may not have peptides with sufficient data points over samples to be used for differential expression analysis (depending on the user-defined filtering criteria in how many replicates peptides should be observed), but do show a strong difference in the number of detected peptides between sample groups. In some proteomics experimental designs, for example a wildtype-knockout APMS study, those are interesting proteins. The DEA based on peptide abundance values (volcano plots above) are the main result for differential testing in MS-DAP but as a situationally useful tool MS-DAP also includes a 'protein detection' z-score, based on the number of times a peptide for each protein was detected per sample group (/experimental condition), as an alternative means of differential testing.

Below figure shows the distribution of these scores with thresholds at 3 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 two	deqms	4537	267	eft2, yef3, eno2, ura2, tdh3, ssb2, ssa2, pgk1, pdc1, hsc82
one vs two	msempire	4537	225	adh1, tdh3, cdc19, pgk1, eno2, tef2, pma1;pma2, pdc1, ura2, ssa2
one vs two	msqrob	4537	337	yef3, eft2, pdc1, eno2, ura2, pgk1, cdc19, ssb2, hsc82, adh1
one vs three	deqms	4527	314	yef3, eno2, ura2, ssb2, tdh3, hsc82, pdc1, ssa2, pma1;pma2, pgk1
one vs three	msempire	4527	274	adh1, tdh3, cdc19, pgk1, eno2, tpi1, gpm1, tef2, pab1, rpl25
one vs three	msqrob	4527	339	yef3, pdc1, eft2, eno2, pgk1, ssb2, ura2, cdc19, ssa2, hsc82
two vs three	deqms	5420	143	ura2, pgk1, ssa2, pma1;pma2, pdc1, eno2, eft2, yef3, ald6, cdc19
two vs three	msempire	5420	177	adh1, tdh3, cdc19, pgk1, eno2, gpm1, tef2, pma1;pma2, ilv5, pdc1
two vs three	msqrob	5420	192	pdc1, eno2, cdc19, pgk1, eft2, ssa2, ura2, gpm1, yef3, ald6

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	#proteins tested	#abs(zscore) >= 3	top10
one vs two	7195	112	cpa2, hem2, aah1, rvb2, gcn20, sec24, rlp7, nug1, wrs1, lap3
one vs three	7199	151	gcn1, rnr1, cpa2, glt1, paa1, ncp1, spt16, wtm1, rpg1, spf1
two vs three	7036	141	gcn1, rasal2, ncl1, dhx35, rpa190, gmnn, tjap1, cbf5, sirt2, sqor

4 log

[info] reading MaxQuant 'txt' folder

[info] Parsing MaxQuant proteinGroups.txt, 5/58442 peptides are not a razor peptide in any protein-group and therefore removed

[info] 57302 target precursors, 54504 (plain)sequences, 7320 proteins

[info] 7641/7641 protein accessions and 7320/7320 protein groups were mapped to provided fasta file(s)

[warning] We advice against numeric shortname values. Preferably, use an alphanumeric string to prevent confusion between sample index and sample names in visualizations. Affected shortname values; 191, 192, 194, 195, 196, 197, 198, 199, 200, 201, 202

[info] contrast: one vs two

[info] contrast: one vs three

[info] contrast: two vs three

[info] using 23 threads for multiprocessing

[progress] caching filter data took 4 seconds

[progress] peptide to protein rollup with MaxLFQ (implementation: iq) took 1 seconds

[progress] peptide to protein rollup with MaxLFQ (implementation: iq) took 1 seconds

[progress] peptide to protein rollup with MaxLFQ (implementation: iq) took 1 seconds

[progress] peptide to protein rollup with MaxLFQ (implementation: iq) took 1 seconds

[progress] peptide to protein rollup with MaxLFQ (implementation: iq) took 1 seconds

[info] filter dataset with settings: min_quant = 3; fraction_quant = 0.75; norm_algorithm = 'vsn&modebetween_protein'; rollup_algorithm = 'maxlfq'
21164/57302 peptides were retained after filtering over all groups
21798/57302 peptides were retained after filtering within contrast: one vs two
21849/57302 peptides were retained after filtering within contrast: one vs three
31125/57302 peptides were retained after filtering within contrast: two vs three
41172/57302 peptides were retained after filtering within each group independently ("by group")

[progress] peptide filtering and normalization took 12 seconds

[info] differential expression analysis for contrast: one vs two

[info] using data from peptide filter: filter by contrast

[progress] peptide to protein rollup with MaxLFQ (implementation: iq) took 1 seconds

[info] log2 foldchange threshold estimated by bootstrap analysis: 0.428

[progress] DEqMS took 1 seconds

[progress] MS-EmpiRe took 48 seconds

[info] msqrob linear regression formulas (these are prioritized. eg; if a model fit fails due to lack of data, the next formula is used); expression ~ (1 | condition) + (1 | sample_id) + (1 | peptide_id) , expression ~ (1 | condition)

