MS-DAP: Mass Spectrometry Downstream Analysis Pipeline

version: 1.2.1 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:

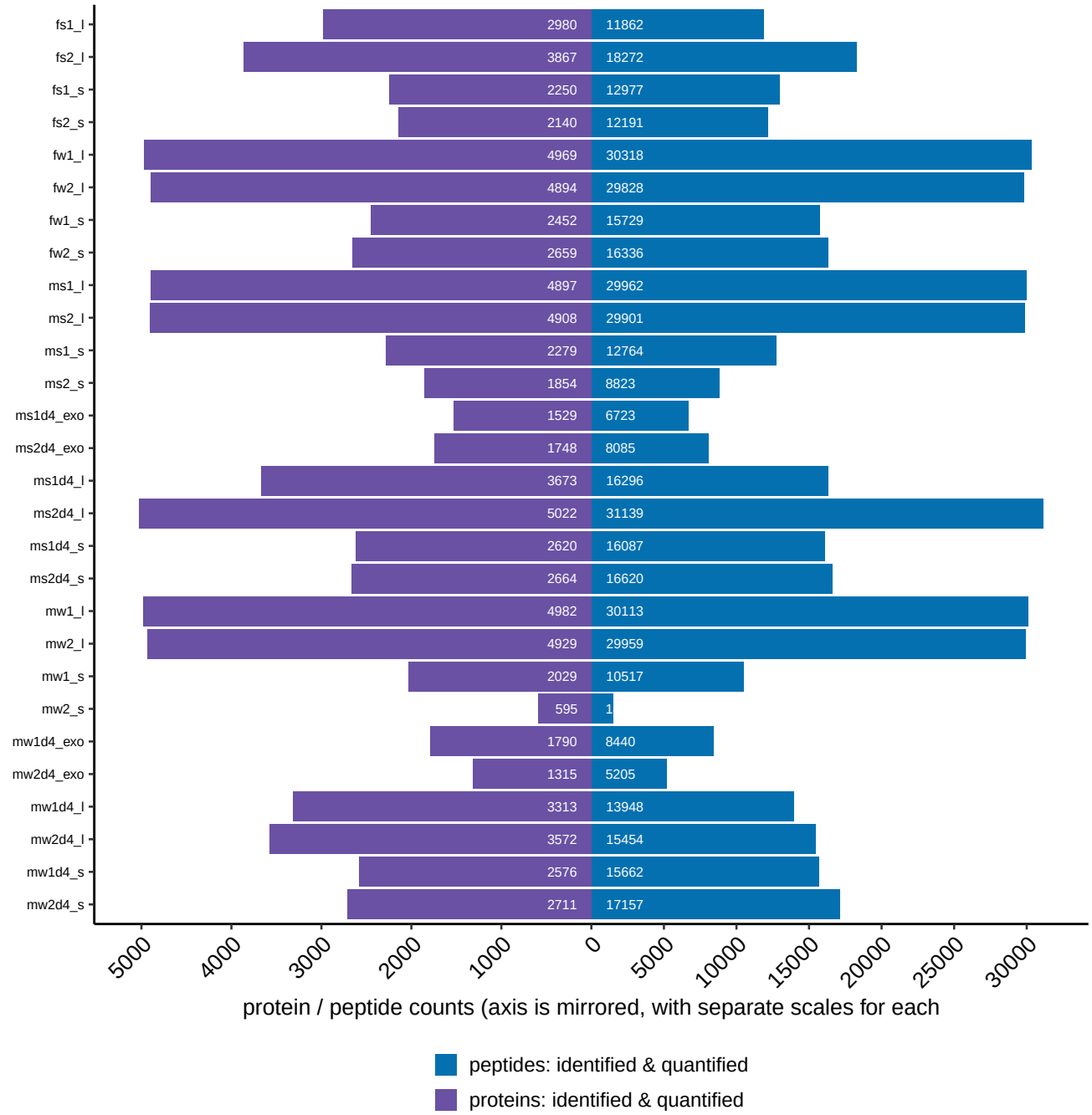
- number of peptides/proteins detected in each sample
- dataset completeness
- local effects in HPLC peptide retention time per sample
- reproducibility of peptide quantification among replicates
- PCA of all samples to visualize clustering

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'.

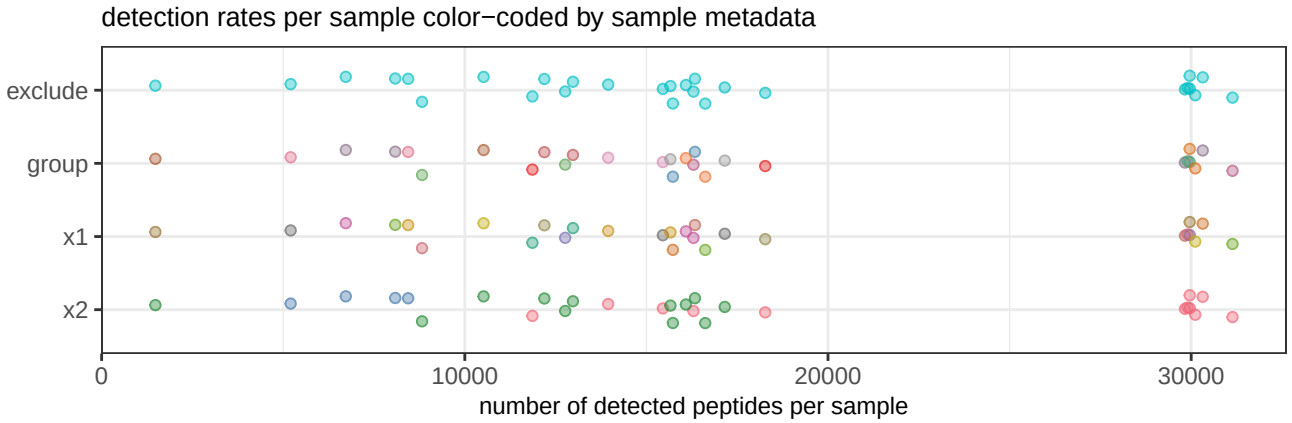

fs1_l
fs2_l
fs1_s
fs2_s
fw1_l
fw2_l
fw1_s
fw2_s
ms1_l
ms2_l
ms1_s
ms2_s
ms1d4_exo
ms2d4_exo
ms1d4_l
ms2d4_l
ms1d4_s
ms2d4_s
mw1_l
mw2_l
mw1_s
mw2_s
mw1d4_exo
mw2d4_exo
mw1d4_l
mw2d4_l
mw1d4_s
mw2d4_s
2980
3867
2250
2140
4969
4894
2452
2659
4897
4908
2279
1854
1529
1748
3673
5022
2620
2664
4982
4929
2029
595
1790
1315
3313
3572
2576
2711
11862
18272
12977
12191
30318
29828
15729
16336
29962
29901
12764
8823
6723
8085
16296
31139
16087
16620
30113
29959
10517
1
8440
5205
13948
15454
15662
17157
5000
4000
3000
2000
1000
0
5000
10000
15000
20000
25000
30000
protein / peptide counts (axis is mirrored, with separate scales for each
peptides: identified & quantified
proteins: identified & quantified

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'.



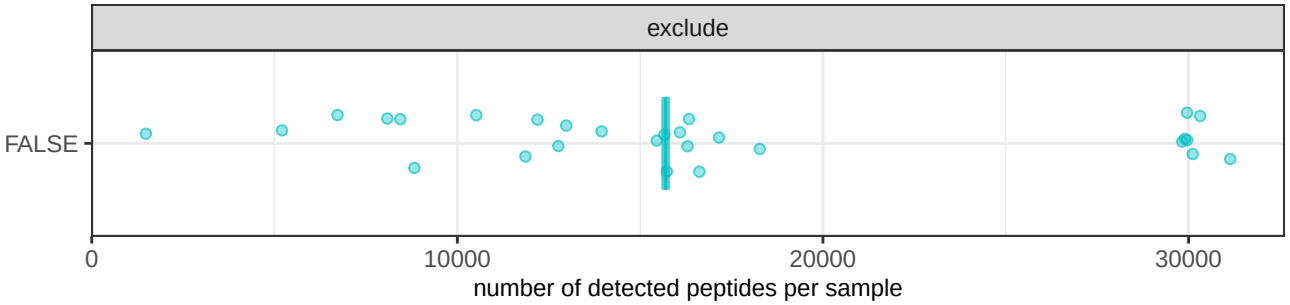
color-coding sample metadata, expanded

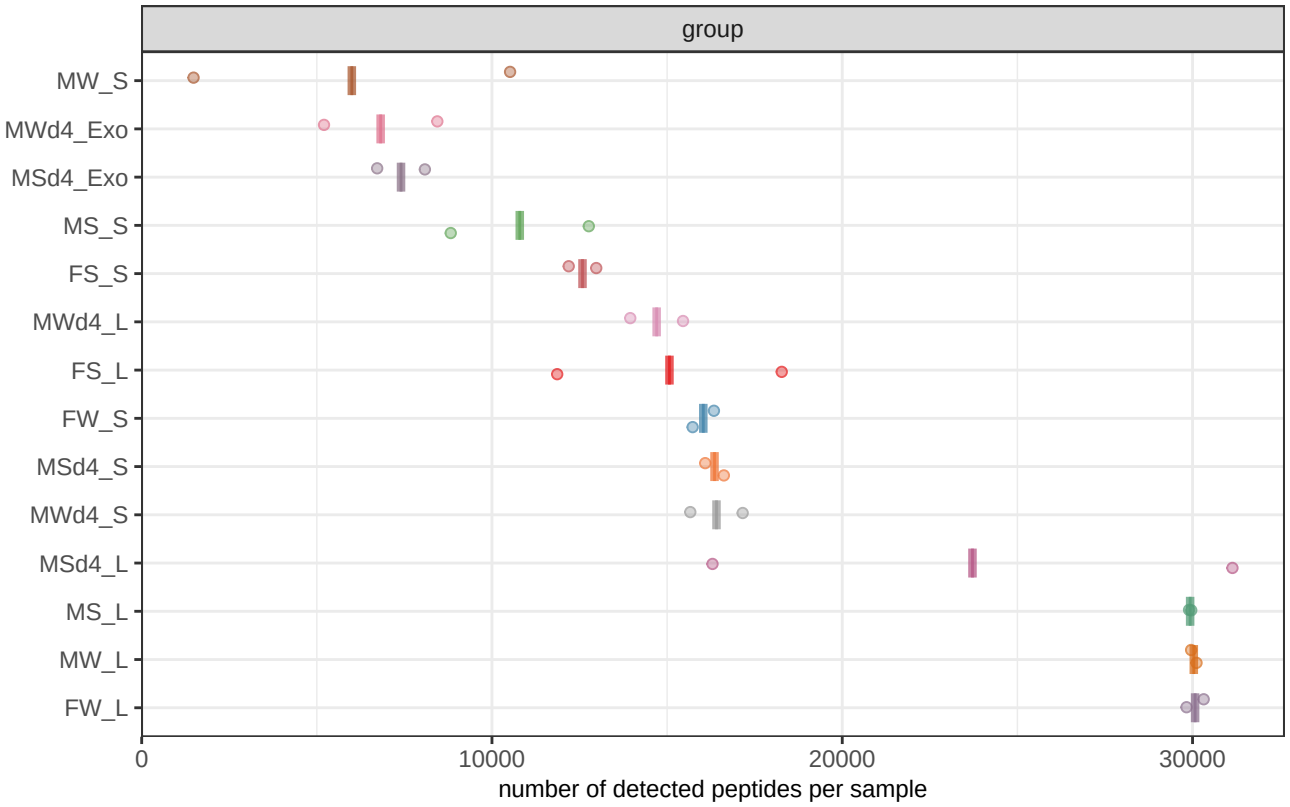
To further detail each sample property, each row in the above figure is now split into separate plots. Thus, a figure is generated for each property in the user-provided metadata (column in the samples table, its name shown in the plot title).

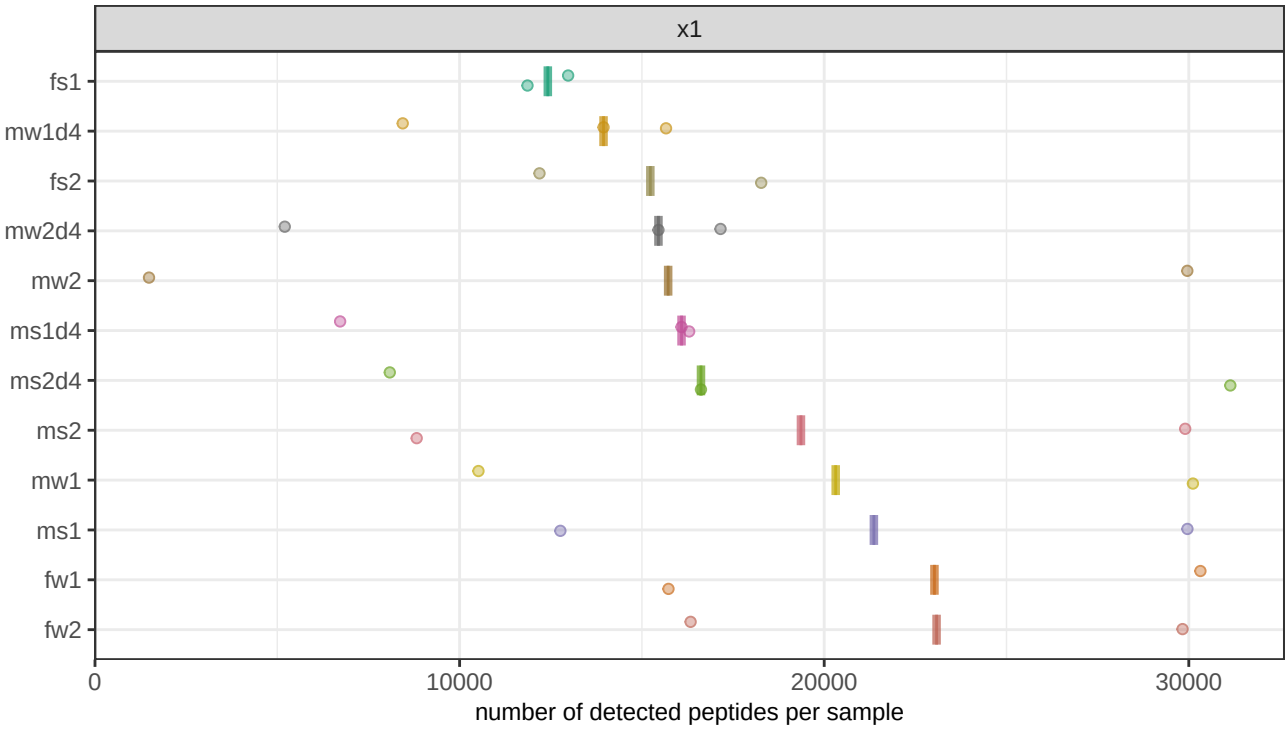
For categorical variables, a scatterplot shows on the y-axis all unique variables while the x-axis depicts the number of detected peptides. Colors are consistent with the above plot. *exclude* samples, if any, are depicted as squares. The median value is shown as a vertical line (thin line = median over 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).

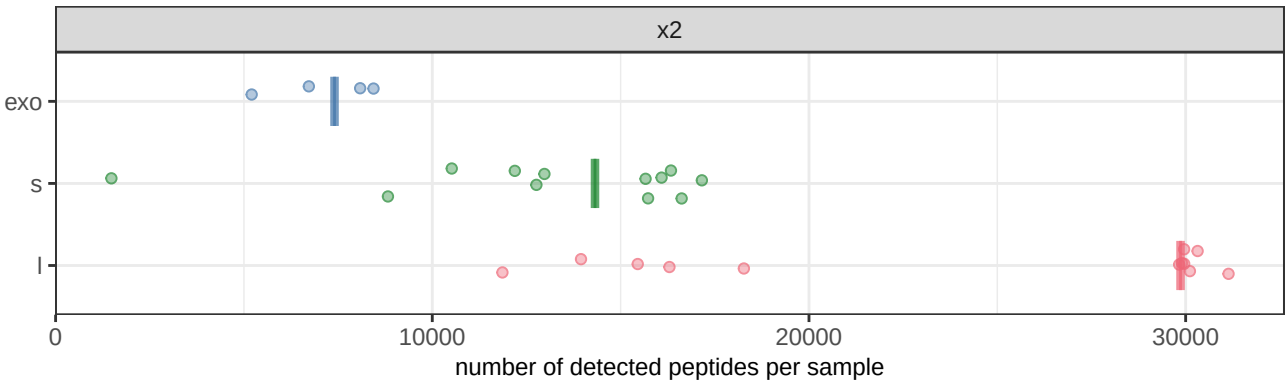
Note that samples flagged as *exclude* are user-provided in the sample metadata table. These are included in data visualizations but excluded from downstream statistical analysis (later part of the report).

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.


exclude
FALSE
0
10000
20000
30000
number of detected peptides per sample


group
MW_S
MWd4_Exo
MSd4_Exo
MS_S
FS_S
MWd4_L
FS_L
FW_S
MSd4_S
MWd4_S
MSd4_L
MS_L
MW_L
FW_L
0
10000
20000
30000
number of detected peptides per sample

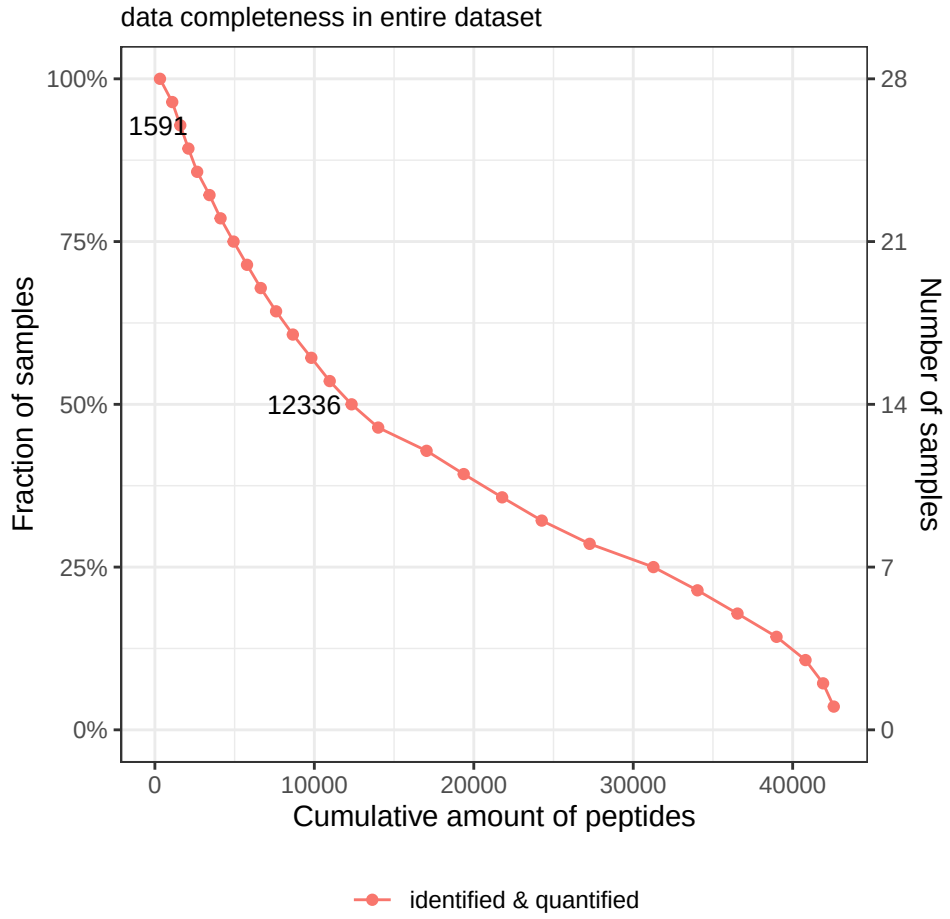

x1
fs1
mw1d4
fs2
mw2d4
mw2
ms1d4
ms2d4
ms2
mw1
ms1
fw1
fw2
0
10000
20000
30000
number of detected peptides per sample


x2
exo
s
l
0
10000
20000
30000
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

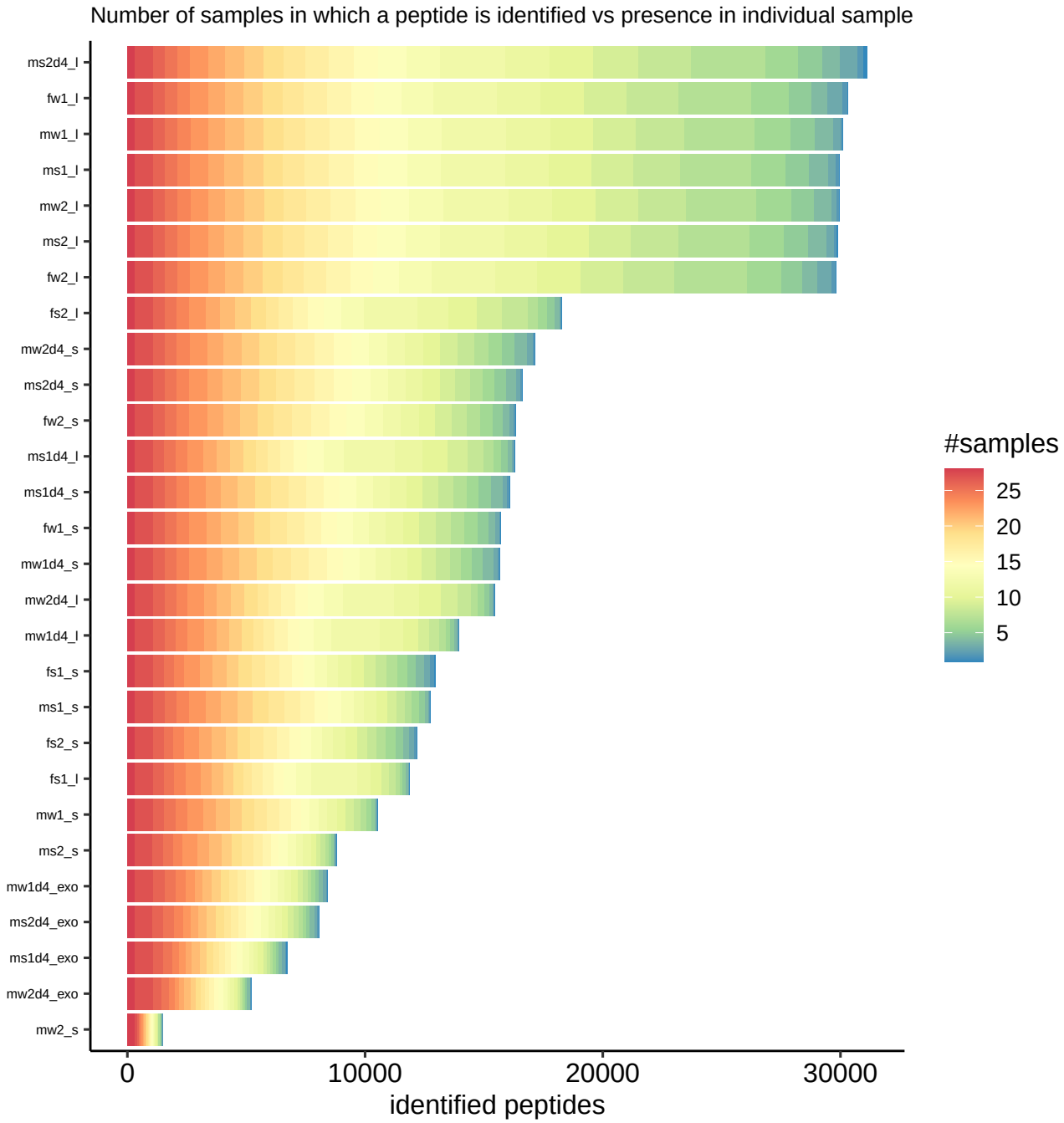


Samples flagged as 'exclude' (by user) are not taken into account in this figure. Exact values are shown for data points matching 90% and 50% of samples to convenience comparison between analyses (e.g. before/after configuring 'exclude' samples, or comparing between experiments of similar protocol).

1.2.2 peptide detection frequency

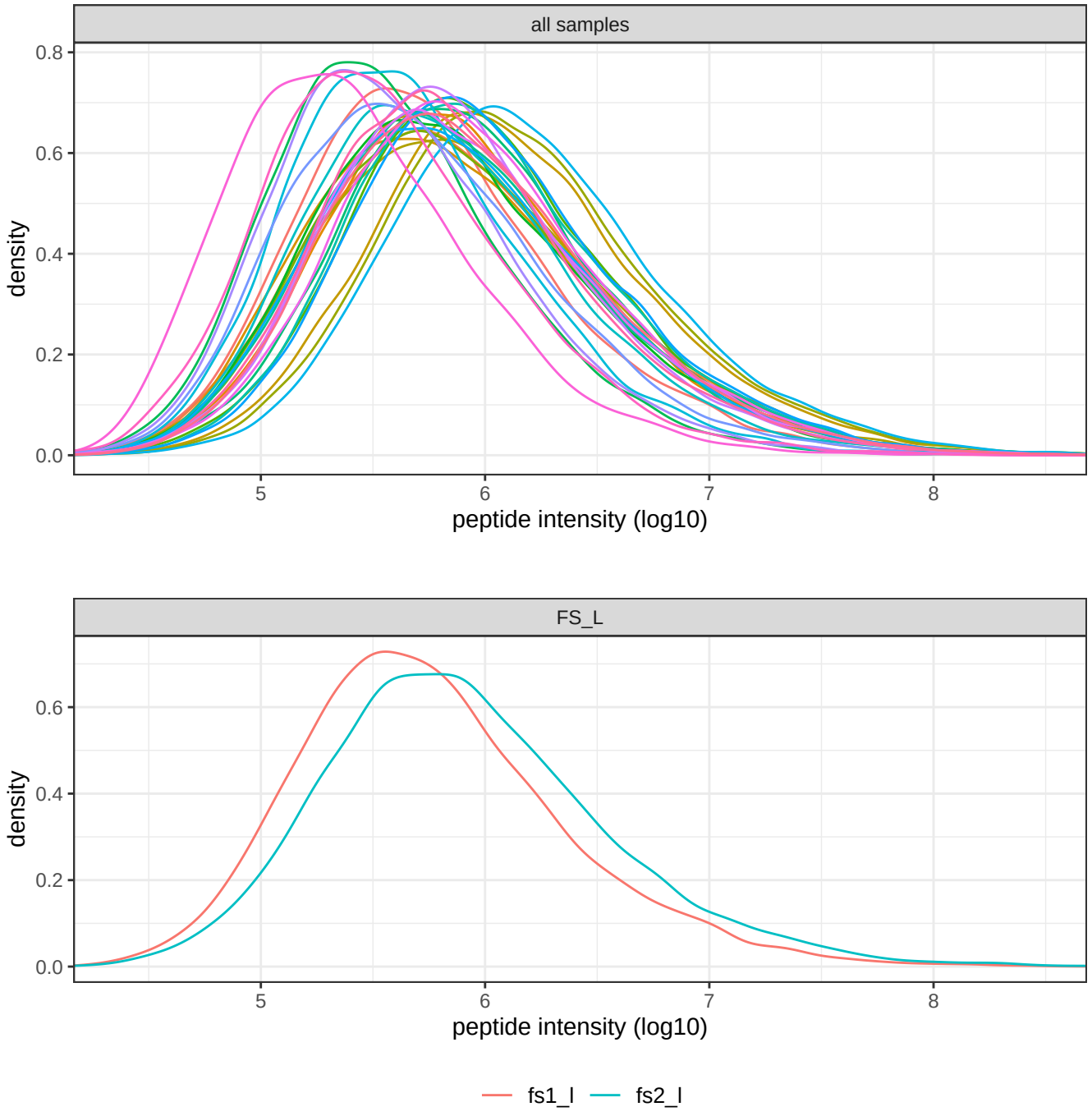
Each identified peptide in a sample is classified and color-coded by the number of other samples where the same peptide is present. Visualization of the amount of peptides that overlap with other samples in the dataset, from peptides identified in most samples (red) to one-hit-wonders (blue), helps identify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

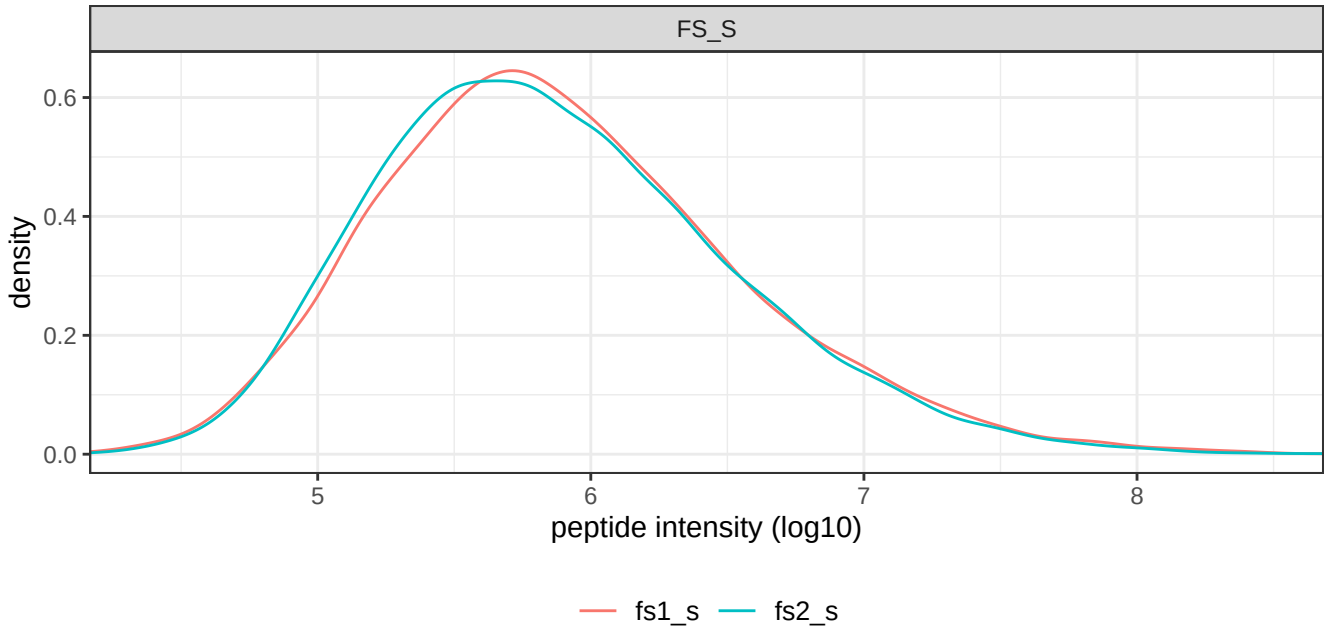
Optimally, the majority of peptides in each sample are red~orange with relatively few uniquely identified peptides (blue~green). Samples are sorted by the total amount of detected peptides.

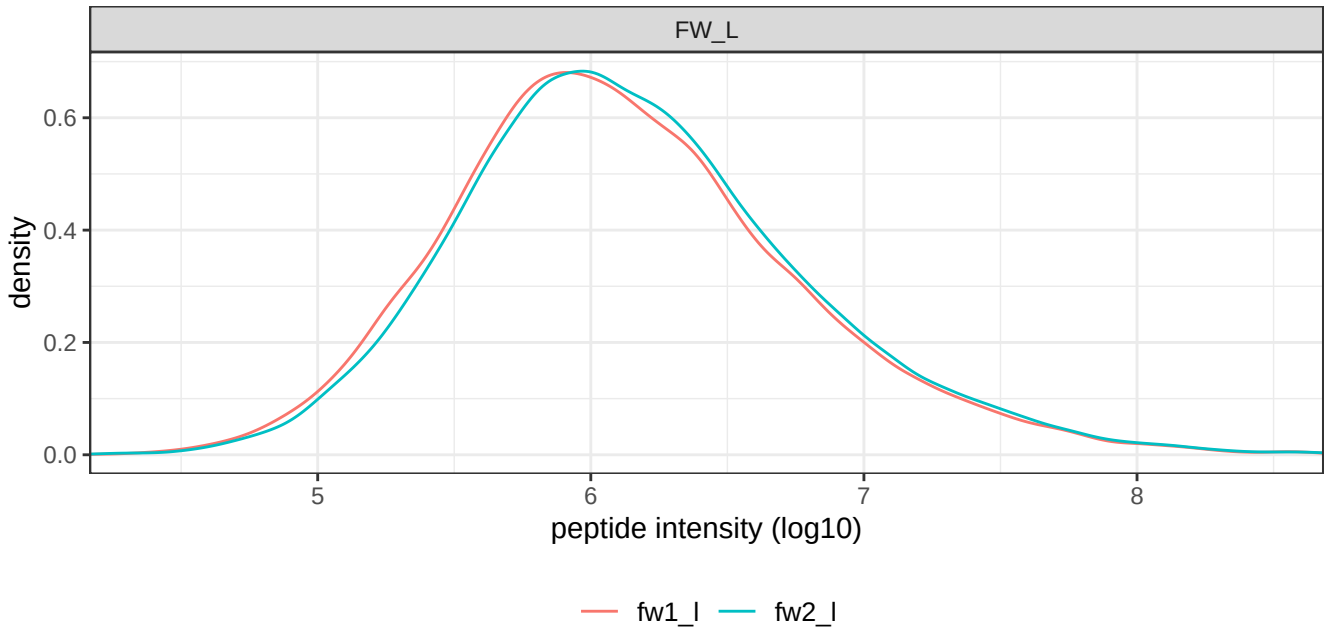


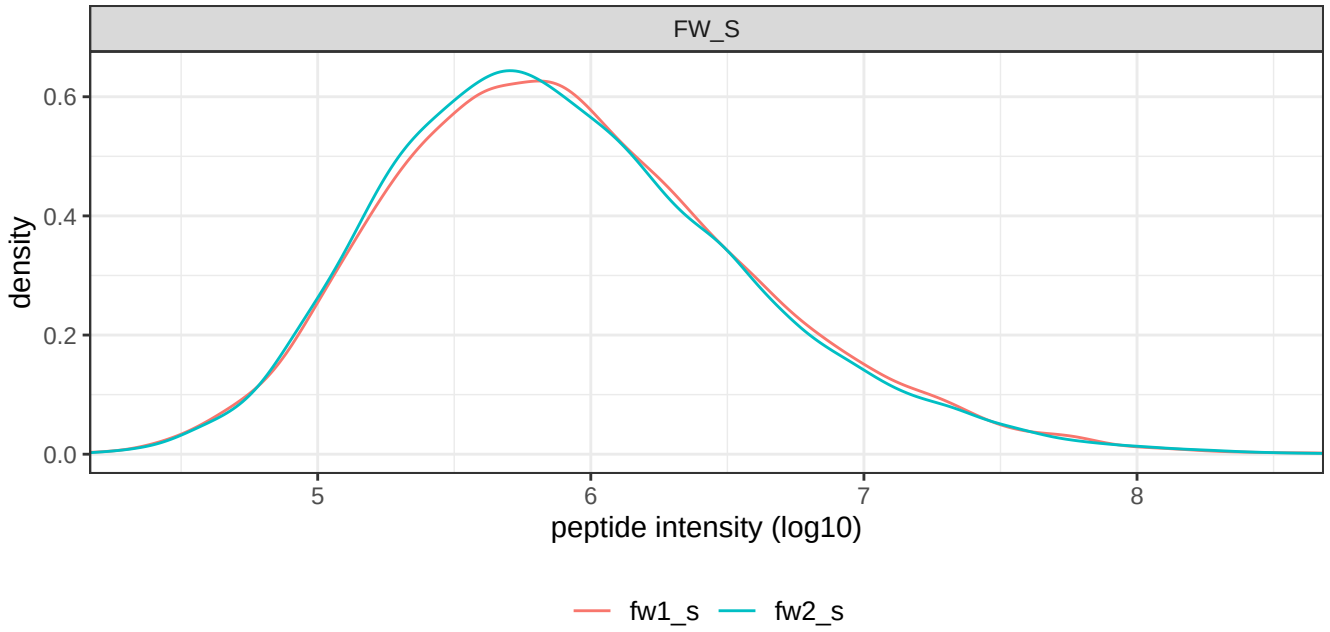
1.3 abundance distributions

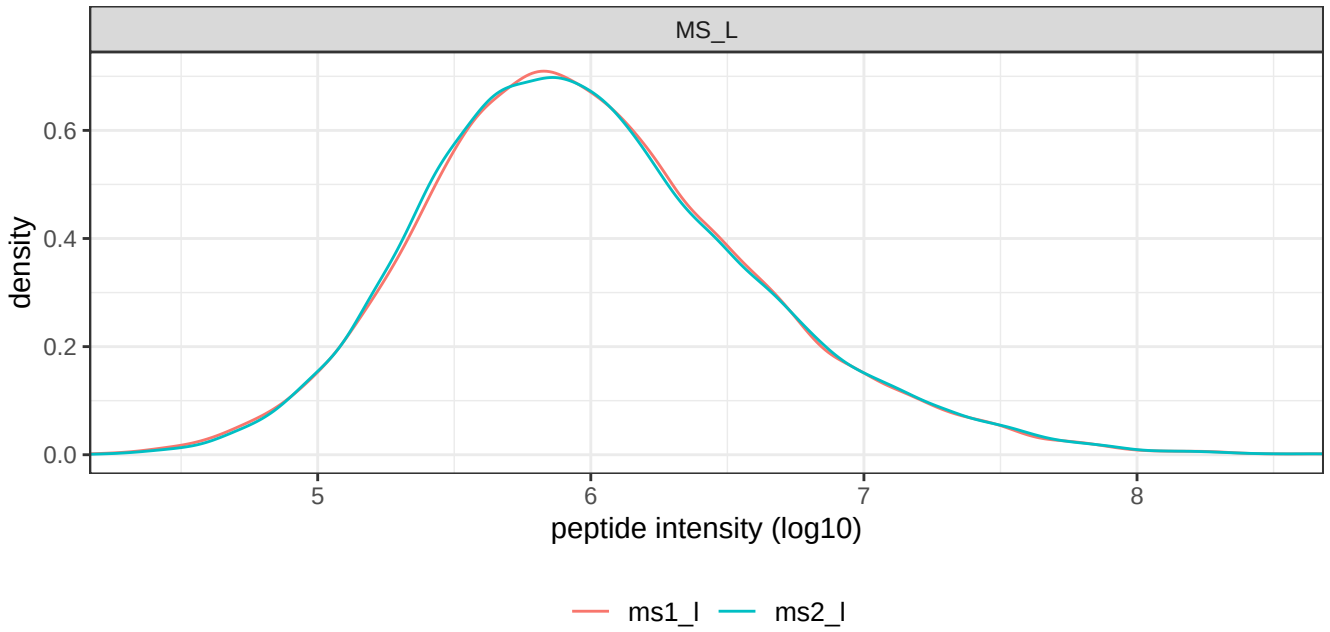
The figures in this subsection are used to identify unexpected mass-spec sensitivity or sample loading differences. Peptide data is shown as provided in input files, so peptide filtering nor intensity normalization has been applied yet (for proper QC, make sure the software that generated the input data 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.

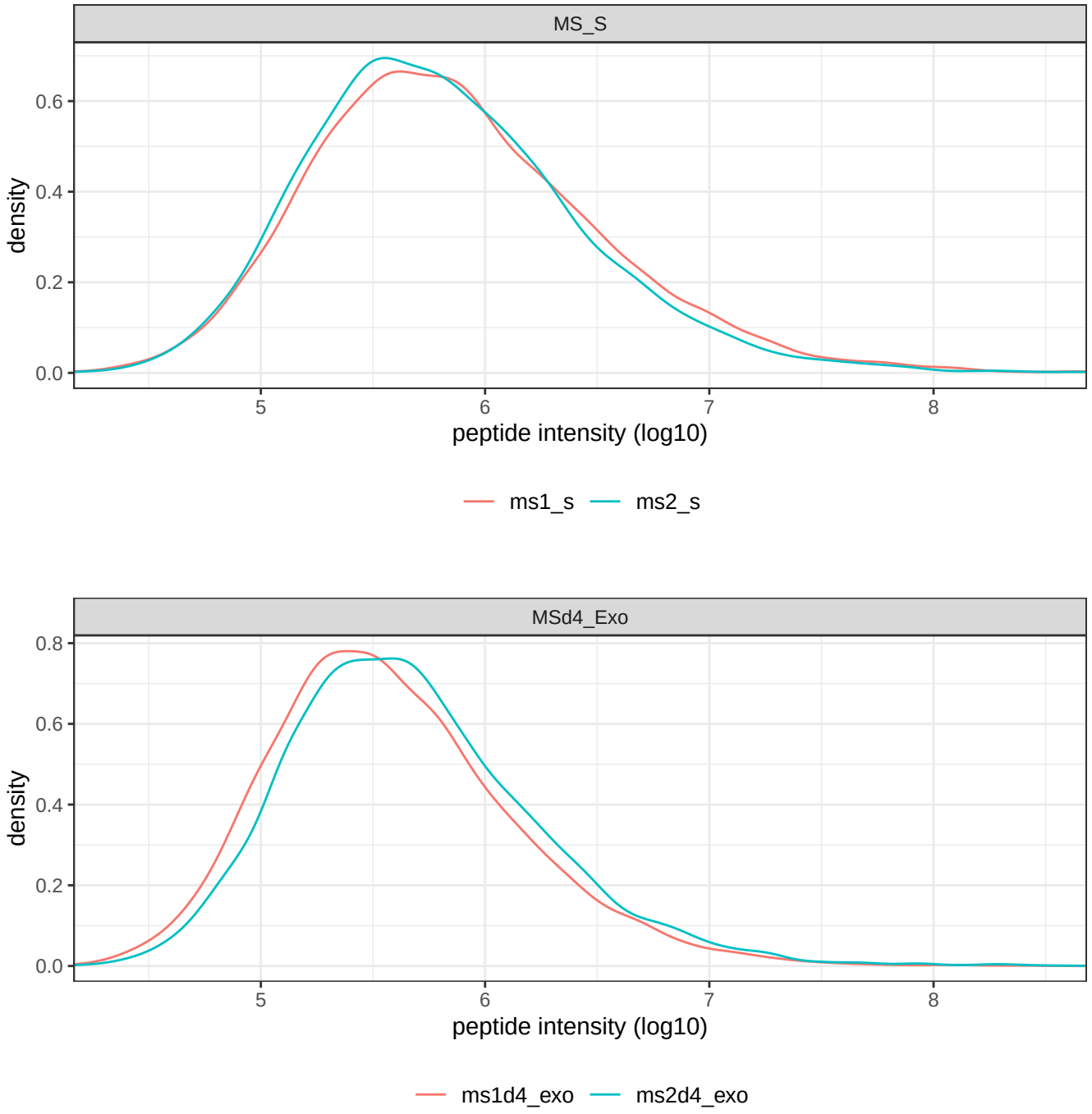



FS_S
density
0.0
0.2
0.4
0.6
5
6
7
8
peptide intensity (log10)
fs1_s
fs2_s

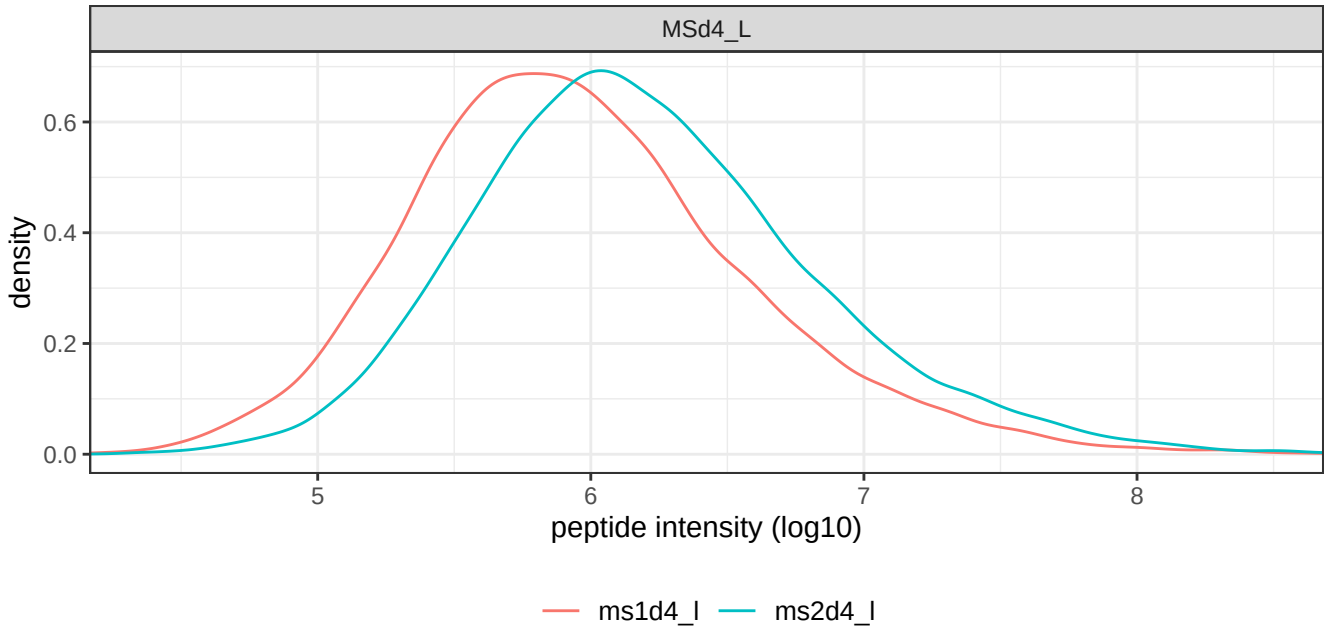

FW_L
density
0.0
0.2
0.4
0.6
5
6
7
8
peptide intensity (log10)
fw1_l
fw2_l

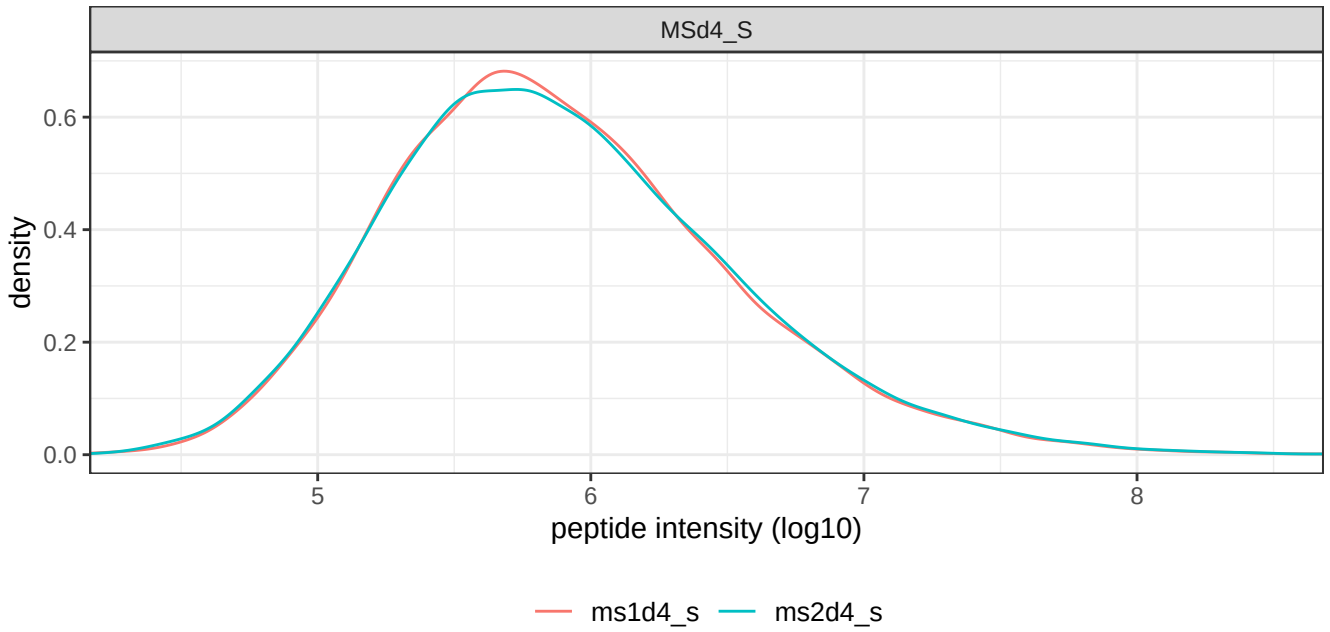

FW_S
density
0.0
0.2
0.4
0.6
5
6
7
8
peptide intensity (log10)
fw1_s
fw2_s

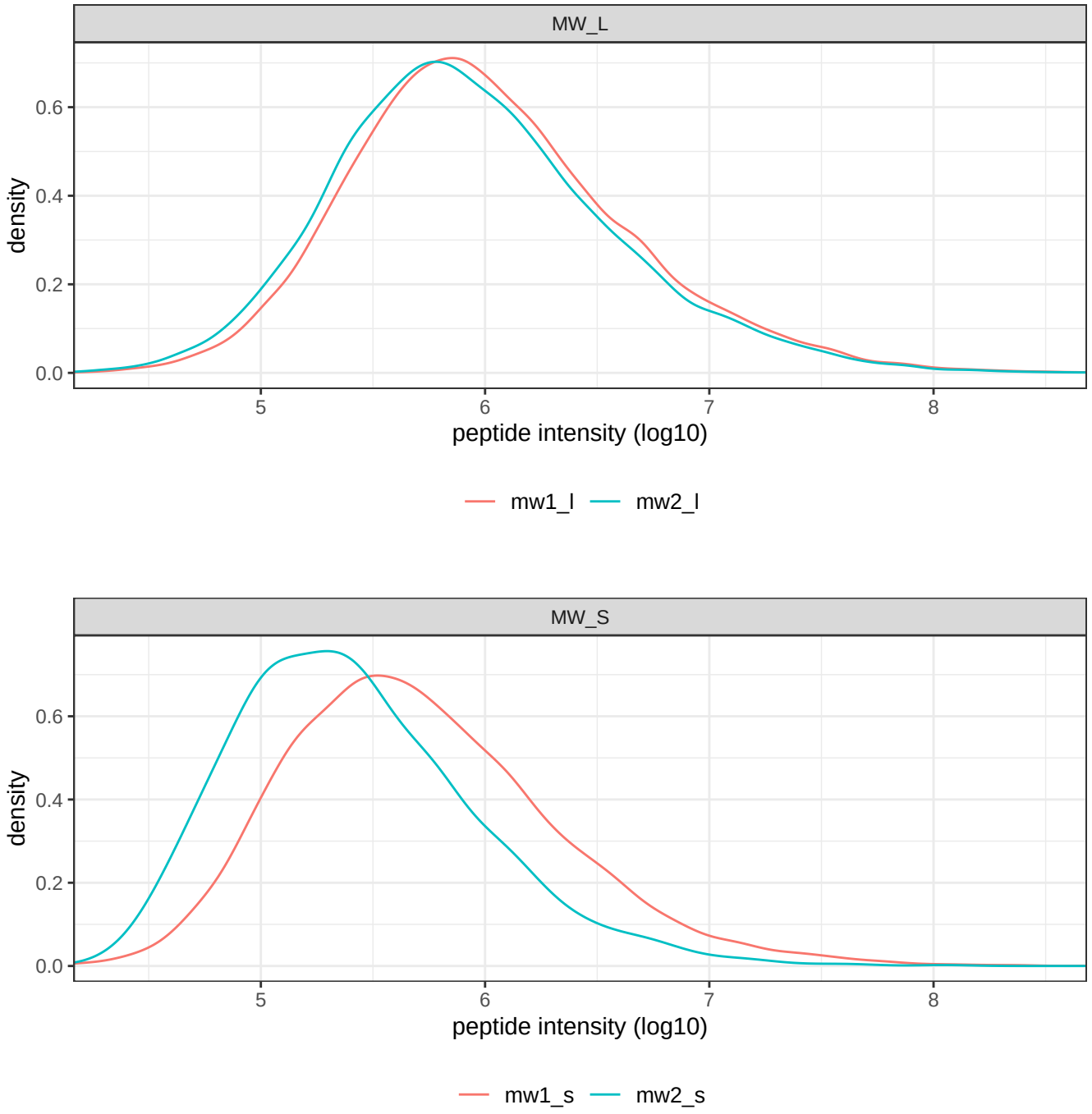

MS_L
density
0.0
0.2
0.4
0.6
5
6
7
8
peptide intensity (log10)
ms1_l
ms2_l


MS_S
density
0.0
0.2
0.4
0.6
5
6
7
8
peptide intensity (log10)
ms1_s
ms2_s

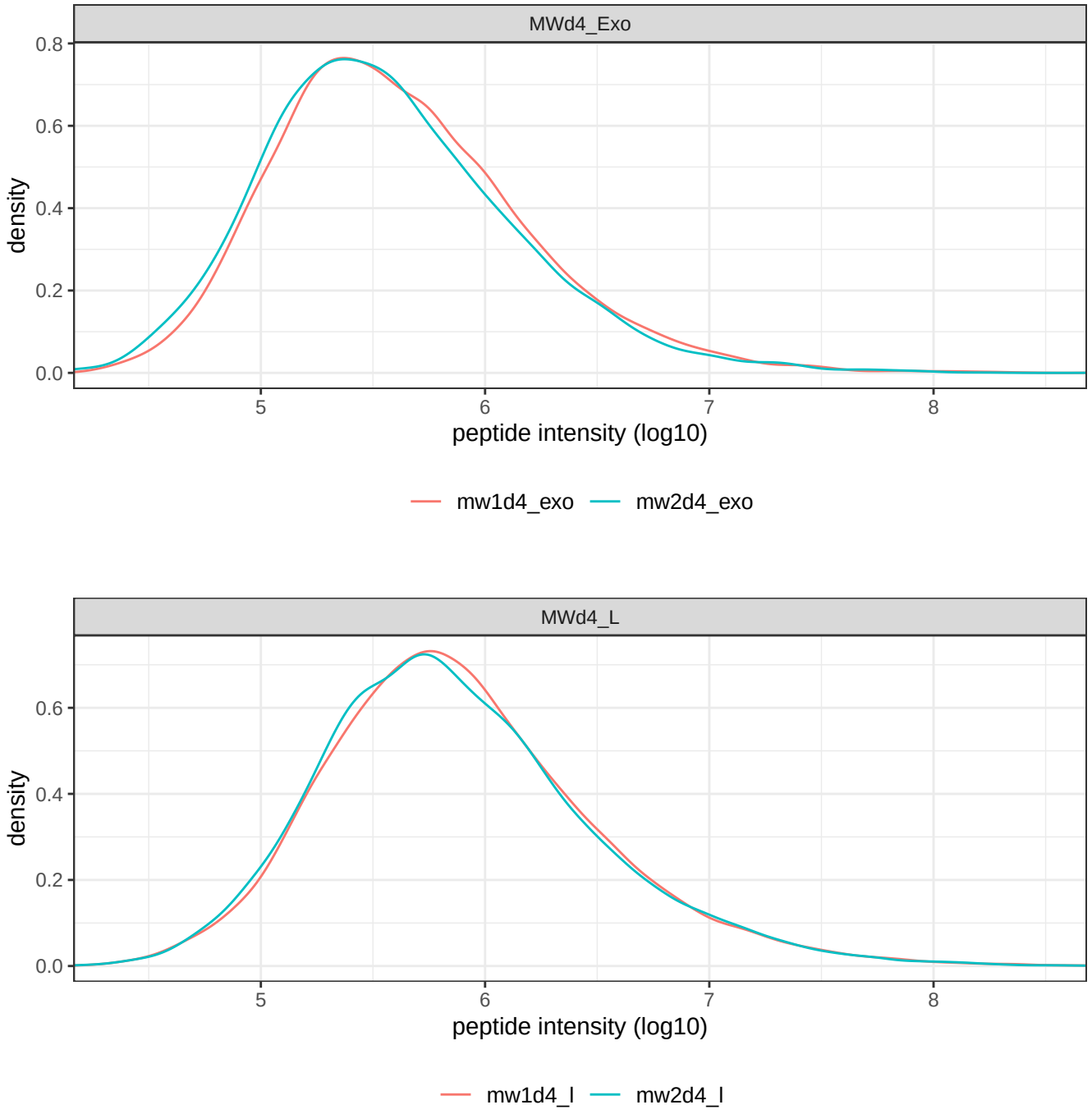
MSd4_Exo
density
0.0
0.2
0.4
0.6
0.8
5
6
7
8
peptide intensity (log10)
ms1d4_exo
ms2d4_exo