[progress] msqrob took 1.1 minutes

[info] differential expression analysis for contrast: one vs three

[info] using data from peptide filter: filter by contrast

[progress] peptide to protein rollup with MaxLFQ (implementation: iq) took 1 seconds

[info] log2 foldchange threshold estimated by bootstrap analysis: 0.468

[progress] DEqMS took 1 seconds

[progress] MS-EmpiRe took 49 seconds

[info] msqrob linear regression formulas (these are prioritized. eg; if a model fit fails due to lack of data, the next formula is used); expression ~ (1 | condition) + (1 | sample_id) + (1 | peptide_id) , expression ~ (1 | condition)

[progress] msqrob took 58 seconds

[info] differential expression analysis for contrast: two vs three

[info] using data from peptide filter: filter by contrast

[progress] peptide to protein rollup with MaxLFQ (implementation: iq) took 1 seconds

[info] log2 foldchange threshold estimated by bootstrap analysis: 0.297

[progress] DEqMS took 1 seconds

[progress] MS-EmpiRe took 1.4 minutes

[info] msqrob linear regression formulas (these are prioritized. eg; if a model fit fails due to lack of data, the next formula is used); expression ~ (1 | condition) + (1 | sample_id) + (1 | peptide_id) , expression ~ (1 | condition)

[progress] msqrob took 1.2 minutes

[info] differential detection analysis: min_samples_observed=2

[progress] peptide to protein rollup with MaxLFQ (implementation: iq) took 1 seconds

[progress] peptide to protein rollup with MaxLFQ (implementation: iq) took 1 seconds

[progress] peptide to protein rollup with MaxLFQ (implementation: iq) took 1 seconds

[progress] peptide to protein rollup with MaxLFQ (implementation: iq) took 1 seconds

[progress] peptide to protein rollup with MaxLFQ (implementation: iq) took 1 seconds

[progress] peptide to protein rollup with MaxLFQ (implementation: iq) took 1 seconds

[progress] creating PDF report…

[progress] peptide to protein rollup with MaxLFQ (implementation: iq) took 1 seconds

[progress] report: constructing plots specific for each contrast

[progress] report: rendering report (this may take a while depending on dataset size)

[progress] RT plots: preparing data took 5 seconds

[progress] RT plots: creating plots took 1 seconds

[info] No data available for CoV leave-one-out computation in sample group ‘one’, skipping plots

[progress] leave-one-out CoV plot computations took 1 seconds

[progress] peptide to protein rollup with MaxLFQ (implementation: iq) took 1 seconds

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 = "E:/DATA/PXD007683/txt_mbr")
dataset = import_fasta(dataset,
  files = c(
    "E:/DATA/PXD007683/fasta/UP000002311_559292.fasta",
    "E:/DATA/PXD007683/fasta/UP000005640_9606.fasta"
  )
)
dataset = import_sample_metadata(
  dataset,
  "E:/DATA/PXD007683/oconnel_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 = 0,
  filter_min_quant = 3,
  filter_fraction_detect = 0,
  filter_fraction_quant = 0.75,
  filter_by_contrast = TRUE,
  filter_topn_peptides = 0,
  filter_min_peptide_per_prot = 1,
  norm_algorithm = c(
    "vsn",
    "modebetween_protein"
  ),
  dea_algorithm = c(
    "deqms",
    "msempire", "msqrob"
  ),
  dea_qvalue_threshold = 0.01,
  dea_log2foldchange_threshold = NA,
  diffdetect_min_samples_observed = 2,
  output_qc_report = TRUE,
  output_abundance_tables = TRUE,
  output_dir = "C:/temp",
  output_within_timestamped_subdirectory = TRUE,
  dump_all_data = 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 4.2.1 (2022-06-23 ucrt)
os	Windows 10 x64 (build 19043)
system	x86_64, mingw32
ui	RStudio
language	(EN)
collate	English_United States.utf8
ctype	English_United States.utf8
tz	Europe/Berlin
date	2022-10-30
rstudio	2022.07.0+548 Spotted Wakerobin (desktop)
pandoc	2.18 @ C:/Program Files/RStudio/bin/quarto/bin/tools/ (via rmarkdown)

System

package	loadedversion	source
dplyr	1.0.9	CRAN (R 4.2.1)
ggplot2	3.3.6	CRAN (R 4.2.1)
msdap	1.0.3	
rlang	1.0.3	CRAN (R 4.2.1)
testthat	3.1.4	CRAN (R 4.2.1)
tibble	3.1.7	CRAN (R 4.2.1)
tidyr	1.2.0	CRAN (R 4.2.1)