MSd4_L
density
0.0
0.2
0.4
0.6
5
6
7
8
peptide intensity (log10)
ms1d4_l
ms2d4_l

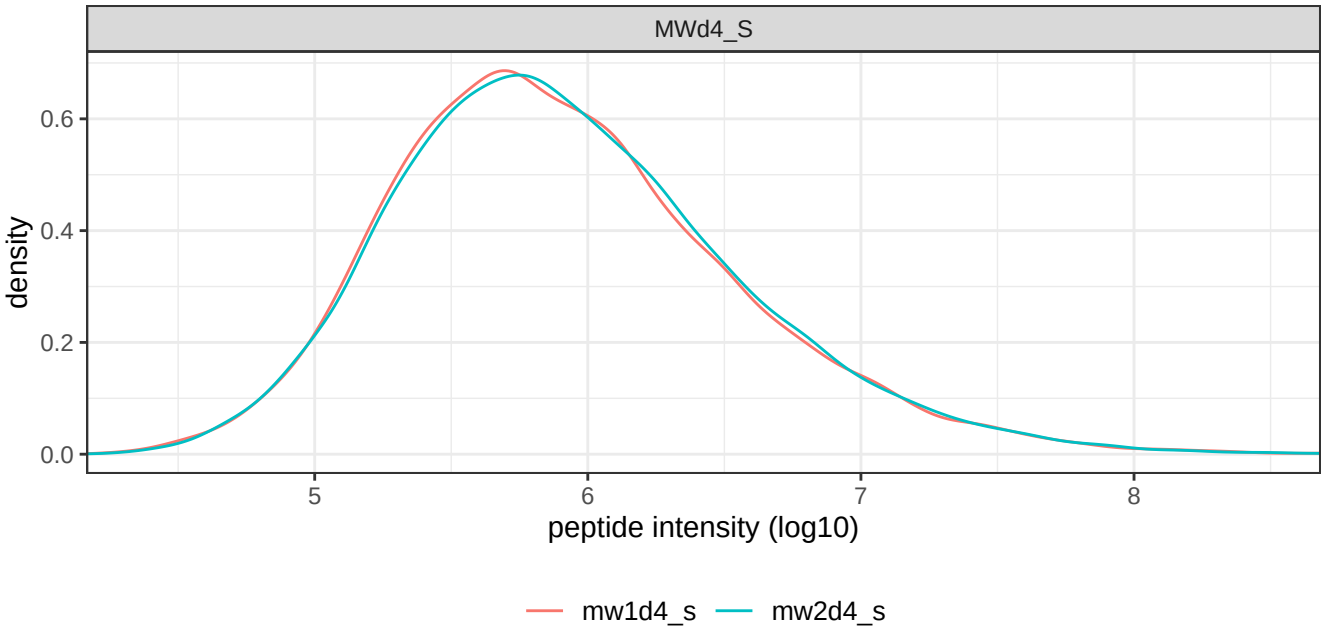

MSd4_S
density
0.0
0.2
0.4
0.6
5
6
7
8
peptide intensity (log10)
ms1d4_s
ms2d4_s


MW_L
density
0.0
0.2
0.4
0.6
5
6
7
8
peptide intensity (log10)
mw1_l
mw2_l

MW_S
density
0.0
0.2
0.4
0.6
5
6
7
8
peptide intensity (log10)
mw1_s
mw2_s


MWd4_Exo
density
0.0
0.2
0.4
0.6
0.8
5
6
7
8
peptide intensity (log10)
mw1d4_exo
mw2d4_exo

MWd4_L
density
0.0
0.2
0.4
0.6
5
6
7
8
peptide intensity (log10)
mw1d4_l
mw2d4_l

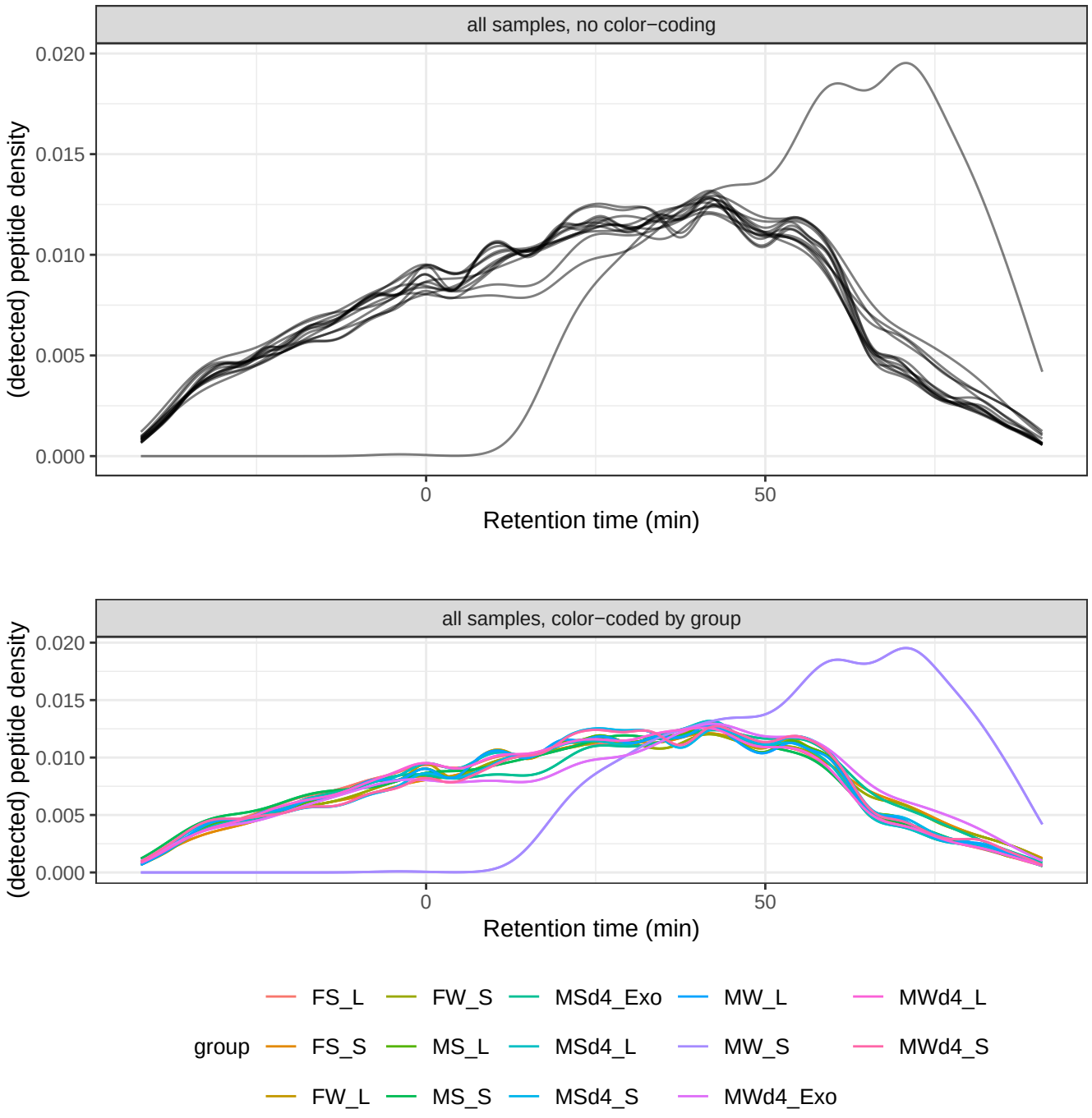

MWd4_S
density
0.0
0.2
0.4
0.6
5
6
7
8
peptide intensity (log10)
mw1d4_s
mw2d4_s

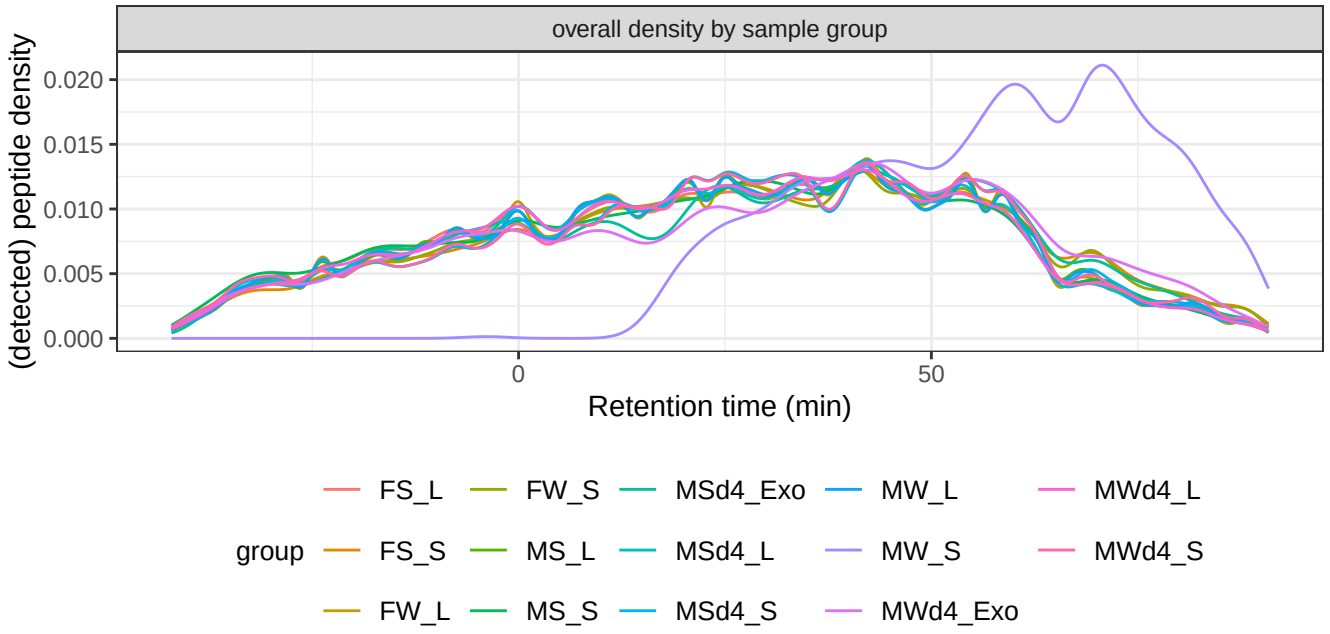
1.4 retention time

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.



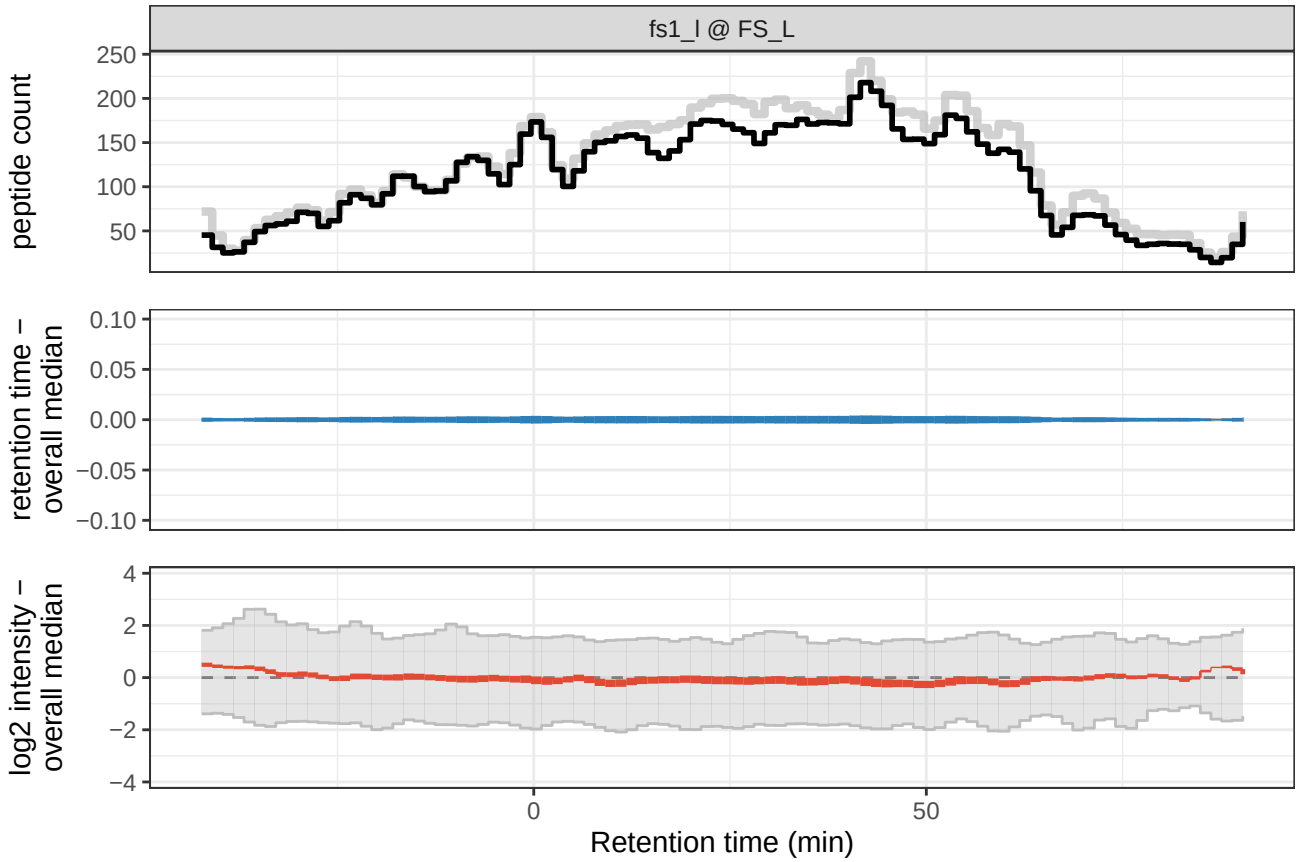


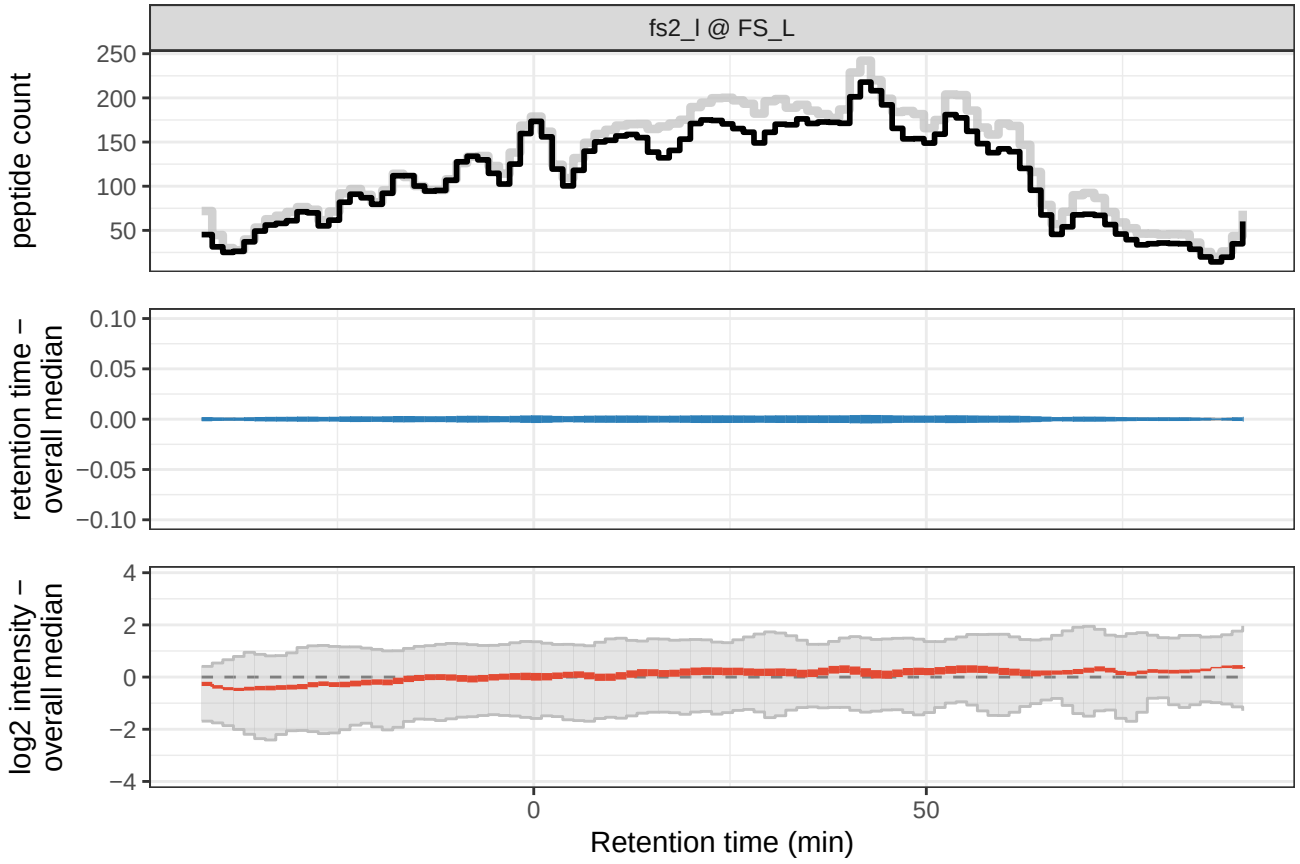
1.4.2 retention time local effects

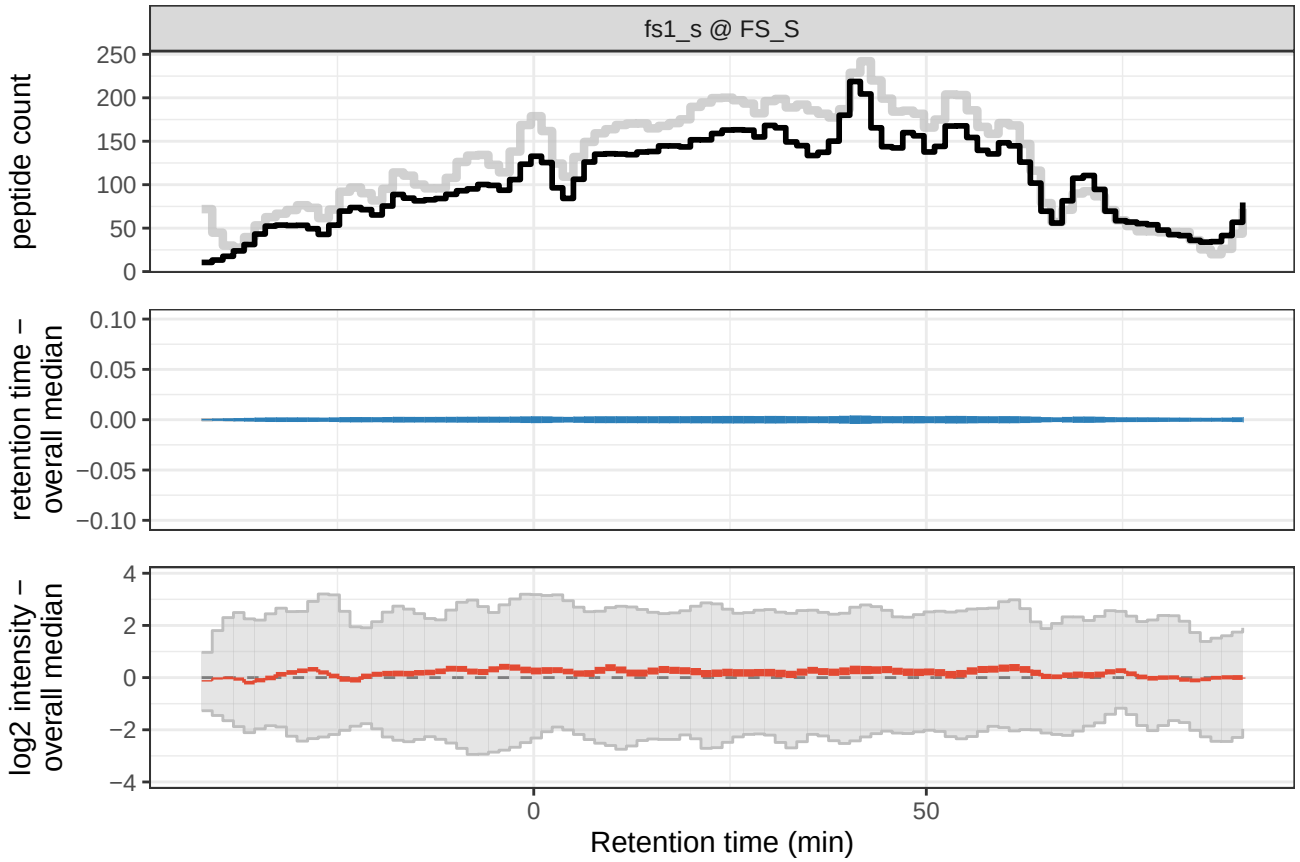
To investigate how each measurement differs from others, we visualize each sample as a 3 panel figure. First, the data is binned across the retention time dimension (x-axis). If a samples was marked as 'exclude' in the provided sample metadata, this is indicated in the plot title.

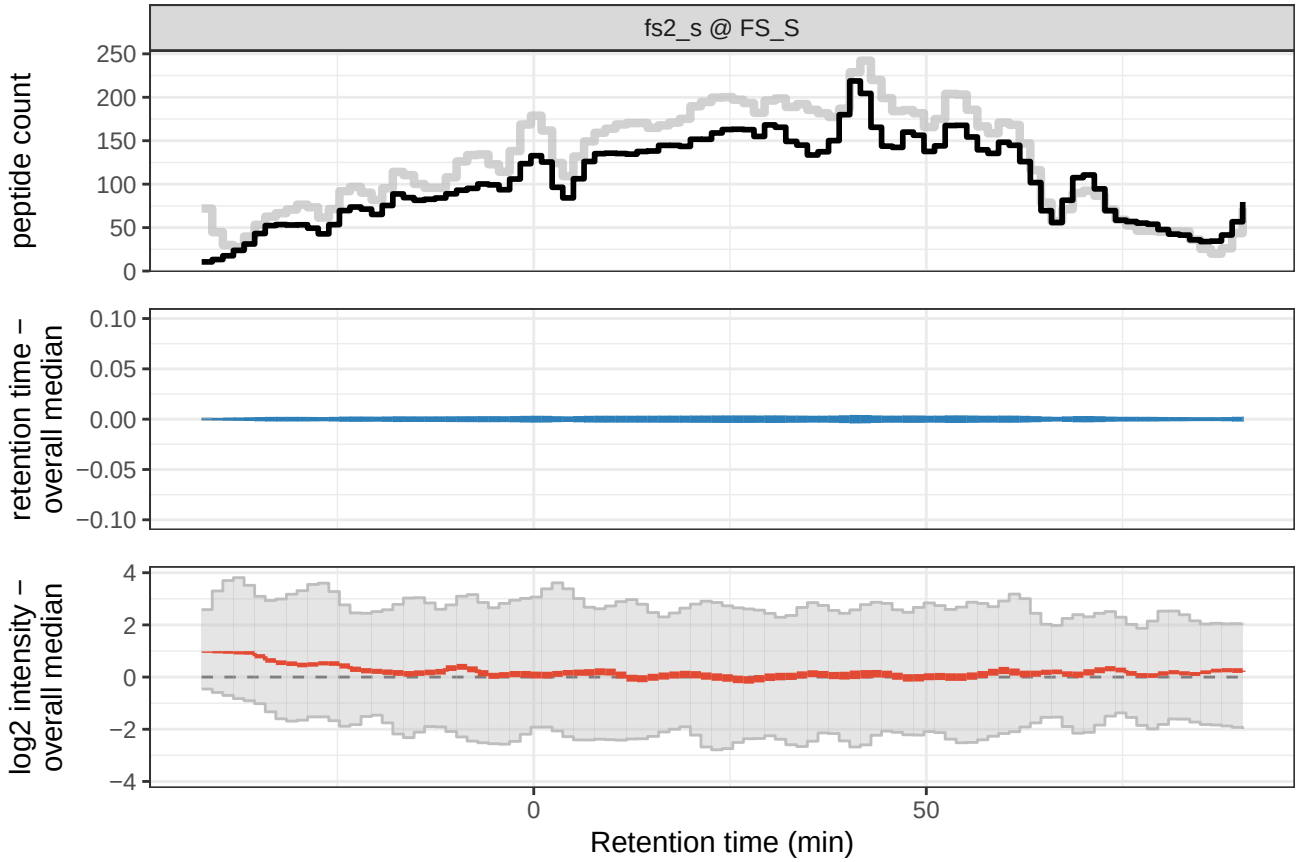
The top panel shows the number of peptides in the input data, e.g. as recognized by the software that generated input for this pipeline, over time (black line). For reference, the grey line shows the median amount over all samples (note; if this is the exact same in all samples, the grey line may not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

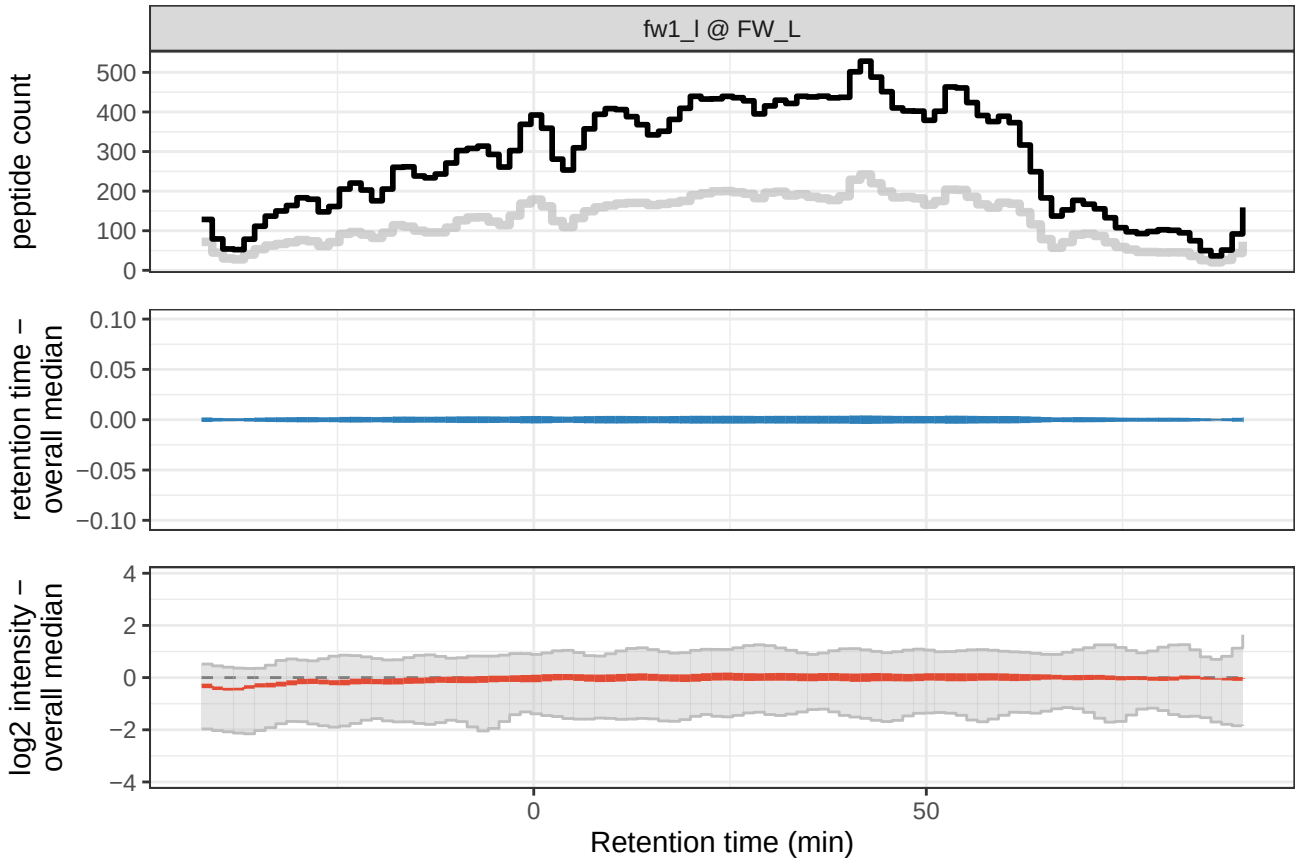
The middle panel indicates whether peptide retention times deviate from their median over all samples (blue line). The grey area depicts the 5% and 95% quantiles, respectively. The line width corresponds to the number of peptides eluting at that time (data from first panel). Analogously, the bottom panel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

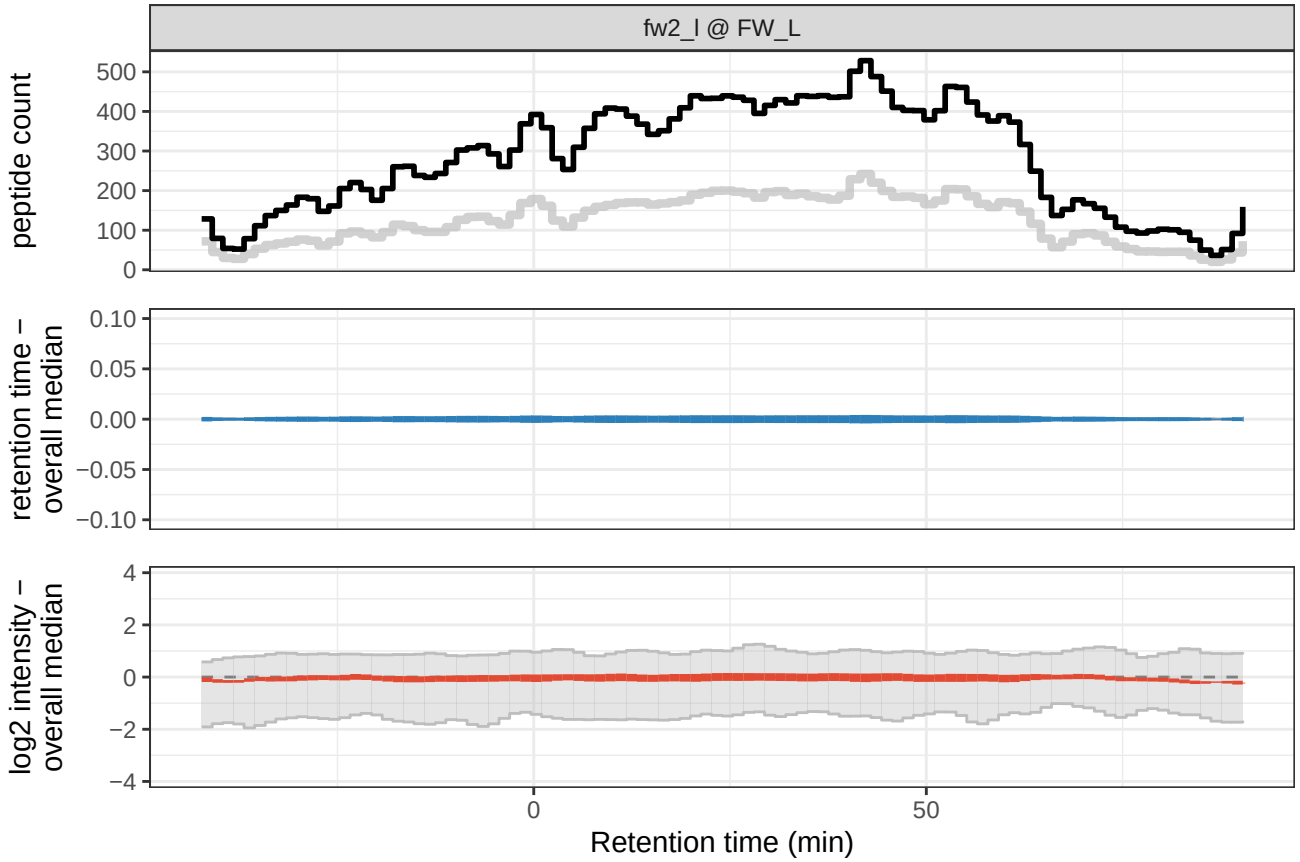

fs1_l @ FS_L
peptide count
250
200
150
100
50
retention time –
overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity –
overall median
4
2
0
−2
−4
0
50
Retention time (min)

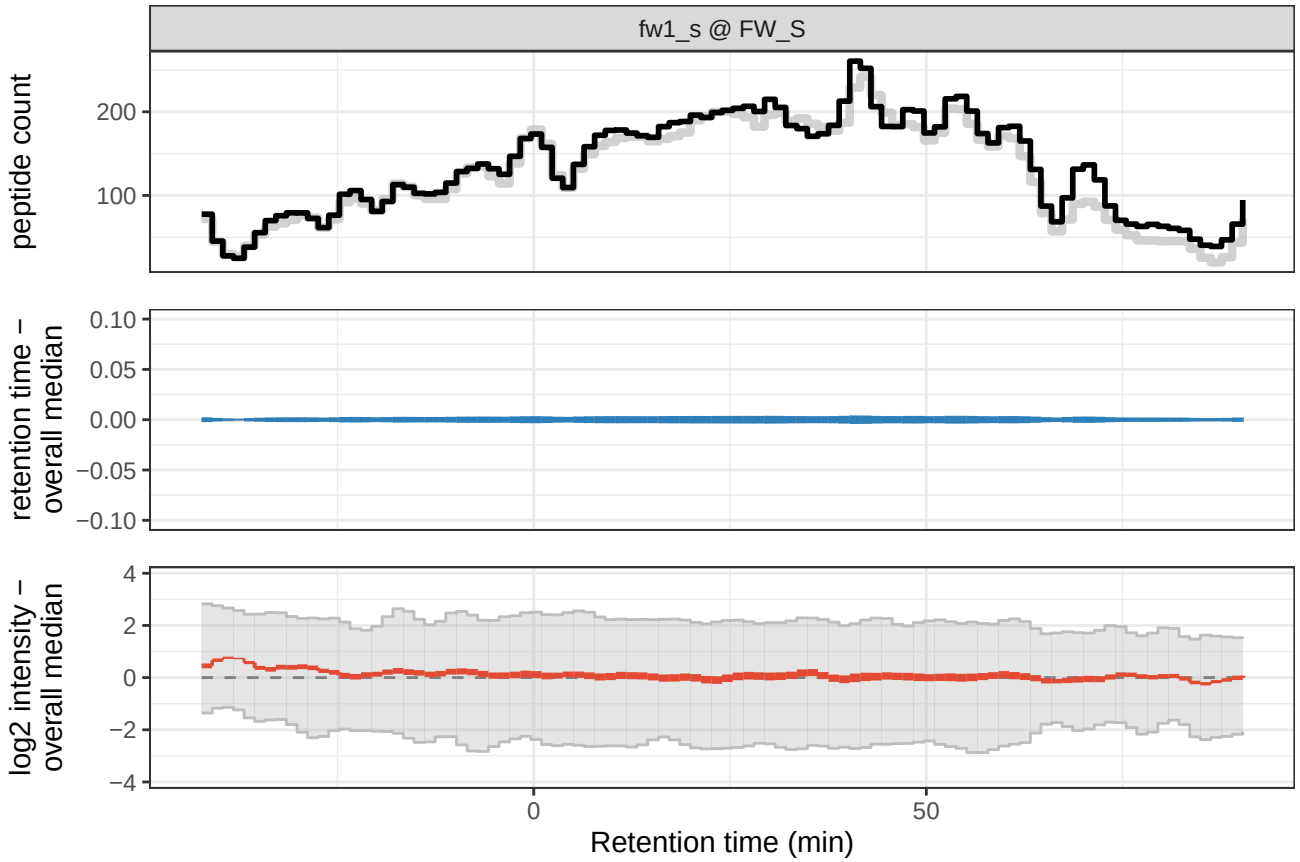

fs2_l @ FS_L
peptide count
250
200
150
100
50
retention time –
overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity –
overall median
4
2
0
−2
−4
0
50
Retention time (min)

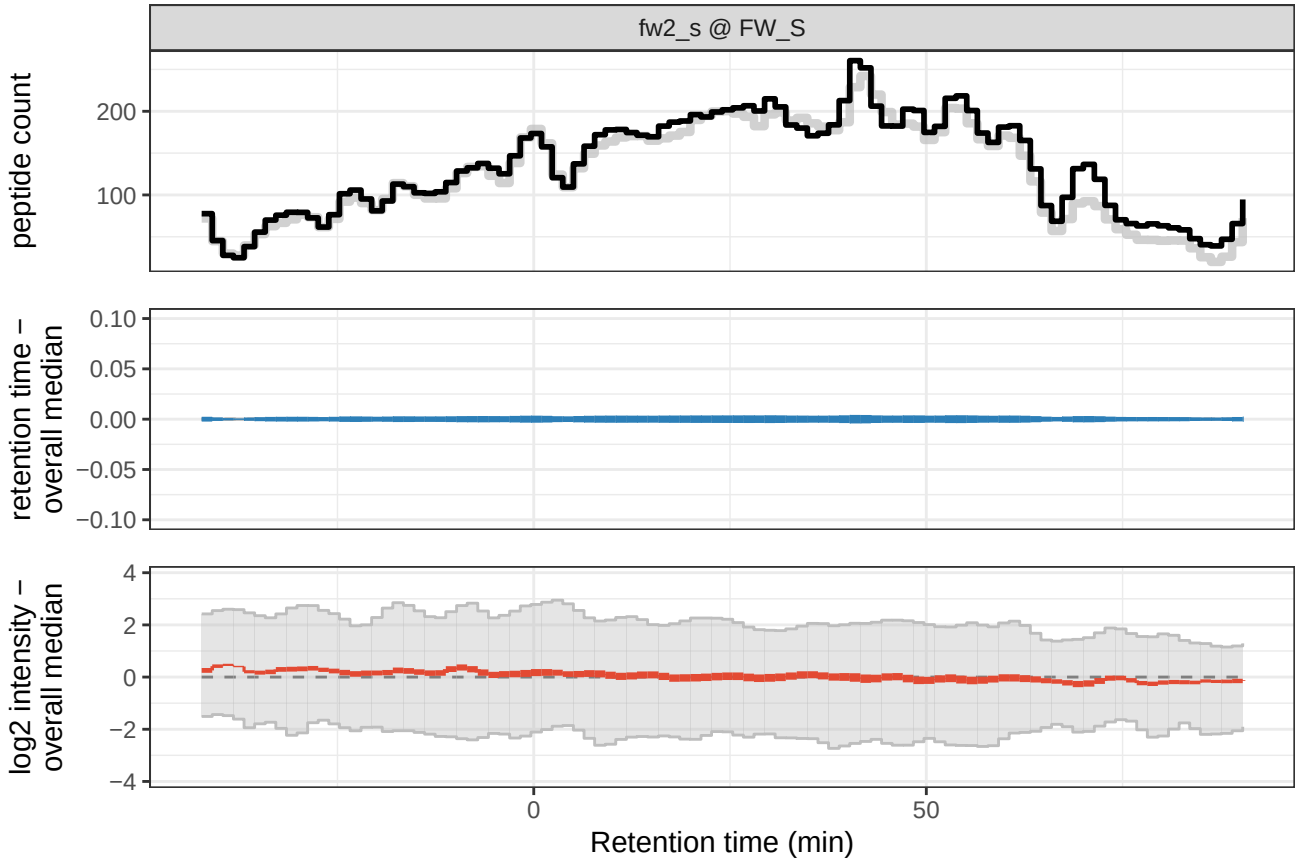

fs1_s @ FS_S
peptide count
250
200
150
100
50
0
retention time –
overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity –
overall median
4
2
0
−2
−4
0
50
Retention time (min)

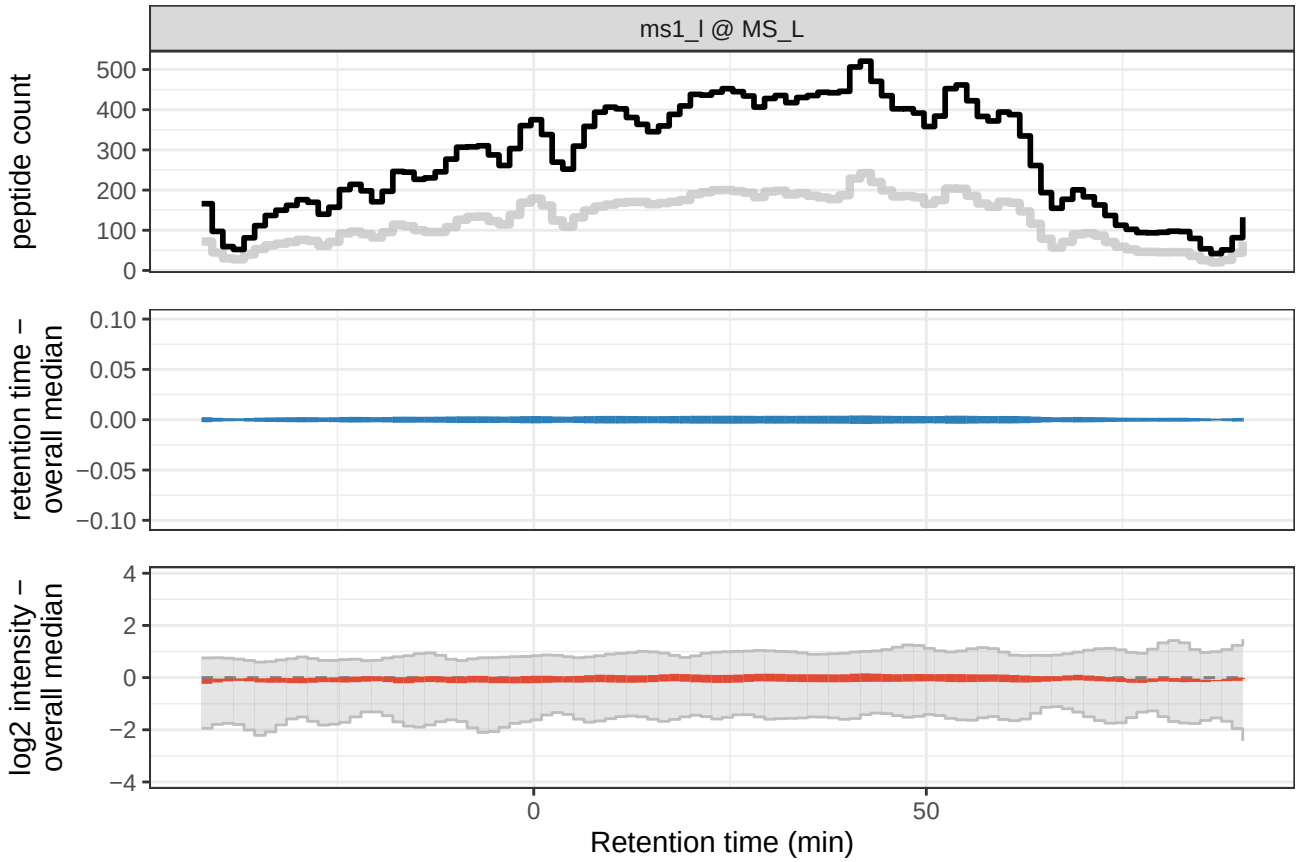

fs2_s @ FS_S
peptide count
250
200
150
100
50
0
retention time –
overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity –
overall median
4
2
0
−2
−4
0
50
Retention time (min)

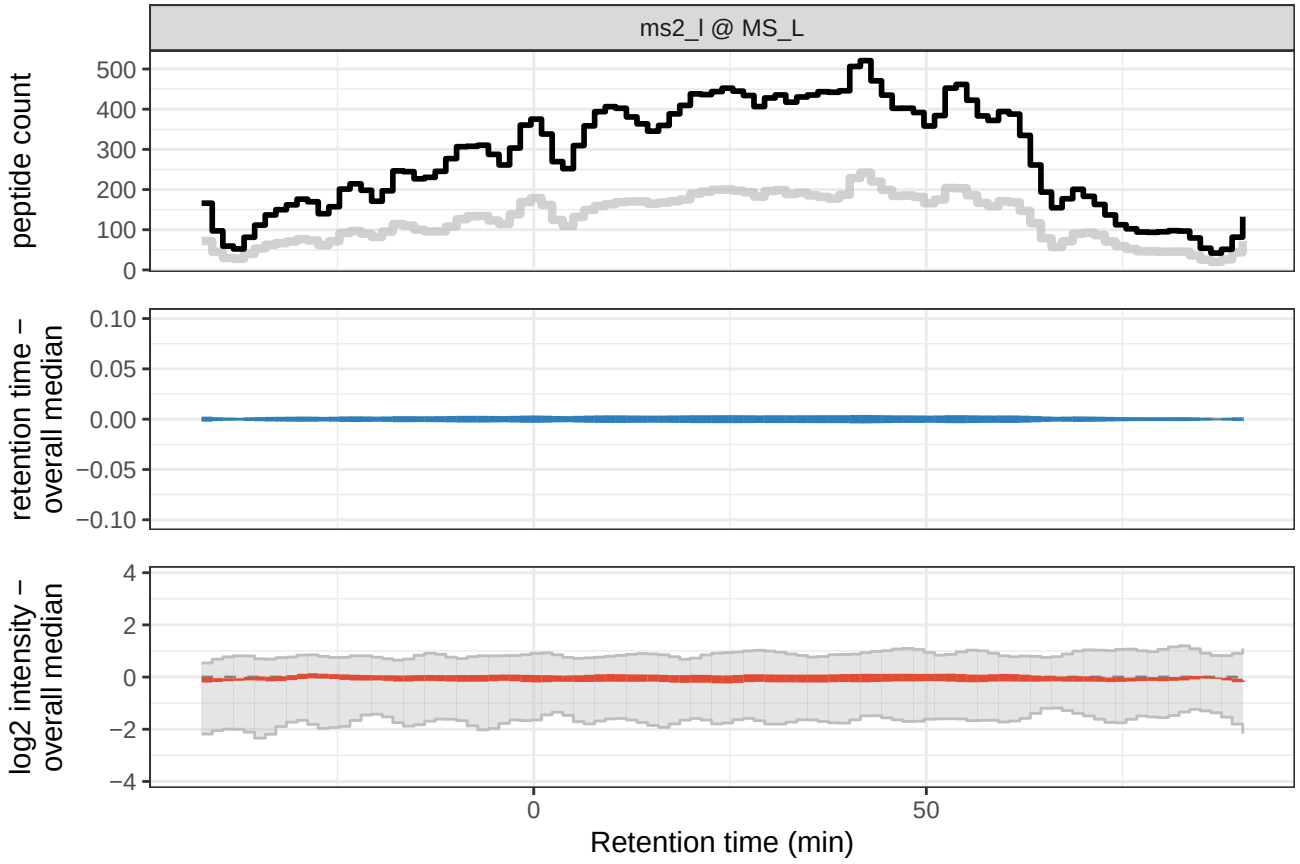

fw1_l @ FW_L
peptide count
500
400
300
200
100
0
retention time –
overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity –
overall median
4
2
0
−2
−4
0
50
Retention time (min)

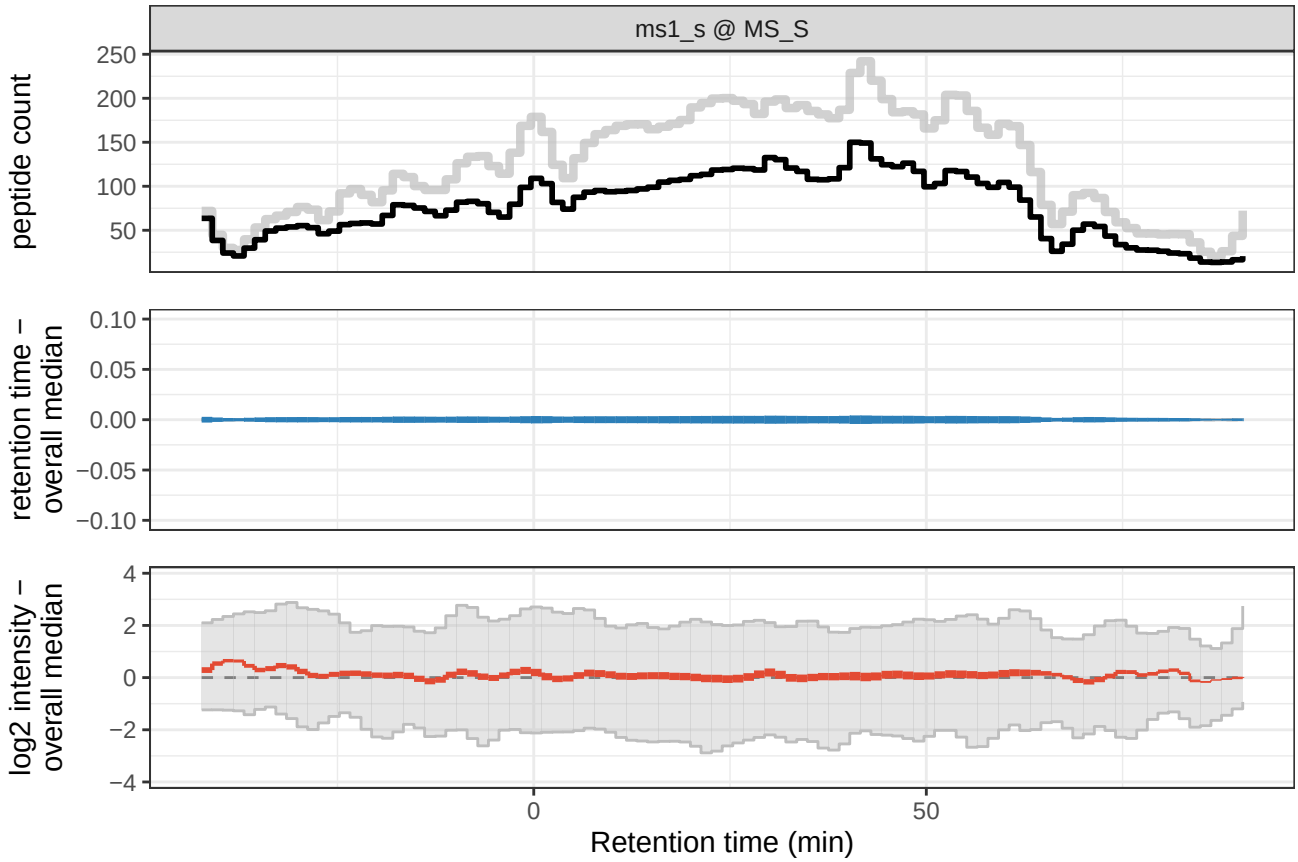

fw2_l @ FW_L
peptide count
500
400
300
200
100
0
retention time –
overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity –
overall median
4
2
0
−2
−4
0
50
Retention time (min)

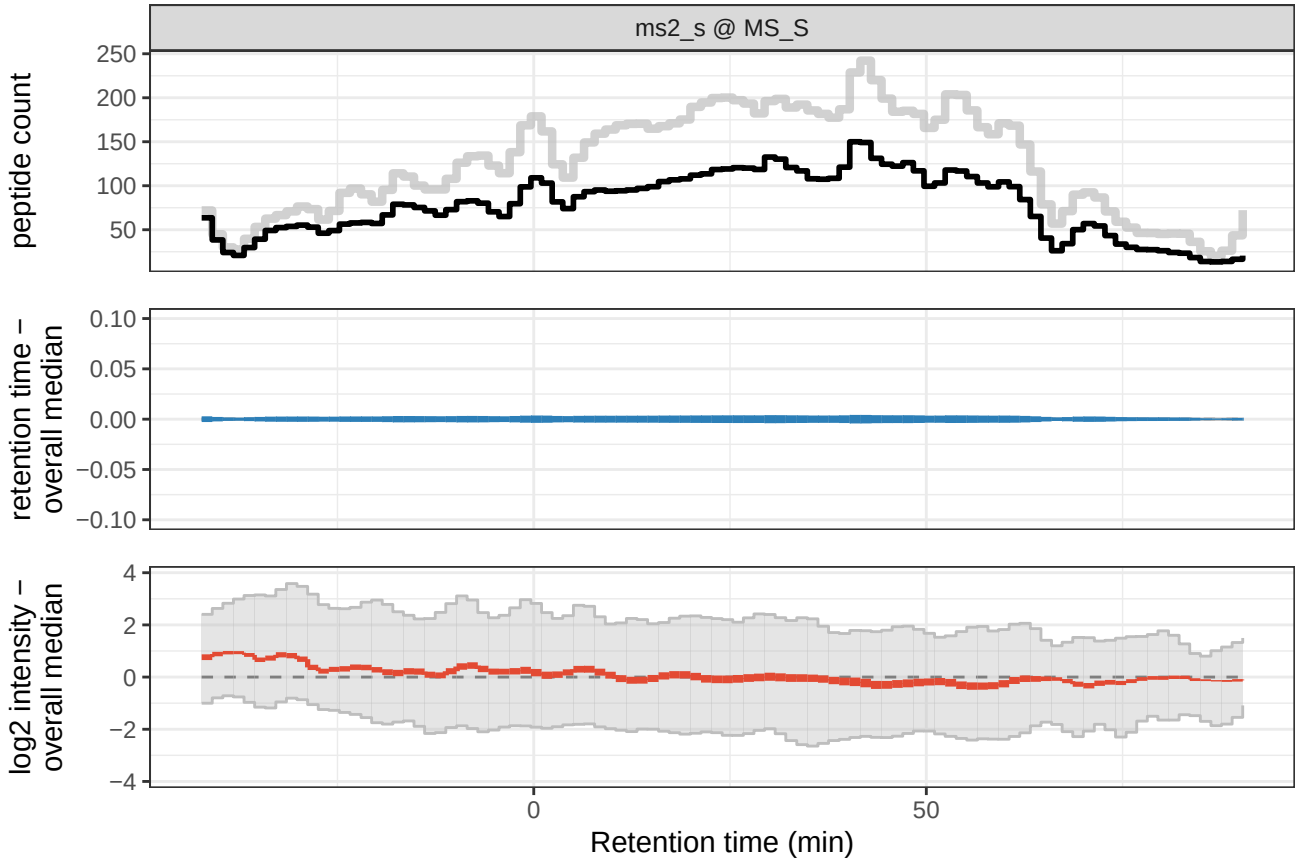

fw1_s @ FW_S
peptide count
200
100
retention time – overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity – overall median
4
2
0
−2
−4
0
50
Retention time (min)

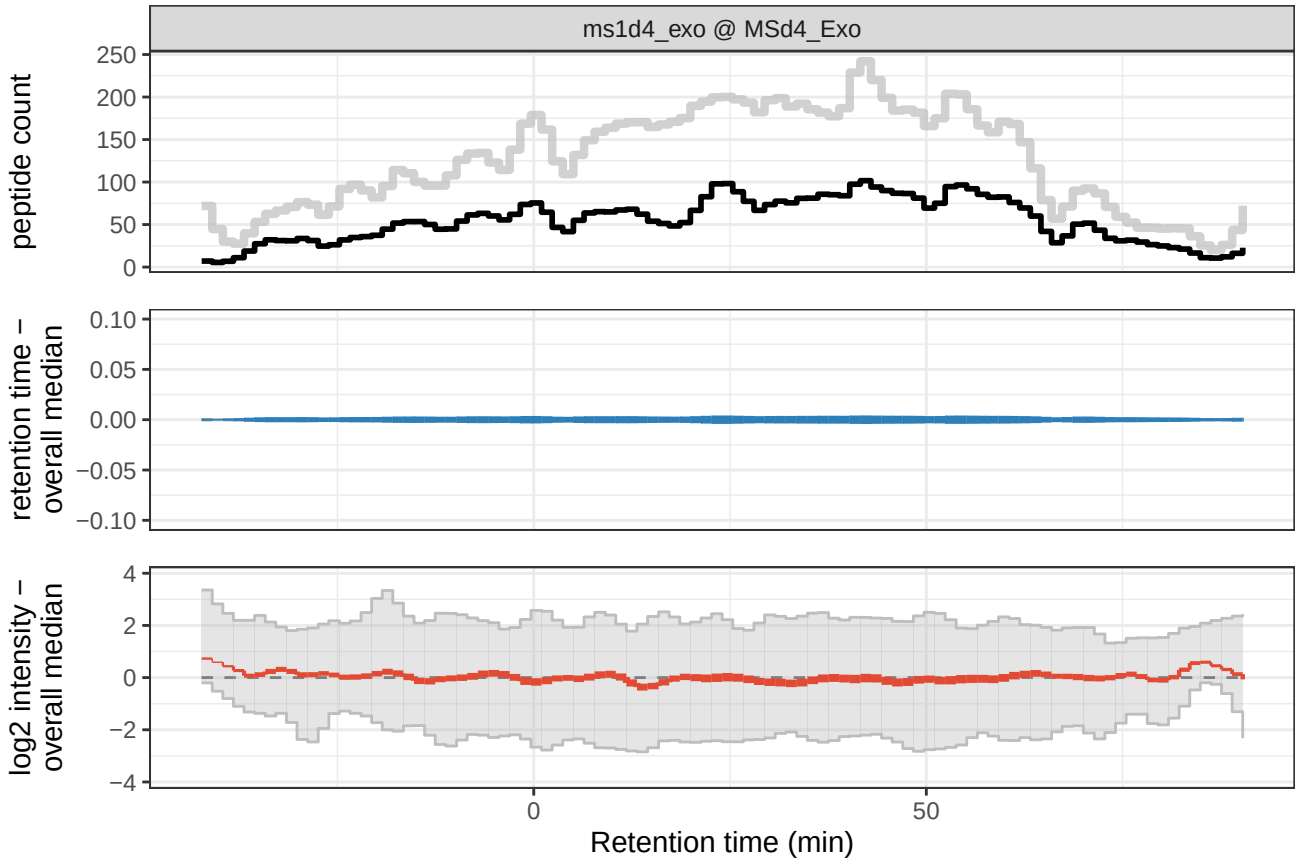

fw2_s @ FW_S
peptide count
200
100
retention time – overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity – overall median
4
2
0
−2
−4
0
50
Retention time (min)

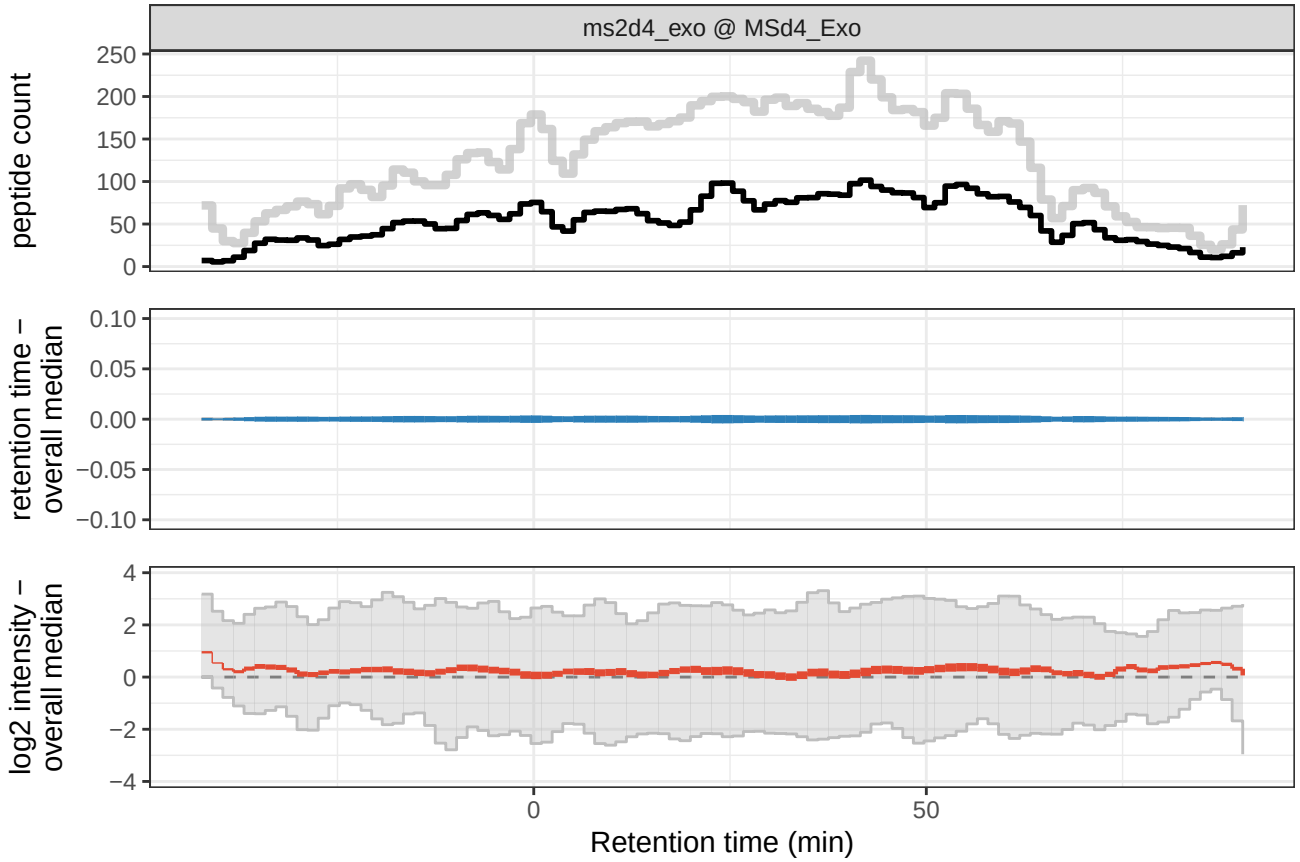

ms1_l @ MS_L
peptide count
500
400
300
200
100
0
retention time – overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity – overall median
4
2
0
−2
−4
0
50
Retention time (min)

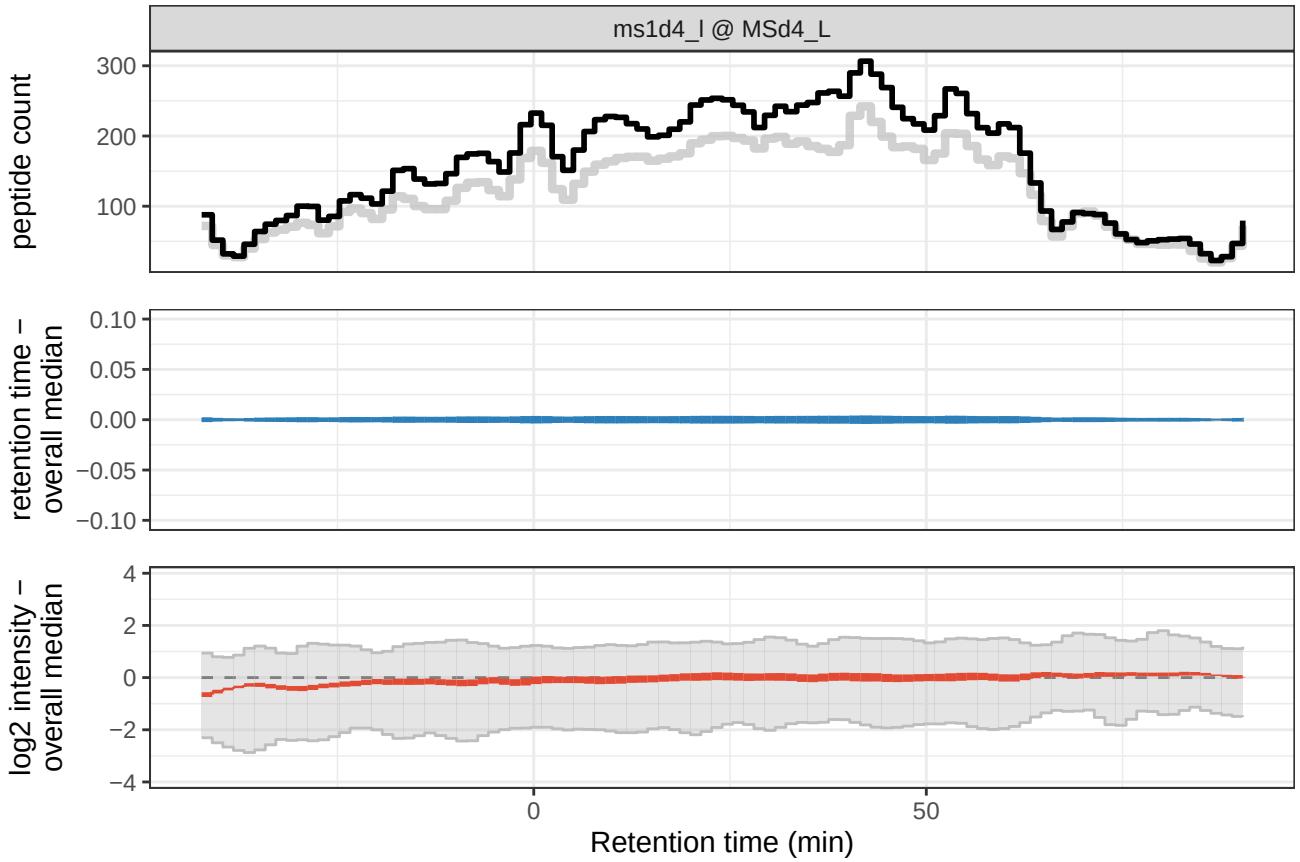

ms2_l @ MS_L
peptide count
500
400
300
200
100
0
retention time – overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity – overall median
4
2
0
−2
−4
0
50
Retention time (min)

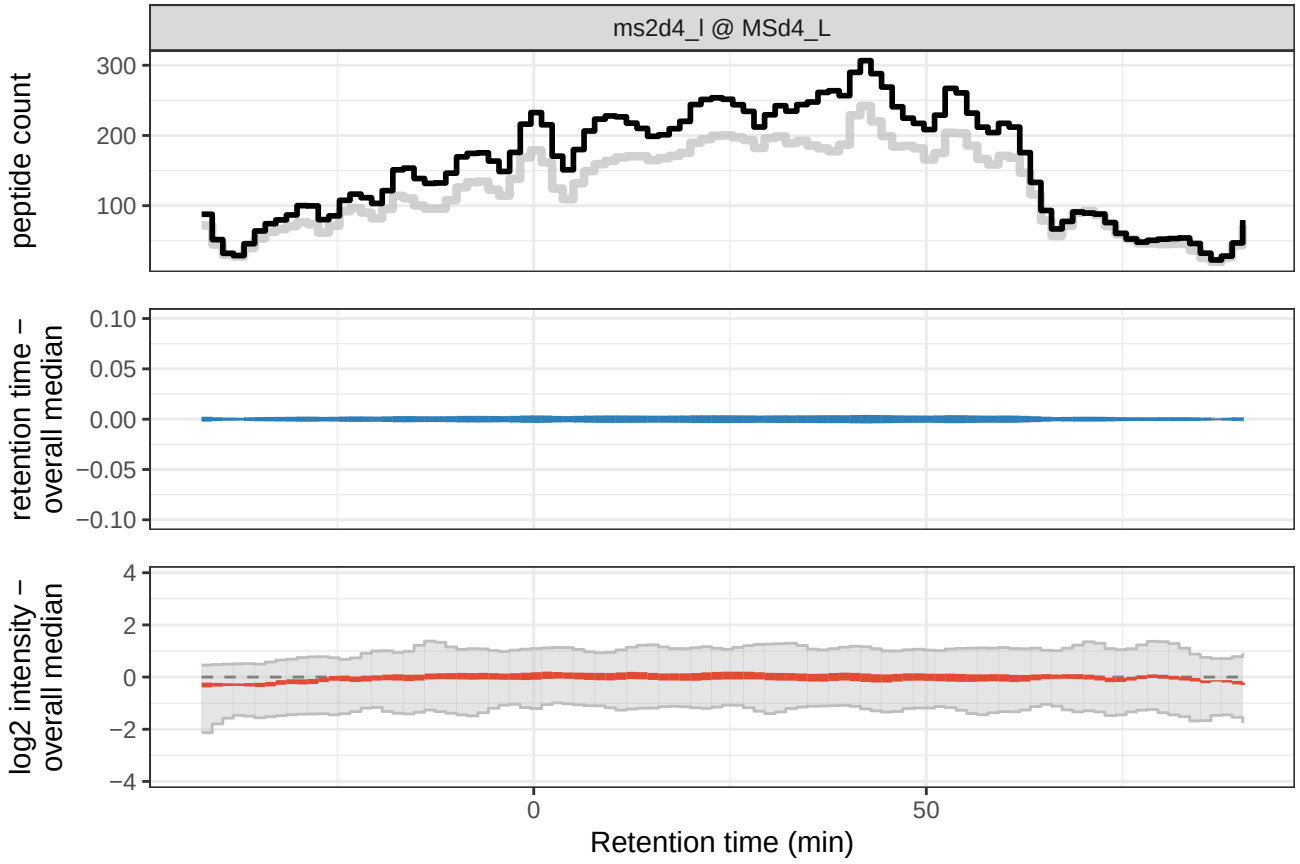

ms1_s @ MS_S
peptide count
250
200
150
100
50
retention time –
overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity –
overall median
4
2
0
−2
−4
0
50
Retention time (min)

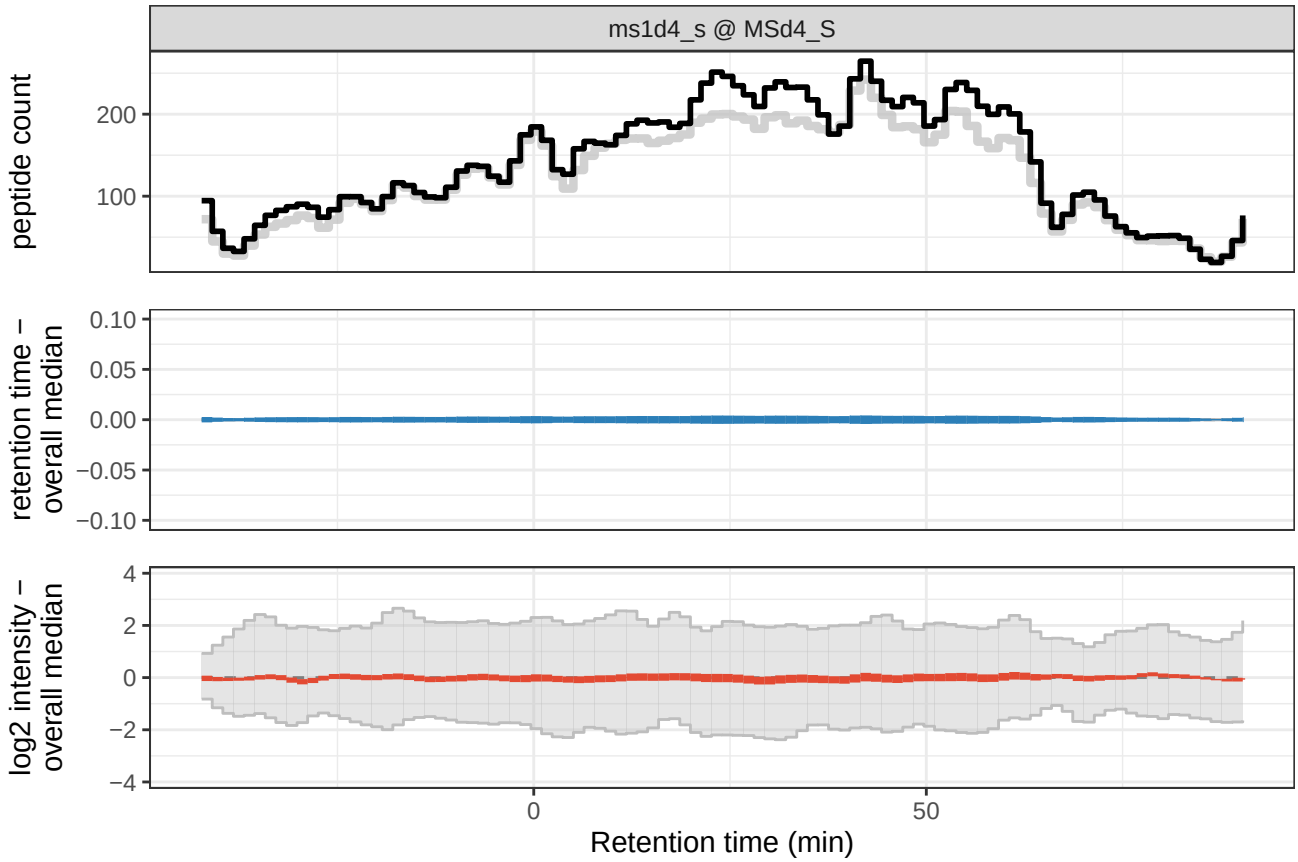

ms2_s @ MS_S
peptide count
250
200
150
100
50
retention time –
overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity –
overall median
4
2
0
−2
−4
0
50
Retention time (min)

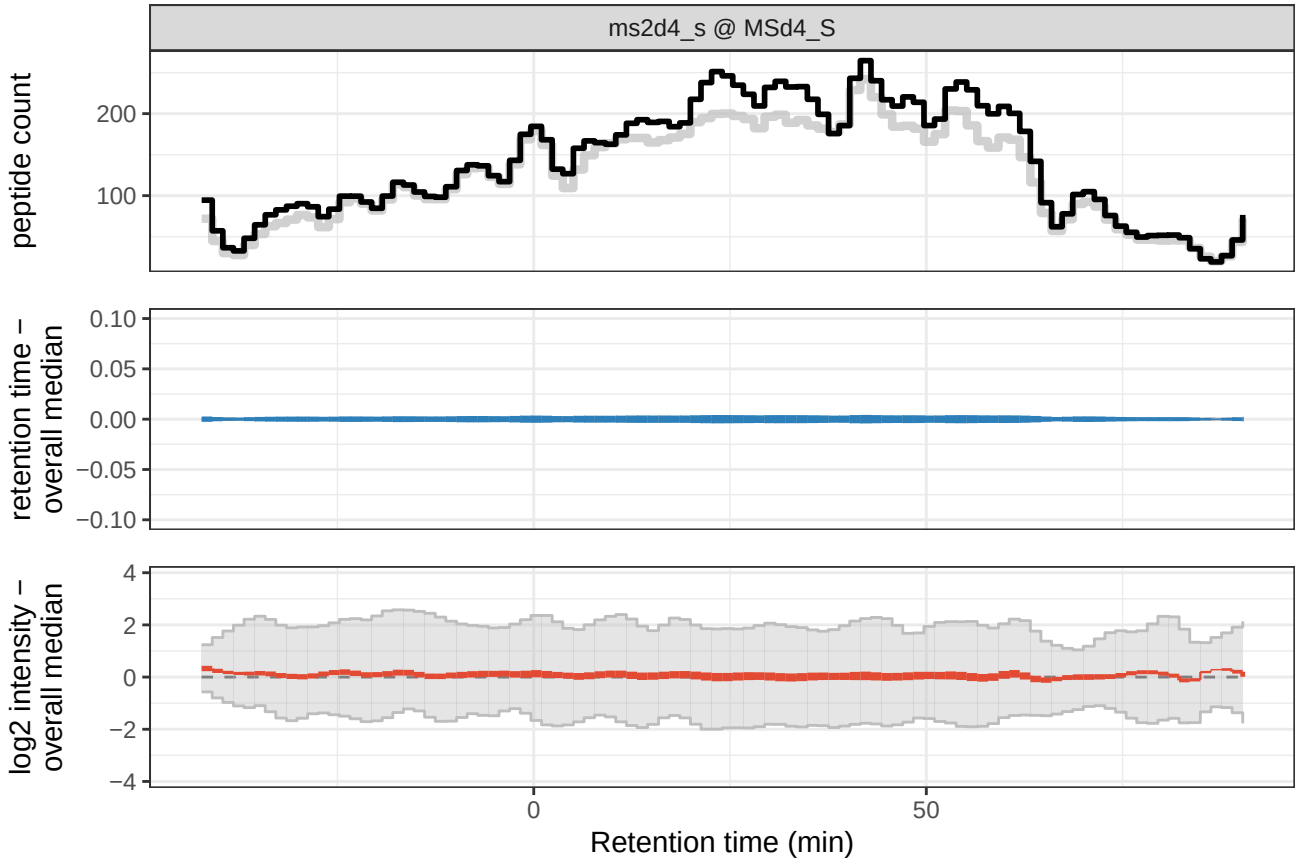

ms1d4_exo @ MSd4_Exo
peptide count
250
200
150
100
50
0
retention time – overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity – overall median
4
2
0
−2
−4
0
50
Retention time (min)

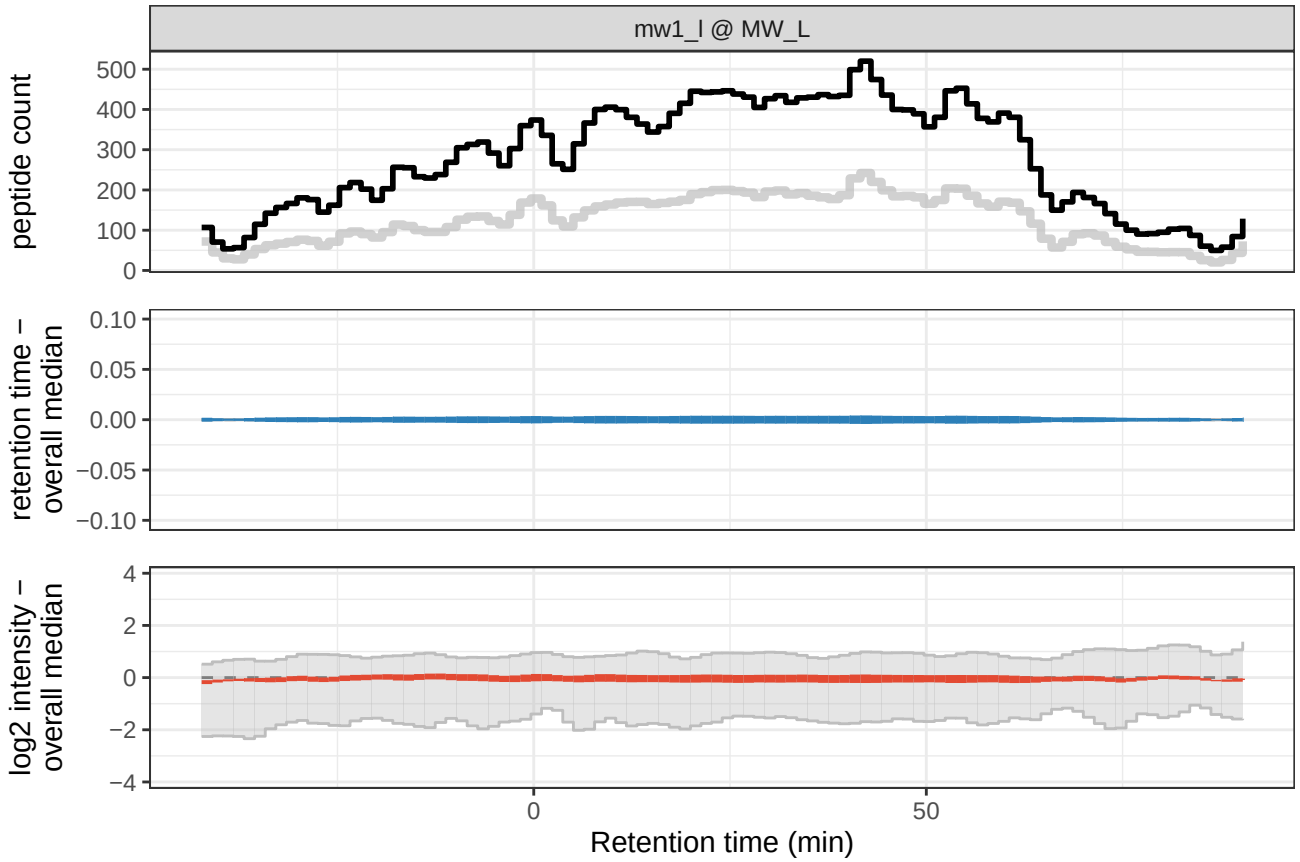

ms2d4_exo @ MSd4_Exo
peptide count
250
200
150
100
50
0
retention time – overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity – overall median
4
2
0
−2
−4
0
50
Retention time (min)

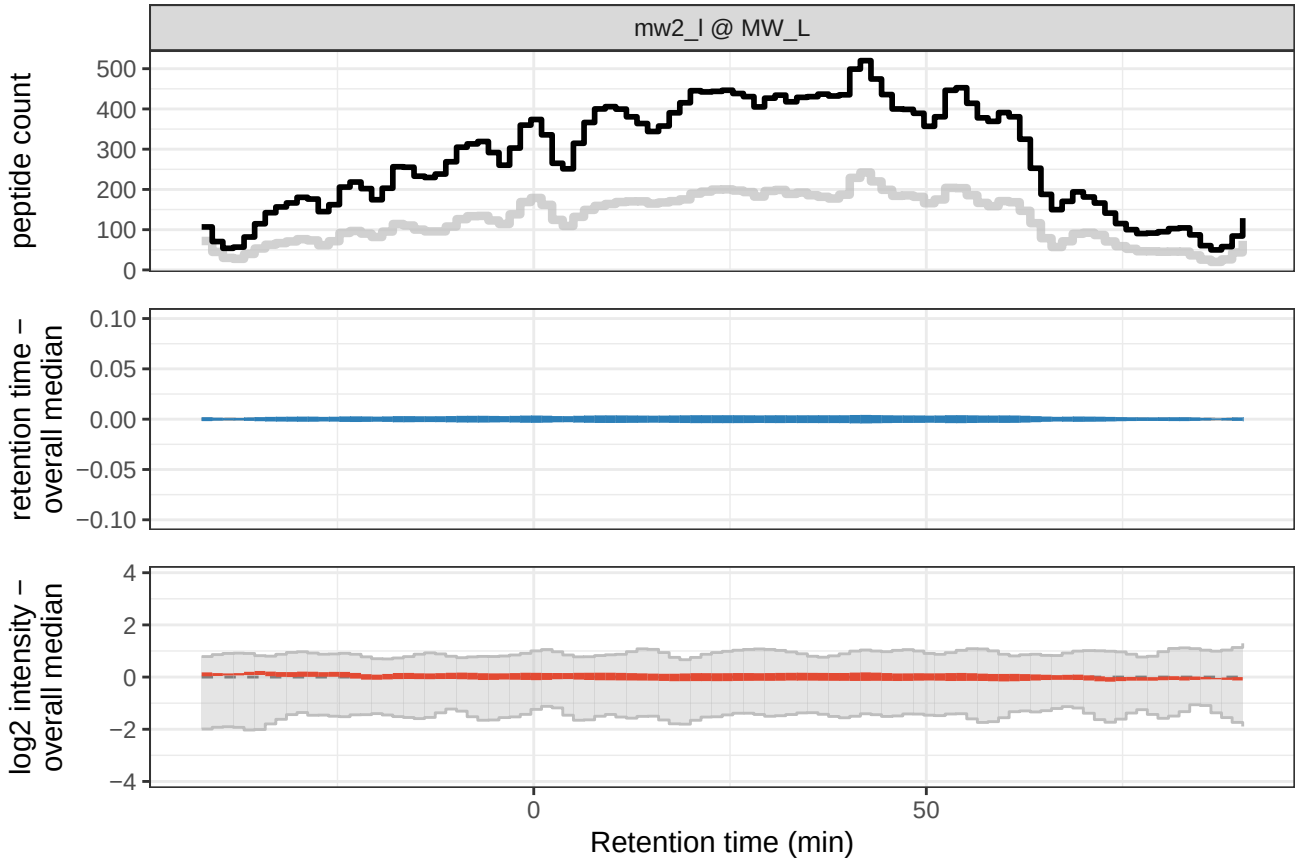

ms1d4_l @ MSd4_L
peptide count
300
200
100
retention time –
overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity –
overall median
4
2
0
−2
−4
0
50
Retention time (min)

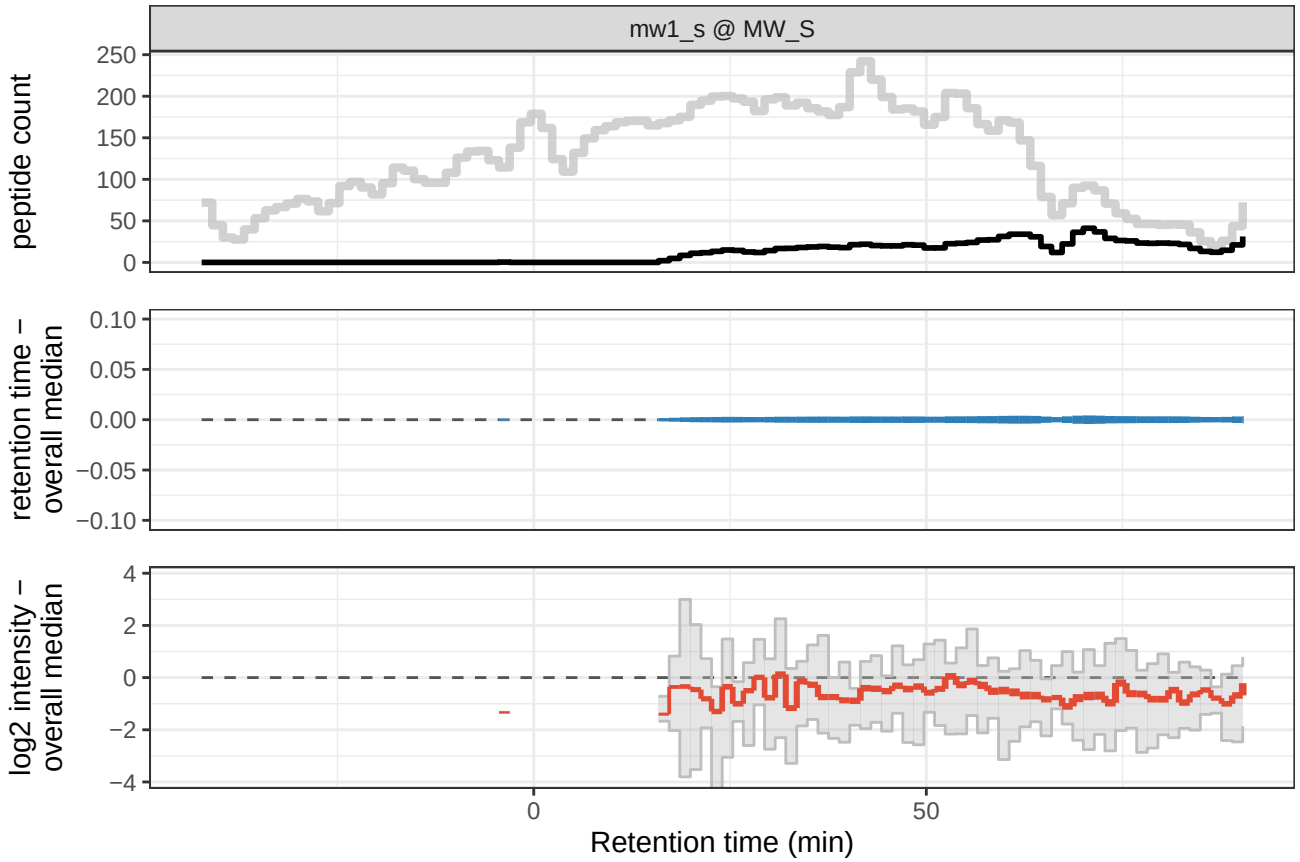

ms2d4_l @ MSd4_L
peptide count
300
200
100
retention time –
overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity –
overall median
4
2
0
−2
−4
0
50
Retention time (min)

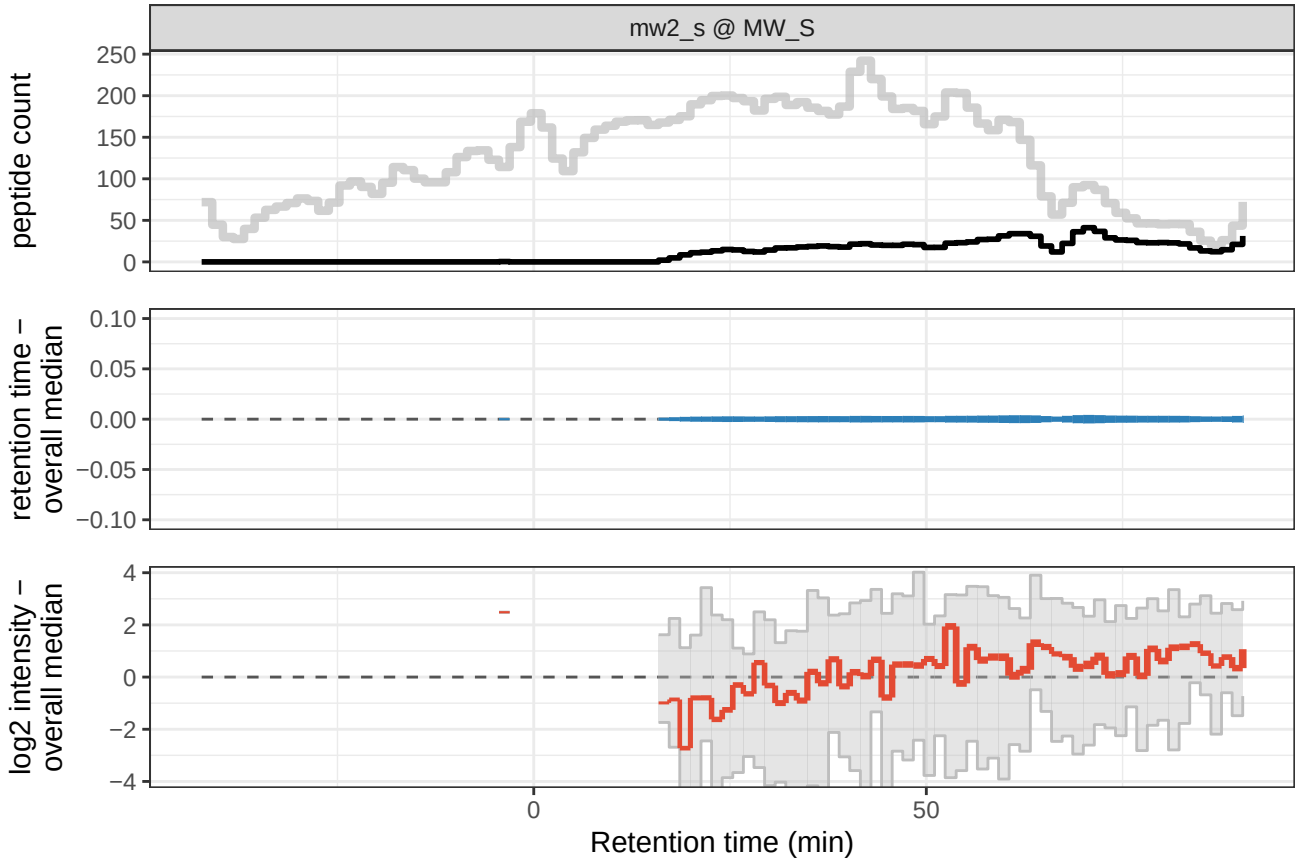

ms1d4_s @ MSd4_S
peptide count
200
100
retention time –
overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity –
overall median
4
2
0
−2
−4
0
50
Retention time (min)

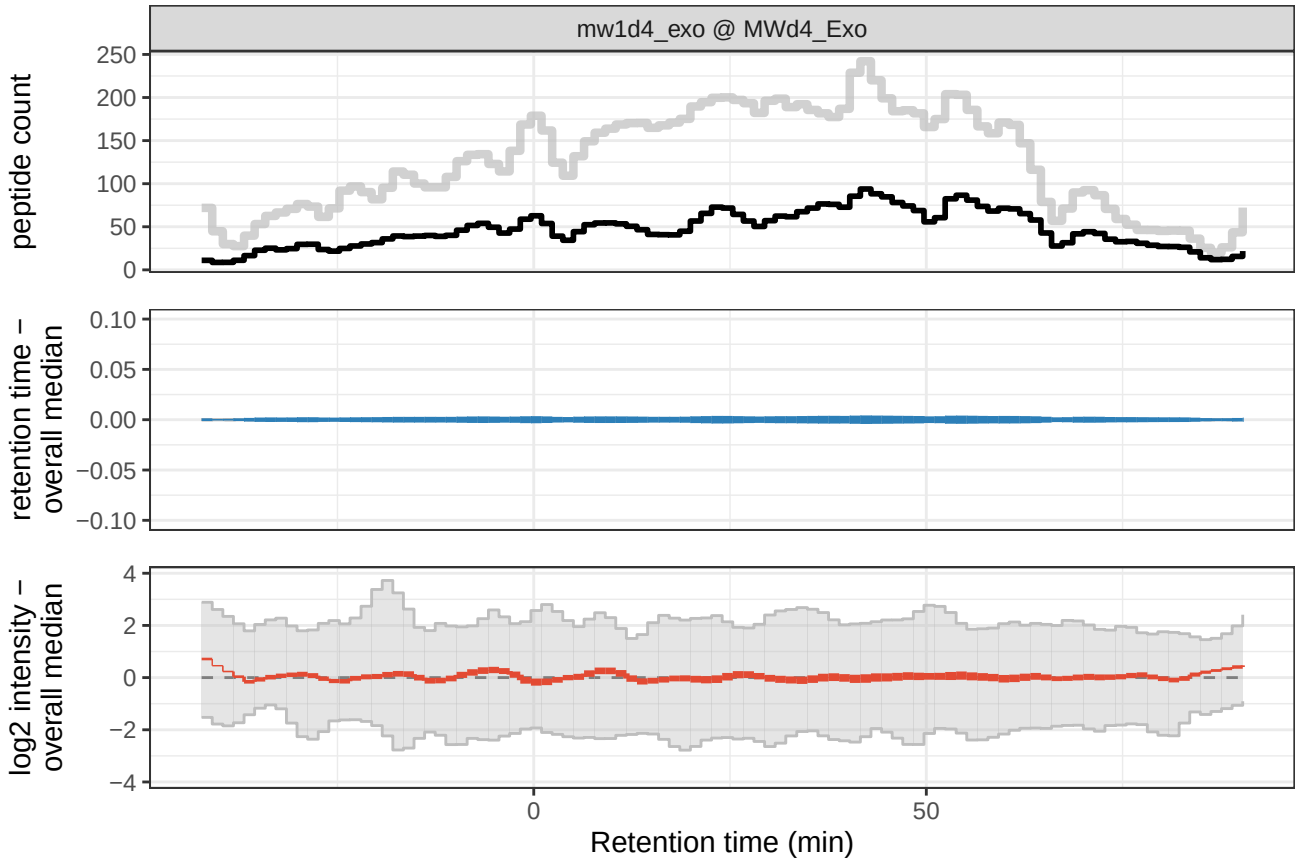

ms2d4_s @ MSd4_S
peptide count
200
100
retention time –
overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity –
overall median
4
2
0
−2
−4
0
50
Retention time (min)

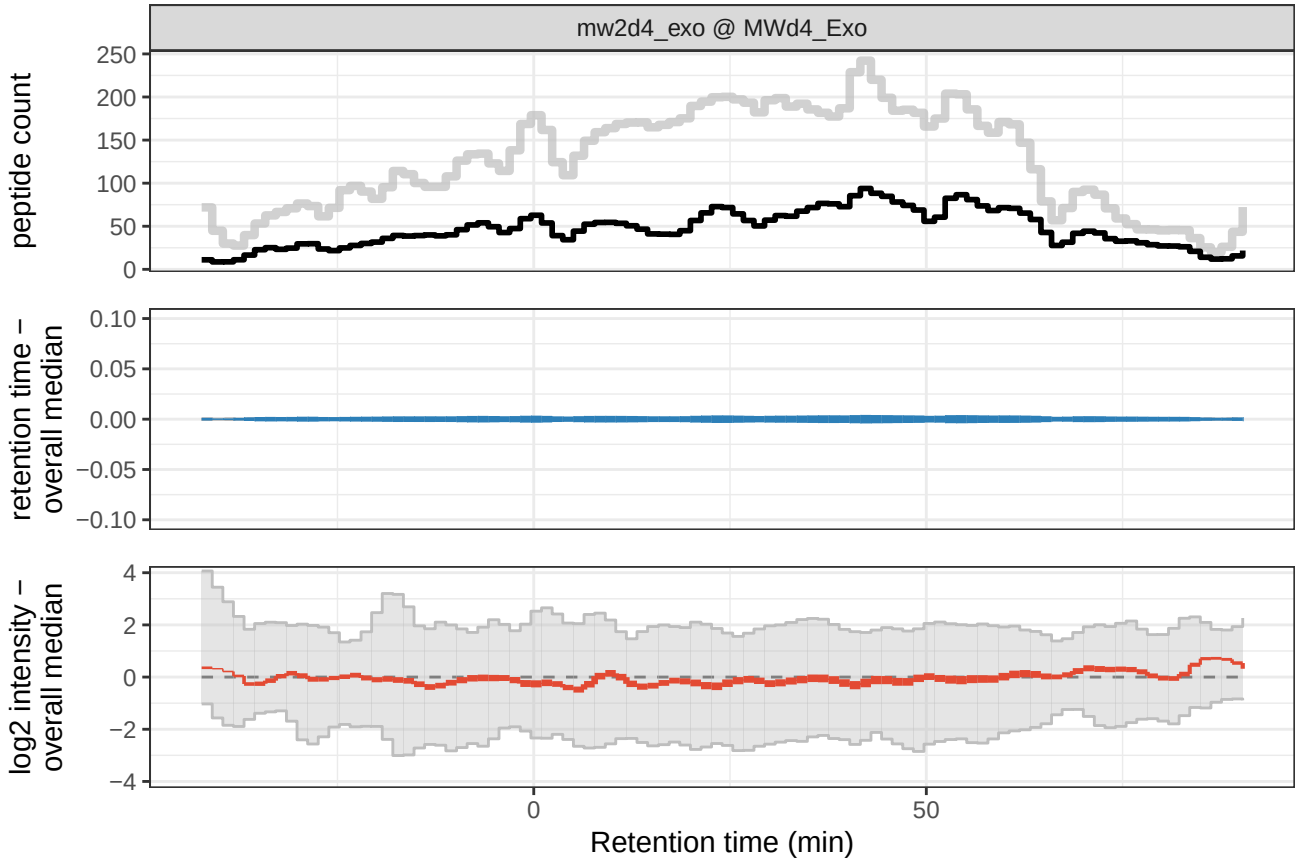

mw1_l @ MW_L
peptide count
500
400
300
200
100
0
retention time –
overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity –
overall median
4
2
0
−2
−4
0
50
Retention time (min)

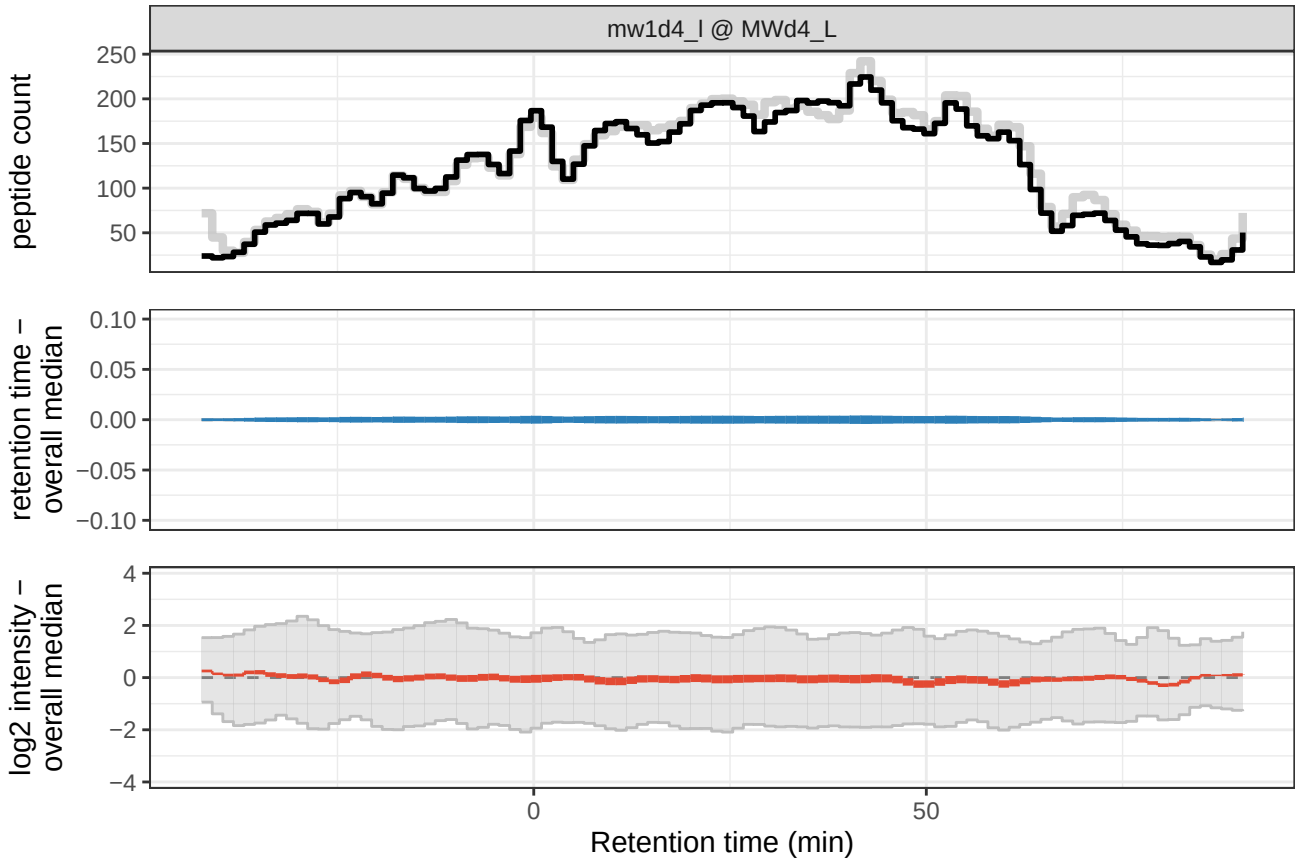

mw2_l @ MW_L
peptide count
500
400
300
200
100
0
retention time –
overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity –
overall median
4
2
0
−2
−4
0
50
Retention time (min)