Attached packages

package	loadedversion	source
abind	1.4-5	CRAN (R 4.2.0)
affy	1.74.0	Bioconductor
affyio	1.66.0	Bioconductor
aod	1.3.2	CRAN (R 4.2.1)
askpass	1.1	CRAN (R 4.2.1)
assertthat	0.2.1	CRAN (R 4.2.1)
backports	1.4.1	CRAN (R 4.2.0)
Biobase	2.56.0	Bioconductor
BiocGenerics	0.42.0	Bioconductor
BiocManager	1.30.18	CRAN (R 4.2.1)
BiocParallel	1.30.3	Bioconductor
bit	4.0.4	CRAN (R 4.2.1)
bit64	4.0.5	CRAN (R 4.2.1)
bitops	1.0-7	CRAN (R 4.2.0)
blob	1.2.3	CRAN (R 4.2.1)
boot	1.3-28	CRAN (R 4.2.1)
brio	1.1.3	CRAN (R 4.2.1)
broom	1.0.0	CRAN (R 4.2.1)
cachem	1.0.6	CRAN (R 4.2.1)
callr	3.7.0	CRAN (R 4.2.1)
car	3.1-0	CRAN (R 4.2.1)
carData	3.0-5	CRAN (R 4.2.1)
caTools	1.18.2	CRAN (R 4.2.1)
cli	3.3.0	CRAN (R 4.2.1)
clue	0.3-61	CRAN (R 4.2.1)
cluster	2.1.3	CRAN (R 4.2.1)
codetools	0.2-18	CRAN (R 4.2.1)
colorspace	2.0-3	CRAN (R 4.2.1)
cowplot	1.1.1	CRAN (R 4.2.1)
crayon	1.5.1	CRAN (R 4.2.1)
data.table	1.14.2	CRAN (R 4.2.1)
DBI	1.1.3	CRAN (R 4.2.1)
DEqMS	1.14.0	bioc_xgit (@a30da3599ce70afb20527ef3c001d4c15a51a662)
desc	1.4.1	CRAN (R 4.2.1)
devtools	2.4.3	CRAN (R 4.2.1)
diann	1.0.1	Github (vdemichev/diann-rpackage@af538f6e2cd5ab715e1381632e17cb8f234ebf53)
digest	0.6.29	CRAN (R 4.2.1)
doParallel	1.0.17	CRAN (R 4.2.1)
doRNG	1.8.2	CRAN (R 4.2.1)
ellipsis	0.3.2	CRAN (R 4.2.1)
evaluate	0.15	CRAN (R 4.2.1)
fansi	1.0.3	CRAN (R 4.2.1)
farver	2.1.1	CRAN (R 4.2.1)
fastmap	1.1.0	CRAN (R 4.2.1)
forcats	0.5.1	CRAN (R 4.2.1)
foreach	1.5.2	CRAN (R 4.2.1)
formatR	1.12	CRAN (R 4.2.1)
fs	1.5.2	CRAN (R 4.2.1)
generics	0.1.3	CRAN (R 4.2.1)
ggpubr	0.4.0	CRAN (R 4.2.1)

package	loadedversion	source
ggrepel	0.9.1	CRAN (R 4.2.1)
ggsignif	0.6.3	CRAN (R 4.2.1)
glue	1.6.2	CRAN (R 4.2.1)
gplots	3.1.3	CRAN (R 4.2.1)
gridExtra	2.3	CRAN (R 4.2.1)
gtable	0.3.0	CRAN (R 4.2.1)
gtools	3.9.2.2	CRAN (R 4.2.1)
hms	1.1.1	CRAN (R 4.2.1)
htmltools	0.5.2	CRAN (R 4.2.1)
impute	1.70.0	Bioconductor
iq	1.9.6	CRAN (R 4.2.1)
IRanges	2.30.0	Bioconductor
iterators	1.0.14	CRAN (R 4.2.1)
itertools	0.1-3	CRAN (R 4.2.1)
KernSmooth	2.23-20	CRAN (R 4.2.1)
knitr	1.39	CRAN (R 4.2.1)
labeling	0.4.2	CRAN (R 4.2.0)
lattice	0.20-45	CRAN (R 4.2.1)
lifecycle	1.0.1	CRAN (R 4.2.1)
limma	3.52.2	Bioconductor
lme4	1.1-30	CRAN (R 4.2.1)
magrittr	2.0.3	CRAN (R 4.2.1)
MALDIquant	1.21	CRAN (R 4.2.1)
MASS	7.3-57	CRAN (R 4.2.1)
Matrix	1.5-1	CRAN (R 4.2.1)
matrixStats	0.62.0	CRAN (R 4.2.1)
memoise	2.0.1	CRAN (R 4.2.1)
mgcv	1.8-40	CRAN (R 4.2.1)
minqa	1.2.4	CRAN (R 4.2.1)
missForest	1.5	CRAN (R 4.2.1)
MsCoreUtils	1.8.0	Bioconductor
msEmpiRe	0.1.0	Github (zimmerlab/MS-EmpiRe@8a85757c8d604014130ee9d379aa8cfeb05e3855)
MSnbase	2.22.0	Bioconductor
munsell	0.5.0	CRAN (R 4.2.1)
mzID	1.34.0	Bioconductor
mzR	2.30.0	Bioconductor
ncdf4	1.19	CRAN (R 4.2.0)
nlme	3.1-157	CRAN (R 4.2.1)
nloptr	2.0.3	CRAN (R 4.2.1)
openssl	2.0.2	CRAN (R 4.2.1)
openxlsx	4.2.5	CRAN (R 4.2.1)
patchwork	1.1.1	CRAN (R 4.2.1)
pbkrtest	0.5.1	CRAN (R 4.2.1)
pcaMethods	1.88.0	Bioconductor
pdftools	3.3.0	CRAN (R 4.2.1)
Peptides	2.4.4	CRAN (R 4.2.1)
pillar	1.7.0	CRAN (R 4.2.1)
pkgbuild	1.3.1	CRAN (R 4.2.1)
pkgconfig	2.0.3	CRAN (R 4.2.1)
pkgload	1.3.0	CRAN (R 4.2.1)