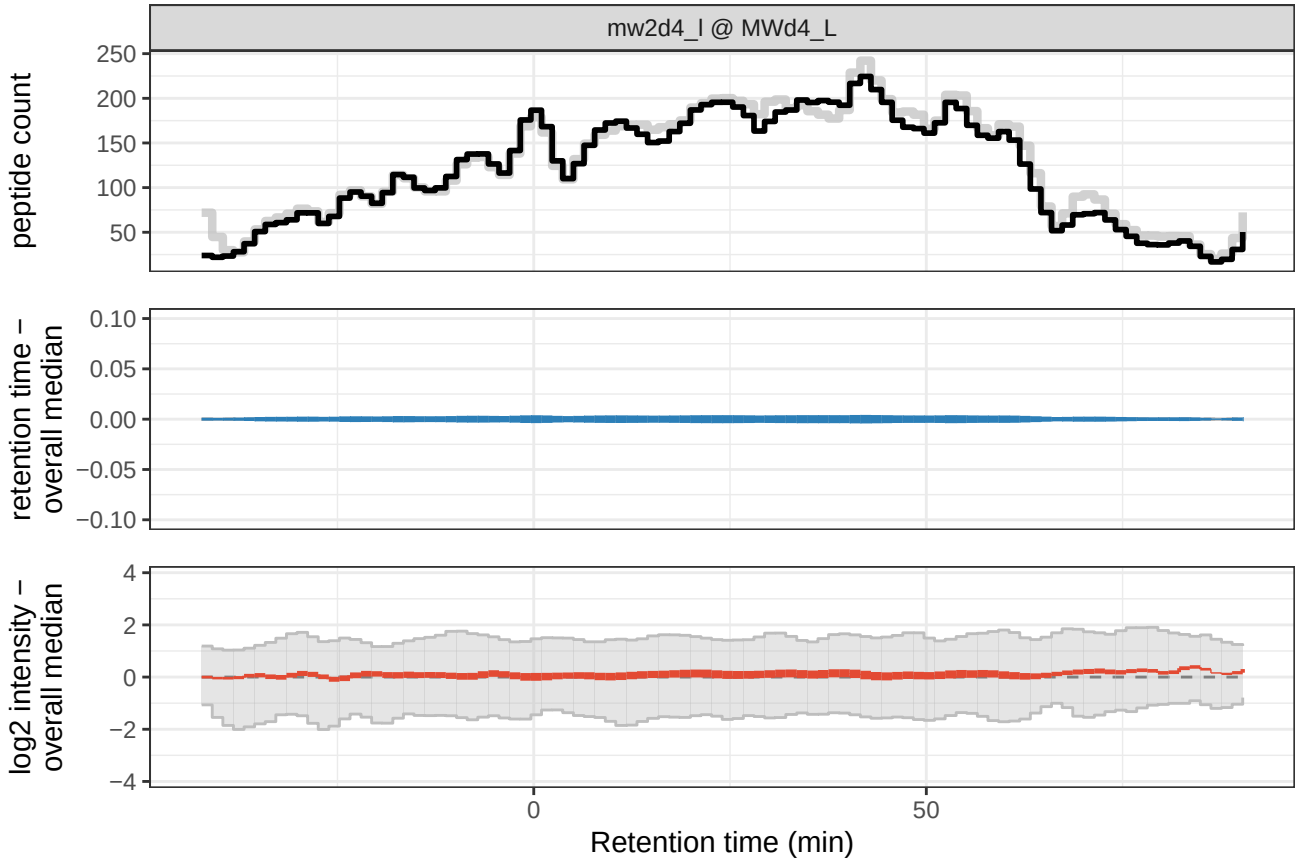

mw1_s @ MW_S
peptide count
250
200
150
100
50
0
retention time – overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity – overall median
4
2
0
−2
−4
0
50
Retention time (min)

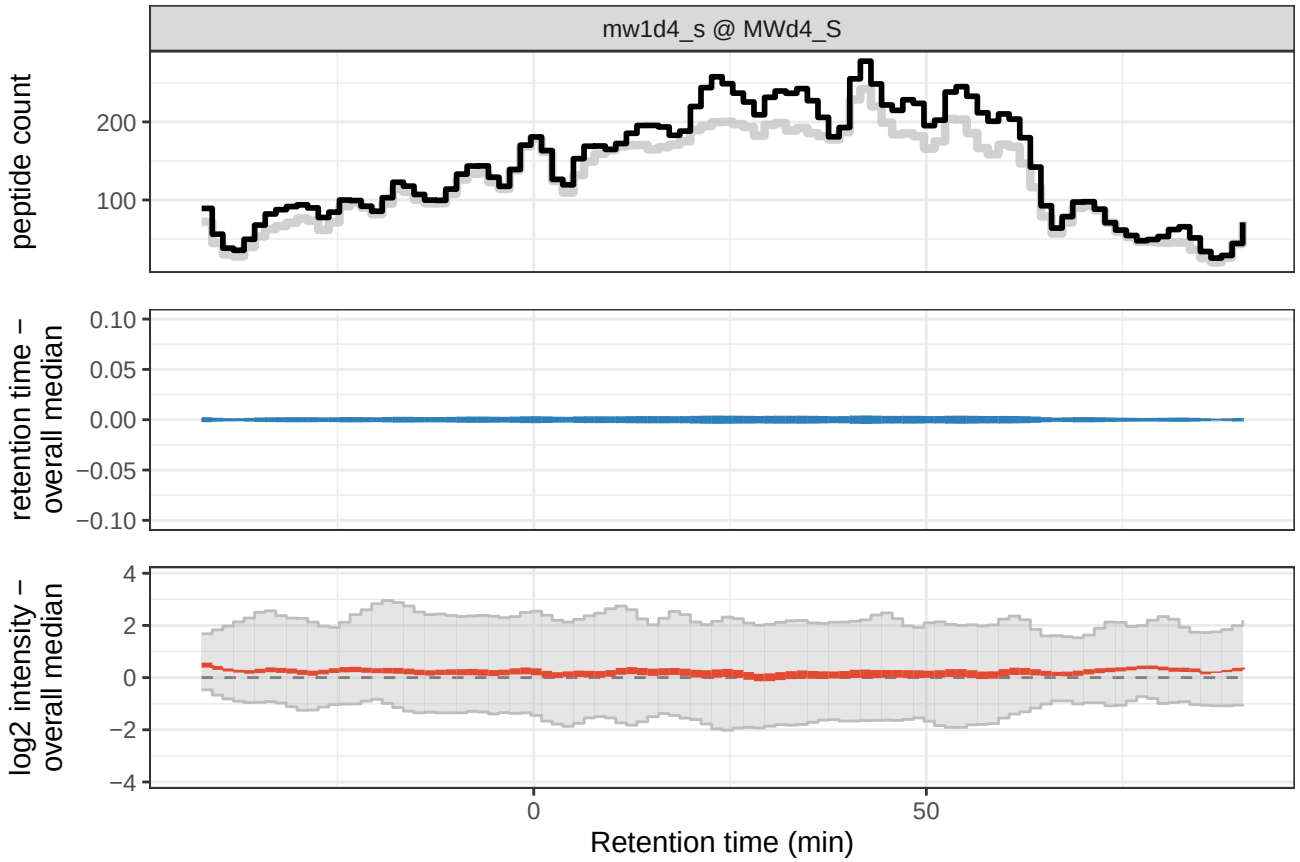

mw2_s @ MW_S
peptide count
250
200
150
100
50
0
retention time – overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity – overall median
4
2
0
−2
−4
0
50
Retention time (min)

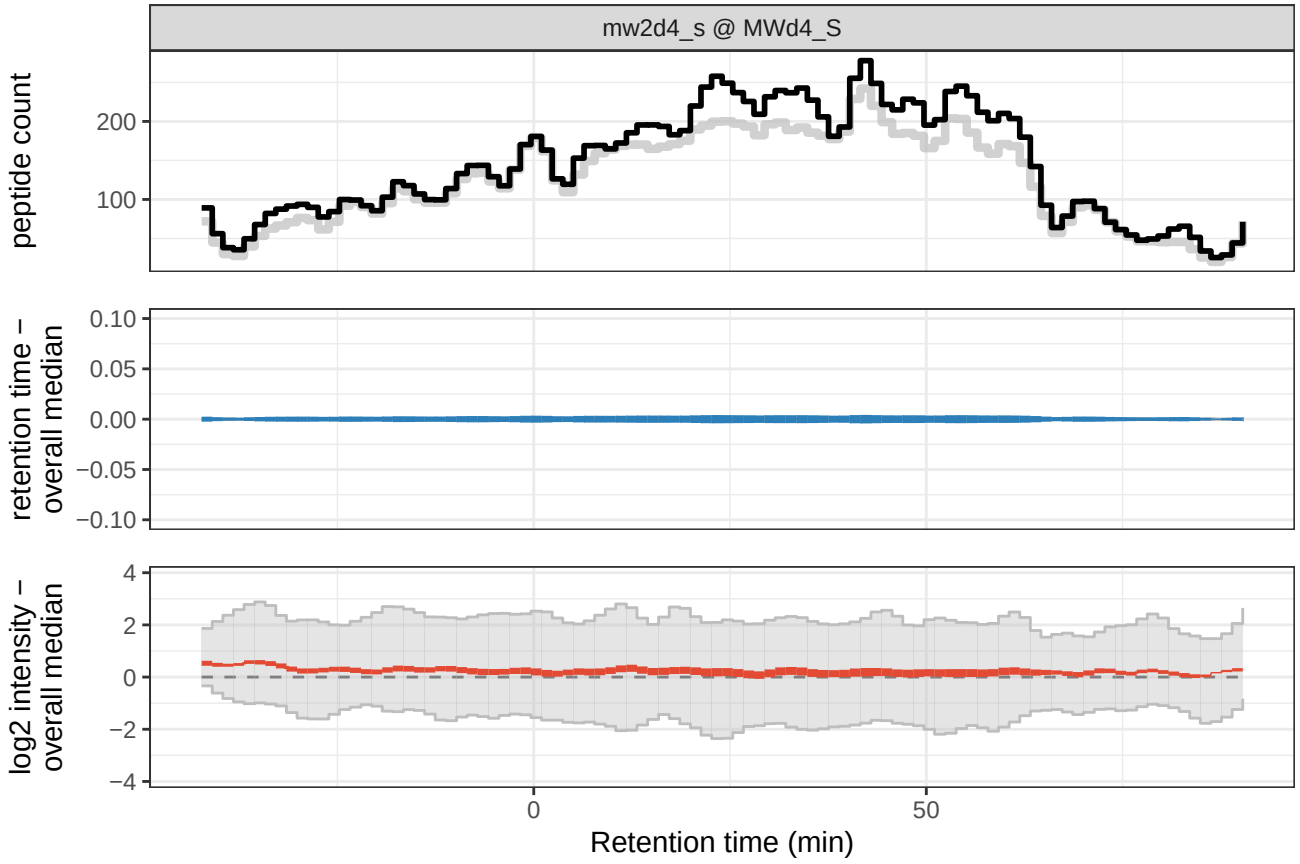

mw1d4_exo @ MWd4_Exo
peptide count
250
200
150
100
50
0
retention time – overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity – overall median
4
2
0
−2
−4
0
50
Retention time (min)


mw2d4_exo @ MWd4_Exo
peptide count
250
200
150
100
50
0
retention time – overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity – overall median
4
2
0
−2
−4
0
50
Retention time (min)


mw1d4_l @ MWd4_L
peptide count
250
200
150
100
50
retention time –
overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity –
overall median
4
2
0
−2
−4
0
50
Retention time (min)


mw2d4_l @ MWd4_L
peptide count
250
200
150
100
50
retention time –
overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity –
overall median
4
2
0
−2
−4
0
50
Retention time (min)


mw1d4_s @ MWd4_S
peptide count
200
100
retention time –
overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity –
overall median
4
2
0
−2
−4
0
50
Retention time (min)


mw2d4_s @ MWd4_S
peptide count
200
100
retention time –
overall median
0.10
0.05
0.00
−0.05
−0.10
log2 intensity –
overall median
4
2
0
−2
−4
0
50
Retention time (min)

1.5 variation among replicates

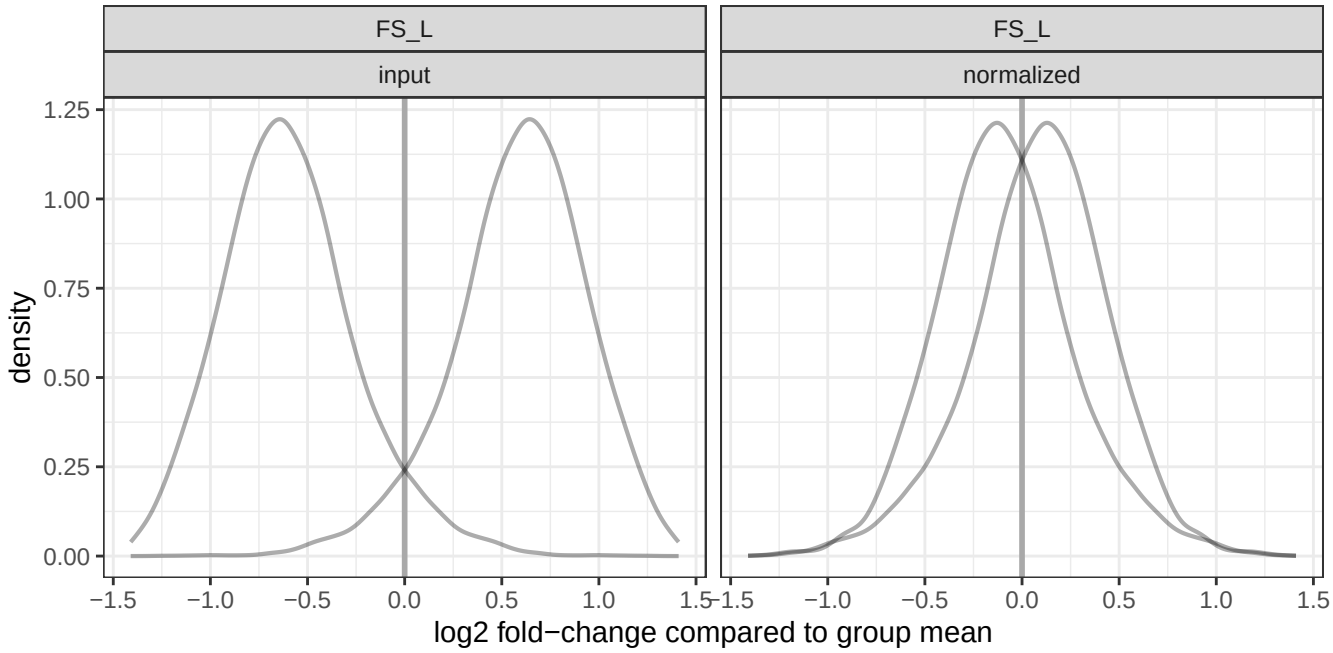
The reproducibility of replicate measurements is expressed in three different analyses. First, the difference between peptide intensities in each sample are compared to the mean value among all replicates (foldchange distributions). Next, the Coefficient of Variation (CoV) is used as a metric for reproducibility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

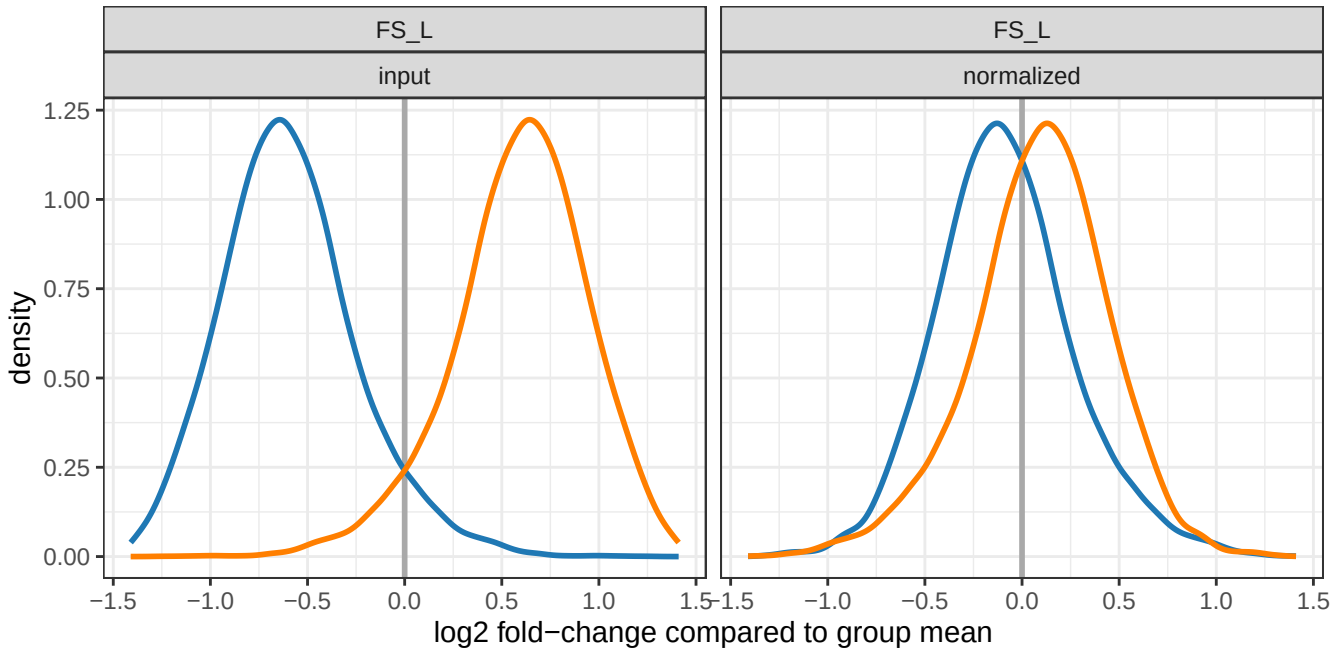
1.5.1 within-group foldchange distributions

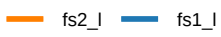
The foldchange of all peptides in a sample is compared to their respective mean value over all samples in the group. This visualizes how strongly each sample deviates from other samples in the same group which helps identify outlier samples. The same data was used as detailed in the "retention time"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

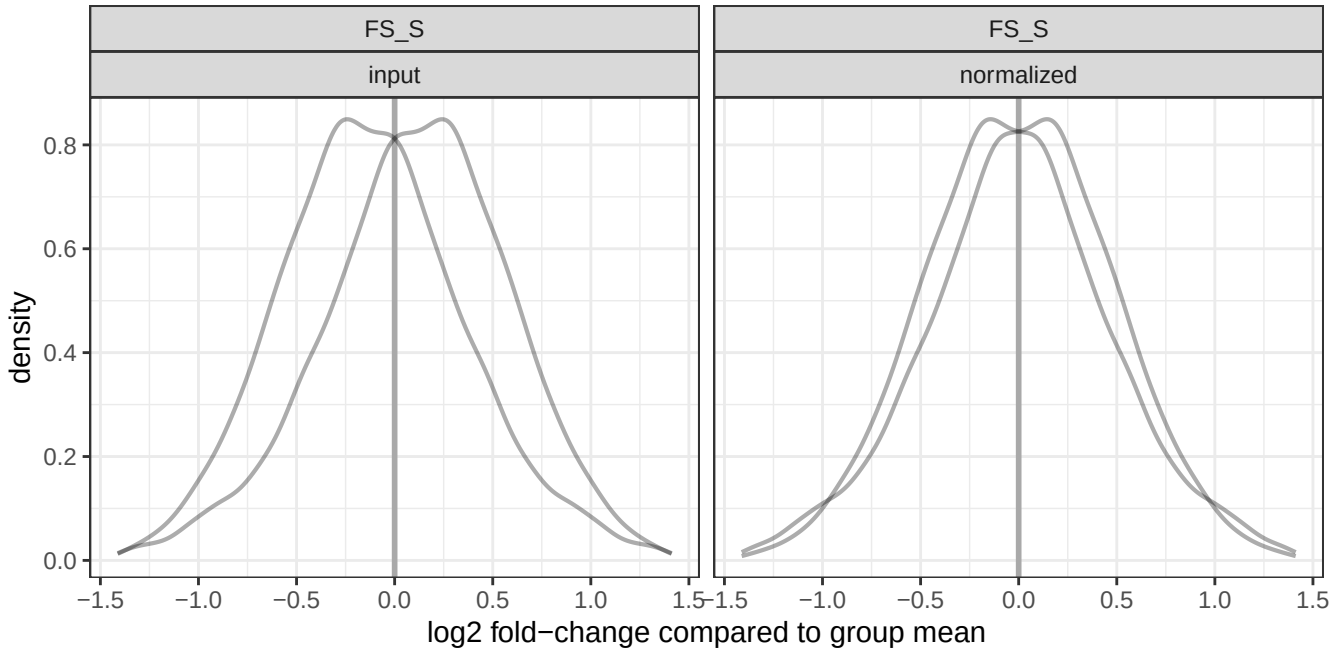
For each sample group, two plots are shown: 1) a basic monochrome plot and 2) a variant that color-codes the top10 'worst' samples based on the standard deviation (sd) of their respective distributions. Note; per sample, the 0.5% quantiles of both top- and bottom-most outliers are disregarded in sd computation. 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.

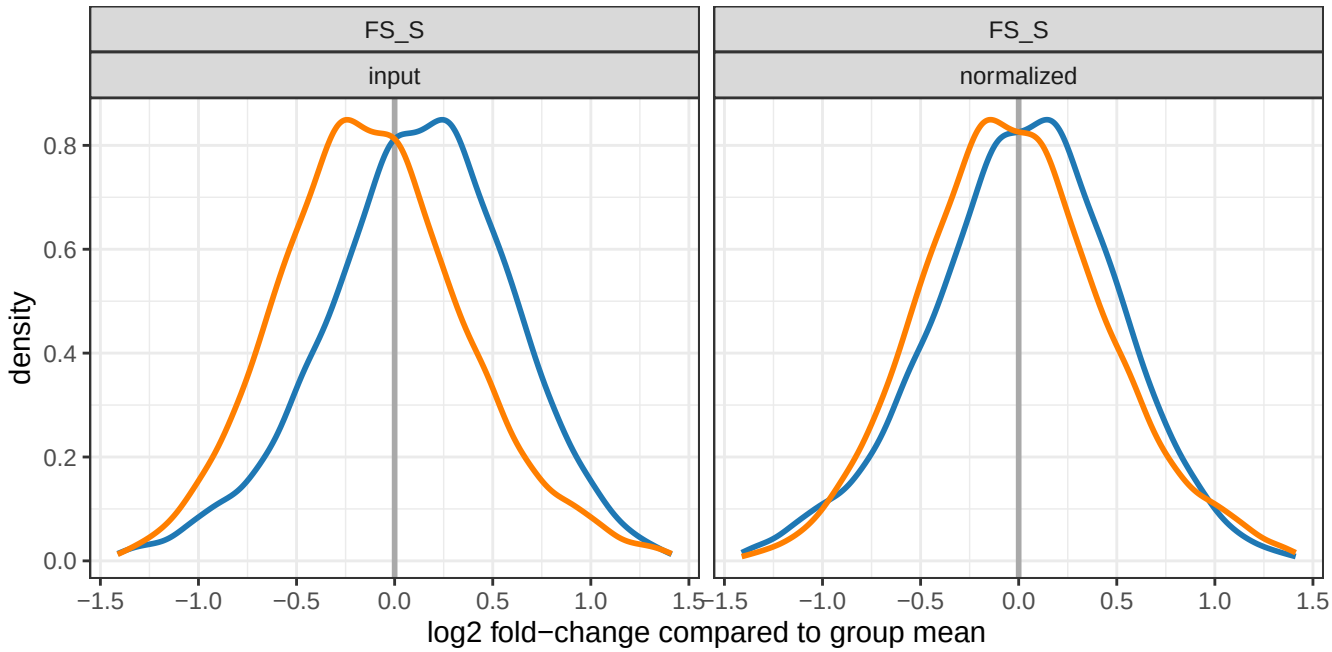
The 'input' panel is based on the peptide intensities as-is (i.e. the user-provided input data from upstream software), the 'normalized' panel shows the exact same samples and peptides after normalization (as specified by user). Samples marked as 'exclude' in the provided sample metadata table are visualized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


FS_L
input
FS_L
normalized
density
1.25
1.00
0.75
0.50
0.25
0.00
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
log2 fold−change compared to group mean

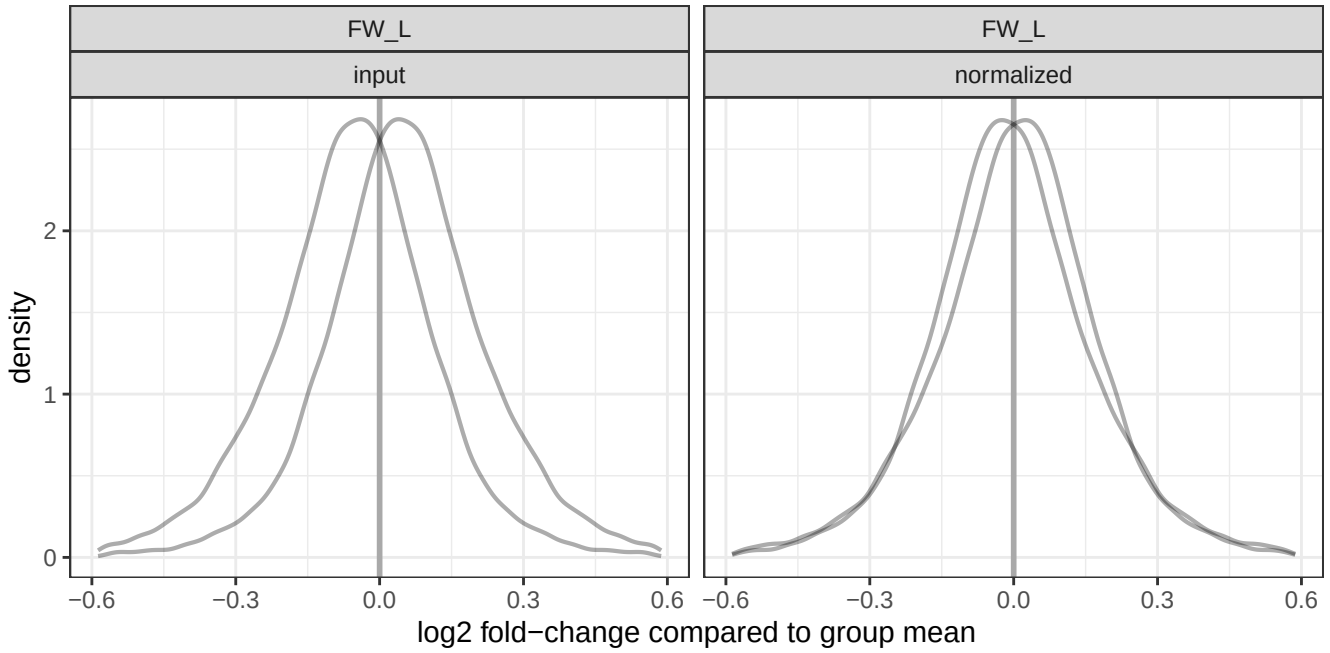

FS_L
input
FS_L
normalized
density
1.25
1.00
0.75
0.50
0.25
0.00
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
log2 fold−change compared to group mean

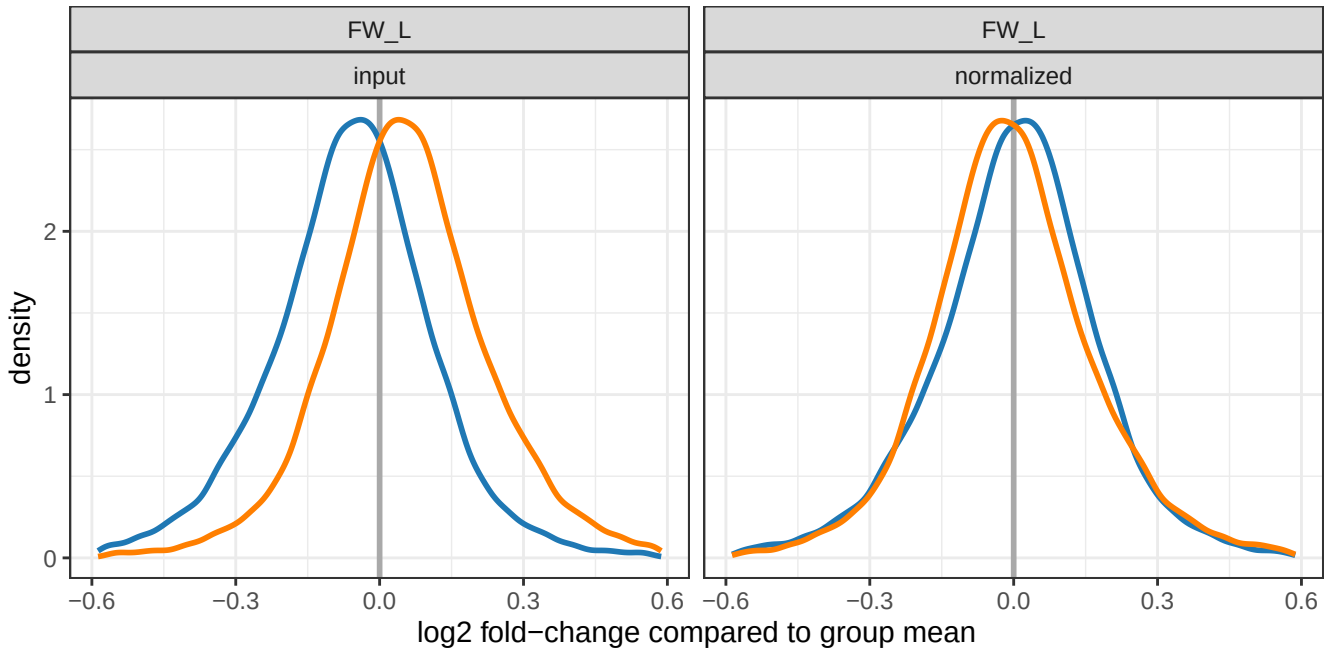

fs2_l
fs1_l

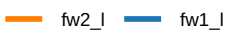

FS_S
input
FS_S
normalized
density
0.0
0.2
0.4
0.6
0.8
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
log2 fold−change compared to group mean

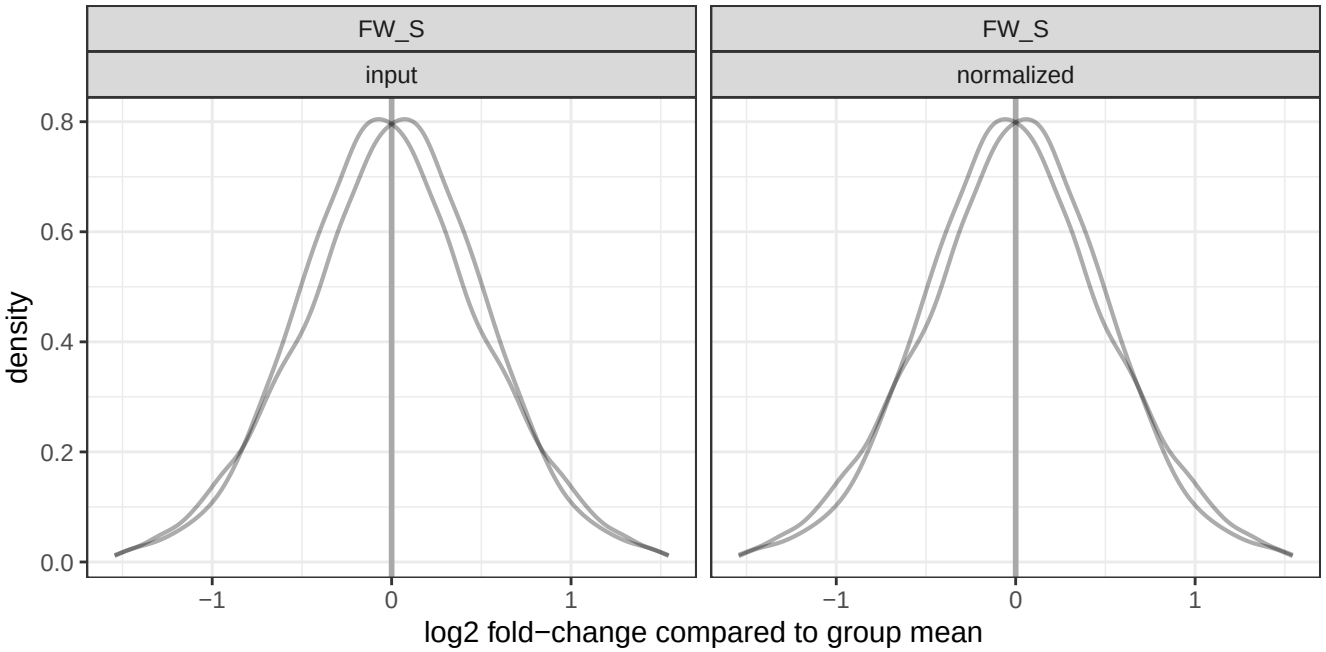

FS_S
input
FS_S
normalized
density
0.0
0.2
0.4
0.6
0.8
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
log2 fold−change compared to group mean

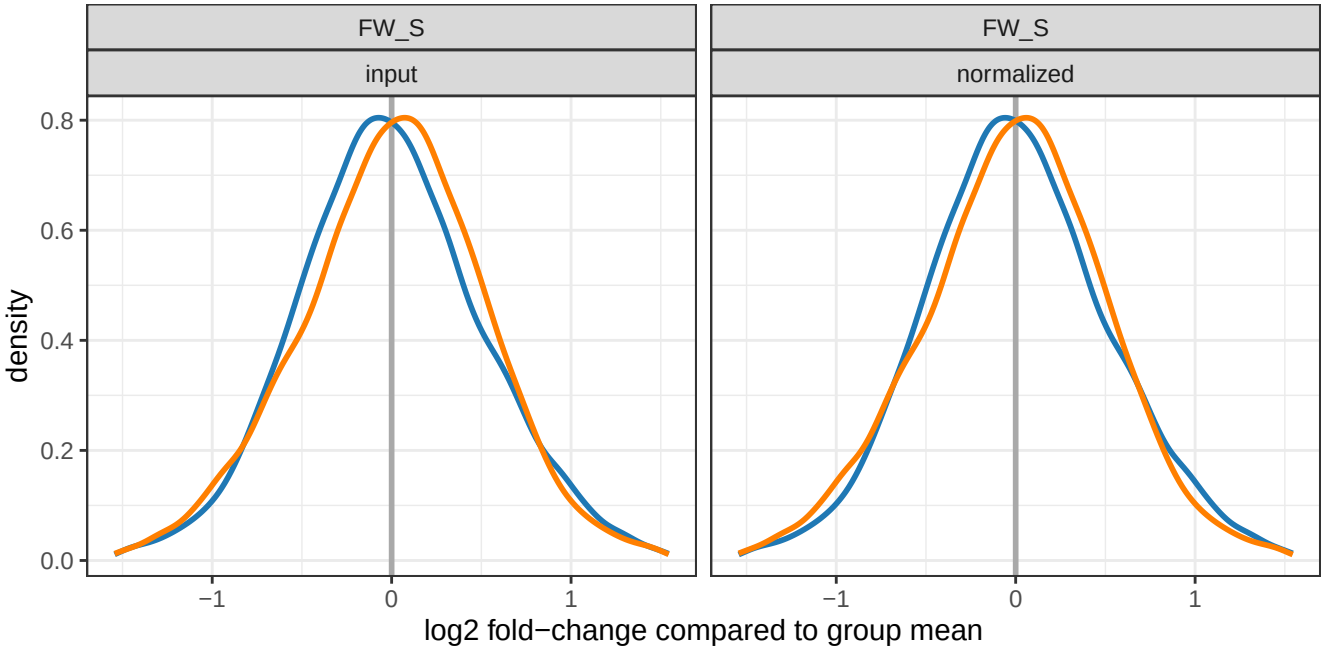

fs2_s
fs1_s


FW_L
input
FW_L
normalized
density
2
1
0
−0.6
−0.3
0.0
0.3
0.6
log2 fold−change compared to group mean

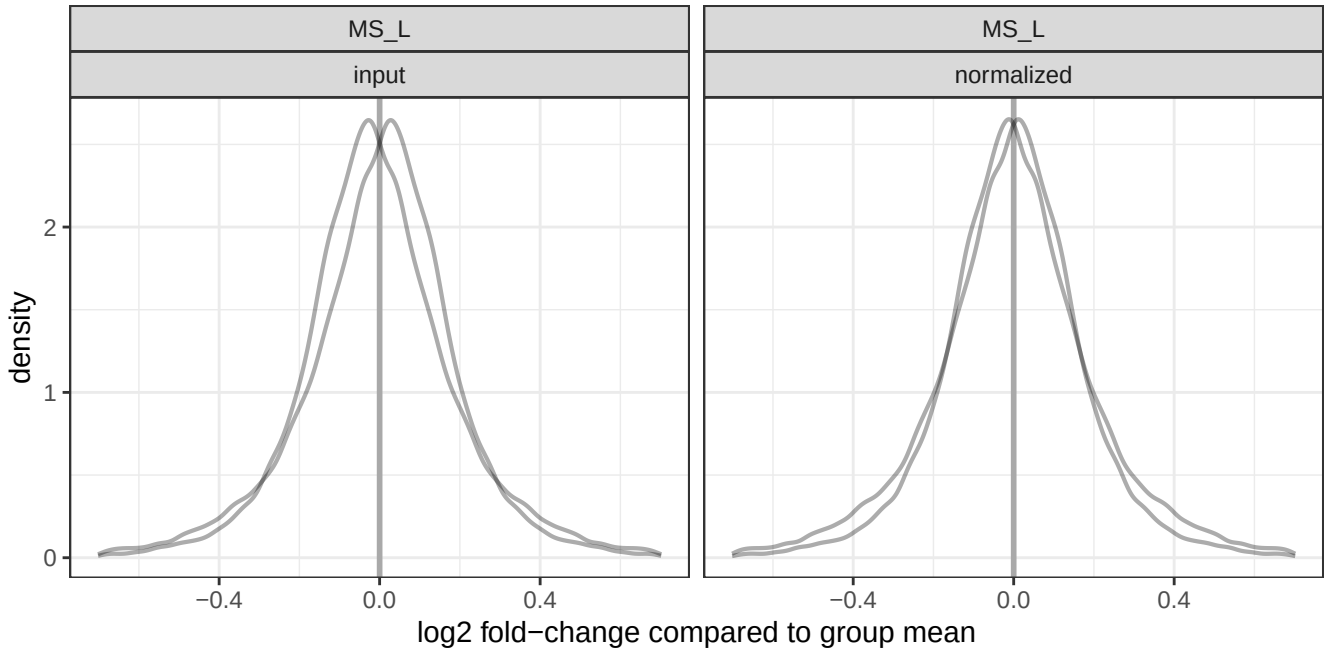

FW_L
input
FW_L
normalized
density
2
1
0
−0.6
−0.3
0.0
0.3
0.6
log2 fold−change compared to group mean

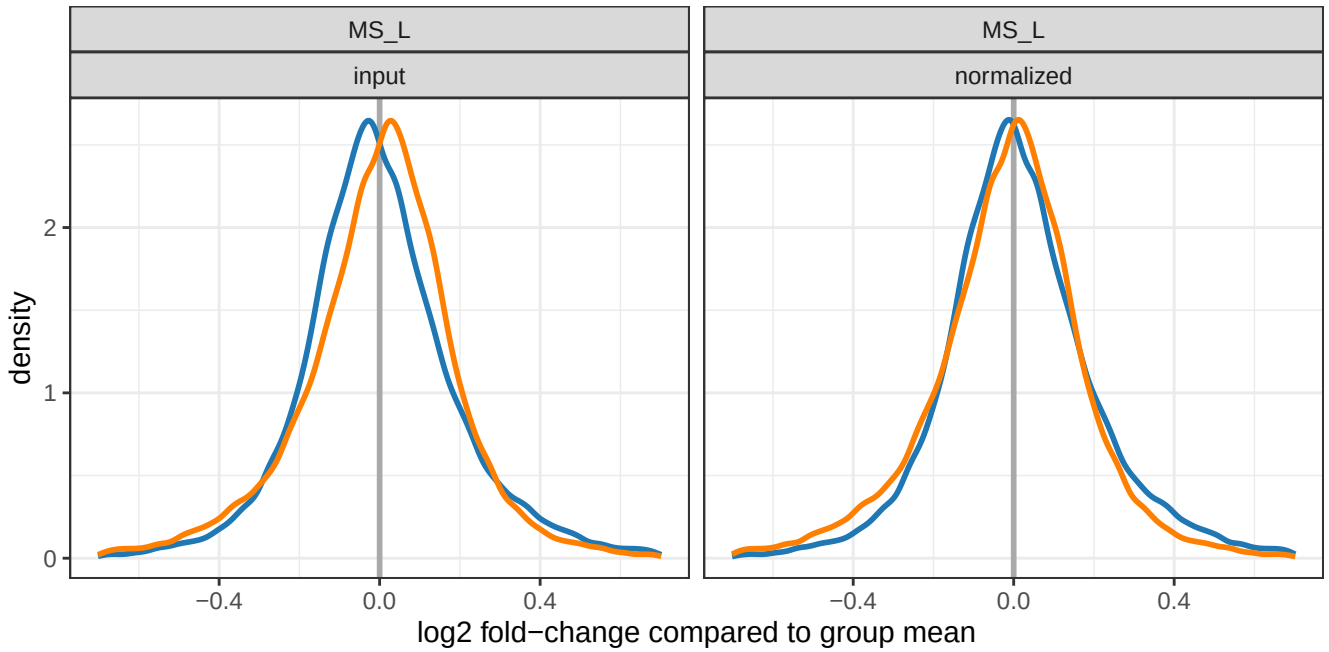

fw2_l
fw1_l

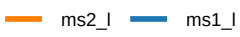

FW_S
input
FW_S
normalized
density
0.0
0.2
0.4
0.6
0.8
−1
0
1
log2 fold−change compared to group mean

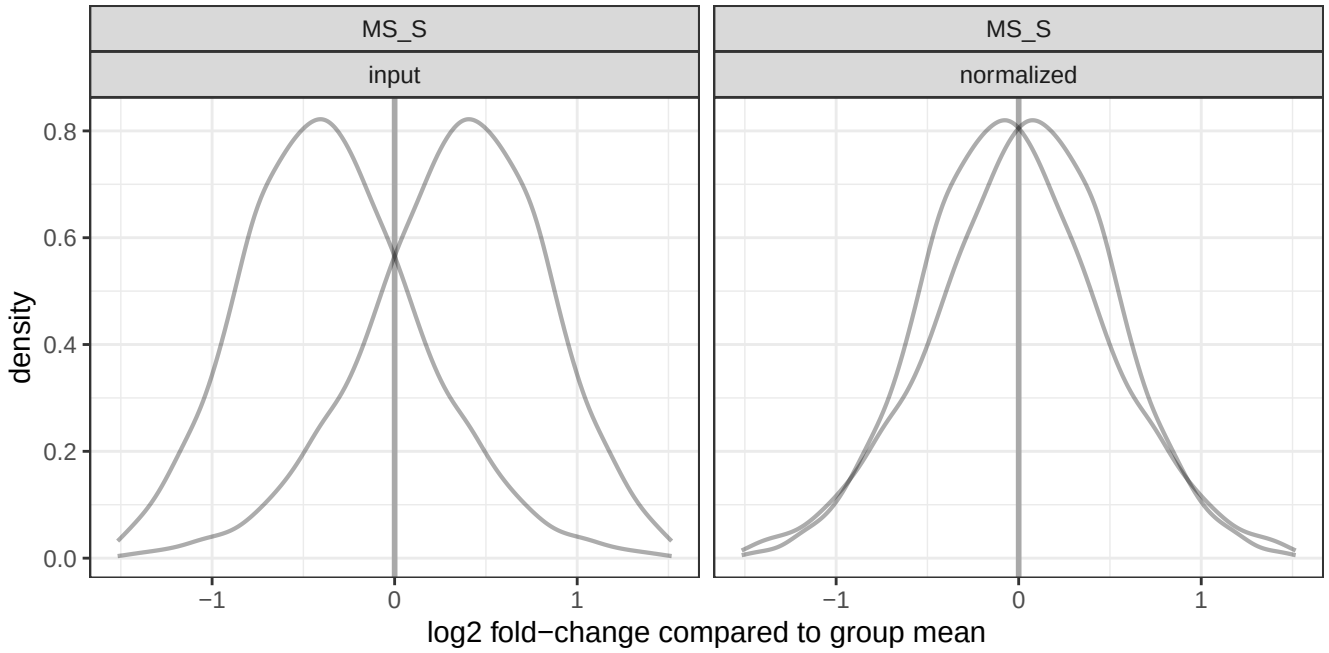

FW_S
input
FW_S
normalized
density
0.0
0.2
0.4
0.6
0.8
−1
0
1
log2 fold−change compared to group mean

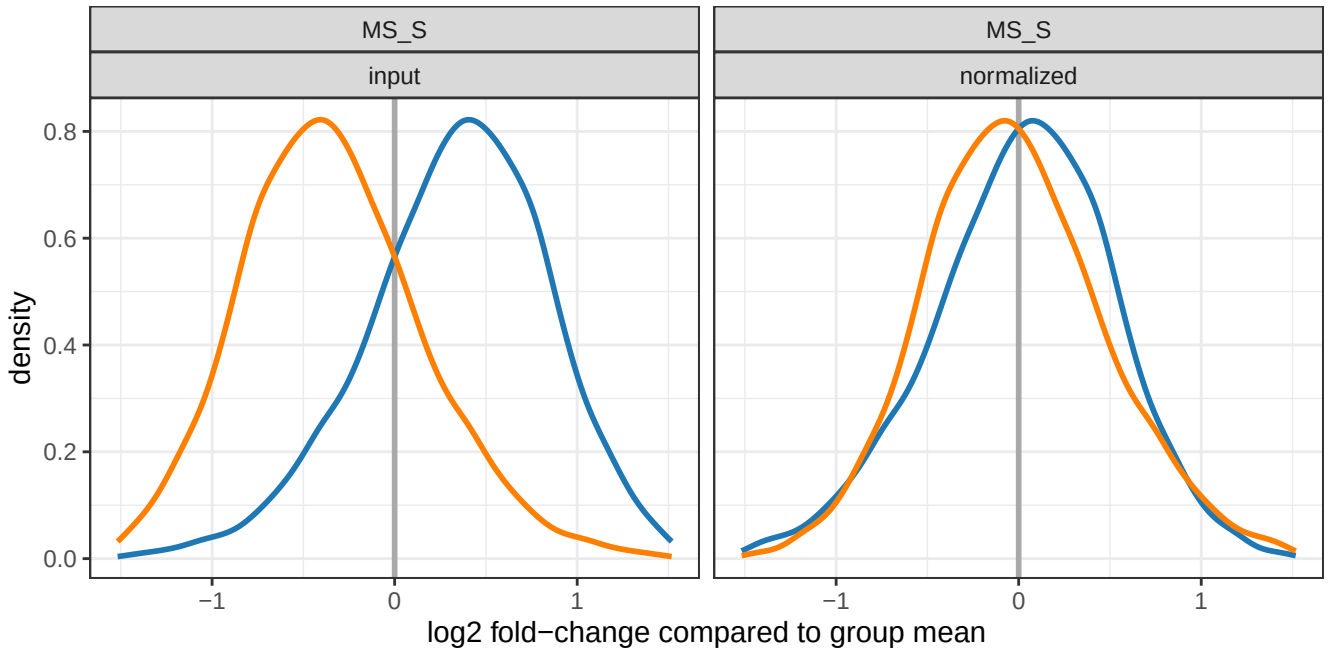

fw2_s
fw1_s


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

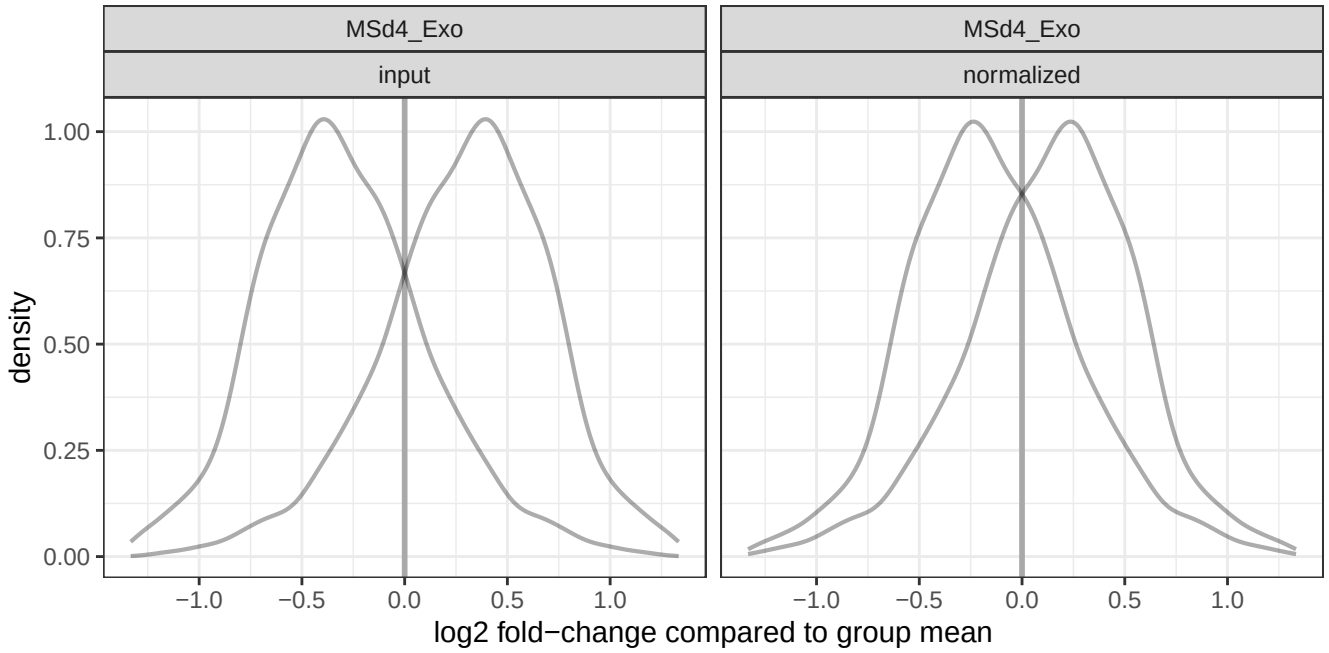

MS_L
input
MS_L
normalized
density
2
1
0
−0.4
0.0
0.4
log2 fold−change compared to group mean

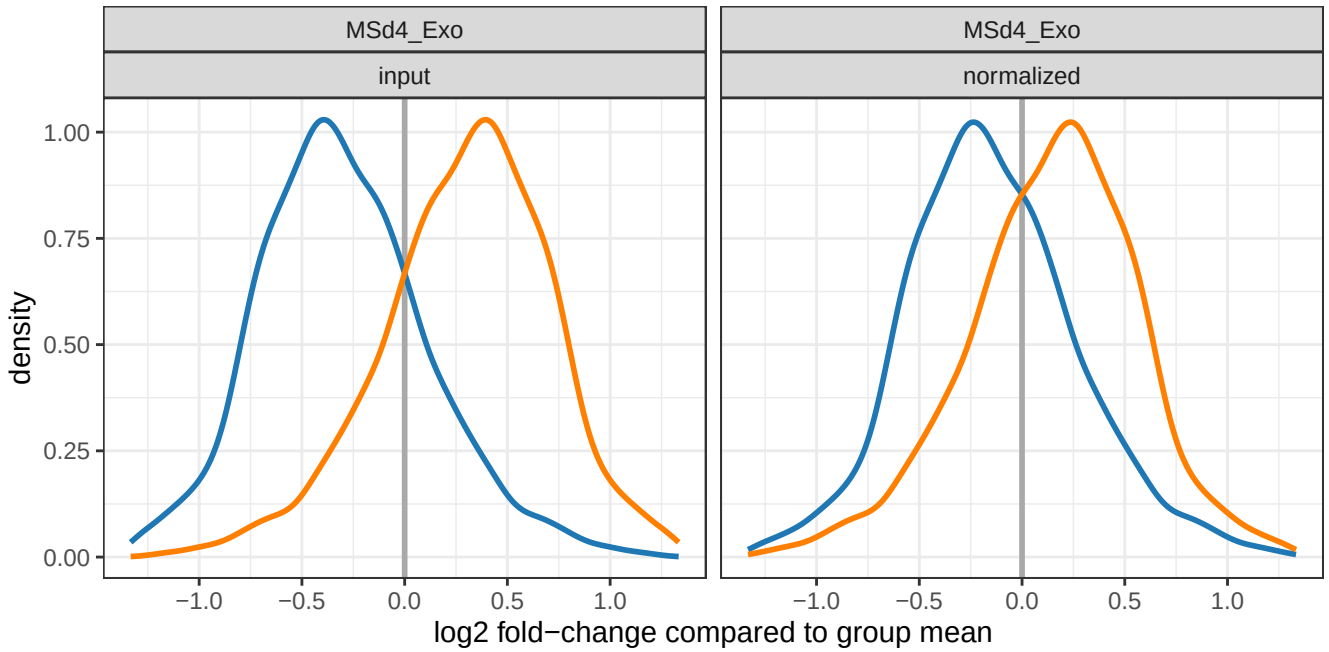

ms2_l
ms1_l


MS_S
input
MS_S
normalized
density
0.0
0.2
0.4
0.6
0.8
−1
0
1
log2 fold−change compared to group mean

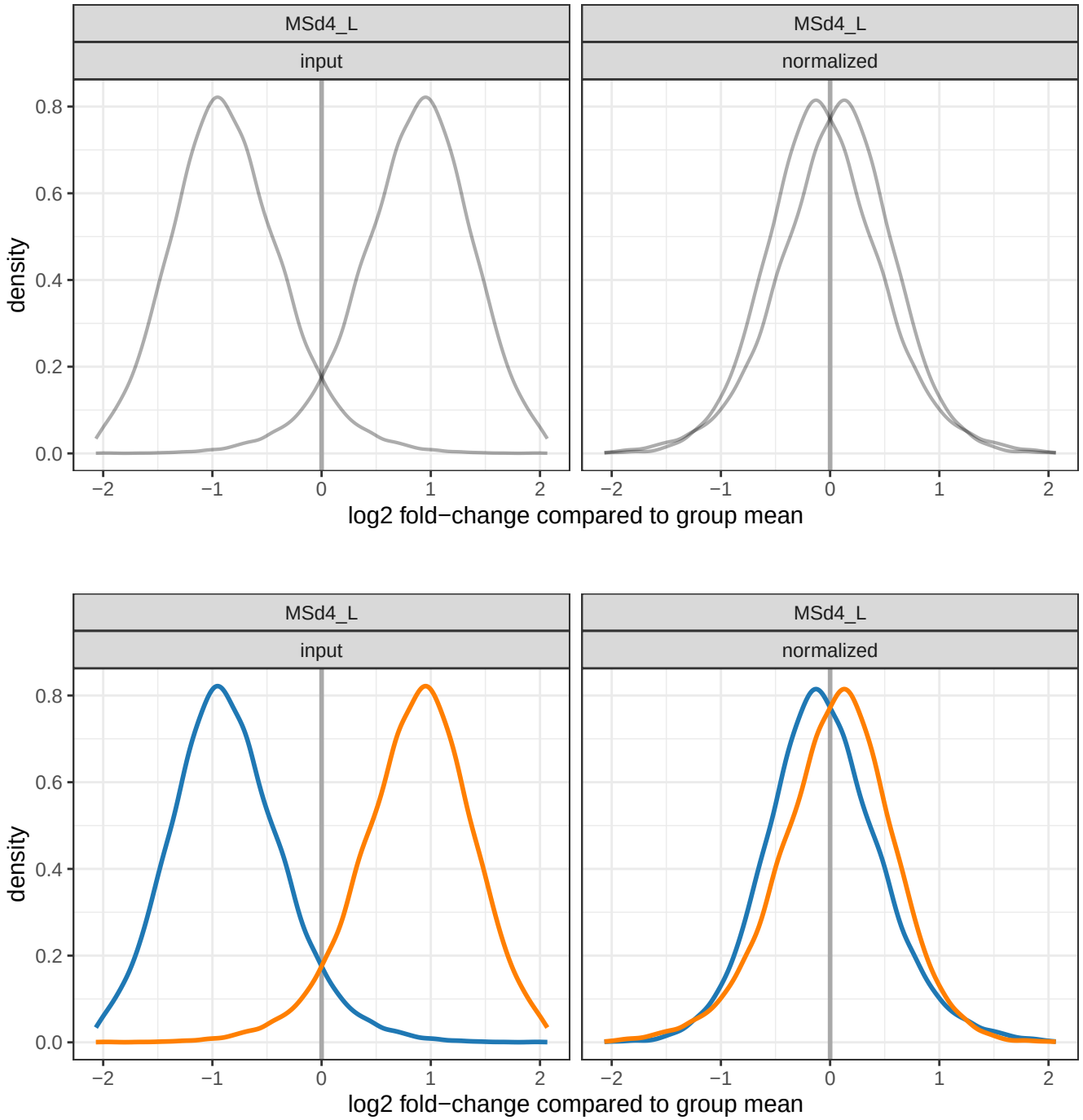

MS_S
input
MS_S
normalized
density
0.0
0.2
0.4
0.6
0.8
−1
0
1
log2 fold−change compared to group mean

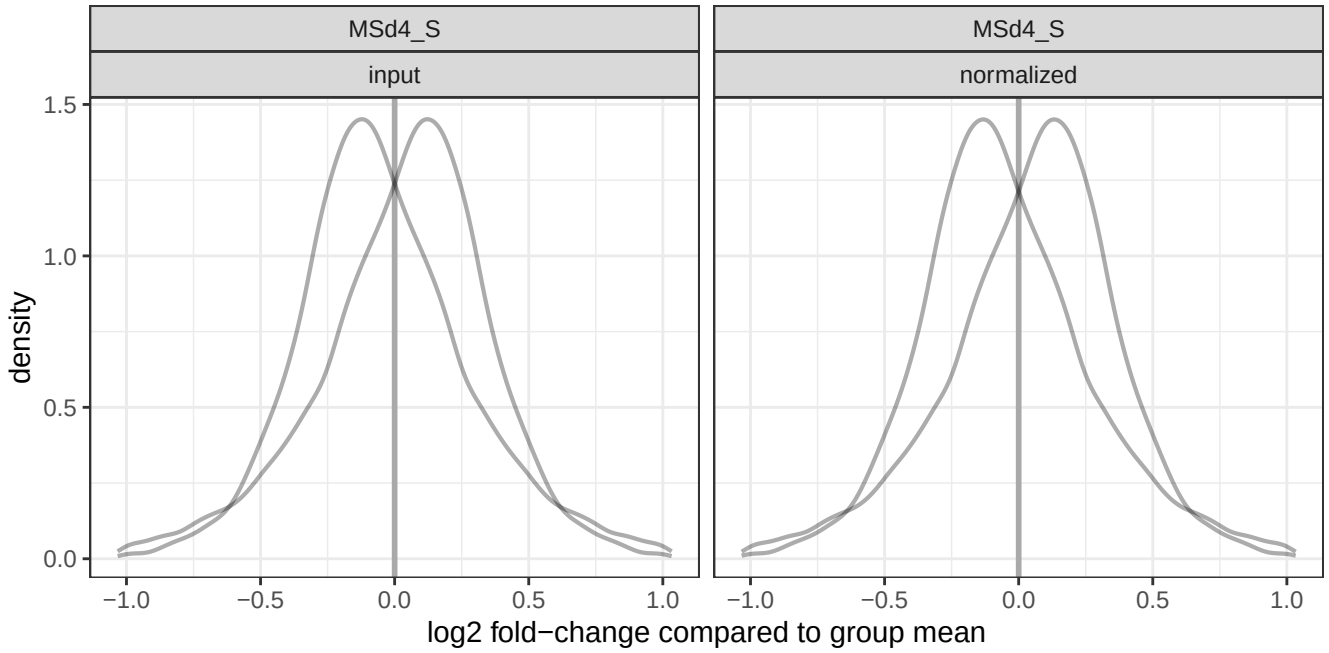

ms2_s
ms1_s

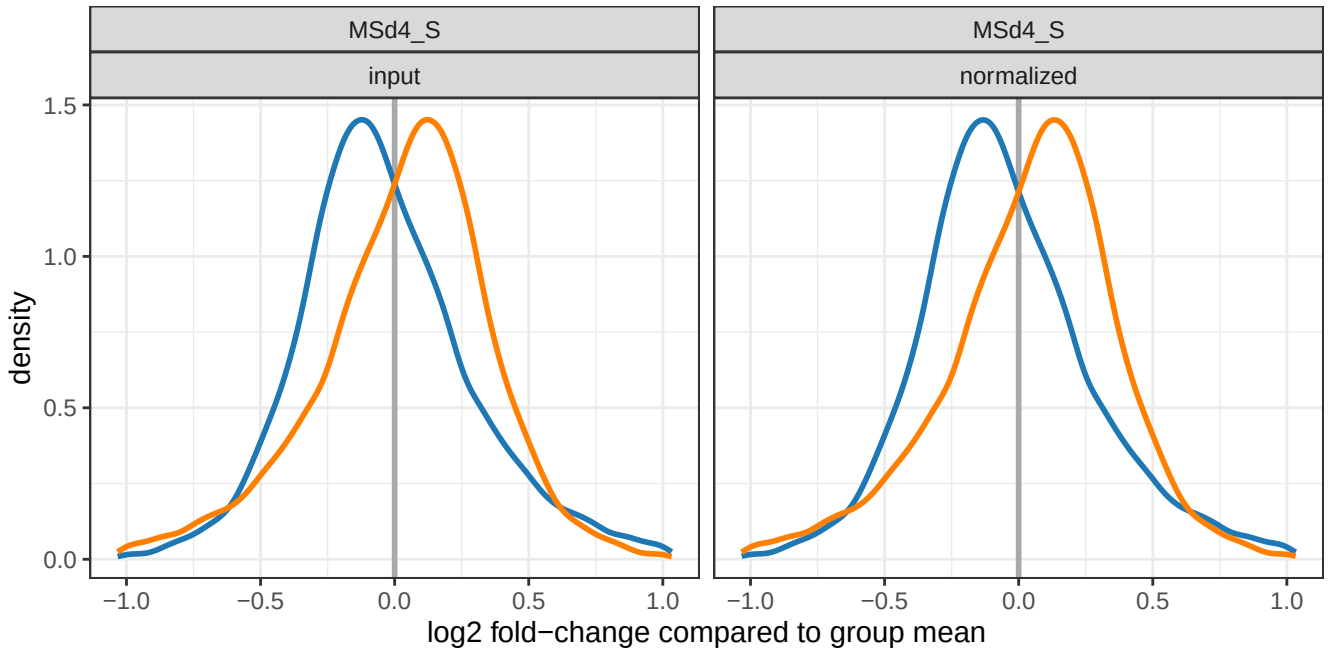

MSd4_Exo
input
MSd4_Exo
normalized
density
1.00
0.75
0.50
0.25
0.00
−1.0
−0.5
0.0
0.5
1.0
log2 fold−change compared to group mean


MSd4_Exo
input
MSd4_Exo
normalized
density
1.00
0.75
0.50
0.25
0.00
−1.0
−0.5
0.0
0.5
1.0
log2 fold−change compared to group mean

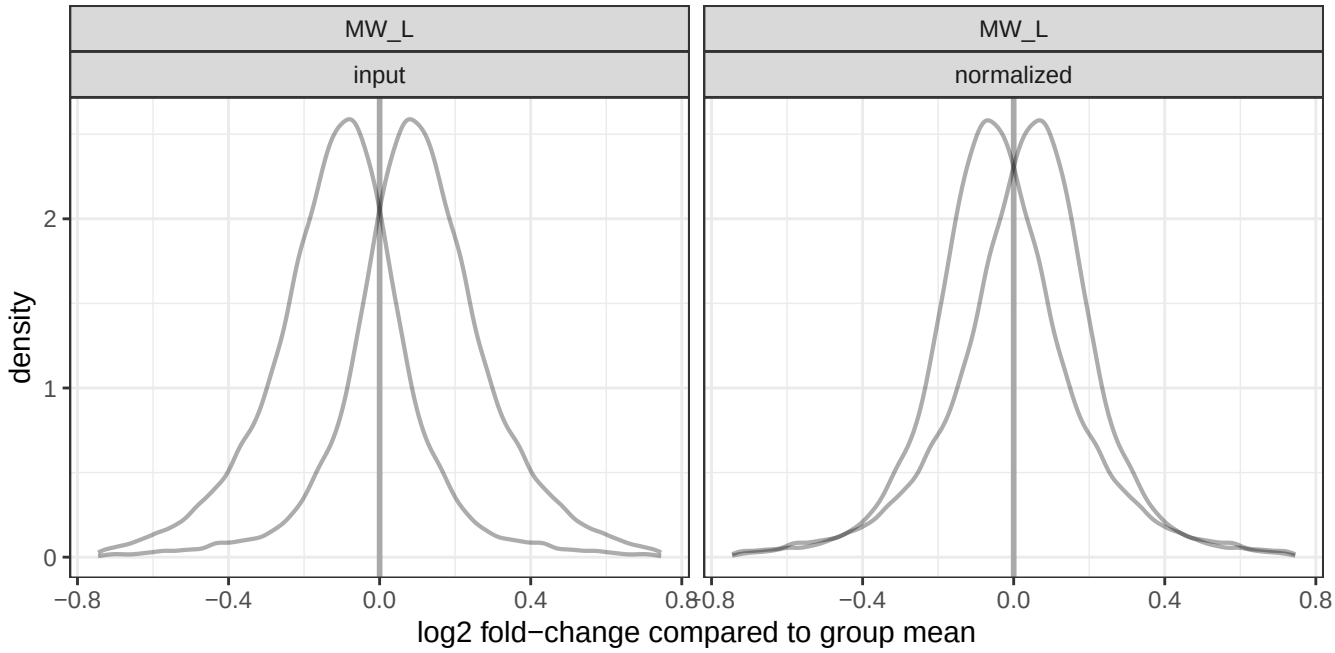

ms2d4_exo
ms1d4_exo

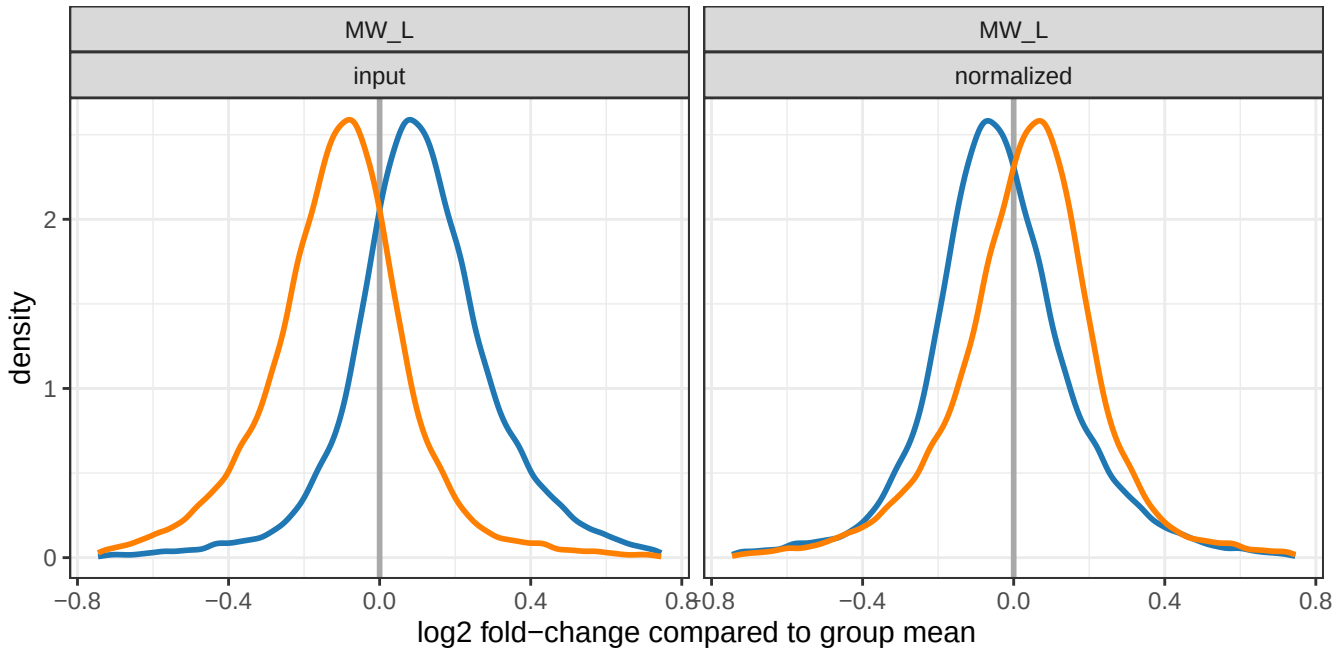

MSd4_L
input
MSd4_L
normalized
density
0.0
0.2
0.4
0.6
0.8
−2
−1
0
1
2
log2 fold−change compared to group mean
MSd4_L
input
MSd4_L
normalized
density
0.0
0.2
0.4
0.6
0.8
−2
−1
0
1
2
log2 fold−change compared to group mean
ms2d4_l
ms1d4_l


MSd4_S
input
MSd4_S
normalized
density
1.5
1.0
0.5
0.0
−1.0
−0.5
0.0
0.5
1.0
log2 fold−change compared to group mean

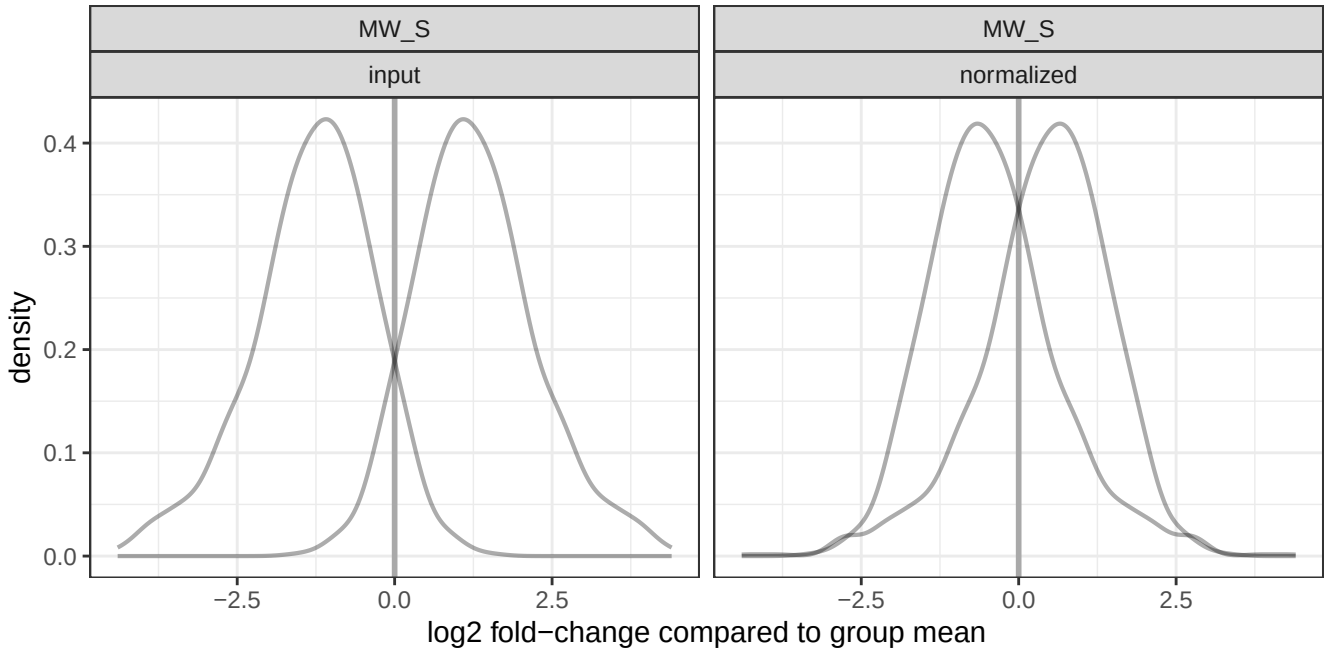

MSd4_S
input
MSd4_S
normalized
density
1.5
1.0
0.5
0.0
−1.0
−0.5
0.0
0.5
1.0
log2 fold−change compared to group mean

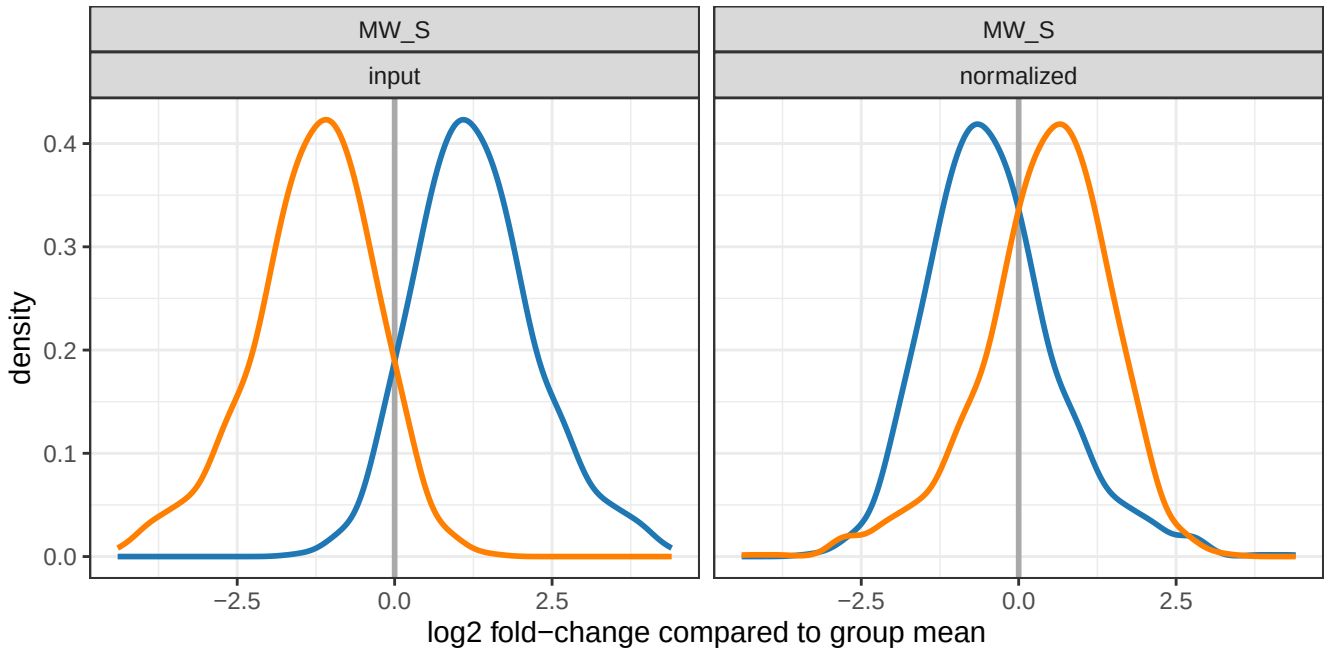

ms2d4_s
ms1d4_s


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

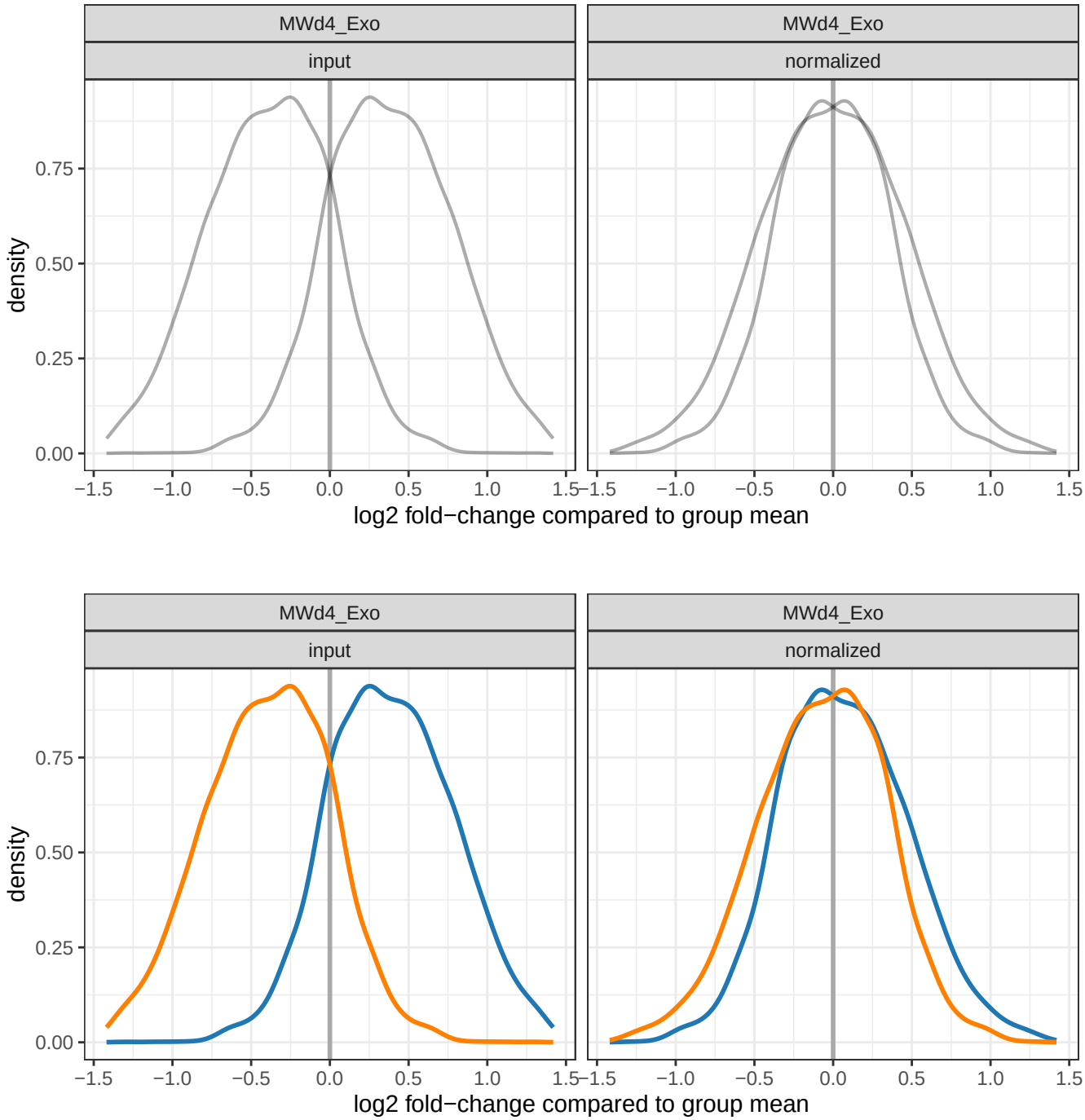

MW_L
input
MW_L
normalized
density
2
1
0
−0.8
−0.4
0.0
0.4
0.8
log2 fold−change compared to group mean


mw2_l
mw1_l


MW_S
input
MW_S
normalized
density
0.4
0.3
0.2
0.1
0.0
−2.5
0.0
2.5
log2 fold−change compared to group mean

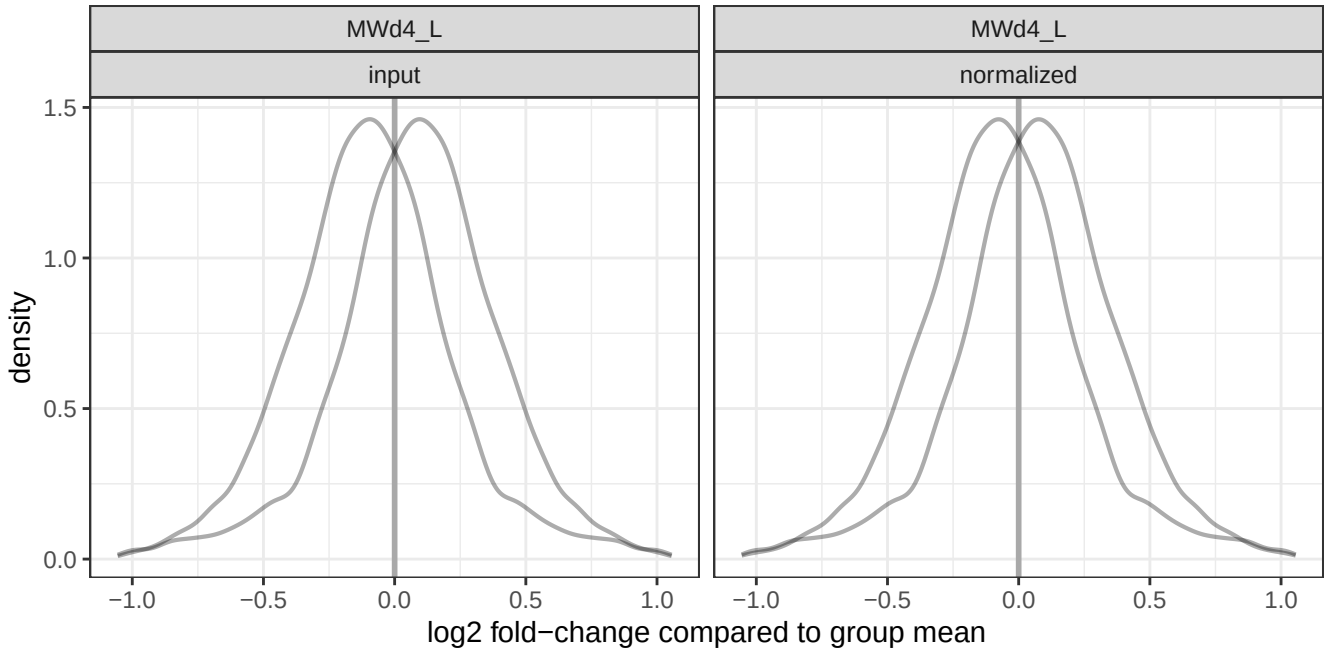

MW_S
input
MW_S
normalized
density
0.4
0.3
0.2
0.1
0.0
−2.5
0.0
2.5
log2 fold−change compared to group mean

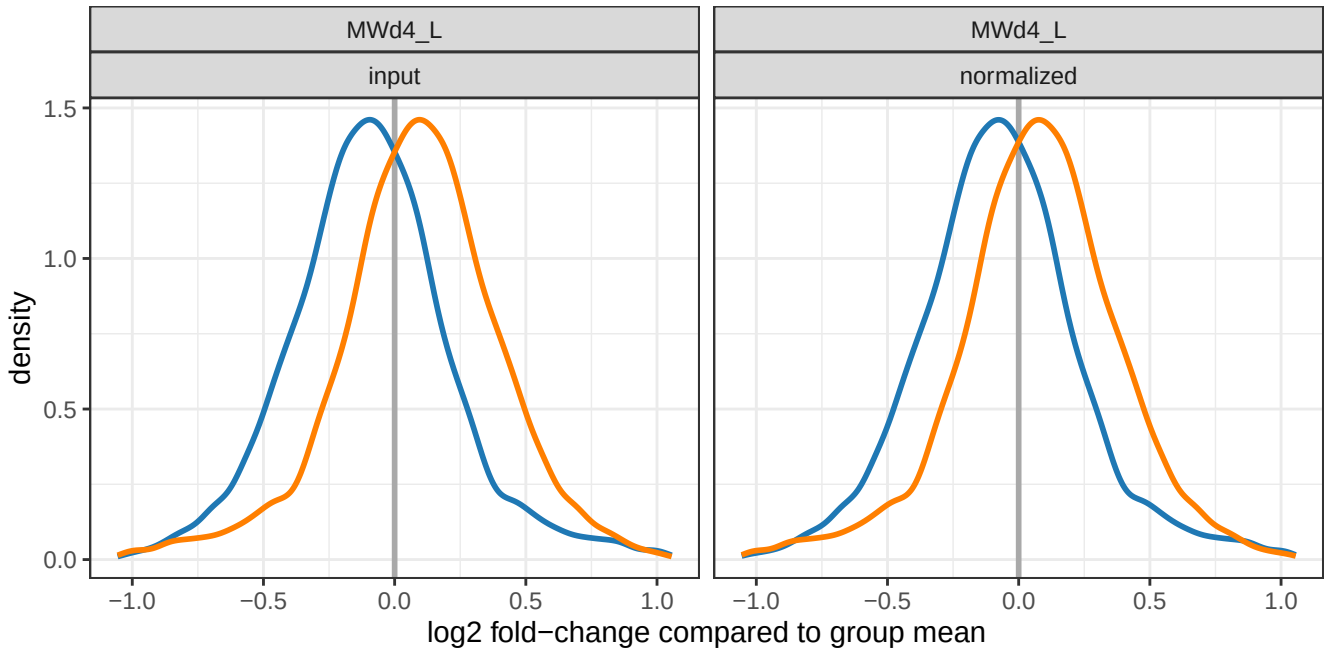

mw2_s
mw1_s


MWd4_Exo
input
MWd4_Exo
normalized
density
0.00
0.25
0.50
0.75
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
log2 fold−change compared to group mean

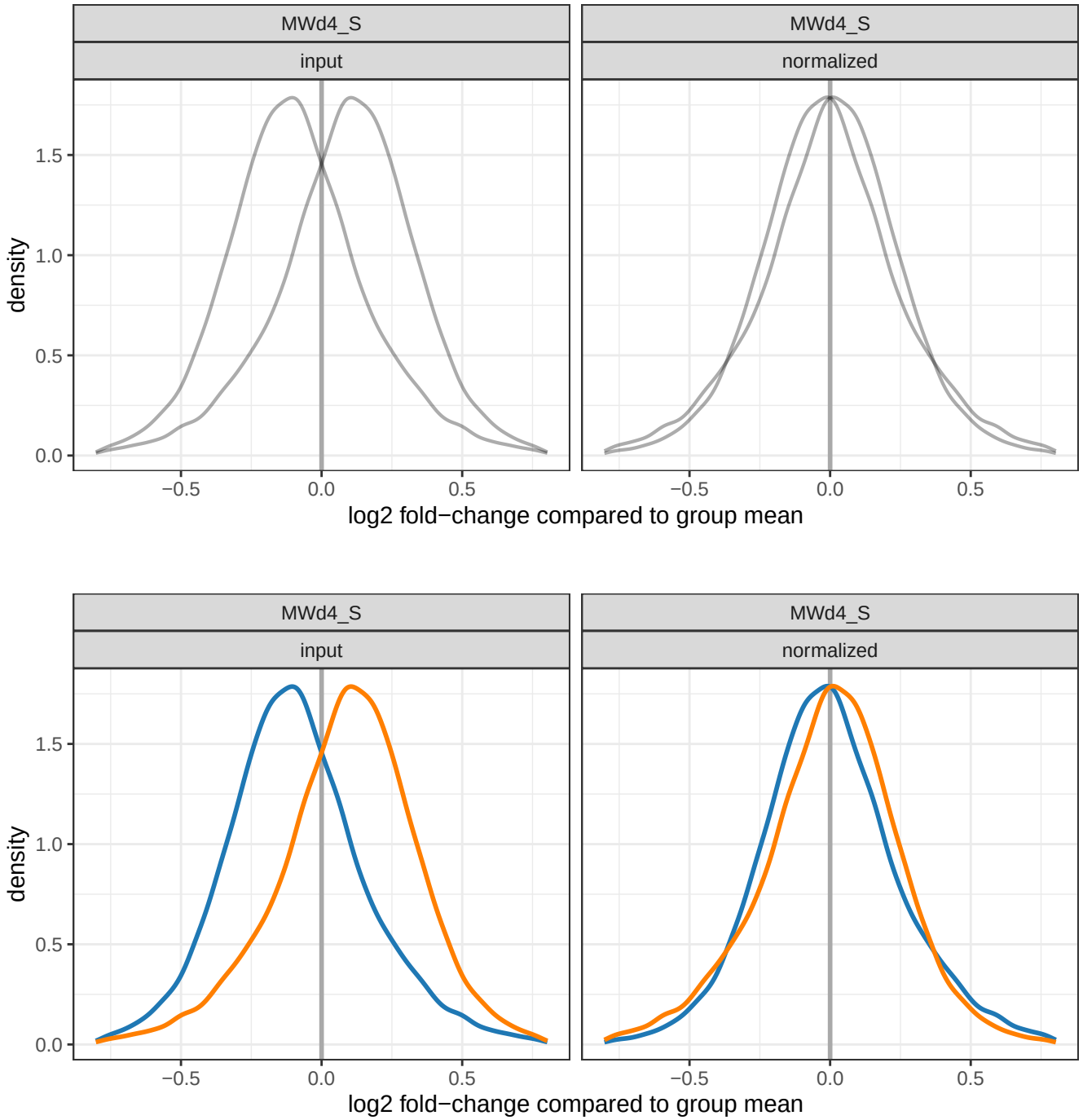
MWd4_Exo
input
MWd4_Exo
normalized
density
0.00
0.25
0.50
0.75
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
log2 fold−change compared to group mean

mw2d4_exo
mw1d4_exo


MWd4_L
input
MWd4_L
normalized
density
1.5
1.0
0.5
0.0
−1.0
−0.5
0.0
0.5
1.0
log2 fold−change compared to group mean


MWd4_L
input
MWd4_L
normalized
density
1.5
1.0
0.5
0.0
−1.0
−0.5
0.0
0.5
1.0
log2 fold−change compared to group mean


mw2d4_l
mw1d4_l


MWd4_S
input
MWd4_S
normalized
density
1.5
1.0
0.5
0.0
−0.5
0.0
0.5
log2 fold−change compared to group mean
MWd4_S
input
MWd4_S
normalized
density
1.5
1.0
0.5
0.0
−0.5
0.0
0.5
log2 fold−change compared to group mean
mw2d4_s
mw1d4_s

top10 outliers per sample group

Group	Sample	Standard Deviation
FS_L	fs1_l	0.342
FS_L	fs2_l	0.342

Group	Sample	Standard Deviation
FS_S	fs2_s	0.479
FS_S	fs1_s	0.479

Group	Sample	Standard Deviation
FW_L	fw1_l	0.168
FW_L	fw2_l	0.168

Group	Sample	Standard Deviation
FW_S	fw1_s	0.517
FW_S	fw2_s	0.517

Group	Sample	Standard Deviation
MS_L	ms1_l	0.191
MS_L	ms2_l	0.191

Group	Sample	Standard Deviation
MS_S	ms1_s	0.494
MS_S	ms2_s	0.494

Group	Sample	Standard Deviation
MSd4_Exo	ms2d4_exo	0.408
MSd4_Exo	ms1d4_exo	0.408

Group	Sample	Standard Deviation
MSd4_L	ms1d4_l	0.509
MSd4_L	ms2d4_l	0.509

Group	Sample	Standard Deviation
MSd4_S	ms1d4_s	0.324
MSd4_S	ms2d4_s	0.324

Group	Sample	Standard Deviation
MW_L	mw1_l	0.191
MW_L	mw2_l	0.191

Group	Sample	Standard Deviation
MW_S	mw1_s	0.983
MW_S	mw2_s	0.983

Group	Sample	Standard Deviation
MWd4_Exo	mw2d4_exo	0.393
MWd4_Exo	mw1d4_exo	0.393

Group	Sample	Standard Deviation
MWd4_L	mw1d4_l	0.311
MWd4_L	mw2d4_l	0.311

Group	Sample	Standard Deviation
MWd4_S	mw1d4_s	0.247
MWd4_S	mw2d4_s	0.247

1.5.2 CoV, leave-one-out

No CoV leave-one-out computations could be made, the dataset lacks sample groups with at least 4 replicate samples.

1.5.3 Coefficient of Variation

No CoV computations could be made, the dataset lacks sample groups with at least 3 replicate samples.

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).

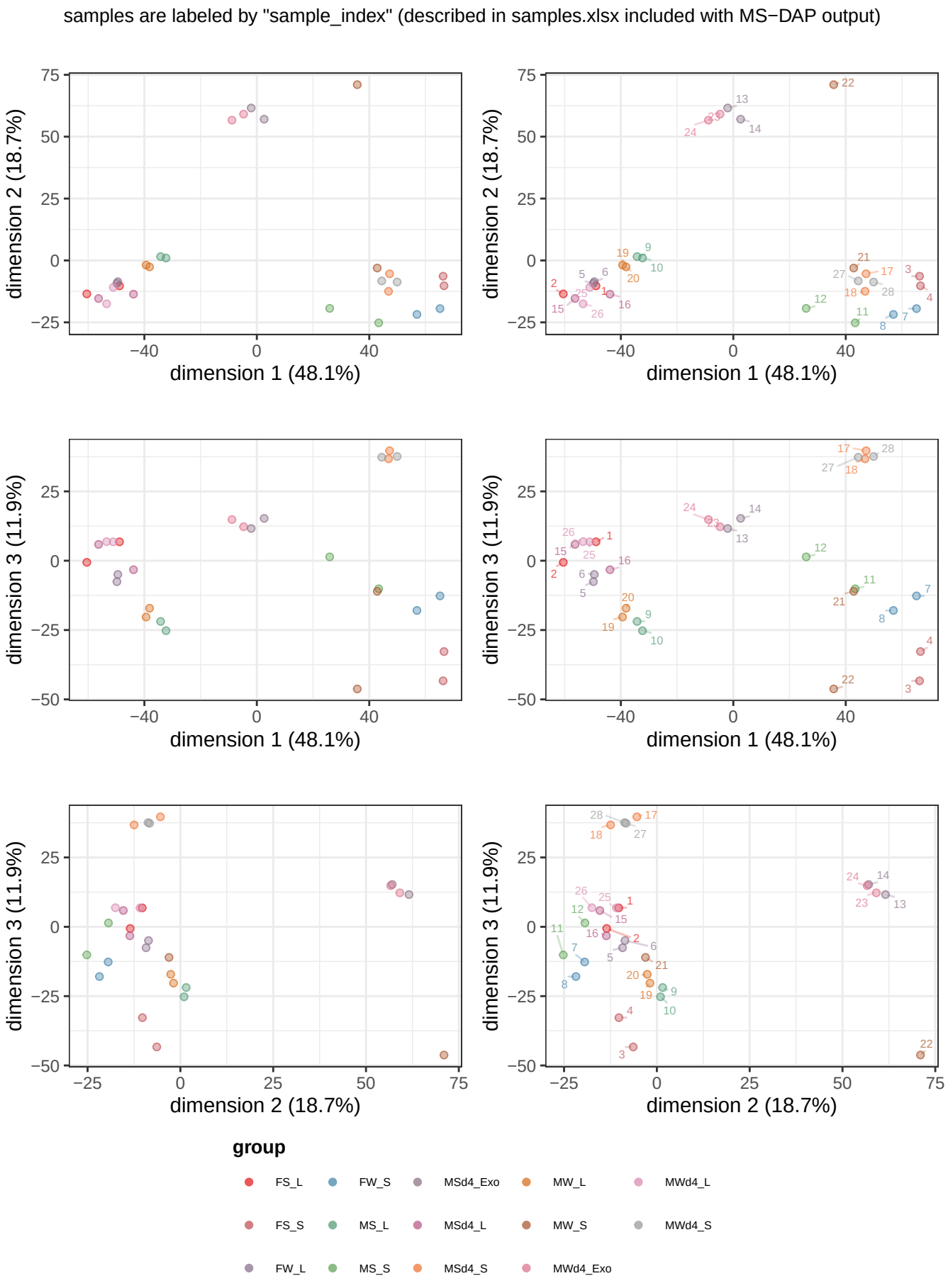
If additional sample metadata was provided, such as experiment batch, sample-prep dates, gel, etc., multiple PCA figures will be generated with respective color-codings. Users are encouraged to provide relevant experiment information as sample metadata and use these figures to search for unexpected batch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

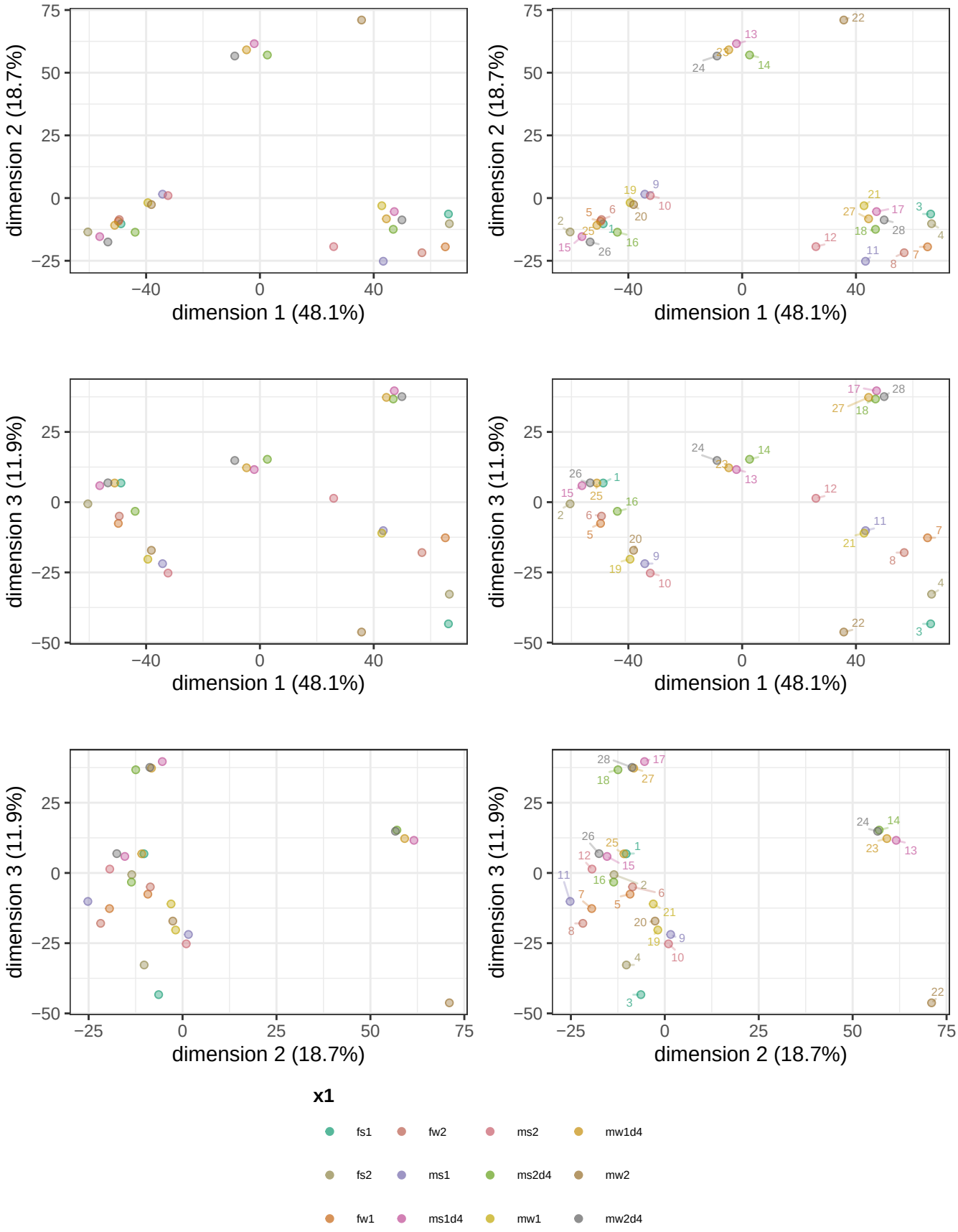
Rationale behind data filter
As mentioned above, the aim of the PCA figures is to identify global effects. To achieve this, we compute sample distances on the subset of peptides identified in each group which prevents rarely detected peptides/proteins from having a disproportionate effect on sample clustering.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

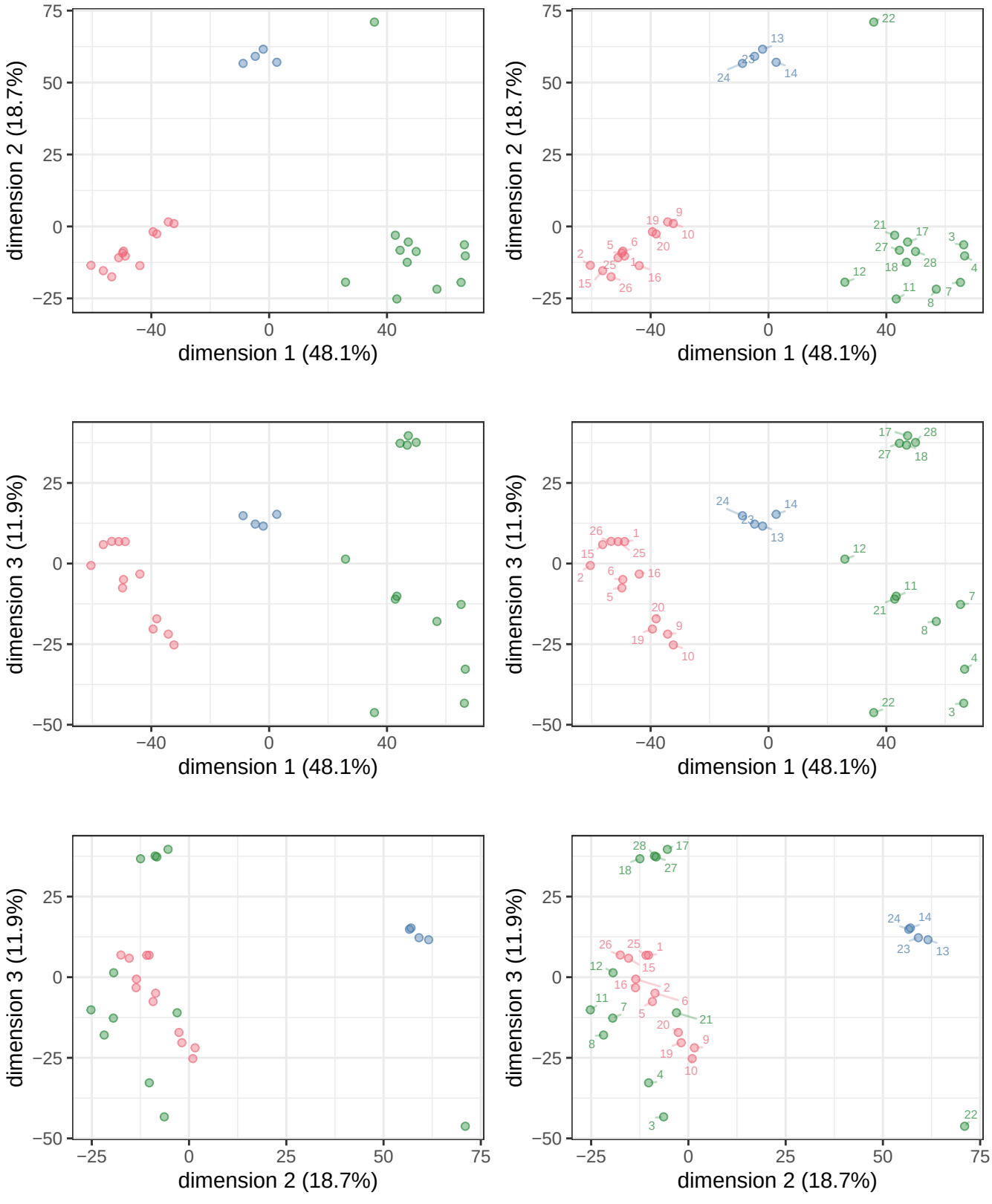
PCA only on samples not flagged as 'exclude', using 2202 peptides