package	loadedversion	source
plyr	1.8.7	CRAN (R 4.2.1)
preprocessCore	1.58.0	Bioconductor
prettyunits	1.1.1	CRAN (R 4.2.1)
pROC	1.18.0	CRAN (R 4.2.1)
processx	3.7.0	CRAN (R 4.2.1)
progress	1.2.2	CRAN (R 4.2.1)
ProtGenerics	1.28.0	Bioconductor
ps	1.7.1	CRAN (R 4.2.1)
purrr	0.3.4	CRAN (R 4.2.1)
qpdf	1.2.0	CRAN (R 4.2.1)
R.cache	0.15.0	CRAN (R 4.2.1)
R.methodsS3	1.8.2	CRAN (R 4.2.0)
R.oo	1.25.0	CRAN (R 4.2.0)
R.utils	2.12.0	CRAN (R 4.2.1)
R6	2.5.1	CRAN (R 4.2.1)
randomForest	4.7-1.1	CRAN (R 4.2.1)
rbibutils	2.2.8	CRAN (R 4.2.1)
RColorBrewer	1.1-3	CRAN (R 4.2.0)
Rcpp	1.0.9	CRAN (R 4.2.1)
RcppEigen	0.3.3.9.2	CRAN (R 4.2.1)
Rdpack	2.3.1	CRAN (R 4.2.1)
readr	2.1.2	CRAN (R 4.2.1)
remotes	2.4.2	CRAN (R 4.2.1)
reshape2	1.4.4	CRAN (R 4.2.1)
RhpcBLASctl	0.21-247.1	CRAN (R 4.2.0)
rmarkdown	2.14	CRAN (R 4.2.1)
rngtools	1.5.2	CRAN (R 4.2.1)
rprojroot	2.0.3	CRAN (R 4.2.1)
RSQLite	2.2.14	CRAN (R 4.2.1)
rstatix	0.7.0	CRAN (R 4.2.1)
rstudioapi	0.13	CRAN (R 4.2.1)
S4Vectors	0.34.0	Bioconductor
scales	1.2.0	CRAN (R 4.2.1)
sessioninfo	1.2.2	CRAN (R 4.2.1)
stringi	1.7.6	CRAN (R 4.2.0)
stringr	1.4.0	CRAN (R 4.2.1)
styler	1.7.0	CRAN (R 4.2.1)
tidyselect	1.1.2	CRAN (R 4.2.1)
tinytex	0.40	CRAN (R 4.2.1)
tzdb	0.3.0	CRAN (R 4.2.1)
usethis	2.1.6	CRAN (R 4.2.1)
utf8	1.2.2	CRAN (R 4.2.1)
variancePartition	1.26.0	bioc_xgit (@b1731297b2f7335ef9155c6ac7df0e246787d9d6)
vctrs	0.4.1	CRAN (R 4.2.1)
viridis	0.6.2	CRAN (R 4.2.1)
viridisLite	0.4.0	CRAN (R 4.2.1)
vroom	1.5.7	CRAN (R 4.2.1)
vsn	3.64.0	Bioconductor
withr	2.5.0	CRAN (R 4.2.1)
xfun	0.31	CRAN (R 4.2.1)

package	loadedversion	source
XML	3.99-0.10	CRAN (R 4.2.0)
xtable	1.8-4	CRAN (R 4.2.1)
yaml	2.3.5	CRAN (R 4.2.0)
zip	2.2.0	CRAN (R 4.2.1)
zlibbioc	1.42.0	Bioconductor

Packages that are not attached