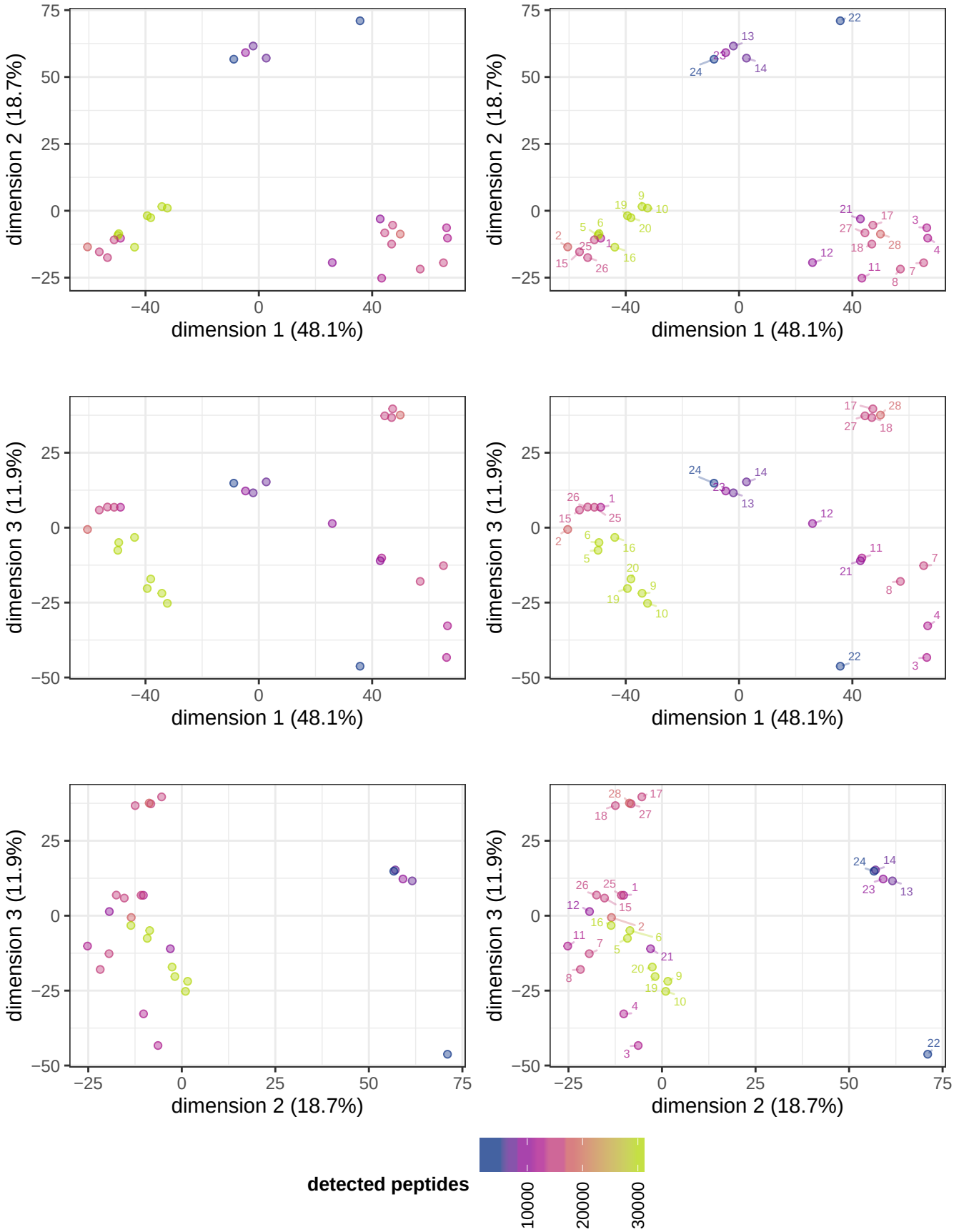
samples are labeled by "sample_index" (described in samples.xlsx included with MS−DAP output)



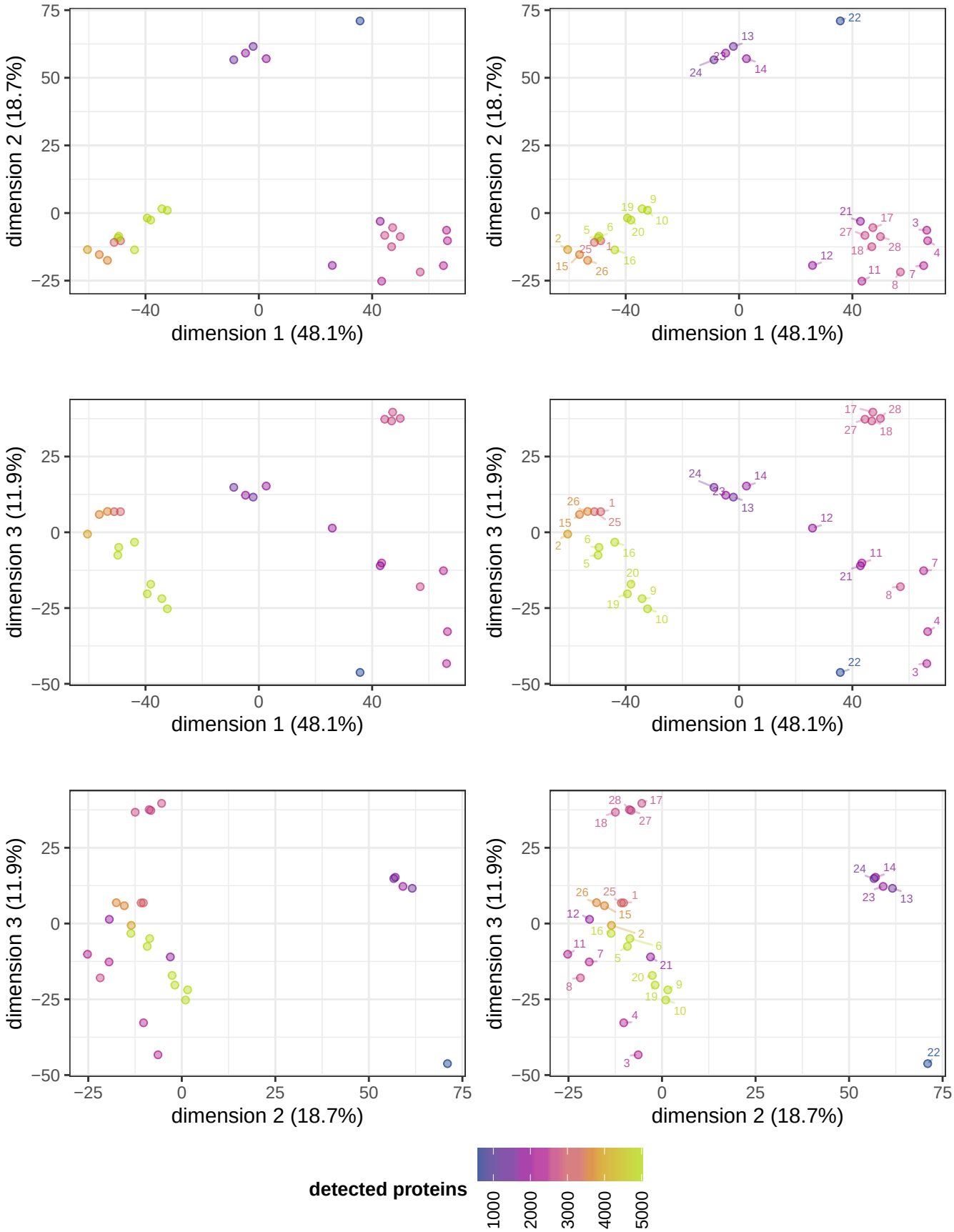
samples are labeled by "sample_index" (described in samples.xlsx included with MS−DAP output)



samples are labeled by "sample_index" (described in samples.xlsx included with MS−DAP output)



samples are labeled by "sample_index" (described in samples.xlsx included with MS−DAP output)



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 MWd4_L vs MSd4_L

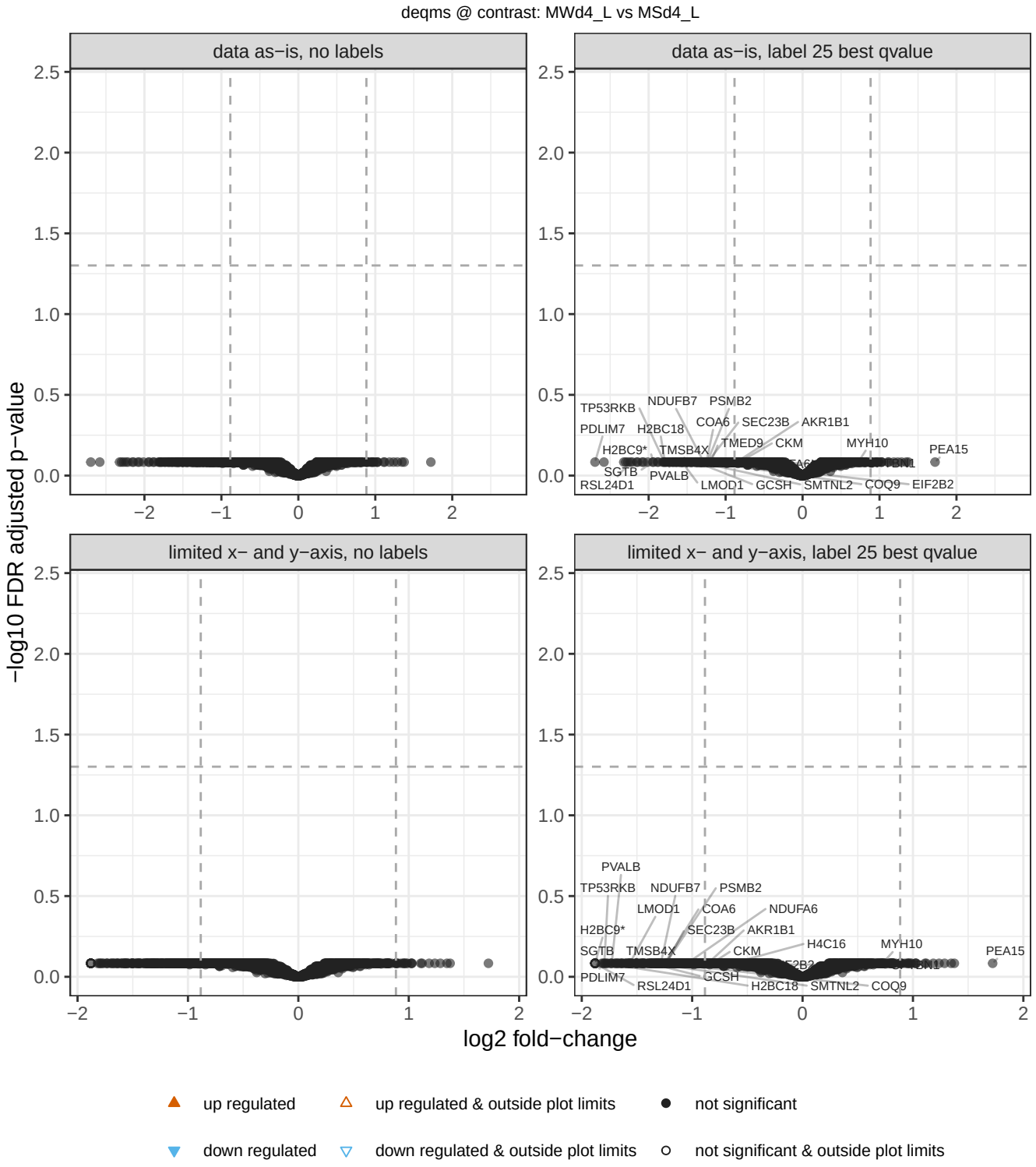
- **user setting:** using 'filter by contrast' peptide filtering approach
- 17163 peptides in 3735 proteins remain in the current contrast after peptide filters and are used for the statistical analysis in this section
- qvalue threshold: 0.05
- log2 foldchange threshold: 0.884

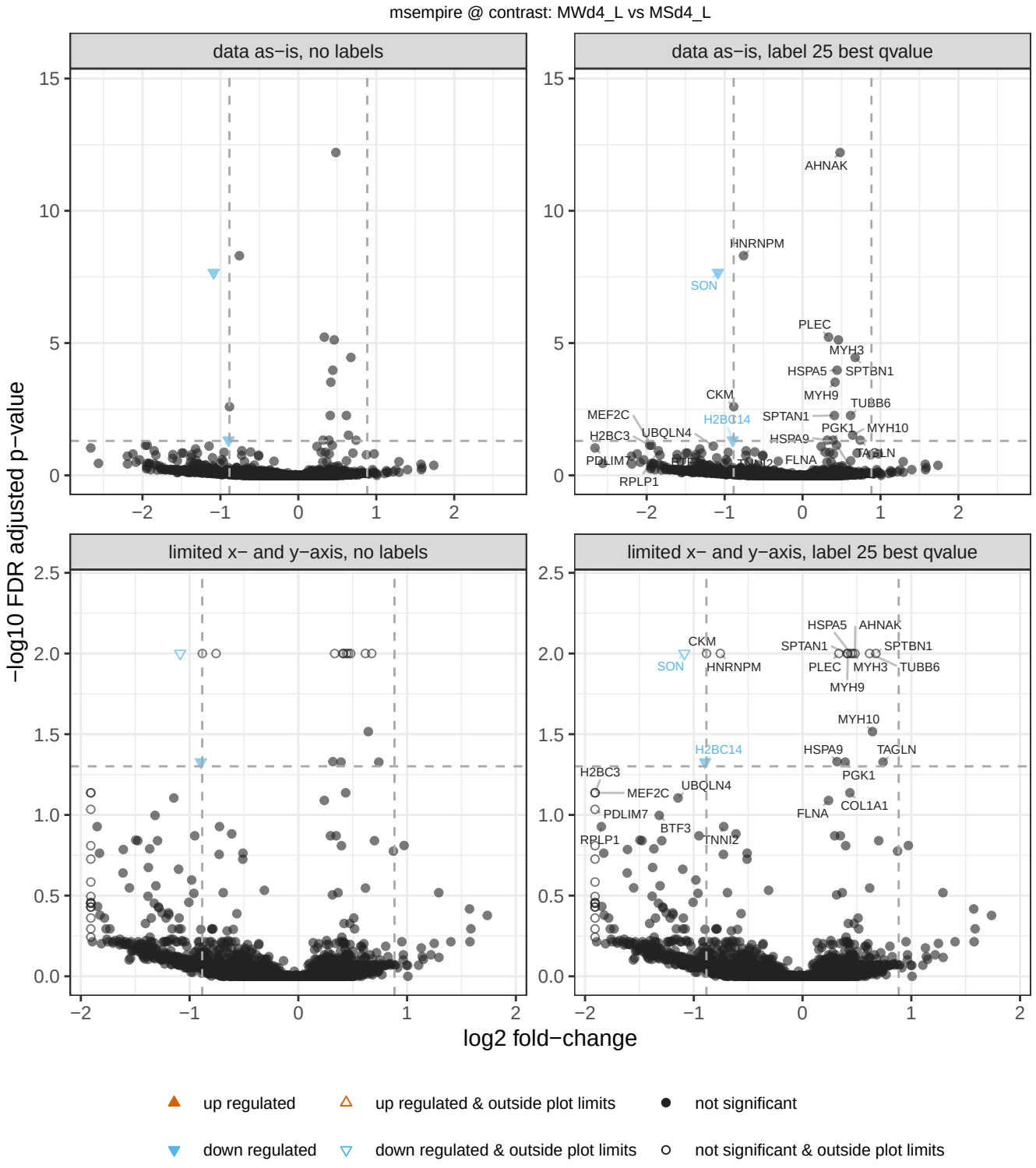
2.1.1 volcano

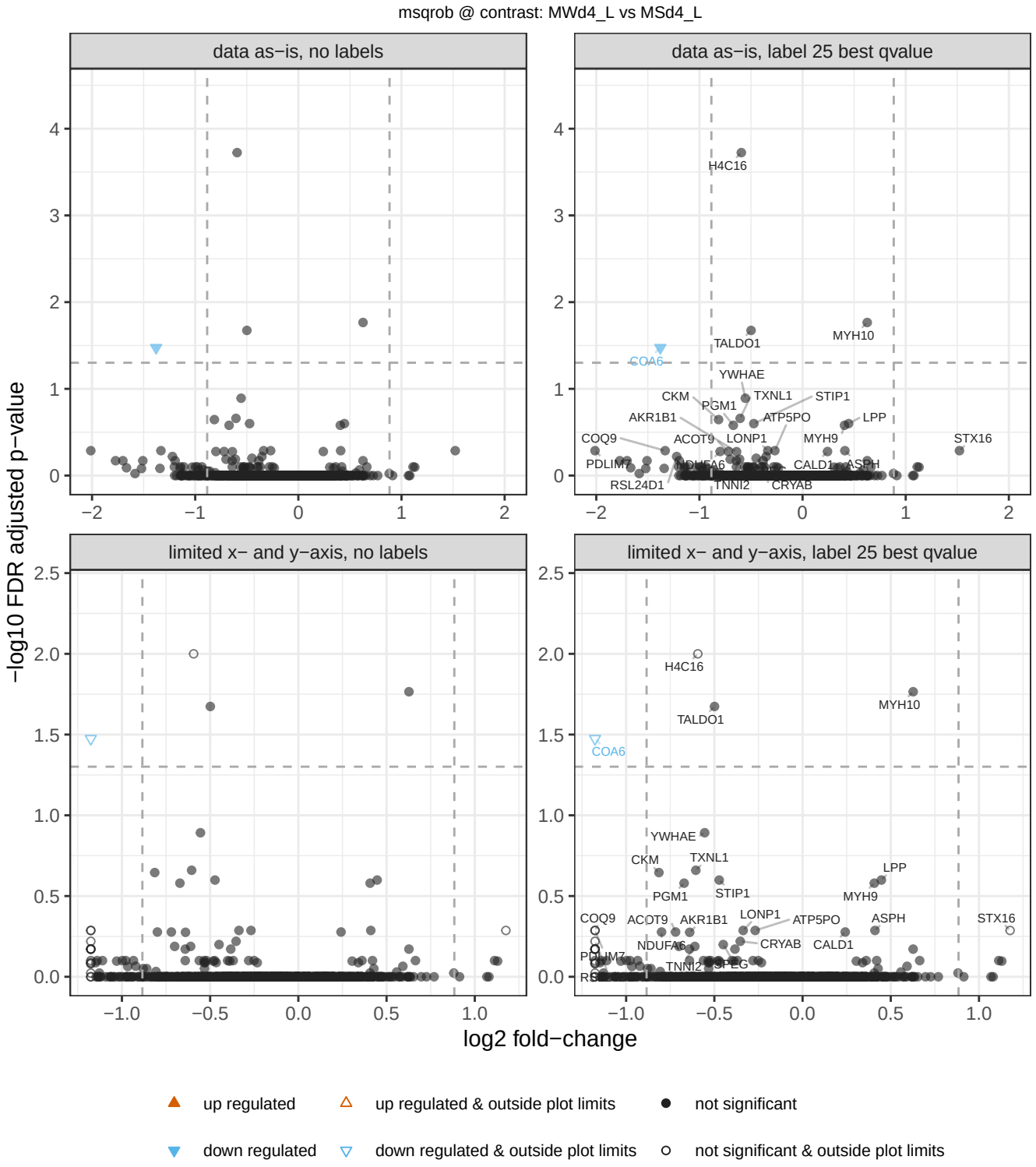
The plot title shows the statistical model and contrast (sample groups in the comparison). Left- and right-side figure panels on each row represent the same figure without and with labels for the 25 proteins with lowest p-value.

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: MWd4_L vs MSd4_L
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
TP53RKB
NDUFB7
PSMB2
PDLIM7
H2BC18
COA6
SEC23B
AKR1B1
H2BC9*
TMSB4X
TMED9
CKM
MYH10
PEA15
SGTB
PVALB
RSL24D1
LMOD1
GCSH
SMTNL2
COQ9
EIF2B2
NDUFA6
H4C16
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits

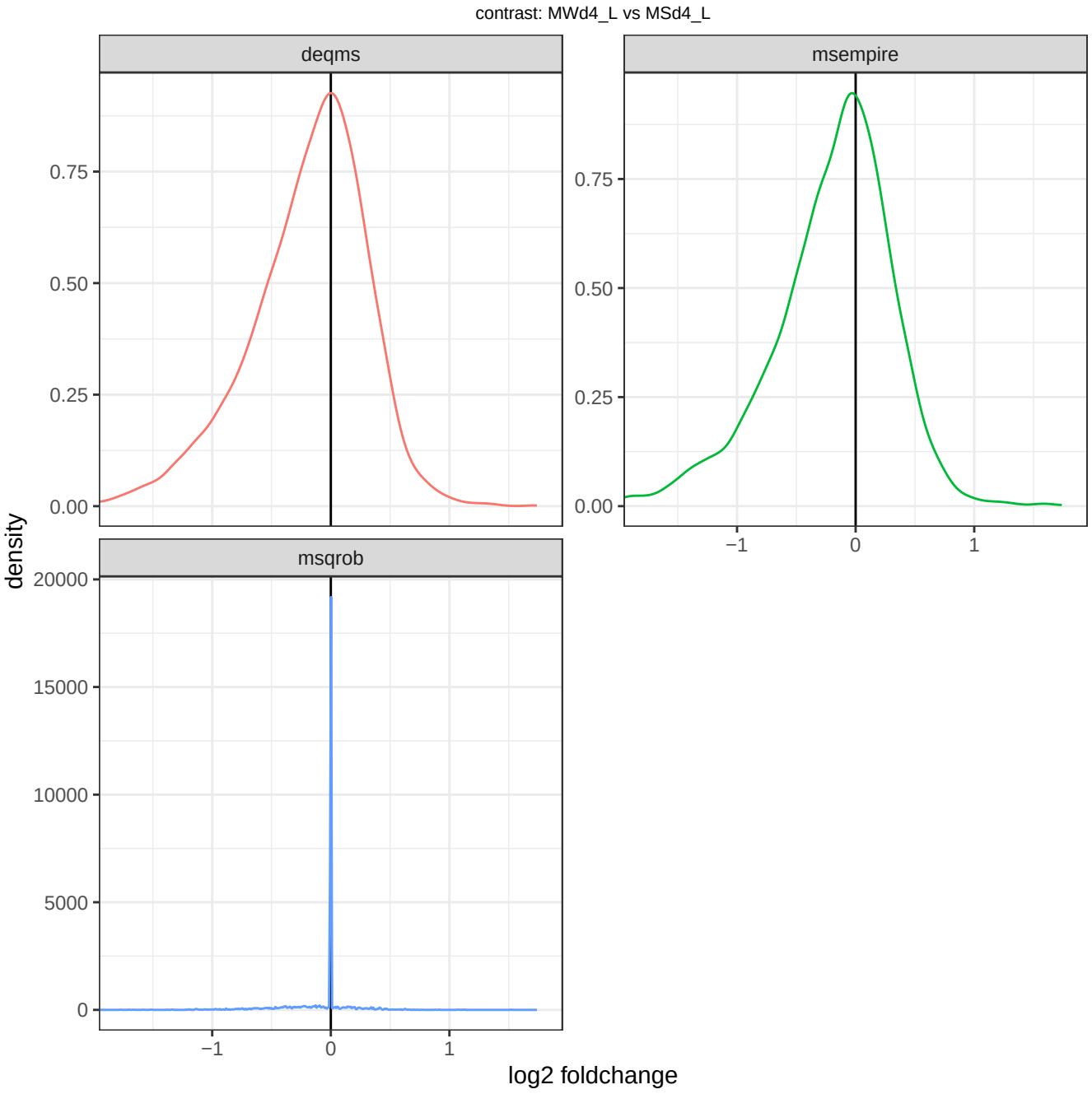

msempire @ contrast: MWd4_L vs MSd4_L
data as−is, no labels
data as−is, label 25 best qvalue
limited x− and y−axis, no labels
limited x− and y−axis, label 25 best qvalue
−log10 FDR adjusted p−value
log2 fold−change
AHNAK
HNRNPM
SON
PLEC
MYH3
HSPA5
SPTBN1
CKM
MYH9
TUBB6
MEF2C
H2BC14
SPTAN1
PGK1
MYH10
H2BC3
UBQLN4
HSPA9
TAGLN
PDLIM7
FLNA
RPLP1
BTF3
TNNI2
COL1A1
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits


msqrob @ contrast: MWd4_L vs MSd4_L
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
H4C16
MYH10
TALDO1
COA6
YWHAE
CKM
TXNL1
STIP1
PGM1
AKR1B1
ATP5PO
LPP
COQ9
ACOT9
LONP1
MYH9
STX16
PDLIM7
NDUFA6
CALD1
ASPH
RSL24D1
TNNI2
CRYAB
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



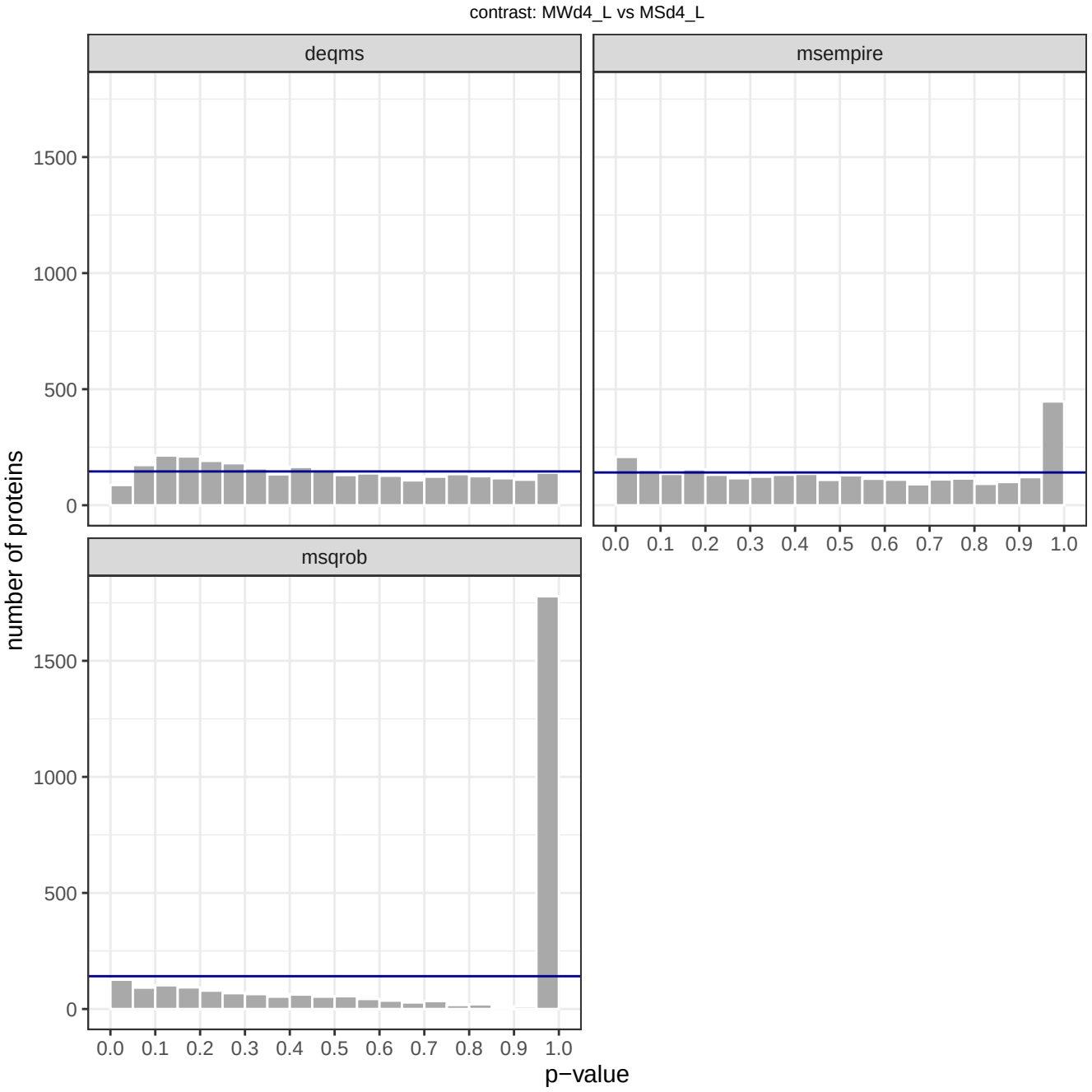
2.1.3 p-value distribution

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

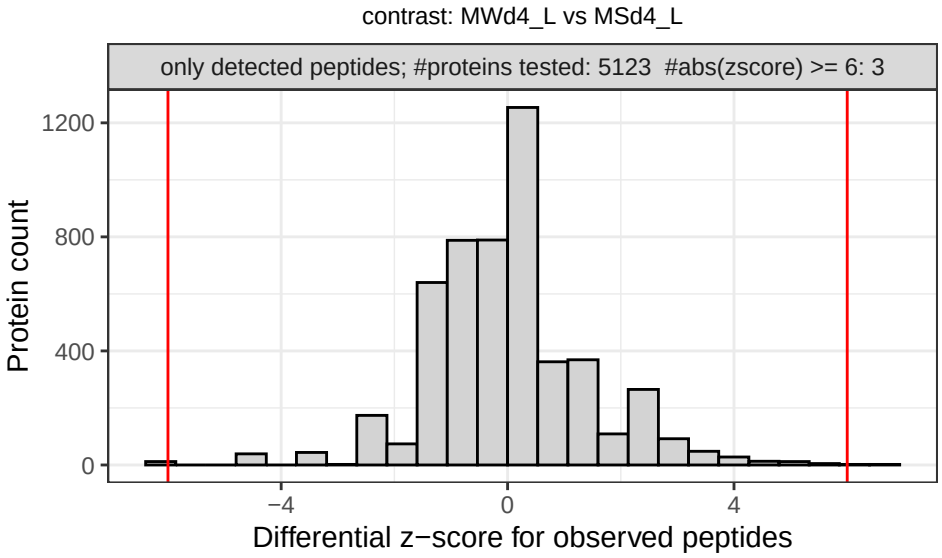


2.1.4 differential detect

Some proteins may not have peptides with sufficient data points over samples to be used for differential expression analysis (DEA), 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 study, those are interesting proteins. For this purpose, a basic metric for differential testing based on observed peptide counts is provided in MS-DAP as a situational tool. Importantly, this approach is less robust than DEA and the criteria to find a reliable set of differentially expressed proteins (by differential testing) might differ between real-world datasets.

As general guidelines for differential detection, the recommended default setting is to filter for proteins that were observed with at least 2 peptides in at least 3 replicates (or 50% of replicates, whichever number is greater). Use the plots of differential detection z-score histograms to observe the overall distribution and start your data exploration at proteins with the strongest z-scores to find desired z-score cutoffs (typically an absolute z-score of 5 or higher, but this is not set in stone for all datasets). This works best for DDA experiments, for DIA only the most extreme values are informative in many cases (e.g. proteins exclusively identified in condition A & found in nearly all replicates of condition A).

Below figure shows the distribution of these scores with thresholds at 6 std. Both the z-scores and the counts these are based upon are available in the statistical result Excel table.



2.2 MWd4_Exo vs MSd4_Exo

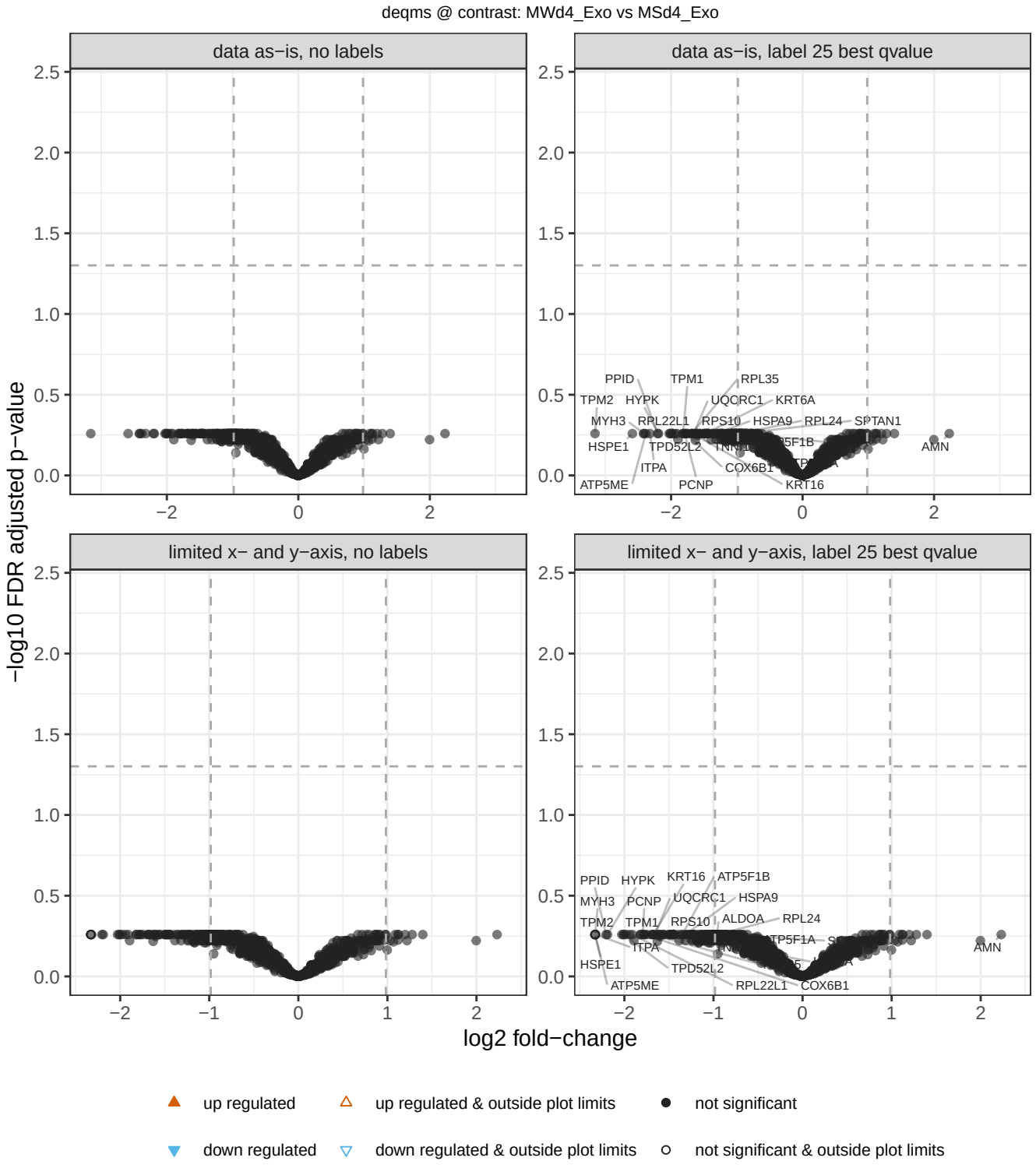
- **user setting:** using 'filter by contrast' peptide filtering approach
- 7351 peptides in 1579 proteins remain in the current contrast after peptide filters and are used for the statistical analysis in this section
- qvalue threshold: 0.05
- log2 foldchange threshold: 0.983

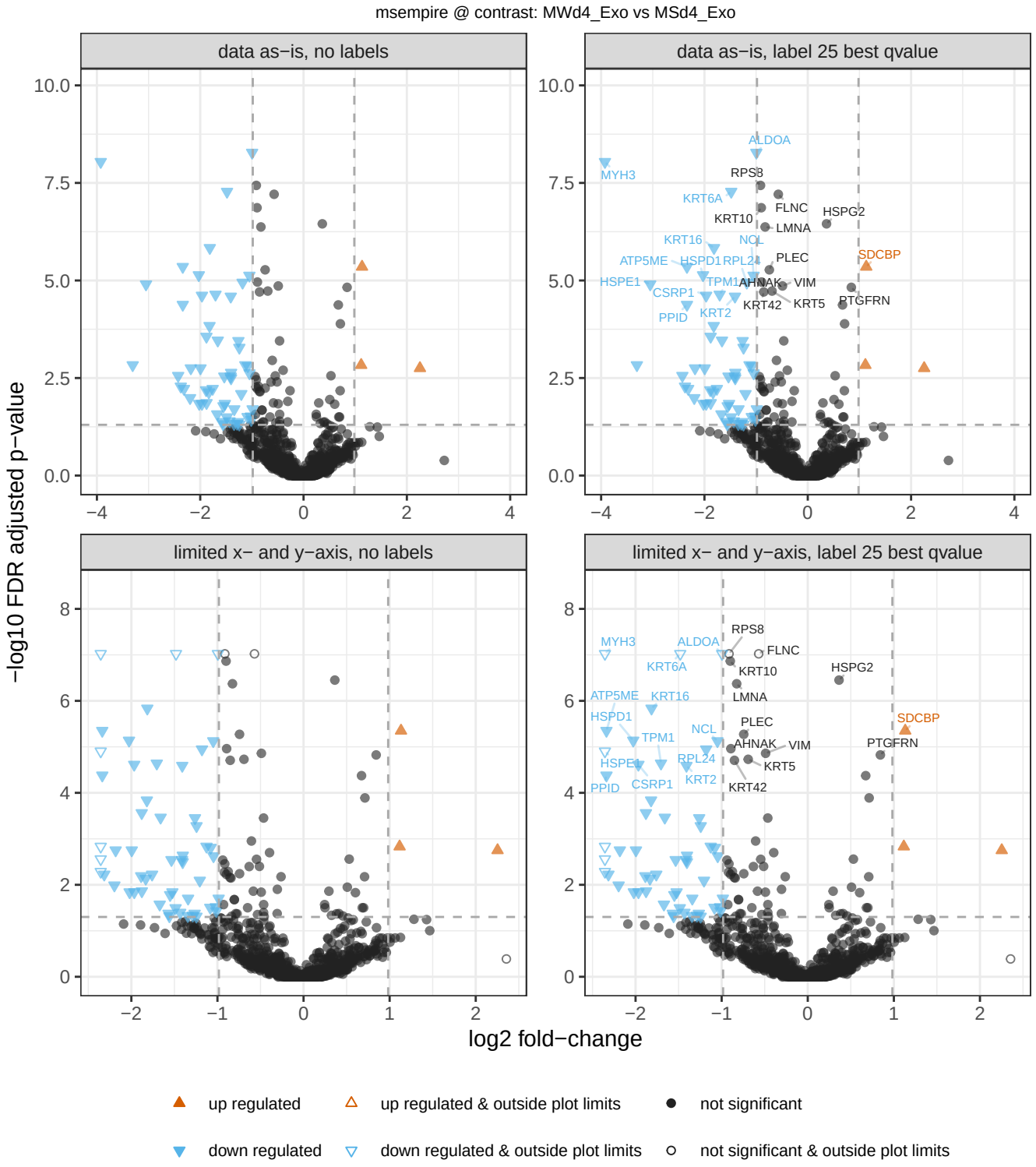
2.2.1 volcano

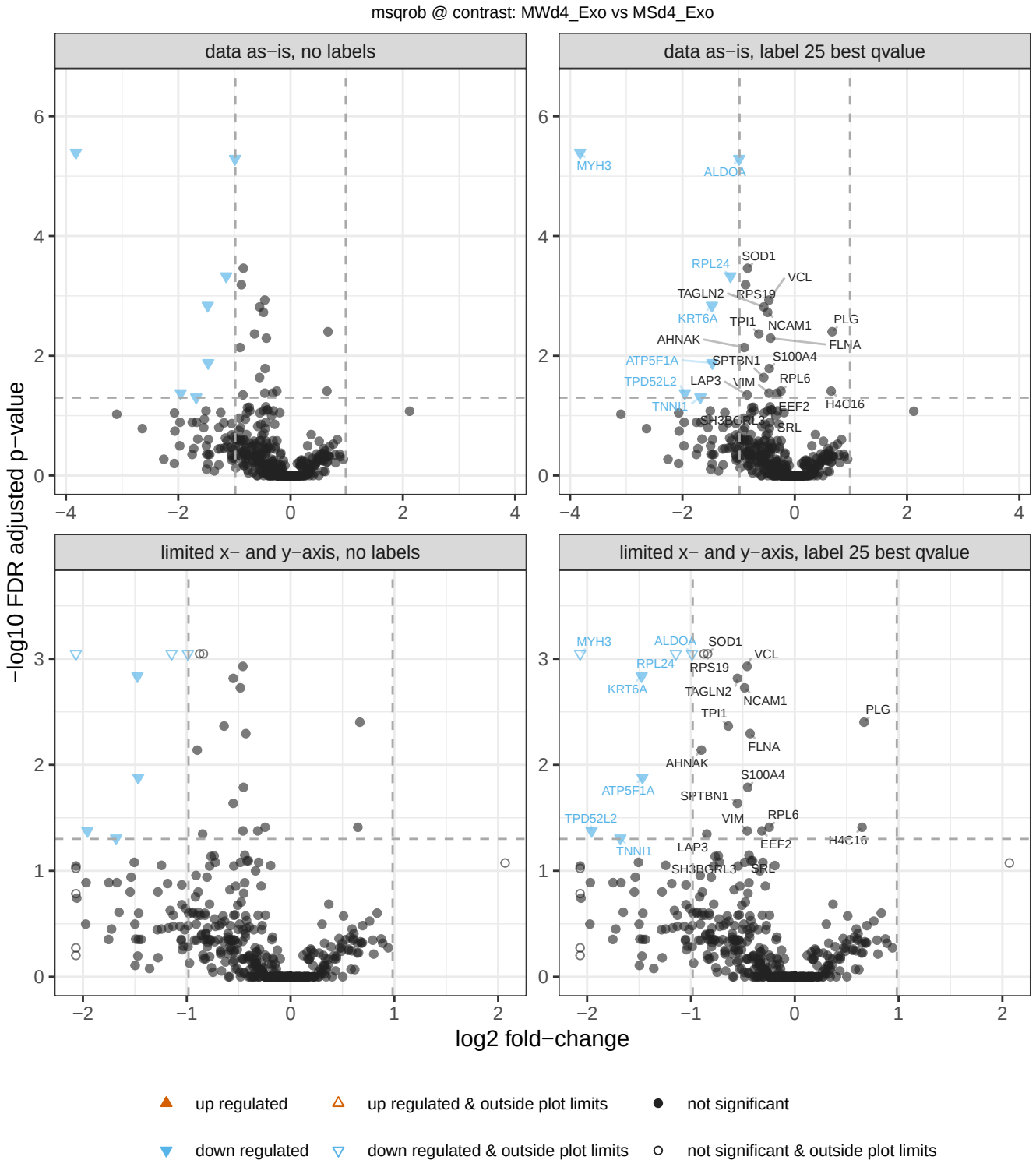
The plot title shows the statistical model and contrast (sample groups in the comparison). Left- and right-side figure panels on each row represent the same figure without and with labels for the 25 proteins with lowest p-value.

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: MWd4_Exo vs MSd4_Exo
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
PPID
TPM1
RPL35
TPM2
HYPK
UQCRC1
KRT6A
MYH3
RPL22L1
RPS10
HSPA9
RPL24
SPTAN1
HSPE1
TPD52L2
AMN
ITPA
COX6B1
ATP5ME
PCNP
KRT16
KRT16
ATP5F1B
PPID
HYPK
MYH3
PCNP
UQCRC1
HSPA9
TPM2
TPM1
RPS10
ALDOA
RPL24
ITPA
AMN
HSPE1
TPD52L2
ATP5ME
RPL22L1
COX6B1
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits

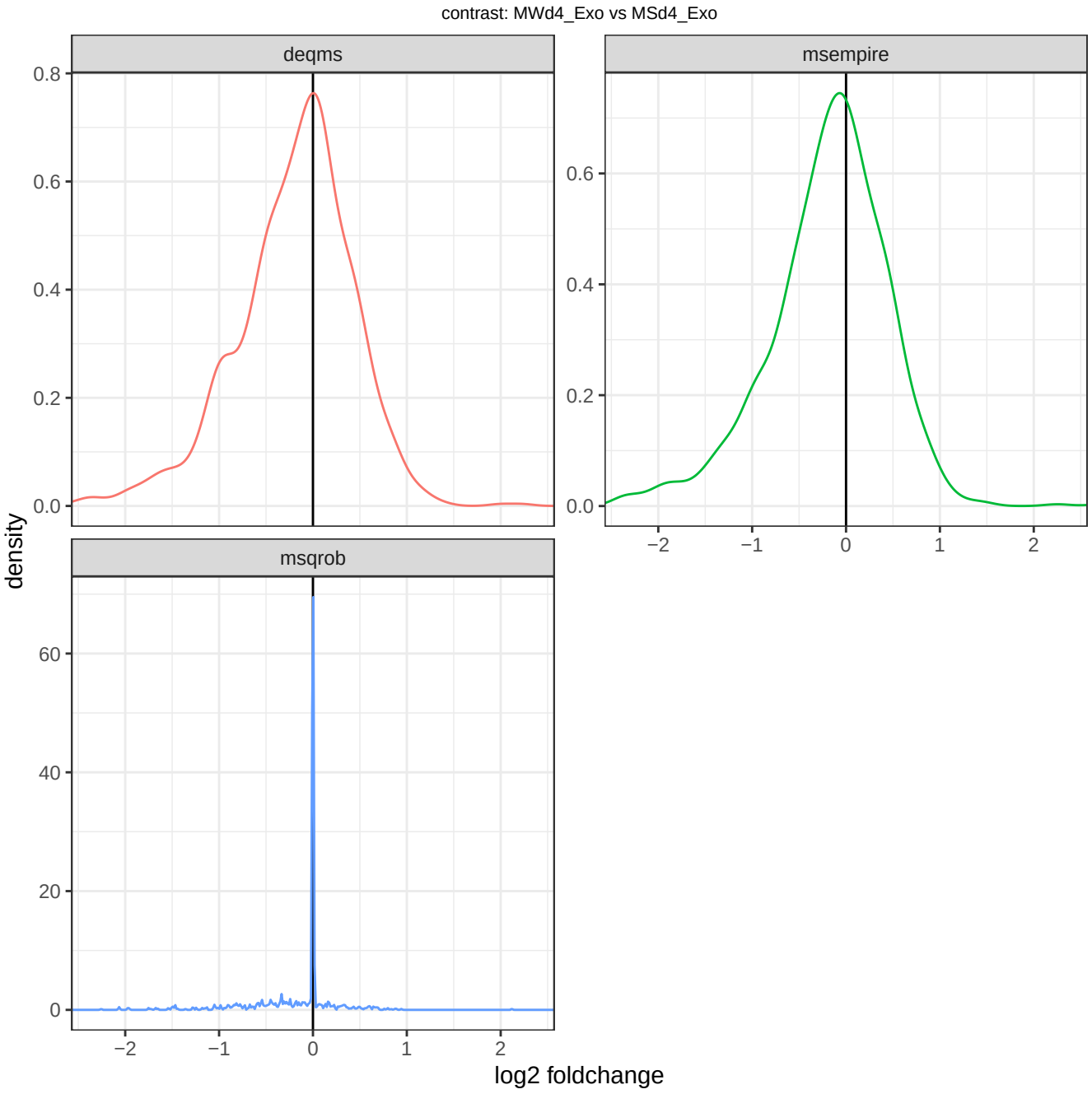

msempire @ contrast: MWd4_Exo vs MSd4_Exo
data as−is, no labels
data as−is, label 25 best qvalue
limited x− and y−axis, no labels
limited x− and y−axis, label 25 best qvalue
−log10 FDR adjusted p−value
log2 fold−change
ALDOA
MYH3
RPS8
KRT6A
FLNC
HSPG2
KRT10
LMNA
KRT16
NCL
ATP5ME
HSPD1
RPL24
PLEC
SDCBP
HSPE1
TPM1
AHNAK
VIM
CSRP1
KRT42
KRT5
PTGFRN
PPID
KRT2
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits


msqrob @ contrast: MWd4_Exo vs MSd4_Exo
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
MYH3
ALDOA
SOD1
RPL24
VCL
TAGLN2
RPS19
KRT6A
TPI1
NCAM1
PLG
AHNAK
FLNA
ATP5F1A
SPTBN1
S100A4
TPD52L2
LAP3
VIM
RPL6
TNNI1
EEF2
H4C16
SH3BGRL3
SRL
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



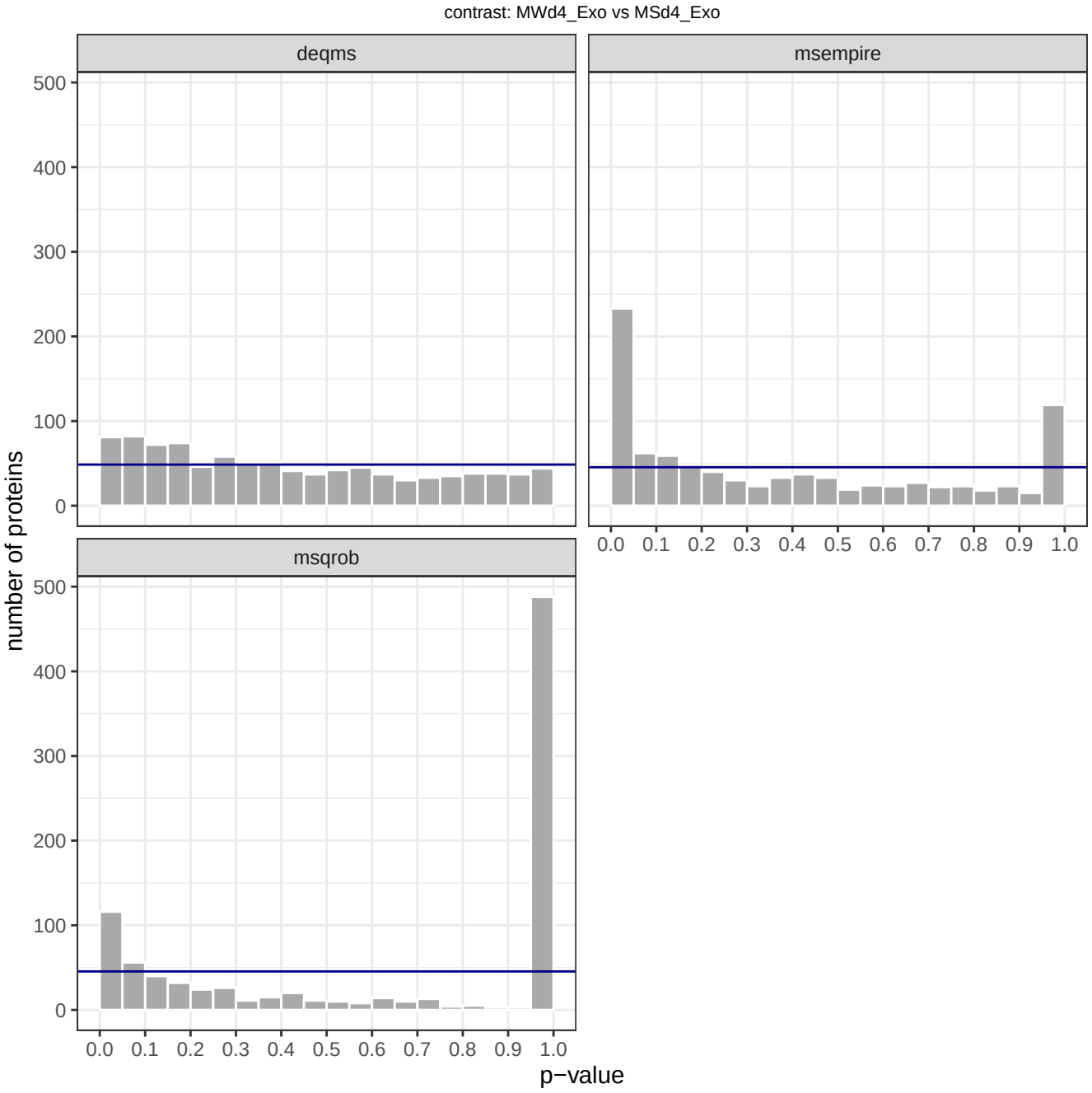
2.2.3 p-value distribution

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

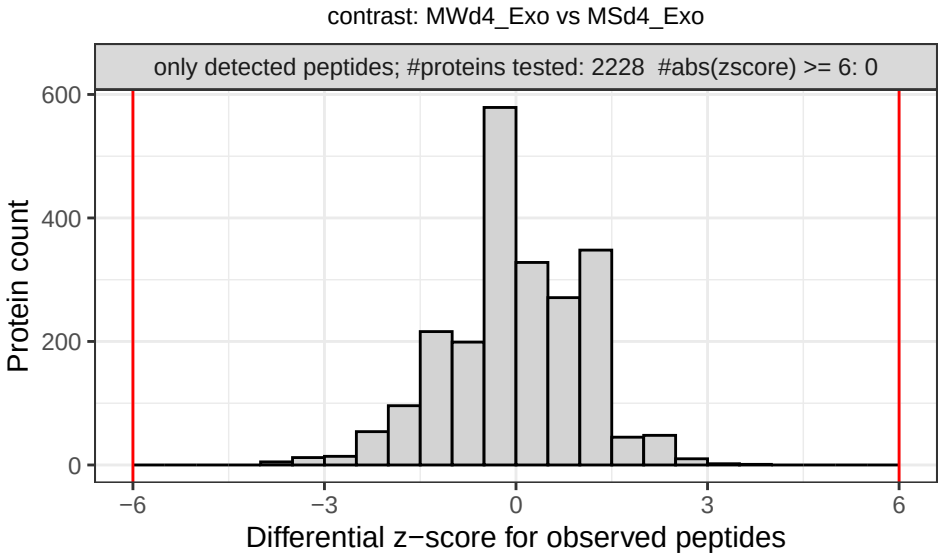


2.2.4 differential detect

Some proteins may not have peptides with sufficient data points over samples to be used for differential expression analysis (DEA), 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 study, those are interesting proteins. For this purpose, a basic metric for differential testing based on observed peptide counts is provided in MS-DAP as a situational tool. Importantly, this approach is less robust than DEA and the criteria to find a reliable set of differentially expressed proteins (by differential testing) might differ between real-world datasets.

As general guidelines for differential detection, the recommended default setting is to filter for proteins that were observed with at least 2 peptides in at least 3 replicates (or 50% of replicates, whichever number is greater). Use the plots of differential detection z-score histograms to observe the overall distribution and start your data exploration at proteins with the strongest z-scores to find desired z-score cutoffs (typically an absolute z-score of 5 or higher, but this is not set in stone for all datasets). This works best for DDA experiments, for DIA only the most extreme values are informative in many cases (e.g. proteins exclusively identified in condition A & found in nearly all replicates of condition A).

Below figure shows the distribution of these scores with thresholds at 6 std. Both the z-scores and the counts these are based upon are available in the statistical result Excel table.



2.3 MWd4_S vs MSd4_S

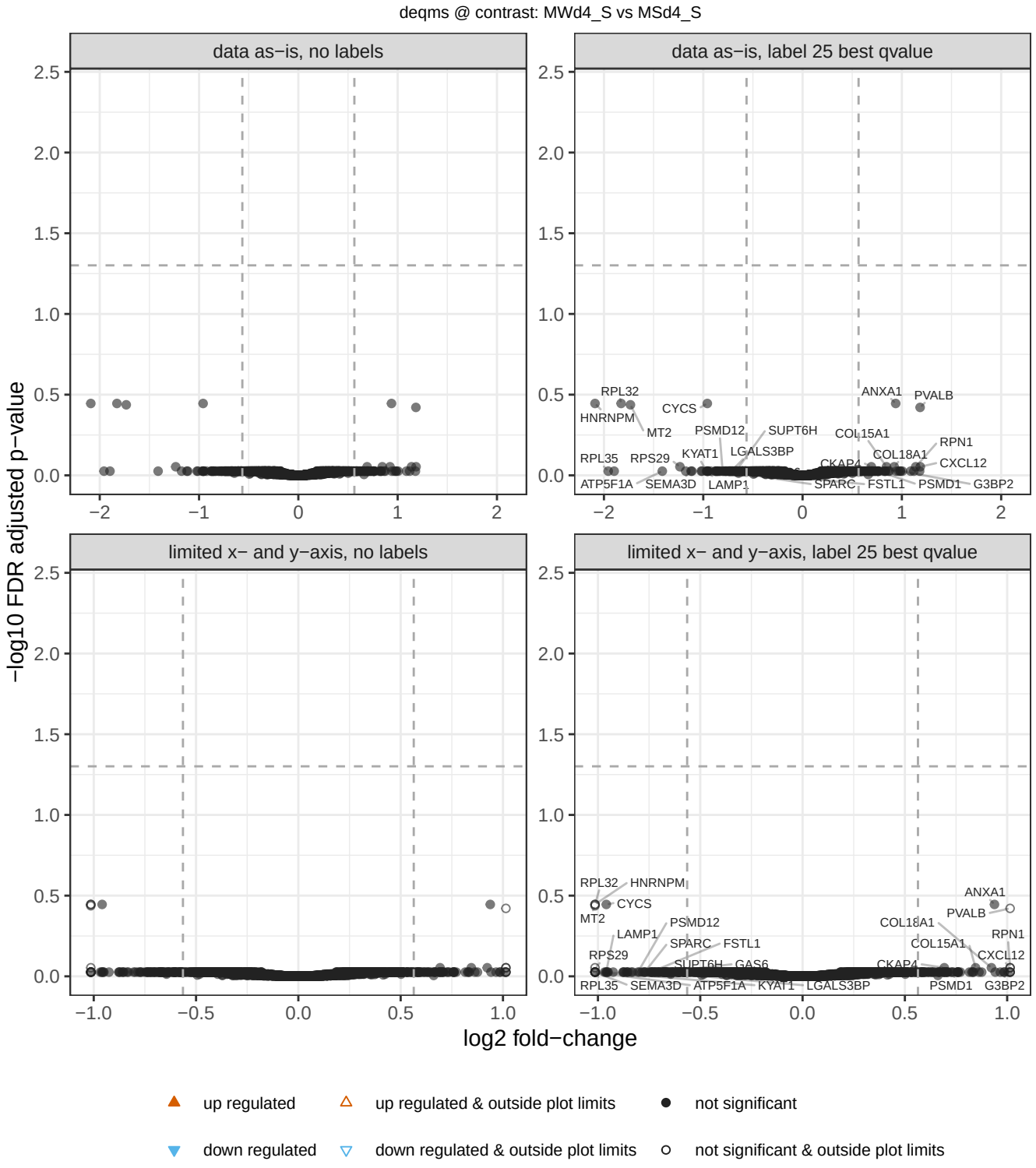
- **user setting:** using 'filter by contrast' peptide filtering approach
- 16772 peptides in 2614 proteins remain in the current contrast after peptide filters and are used for the statistical analysis in this section
- qvalue threshold: 0.05
- log2 foldchange threshold: 0.564

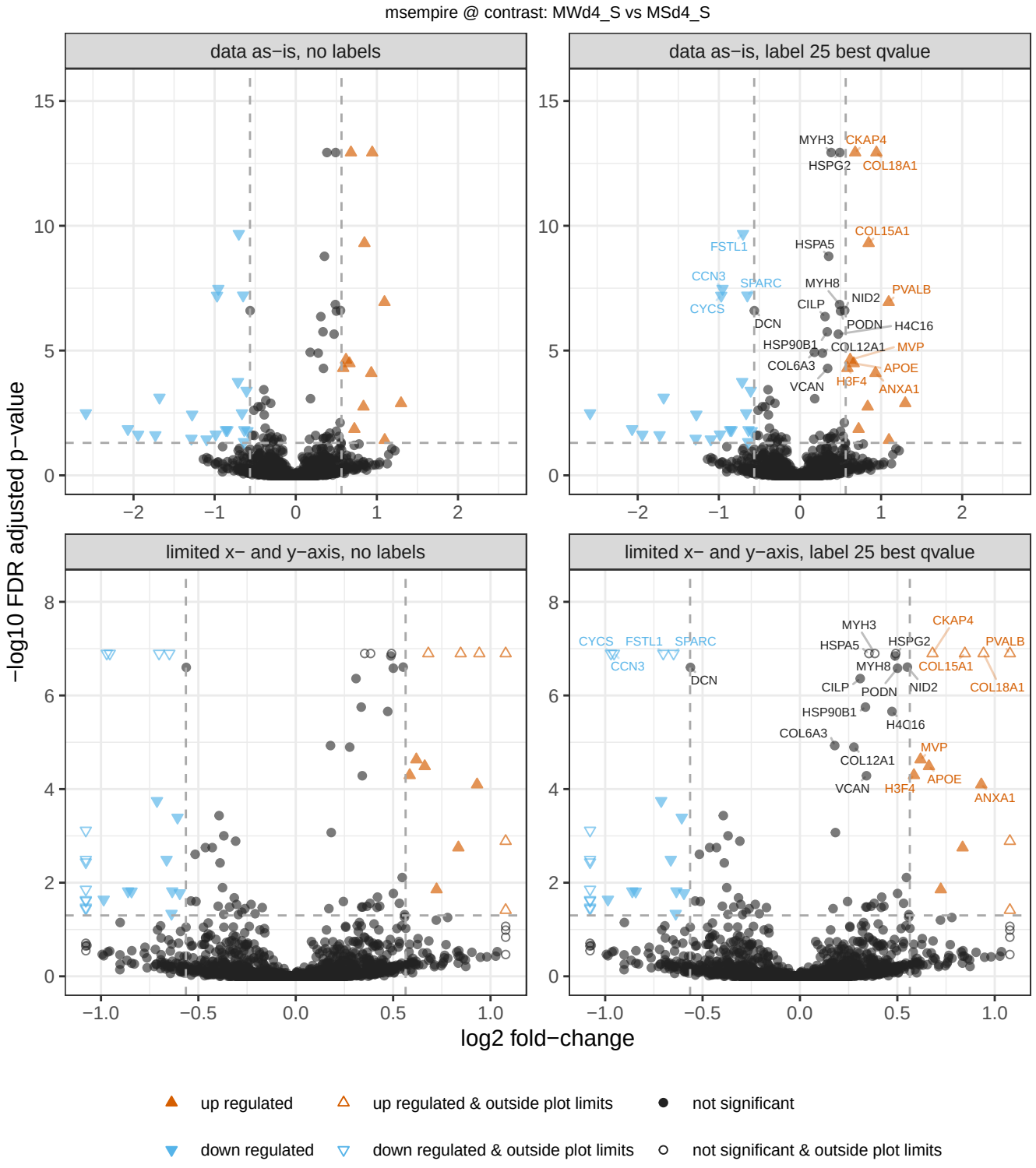
2.3.1 volcano

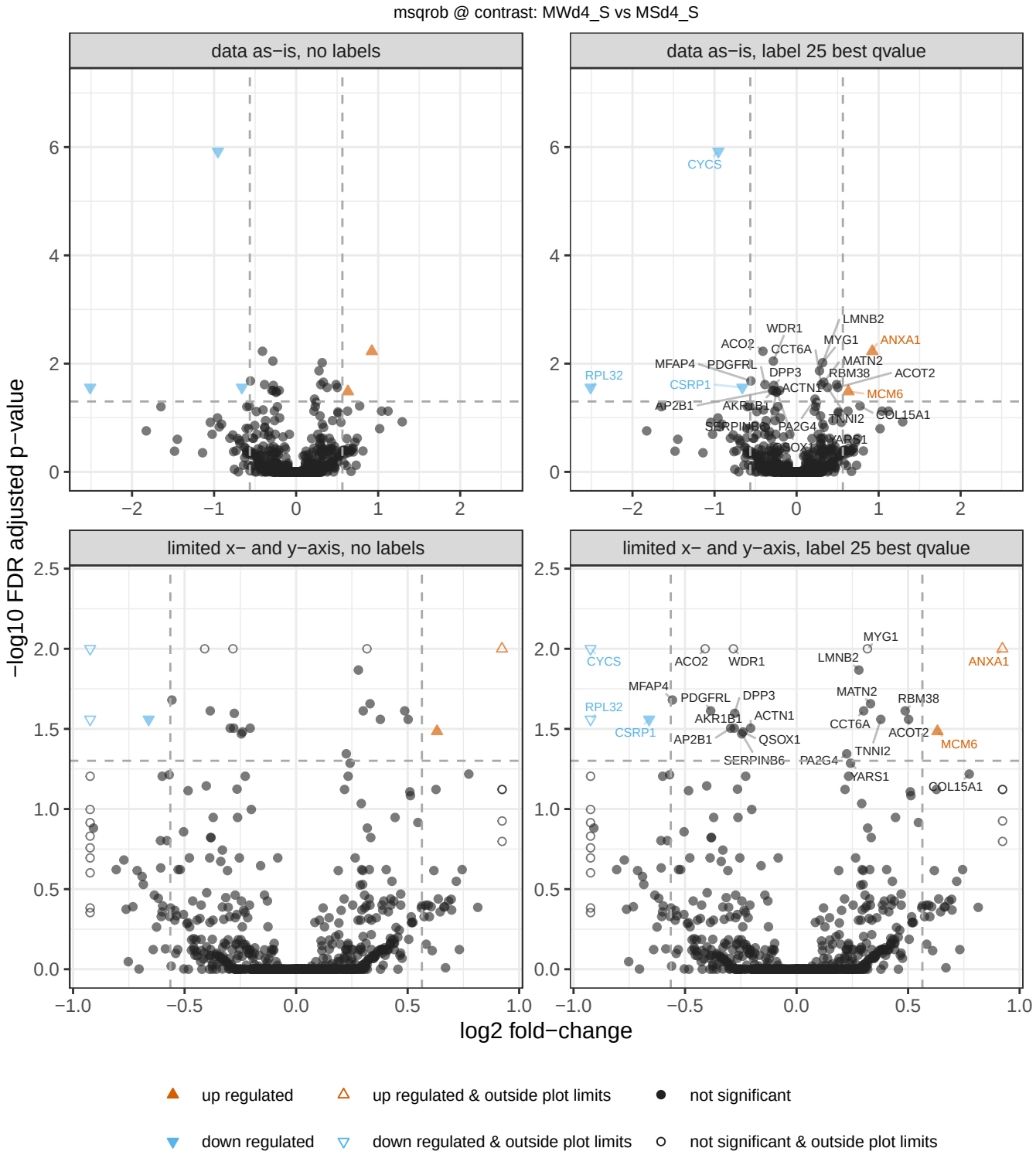
The plot title shows the statistical model and contrast (sample groups in the comparison). Left- and right-side figure panels on each row represent the same figure without and with labels for the 25 proteins with lowest p-value.

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: MWd4_S vs MSd4_S
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
RPL32
HNRNPM
CYCS
MT2
PSMD12
SUPT6H
COL15A1
ANXA1
PVALB
RPN1
RPL35
RPS29
KYAT1
LGALS3BP
COL18A1
CXCL12
CKAP4
ATP5F1A
SEMA3D
LAMP1
SPARC
FSTL1
PSMD1
G3BP2
GAS6
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits

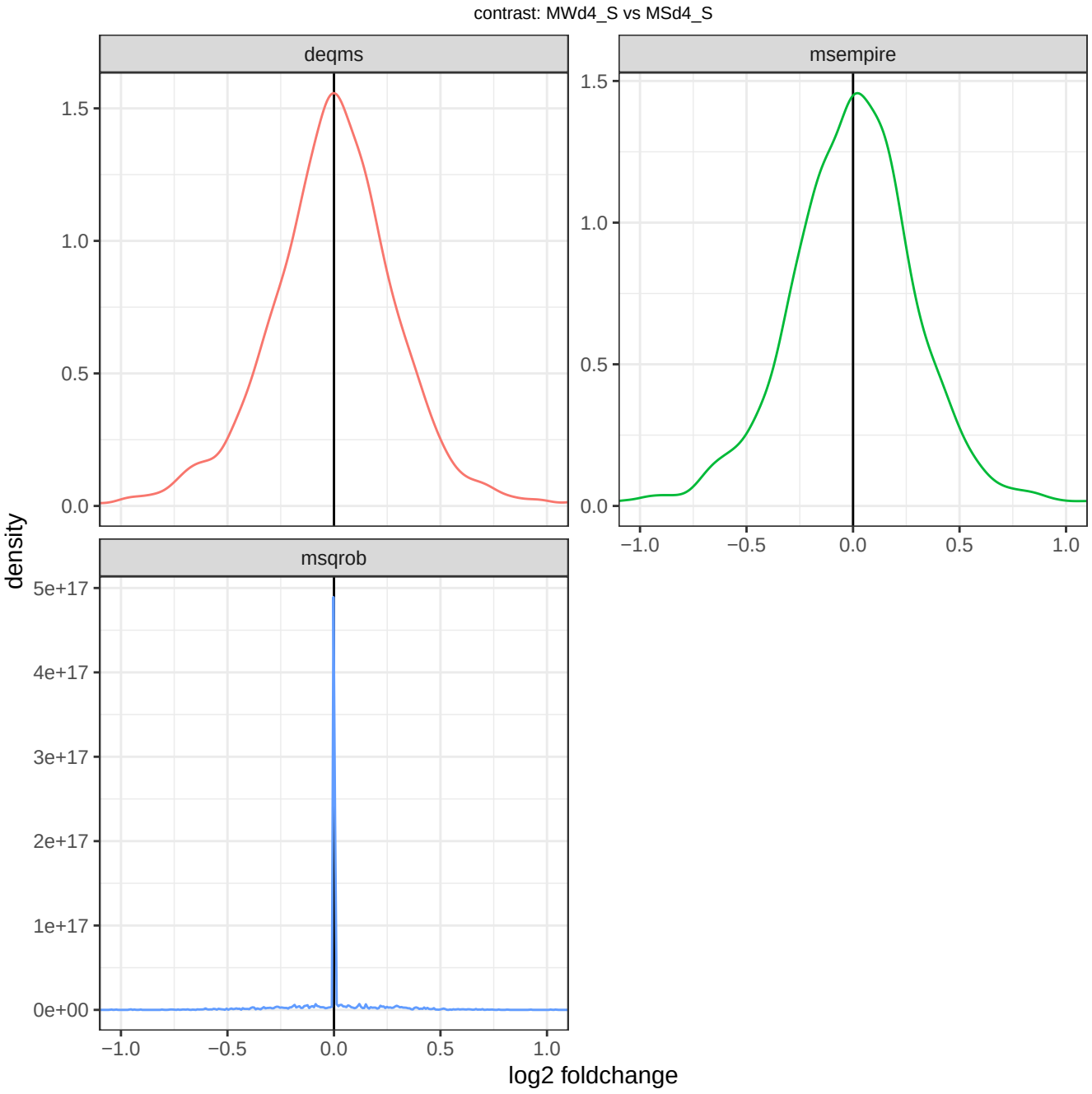

msempire @ contrast: MWd4_S vs MSd4_S
data as−is, no labels
data as−is, label 25 best qvalue
limited x− and y−axis, no labels
limited x− and y−axis, label 25 best qvalue
−log10 FDR adjusted p−value
log2 fold−change
MYH3
CKAP4
HSPG2
COL18A1
COL15A1
FSTL1
HSPA5
CCN3
SPARC
MYH8
PVALB
CILP
NID2
CYCS
DCN
PODN
H4C16
HSP90B1
COL12A1
MVP
COL6A3
APOE
H3F4
VCAN
ANXA1
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits


msqrob @ contrast: MWd4_S vs MSd4_S
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
CYCS
RPL32
CSRP1
AP2B1
MFAP4
PDGFRL
ACO2
WDR1
CCT6A
DPP3
ACTN1
AKR1B1
QSOX1
SERPINB6
PA2G4
LMNB2
MYG1
ANXA1
MATN2
RBM38
ACOT2
MCM6
TNNI2
COL15A1
YARS1
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



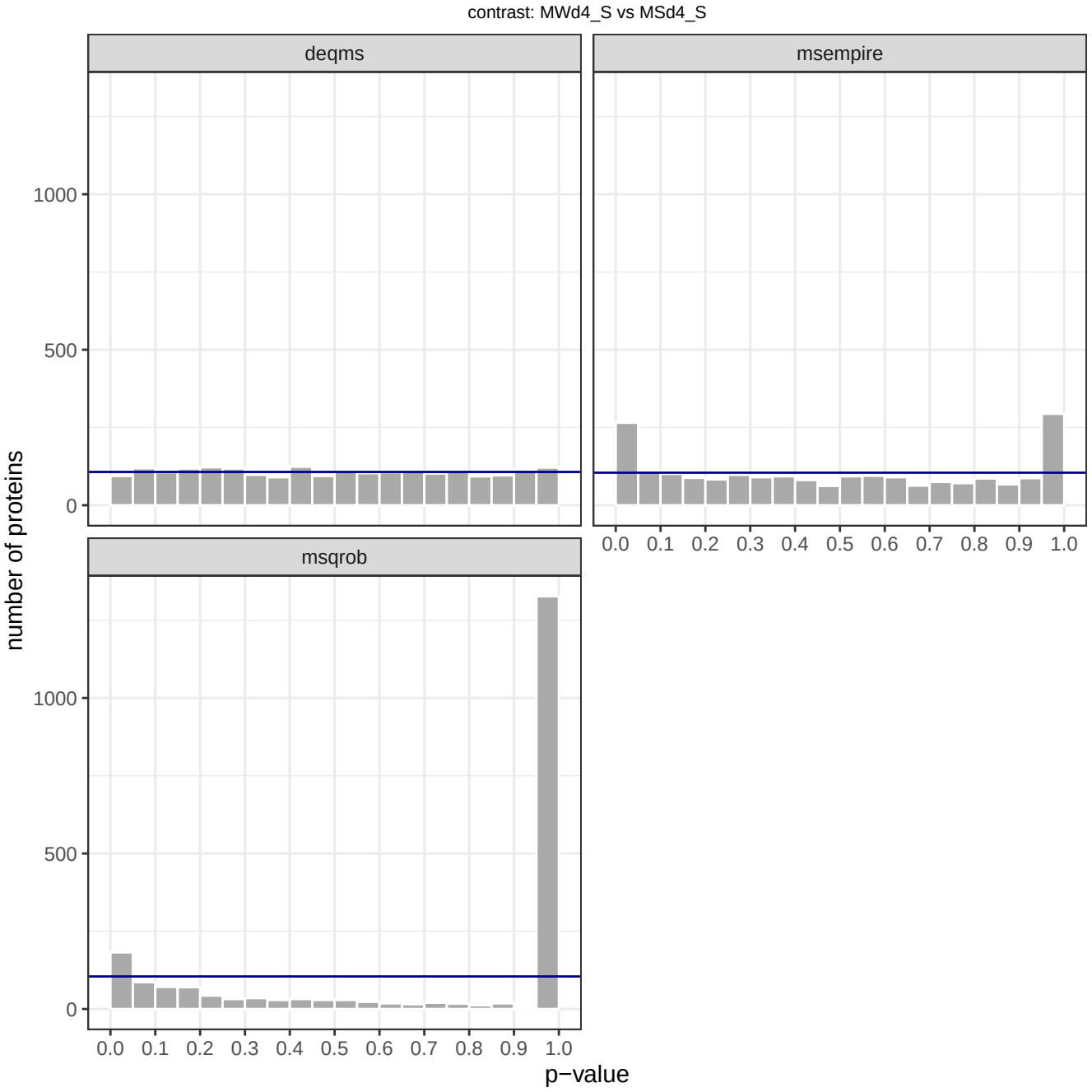
2.3.3 p-value distribution

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

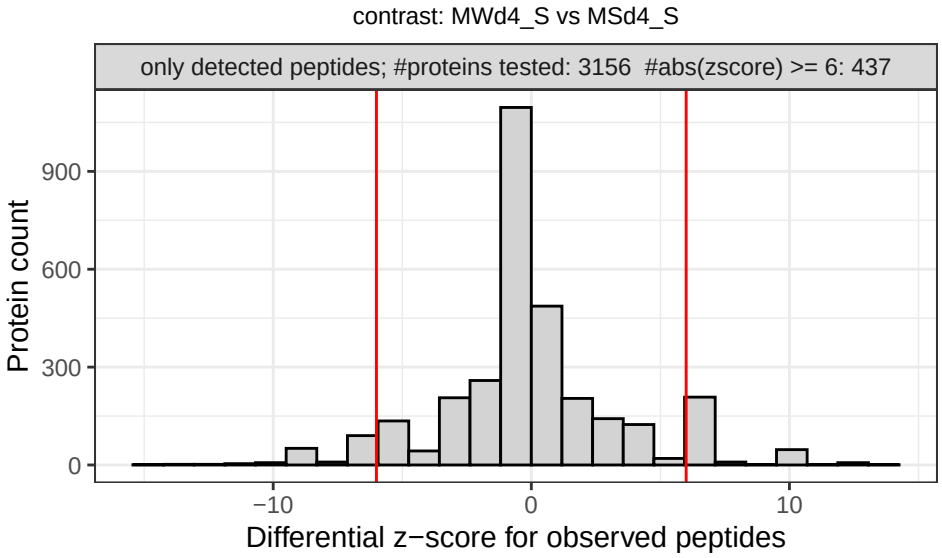


2.3.4 differential detect

Some proteins may not have peptides with sufficient data points over samples to be used for differential expression analysis (DEA), 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 study, those are interesting proteins. For this purpose, a basic metric for differential testing based on observed peptide counts is provided in MS-DAP as a situational tool. Importantly, this approach is less robust than DEA and the criteria to find a reliable set of differentially expressed proteins (by differential testing) might differ between real-world datasets.

As general guidelines for differential detection, the recommended default setting is to filter for proteins that were observed with at least 2 peptides in at least 3 replicates (or 50% of replicates, whichever number is greater). Use the plots of differential detection z-score histograms to observe the overall distribution and start your data exploration at proteins with the strongest z-scores to find desired z-score cutoffs (typically an absolute z-score of 5 or higher, but this is not set in stone for all datasets). This works best for DDA experiments, for DIA only the most extreme values are informative in many cases (e.g. proteins exclusively identified in condition A & found in nearly all replicates of condition A).

Below figure shows the distribution of these scores with thresholds at 6 std. Both the z-scores and the counts these are based upon are available in the statistical result Excel table.



2.4 FW_L vs FS_L

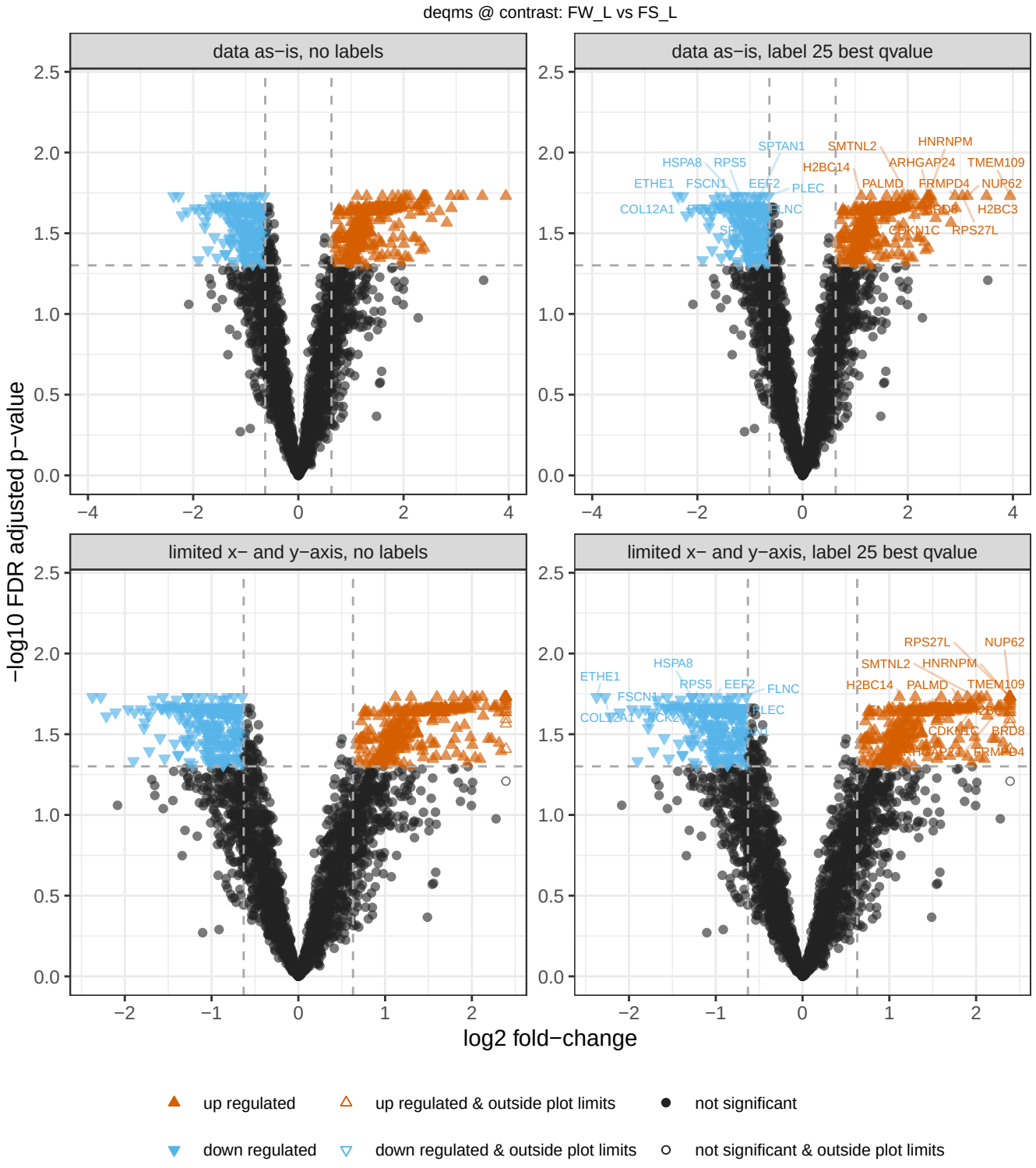
- **user setting:** using 'filter by contrast' peptide filtering approach
- 18456 peptides in 3874 proteins remain in the current contrast after peptide filters and are used for the statistical analysis in this section
- qvalue threshold: 0.05
- log2 foldchange threshold: 0.63

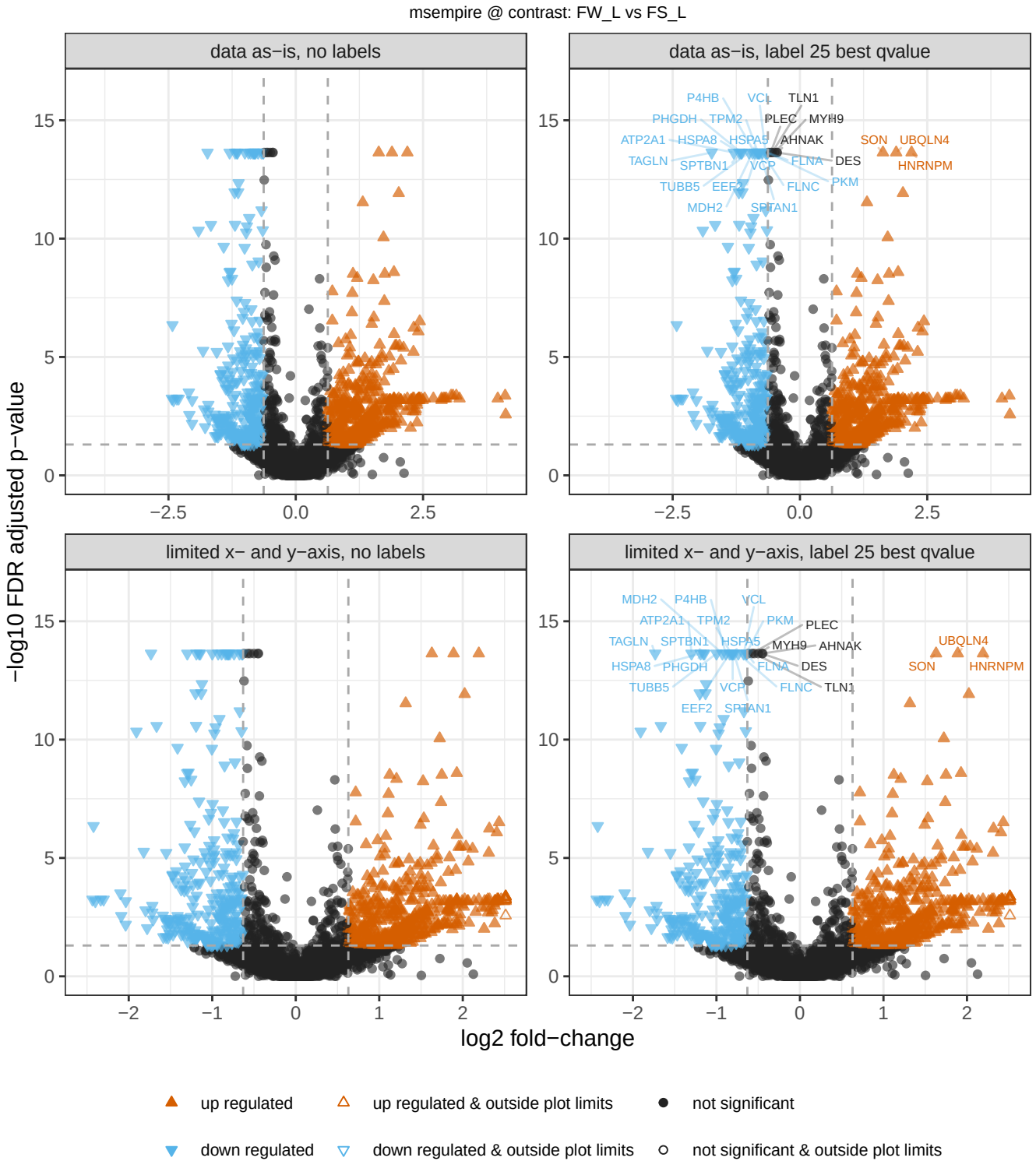
2.4.1 volcano

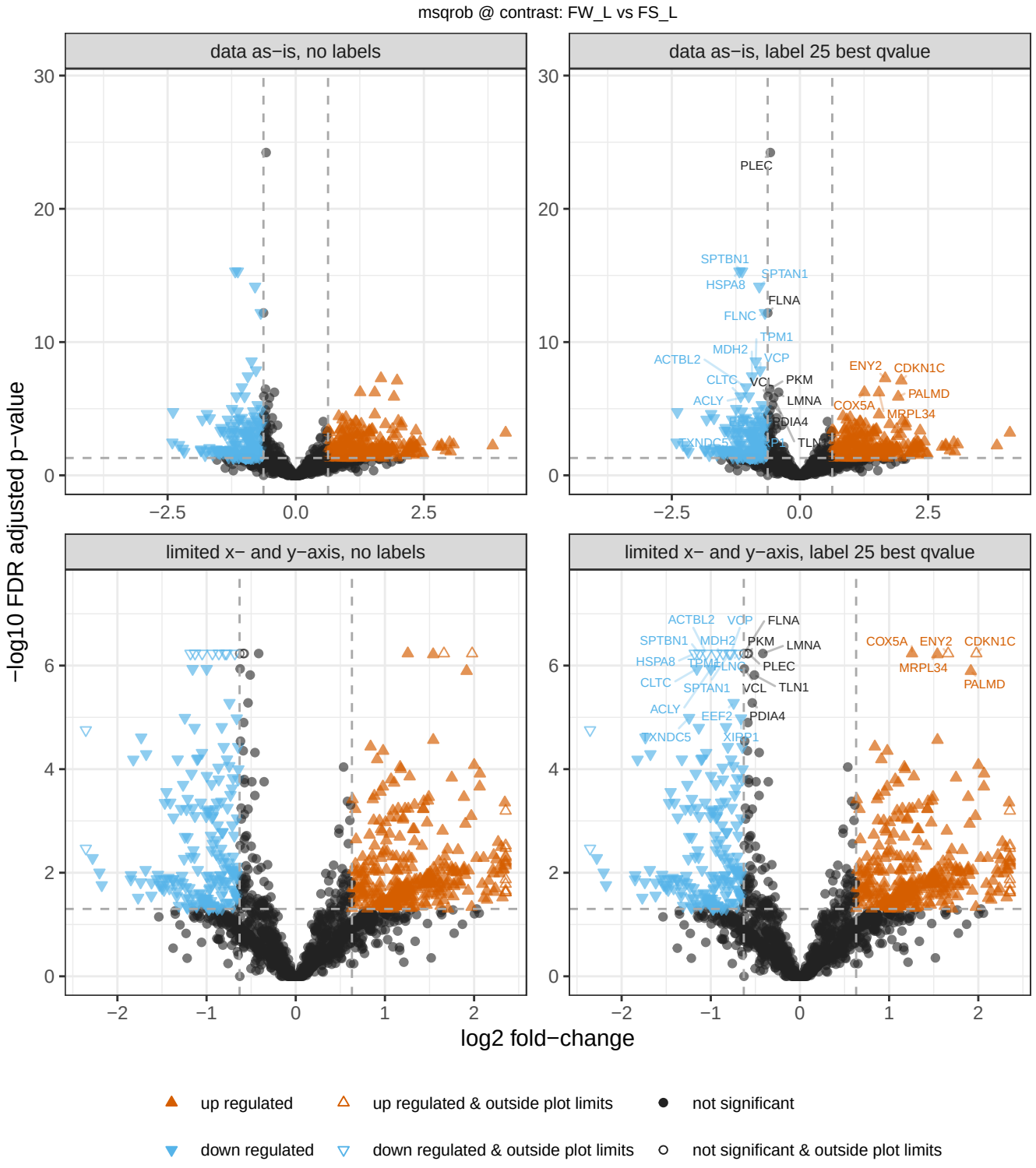
The plot title shows the statistical model and contrast (sample groups in the comparison). Left- and right-side figure panels on each row represent the same figure without and with labels for the 25 proteins with lowest p-value.

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: FW_L vs FS_L
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
SPTAN1
SMTNL2
HNRNPM
HSPA8
RPS5
ARHGAP24
TMEM109
H2BC14
ETHE1
FSCN1
EEF2
PLEC
PALMD
FRMPD4
NUP62
COL12A1
FLNC
BRD8
H2BC3
CDKN1C
RPS27L
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits

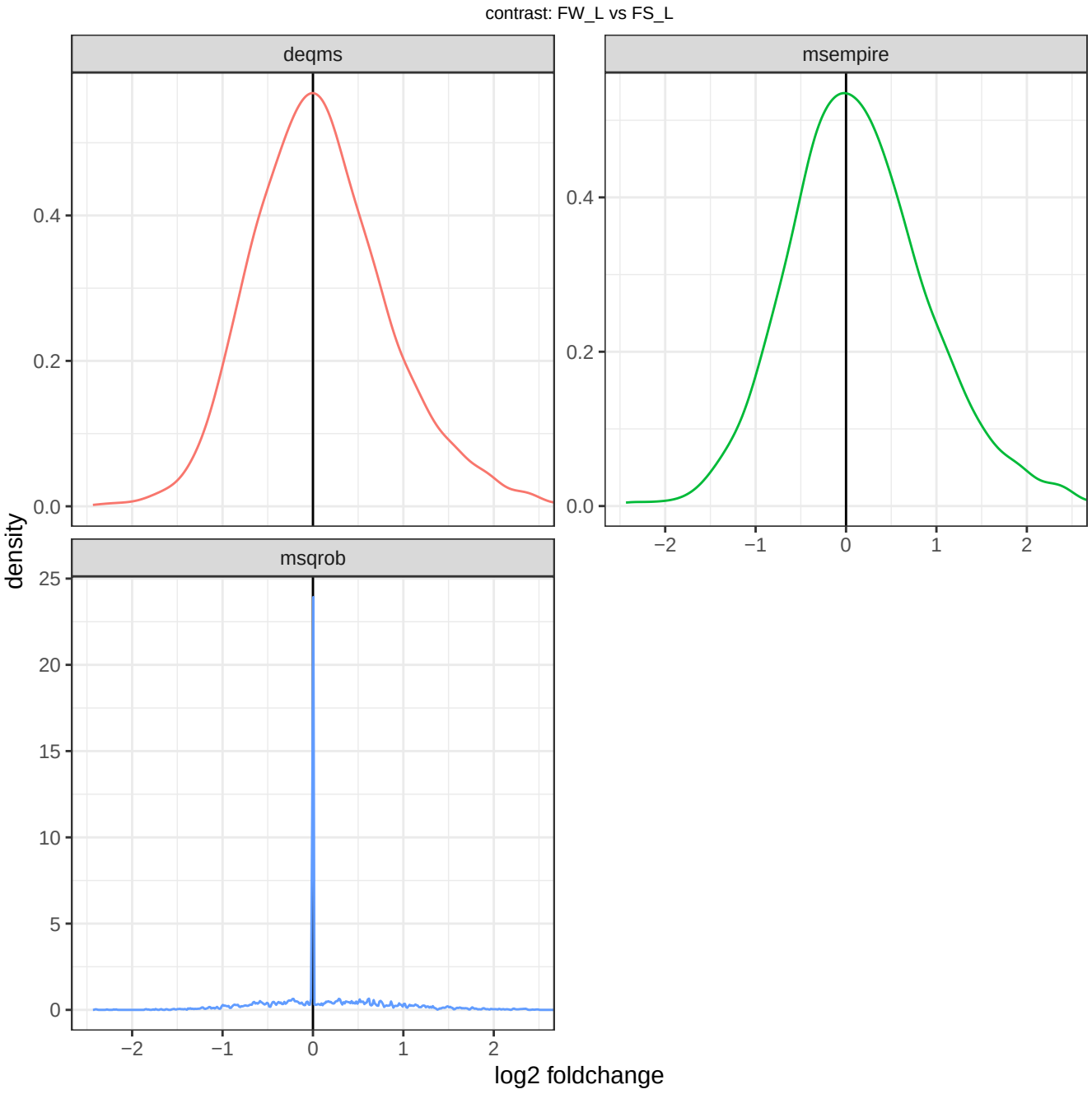

msempire @ contrast: FW_L vs FS_L
data as−is, no labels
data as−is, label 25 best qvalue
limited x− and y−axis, no labels
limited x− and y−axis, label 25 best qvalue
−log10 FDR adjusted p−value
log2 fold−change
P4HB
VCL
TLN1
PHGDH
TPM2
PLEC
MYH9
ATP2A1
HSPA8
HSPA5
AHNAK
SON
UBQLN4
TAGLN
SPTBN1
VCP
FLNA
DES
HNRNPM
TUBB5
EEF2
FLNC
PKM
MDH2
SPTAN1
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits


msqrob @ contrast: FW_L vs FS_L
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
PLEC
SPTBN1
SPTAN1
HSPA8
FLNA
FLNC
TPM1
MDH2
VCP
ACTBL2
CLTC
VCL
PKM
ACLY
LMNA
PDIA4
TXNDC5
XIRP1
TLN1
ENY2
CDKN1C
PALMD
COX5A
MRPL34
EEF2
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits

2.4.2 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



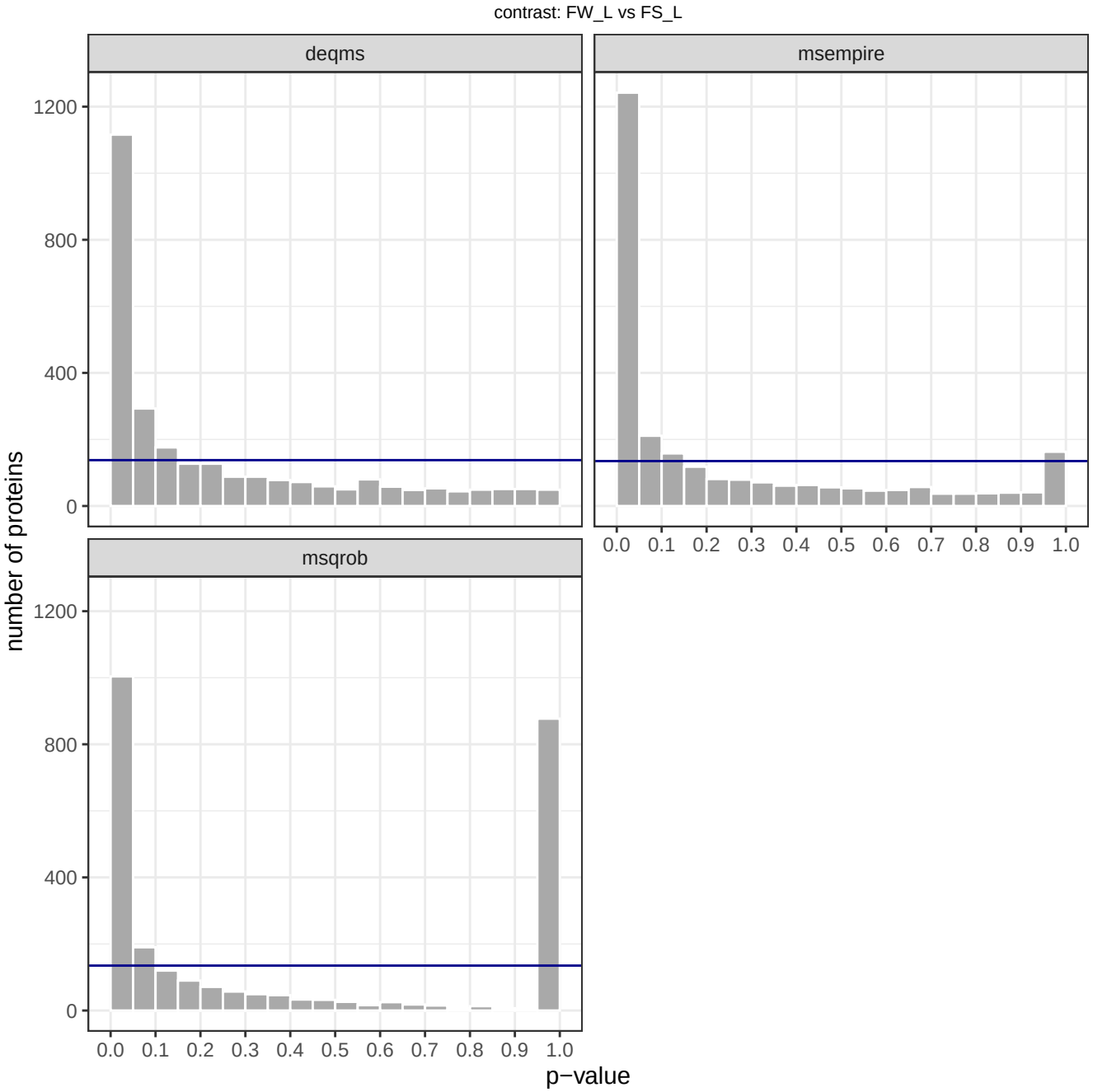
2.4.3 p-value distribution

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

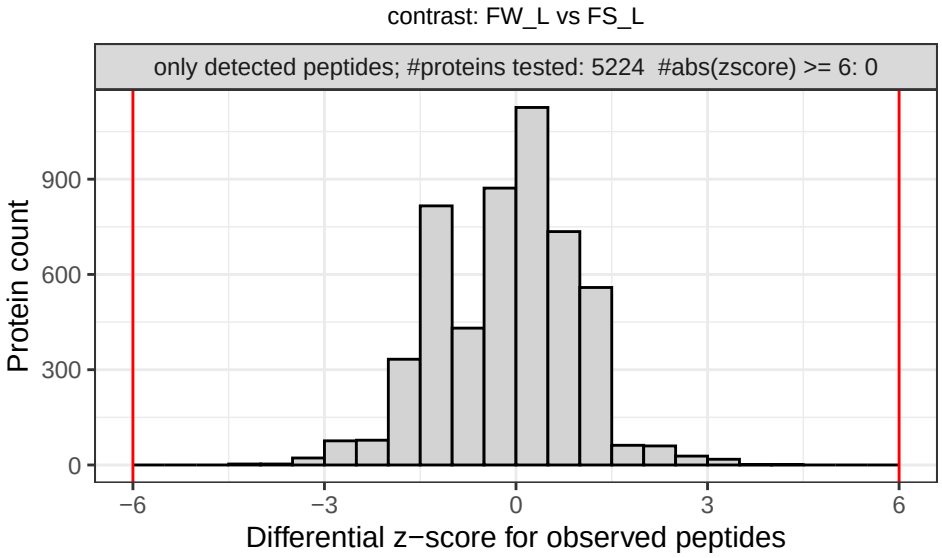


2.4.4 differential detect

Some proteins may not have peptides with sufficient data points over samples to be used for differential expression analysis (DEA), 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 study, those are interesting proteins. For this purpose, a basic metric for differential testing based on observed peptide counts is provided in MS-DAP as a situational tool. Importantly, this approach is less robust than DEA and the criteria to find a reliable set of differentially expressed proteins (by differential testing) might differ between real-world datasets.

As general guidelines for differential detection, the recommended default setting is to filter for proteins that were observed with at least 2 peptides in at least 3 replicates (or 50% of replicates, whichever number is greater). Use the plots of differential detection z-score histograms to observe the overall distribution and start your data exploration at proteins with the strongest z-scores to find desired z-score cutoffs (typically an absolute z-score of 5 or higher, but this is not set in stone for all datasets). This works best for DDA experiments, for DIA only the most extreme values are informative in many cases (e.g. proteins exclusively identified in condition A & found in nearly all replicates of condition A).

Below figure shows the distribution of these scores with thresholds at 6 std. Both the z-scores and the counts these are based upon are available in the statistical result Excel table.



2.5 FW_S vs FS_S

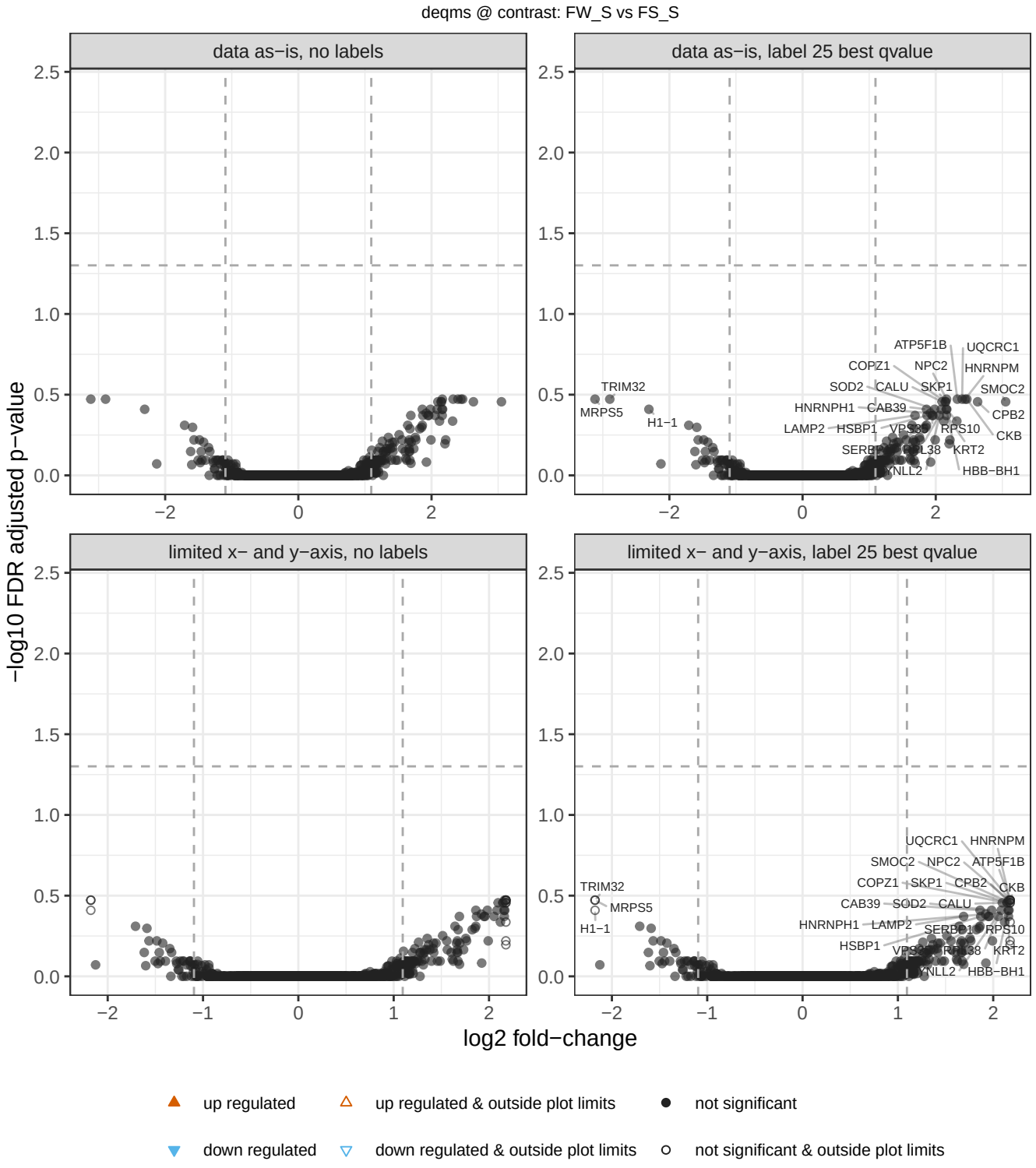
- **user setting:** using 'filter by contrast' peptide filtering approach
- 13506 peptides in 2252 proteins remain in the current contrast after peptide filters and are used for the statistical analysis in this section
- qvalue threshold: 0.05
- log2 foldchange threshold: 1.095

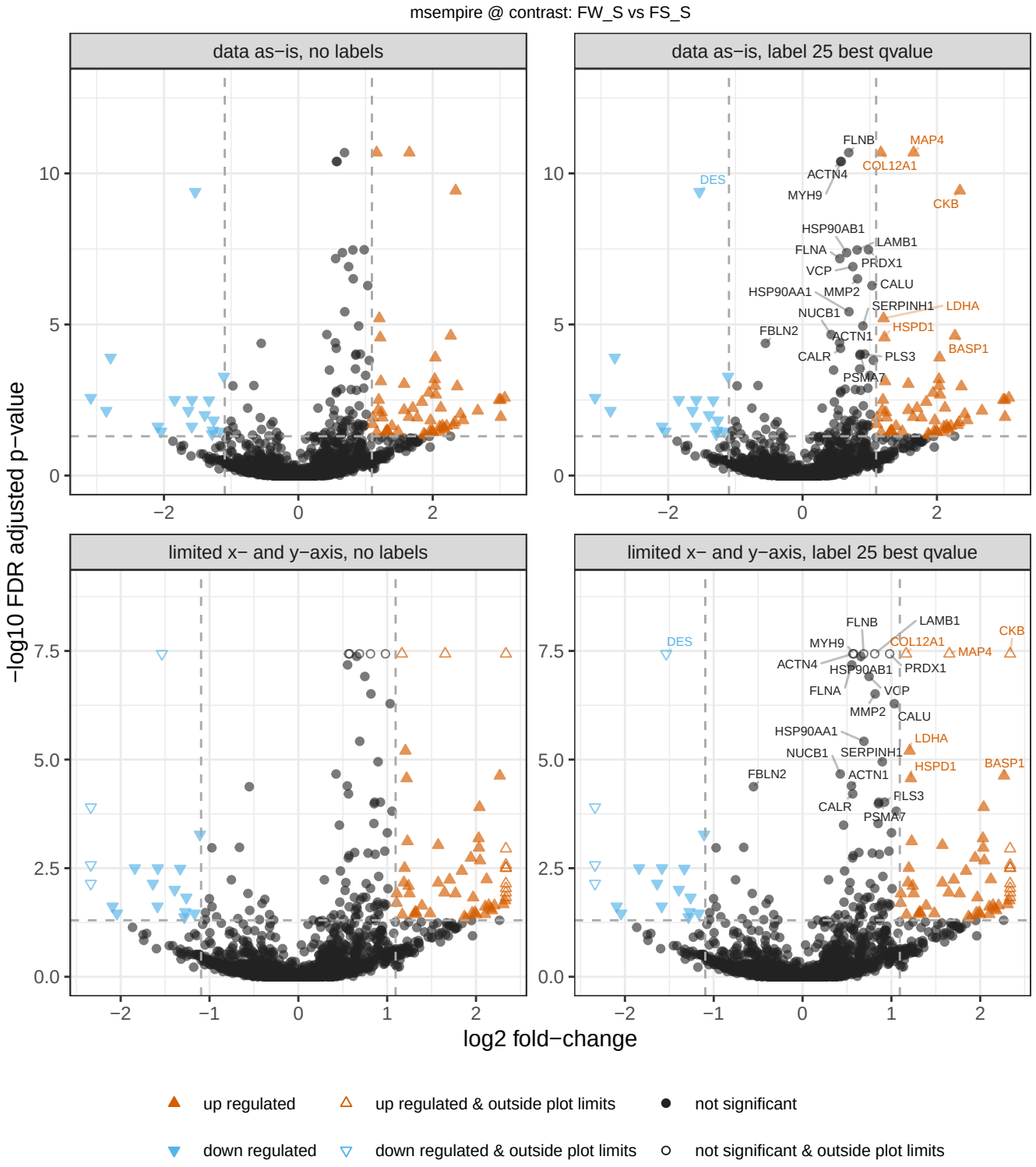
2.5.1 volcano

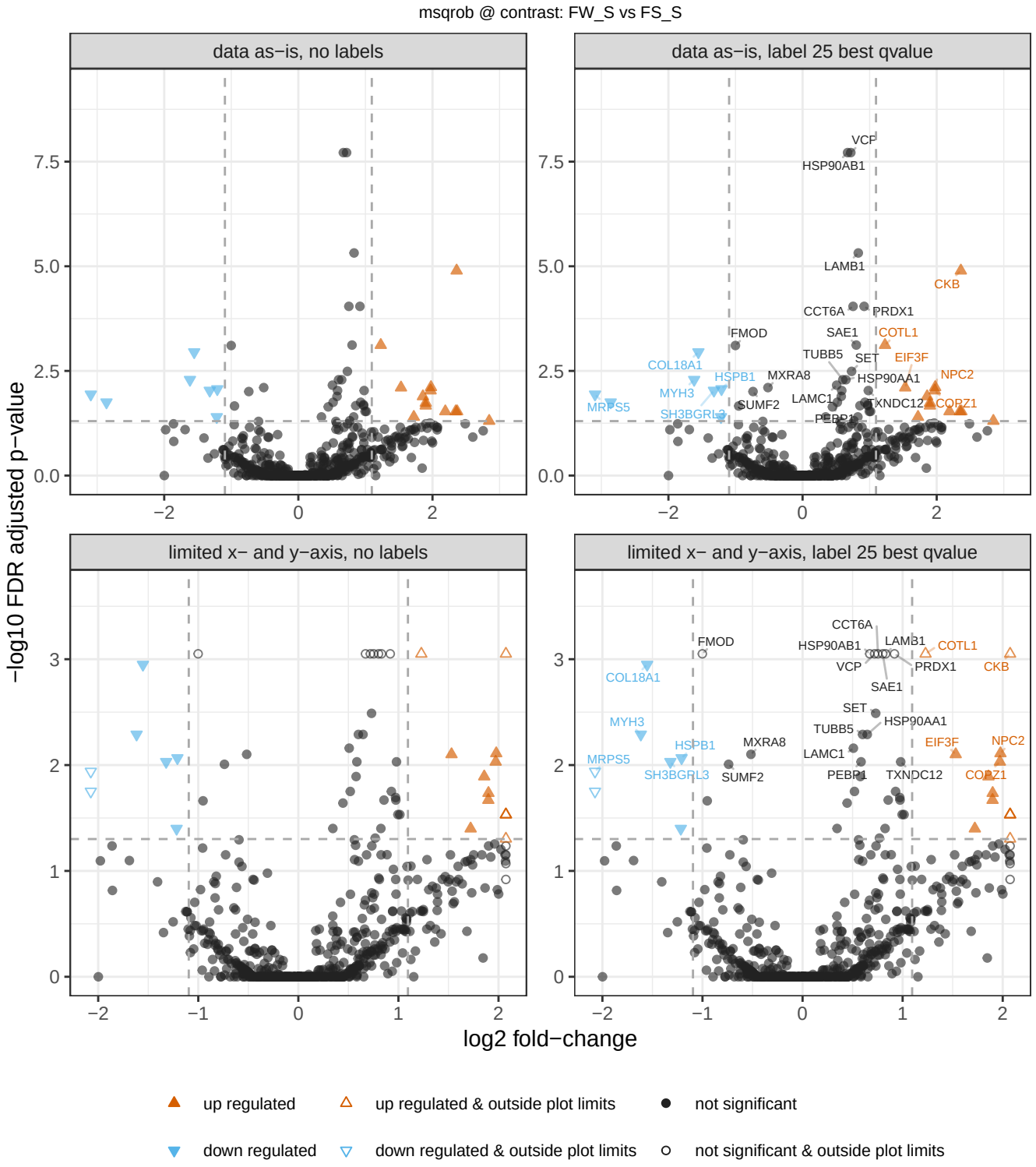
The plot title shows the statistical model and contrast (sample groups in the comparison). Left- and right-side figure panels on each row represent the same figure without and with labels for the 25 proteins with lowest p-value.

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: FW_S vs FS_S
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
TRIM32
MRPS5
H1−1
ATP5F1B
UQCRC1
COPZ1
NPC2
HNRNPM
SOD2
CALU
SKP1
SMOC2
HNRNPH1
CAB39
CPB2
LAMP2
HSBP1
VPS35
RPS10
CKB
SERBP1
RPL38
KRT2
YNLL2
HBB−BH1
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits

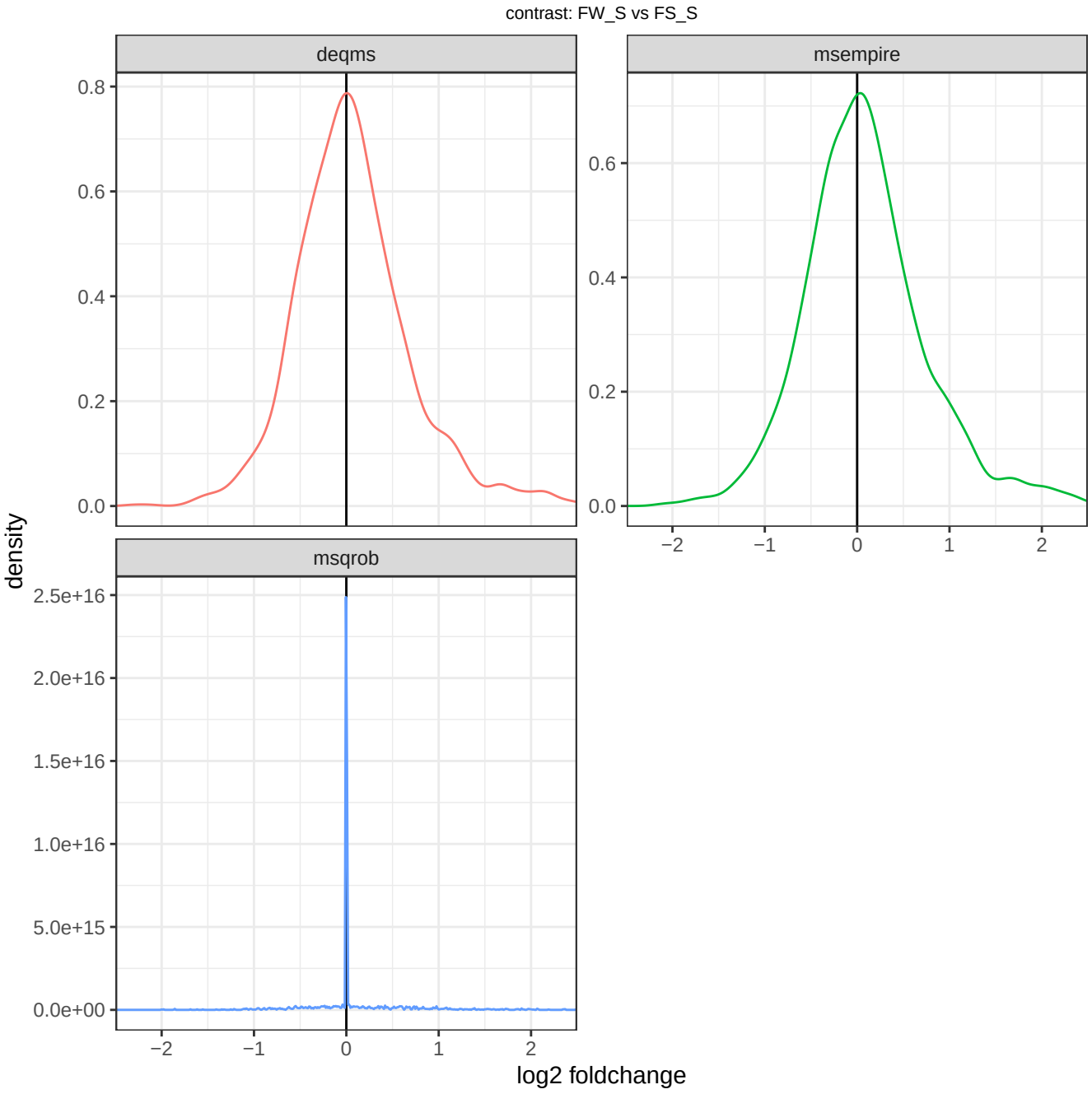

msempire @ contrast: FW_S vs FS_S
data as−is, no labels
data as−is, label 25 best qvalue
limited x− and y−axis, no labels
limited x− and y−axis, label 25 best qvalue
−log10 FDR adjusted p−value
log2 fold−change
FLNB
MAP4
COL12A1
ACTN4
DES
MYH9
CKB
HSP90AB1
LAMB1
FLNA
PRDX1
VCP
CALU
HSP90AA1
MMP2
SERPINH1
LDHA
NUCB1
HSPD1
FBLN2
ACTN1
BASP1
CALR
PLS3
PSMA7
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits


msqrob @ contrast: FW_S vs FS_S
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
VCP
HSP90AB1
LAMB1
CKB
CCT6A
PRDX1
FMOD
SAE1
COTL1
TUBB5
SET
EIF3F
COL18A1
HSPB1
MXRA8
HSP90AA1
NPC2
MYH3
LAMC1
MRPS5
SUMF2
TXNDC12
CORZ1
SH3BGRL3
PEBP1
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits

2.5.2 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



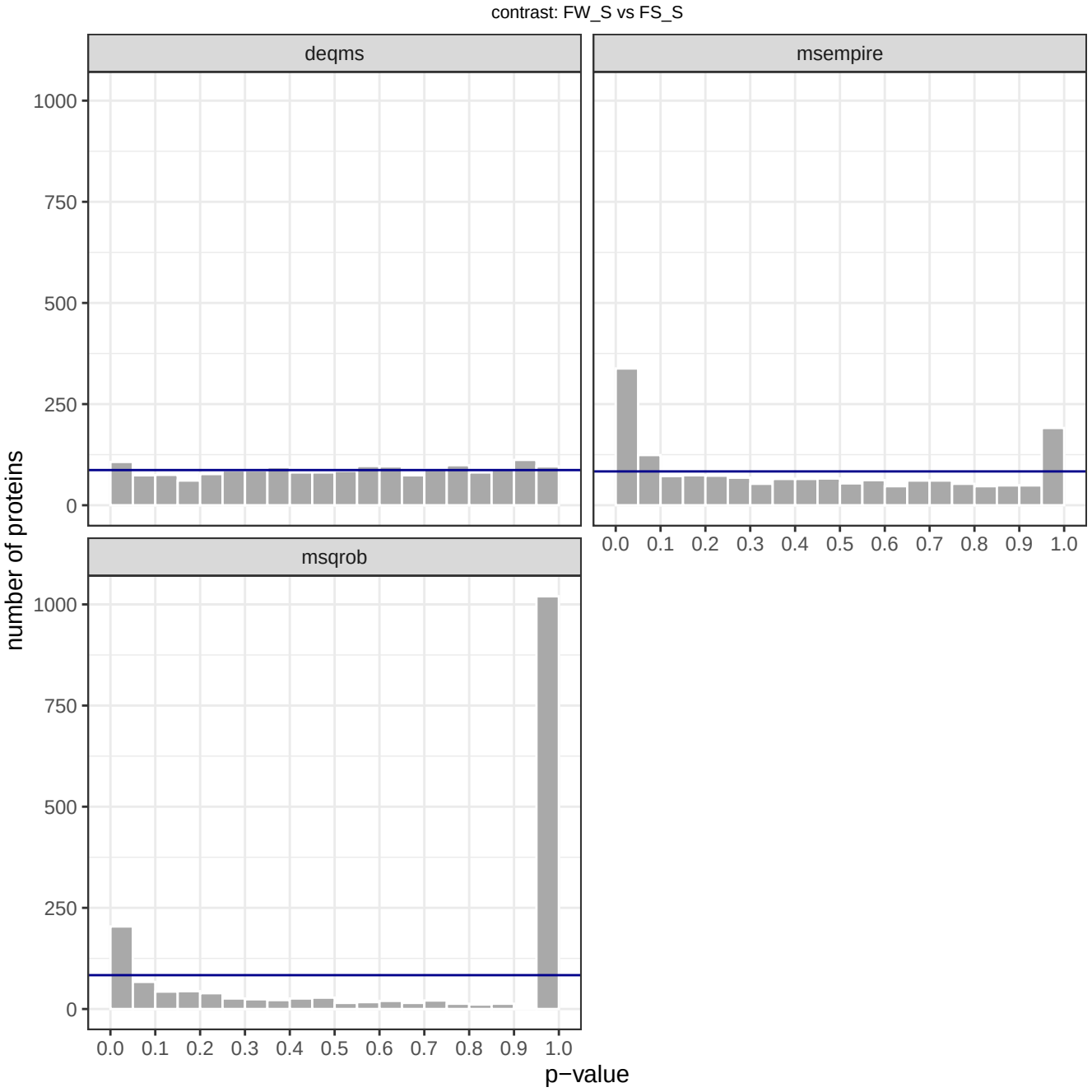
2.5.3 p-value distribution

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

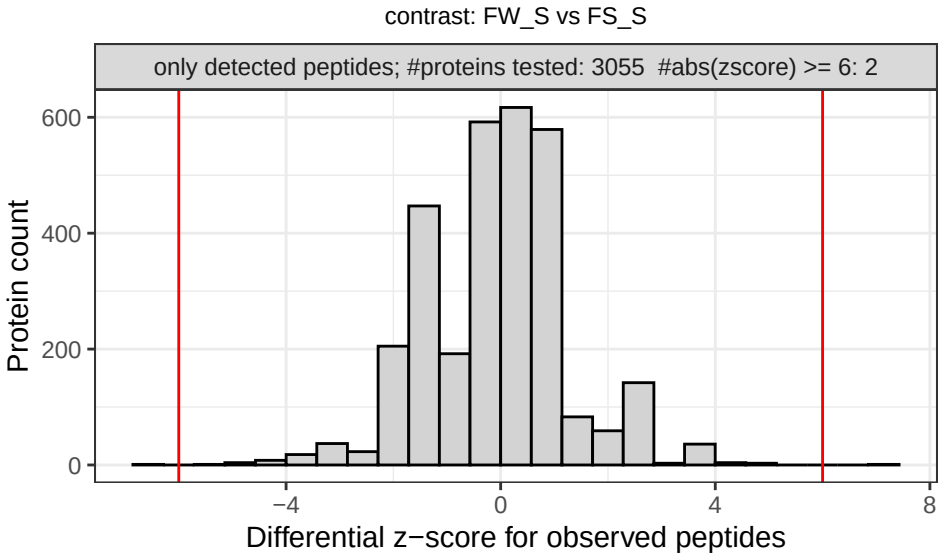


2.5.4 differential detect

Some proteins may not have peptides with sufficient data points over samples to be used for differential expression analysis (DEA), 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 study, those are interesting proteins. For this purpose, a basic metric for differential testing based on observed peptide counts is provided in MS-DAP as a situational tool. Importantly, this approach is less robust than DEA and the criteria to find a reliable set of differentially expressed proteins (by differential testing) might differ between real-world datasets.

As general guidelines for differential detection, the recommended default setting is to filter for proteins that were observed with at least 2 peptides in at least 3 replicates (or 50% of replicates, whichever number is greater). Use the plots of differential detection z-score histograms to observe the overall distribution and start your data exploration at proteins with the strongest z-scores to find desired z-score cutoffs (typically an absolute z-score of 5 or higher, but this is not set in stone for all datasets). This works best for DDA experiments, for DIA only the most extreme values are informative in many cases (e.g. proteins exclusively identified in condition A & found in nearly all replicates of condition A).

Below figure shows the distribution of these scores with thresholds at 6 std. Both the z-scores and the counts these are based upon are available in the statistical result Excel table.



2.6 MW_L vs MS_L

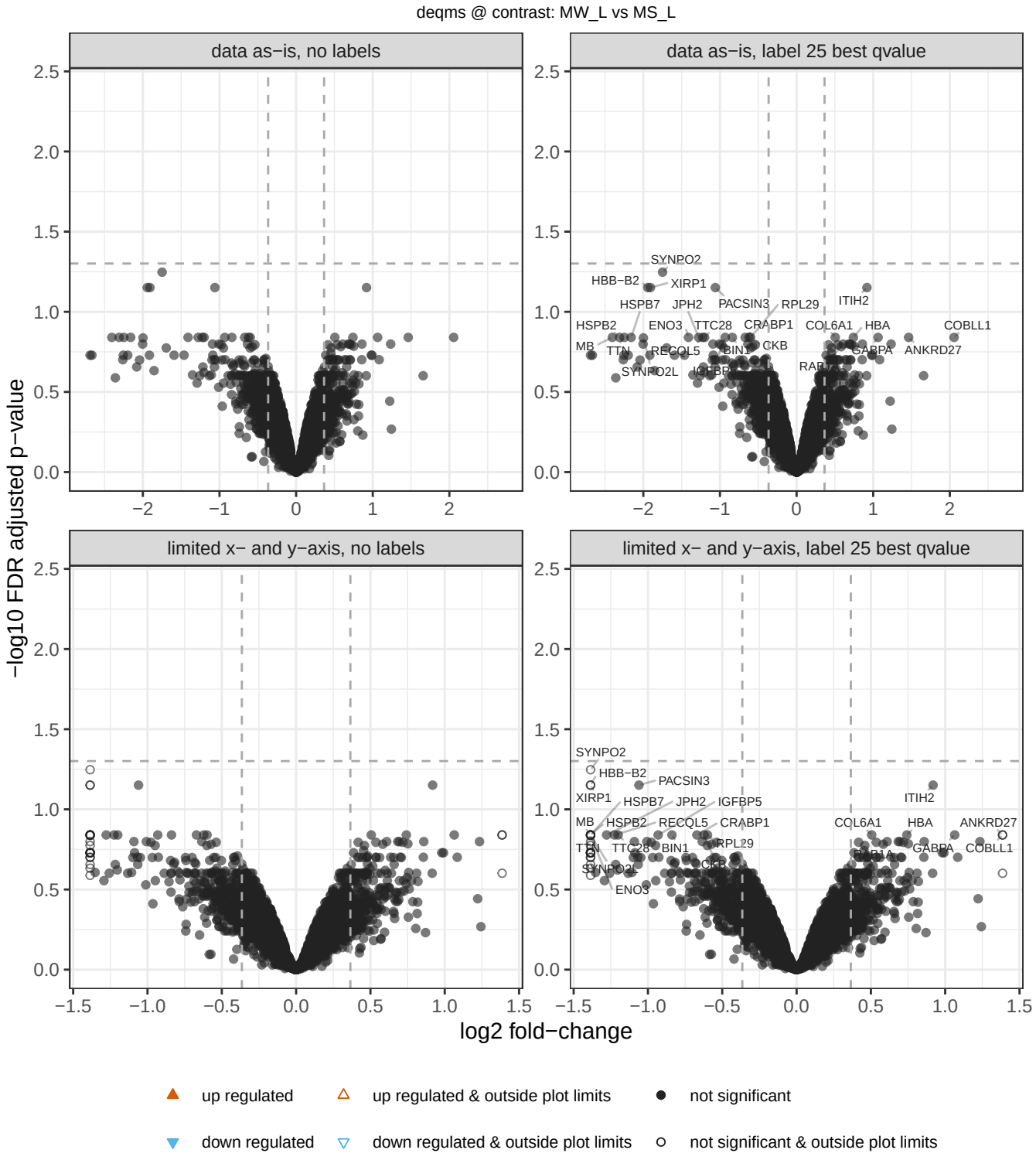
- **user setting:** using 'filter by contrast' peptide filtering approach
- 30414 peptides in 4948 proteins remain in the current contrast after peptide filters and are used for the statistical analysis in this section
- qvalue threshold: 0.05
- log2 foldchange threshold: 0.365

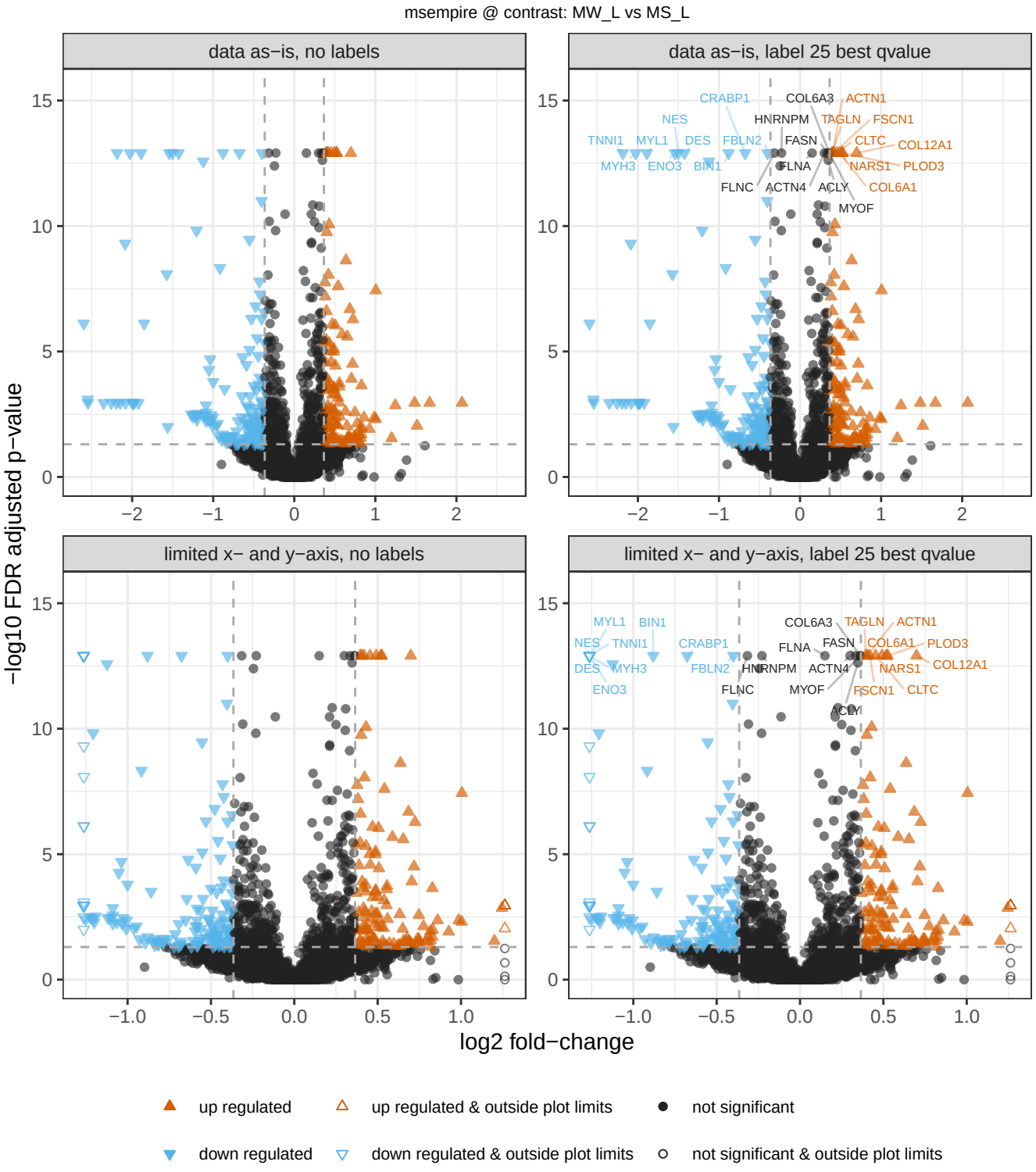
2.6.1 volcano

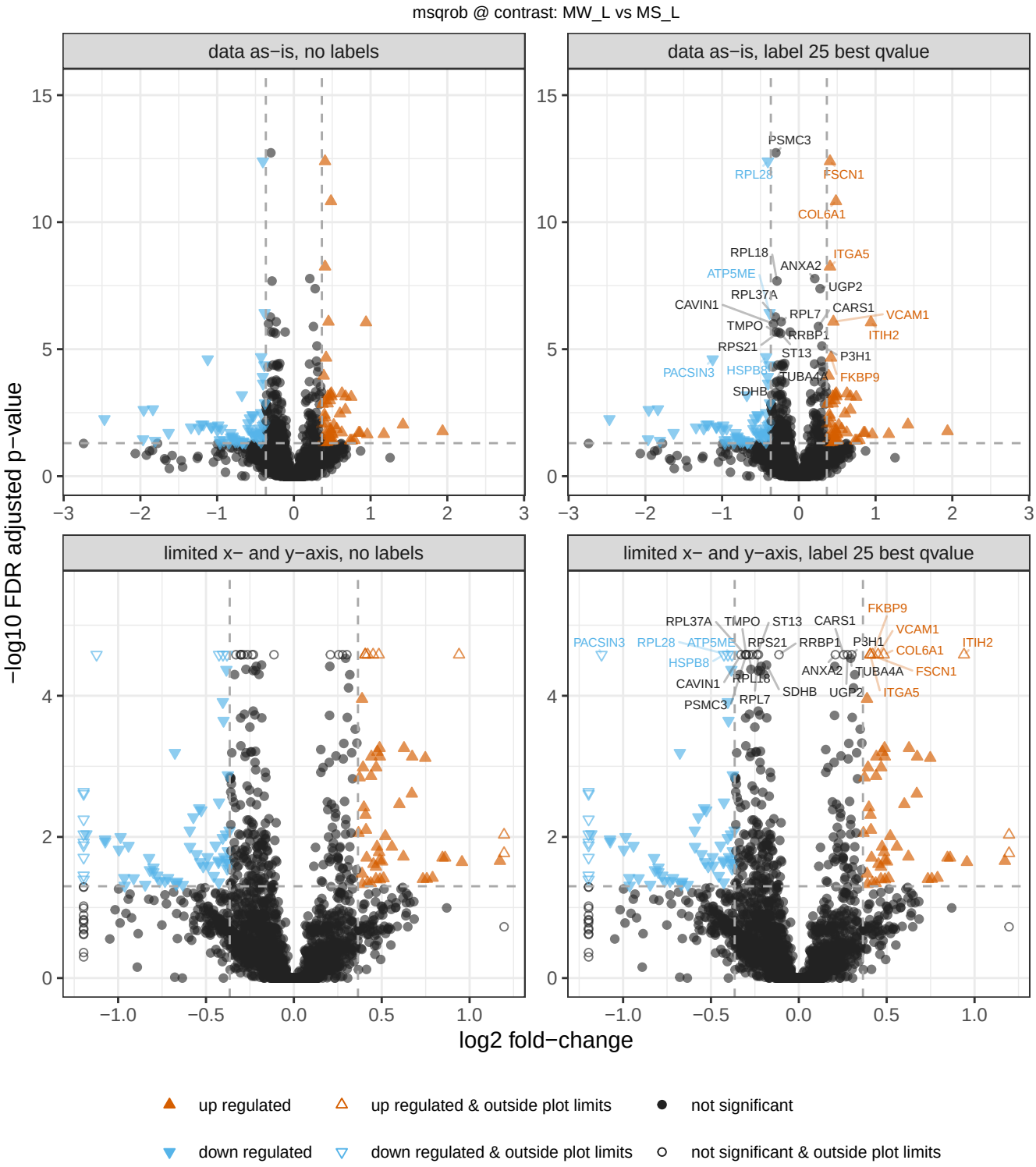
The plot title shows the statistical model and contrast (sample groups in the comparison). Left- and right-side figure panels on each row represent the same figure without and with labels for the 25 proteins with lowest p-value.

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: MW_L vs MS_L
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
SYNPO2
HBB−B2
XIRP1
HSPB7
JPH2
PACSIN3
RPL29
ITIH2
HSPB2
ENO3
TTC28
CRABP1
COL6A1
HBA
COBLL1
MB
TTN
RECQL5
BIN1
CKB
GABRA
ANKRD27
SYNPO2L
IGFBP5
RABL6
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits

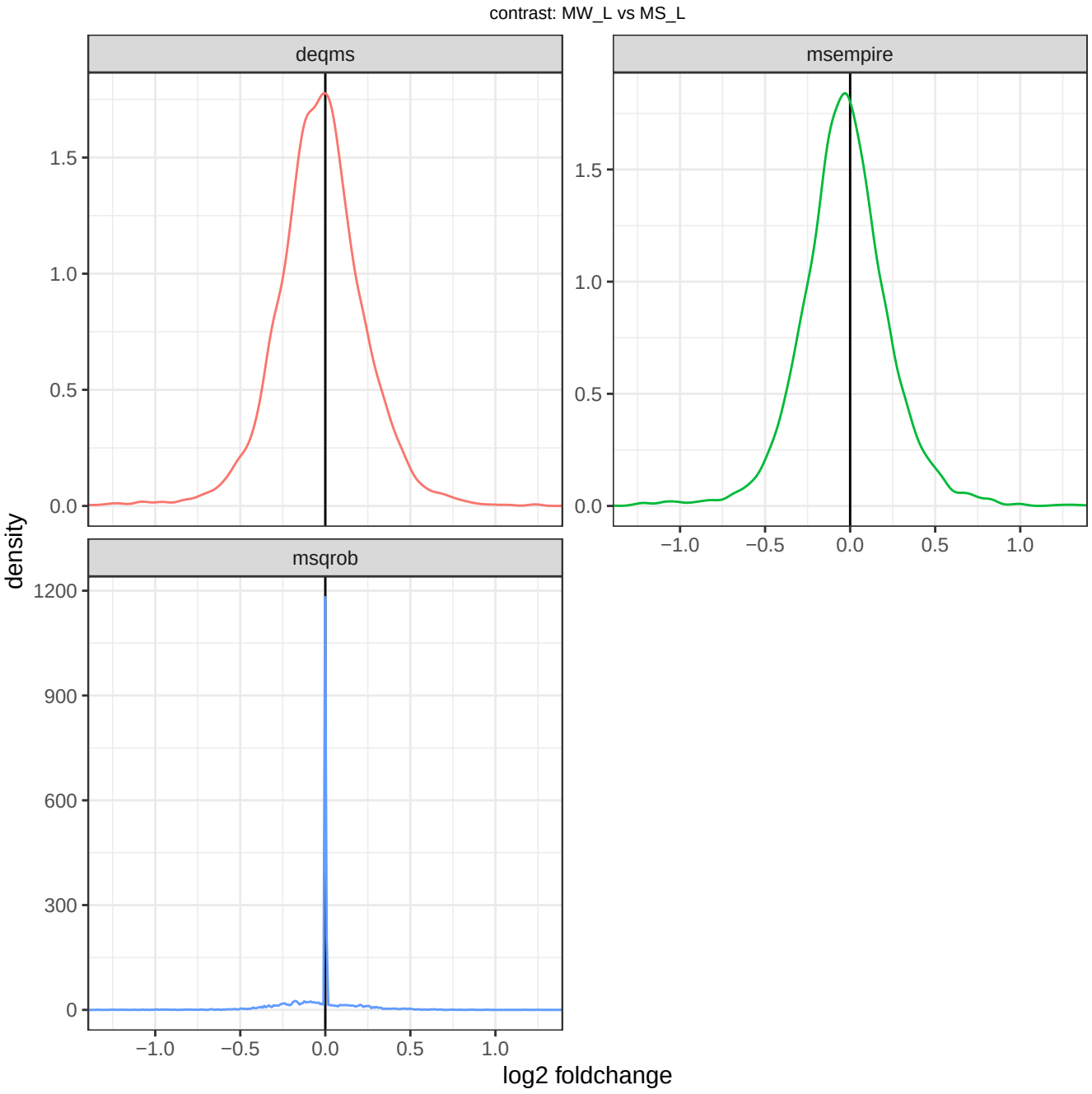

msempire @ contrast: MW_L vs MS_L
data as−is, no labels
data as−is, label 25 best qvalue
limited x− and y−axis, no labels
limited x− and y−axis, label 25 best qvalue
−log10 FDR adjusted p−value
log2 fold−change
CRABP1
NES
TNNI1
MYL1
DES
FBLN2
MYH3
ENO3
BIN1
COL6A3
ACTN1
HNRNPM
TAGLN
FSCN1
FASN
CLTC
COL12A1
FLNA
NARS1
PLOD3
FLNC
ACTN4
ACLY
COL6A1
MYOF
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits


msqrob @ contrast: MW_L vs MS_L
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
PSMC3
RPL28
FSCN1
COL6A1
RPL18
ITGA5
ANXA2
ATP5ME
UGP2
RPL37A
CAVIN1
CARS1
RPL7
VCAM1
TMPO
RRBP1
ITIH2
RPS21
ST13
P3H1
PACSIN3
HSPB8
TUBA4A
FKBP9
SDHB
RPS21
RRBP1
SDHB
UGP2
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits

2.6.2 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



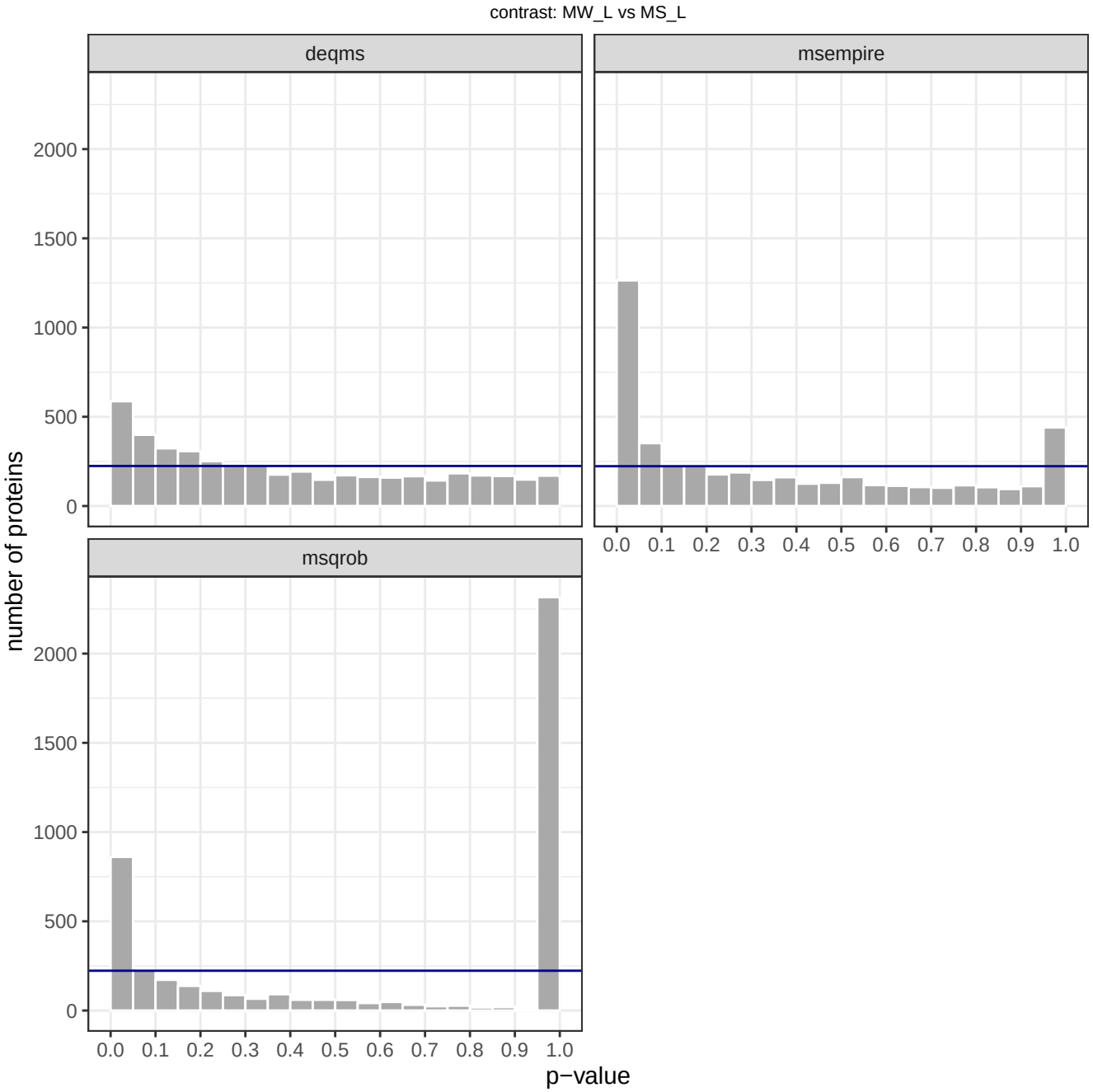
2.6.3 p-value distribution

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

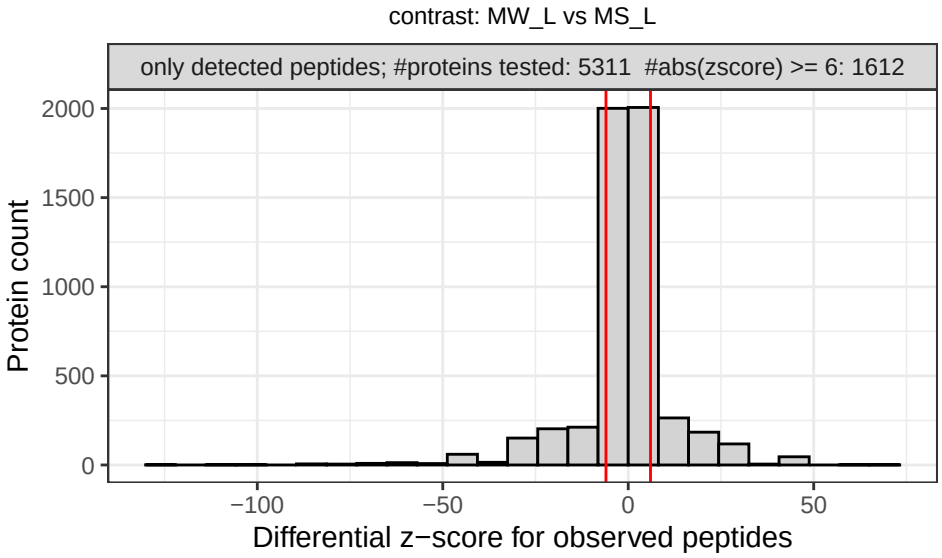


2.6.4 differential detect

Some proteins may not have peptides with sufficient data points over samples to be used for differential expression analysis (DEA), 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 study, those are interesting proteins. For this purpose, a basic metric for differential testing based on observed peptide counts is provided in MS-DAP as a situational tool. Importantly, this approach is less robust than DEA and the criteria to find a reliable set of differentially expressed proteins (by differential testing) might differ between real-world datasets.

As general guidelines for differential detection, the recommended default setting is to filter for proteins that were observed with at least 2 peptides in at least 3 replicates (or 50% of replicates, whichever number is greater). Use the plots of differential detection z-score histograms to observe the overall distribution and start your data exploration at proteins with the strongest z-scores to find desired z-score cutoffs (typically an absolute z-score of 5 or higher, but this is not set in stone for all datasets). This works best for DDA experiments, for DIA only the most extreme values are informative in many cases (e.g. proteins exclusively identified in condition A & found in nearly all replicates of condition A).

Below figure shows the distribution of these scores with thresholds at 6 std. Both the z-scores and the counts these are based upon are available in the statistical result Excel table.



2.7 MW_S vs MS_S

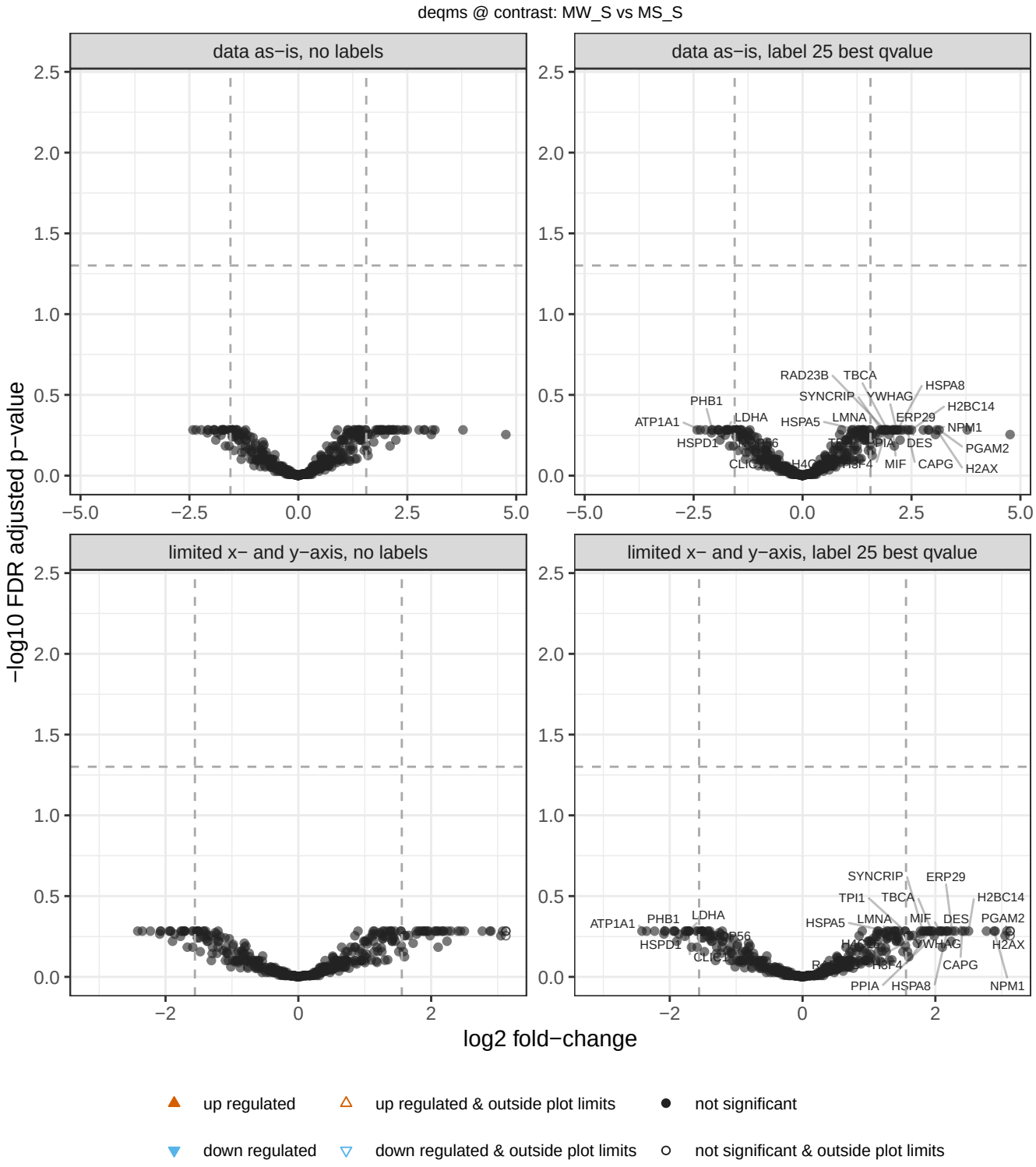
- **user setting:** using 'filter by contrast' peptide filtering approach
- 8784 peptides in 1721 proteins remain in the current contrast after peptide filters and are used for the statistical analysis in this section
- qvalue threshold: 0.05
- log2 foldchange threshold: 1.558

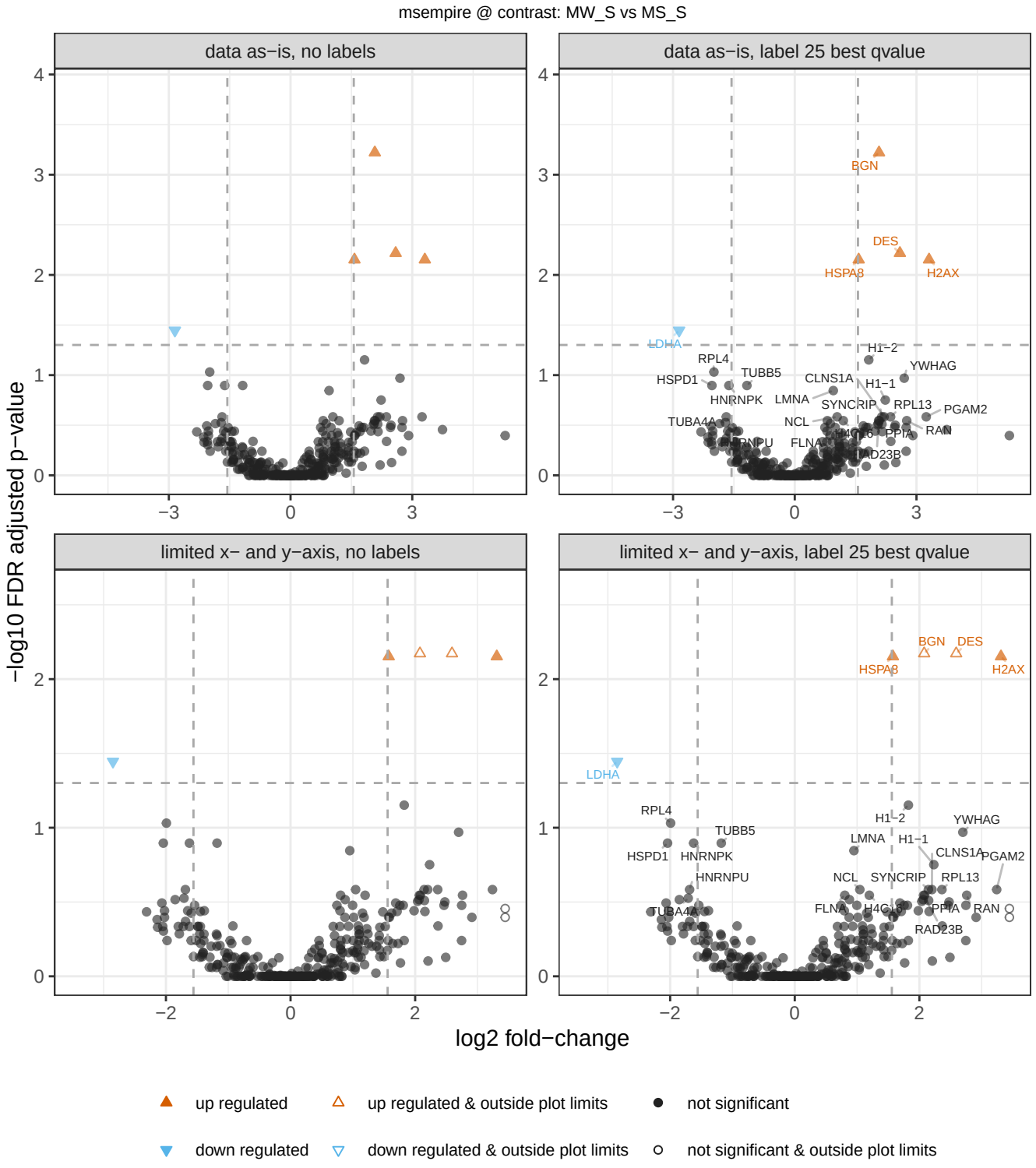
2.7.1 volcano

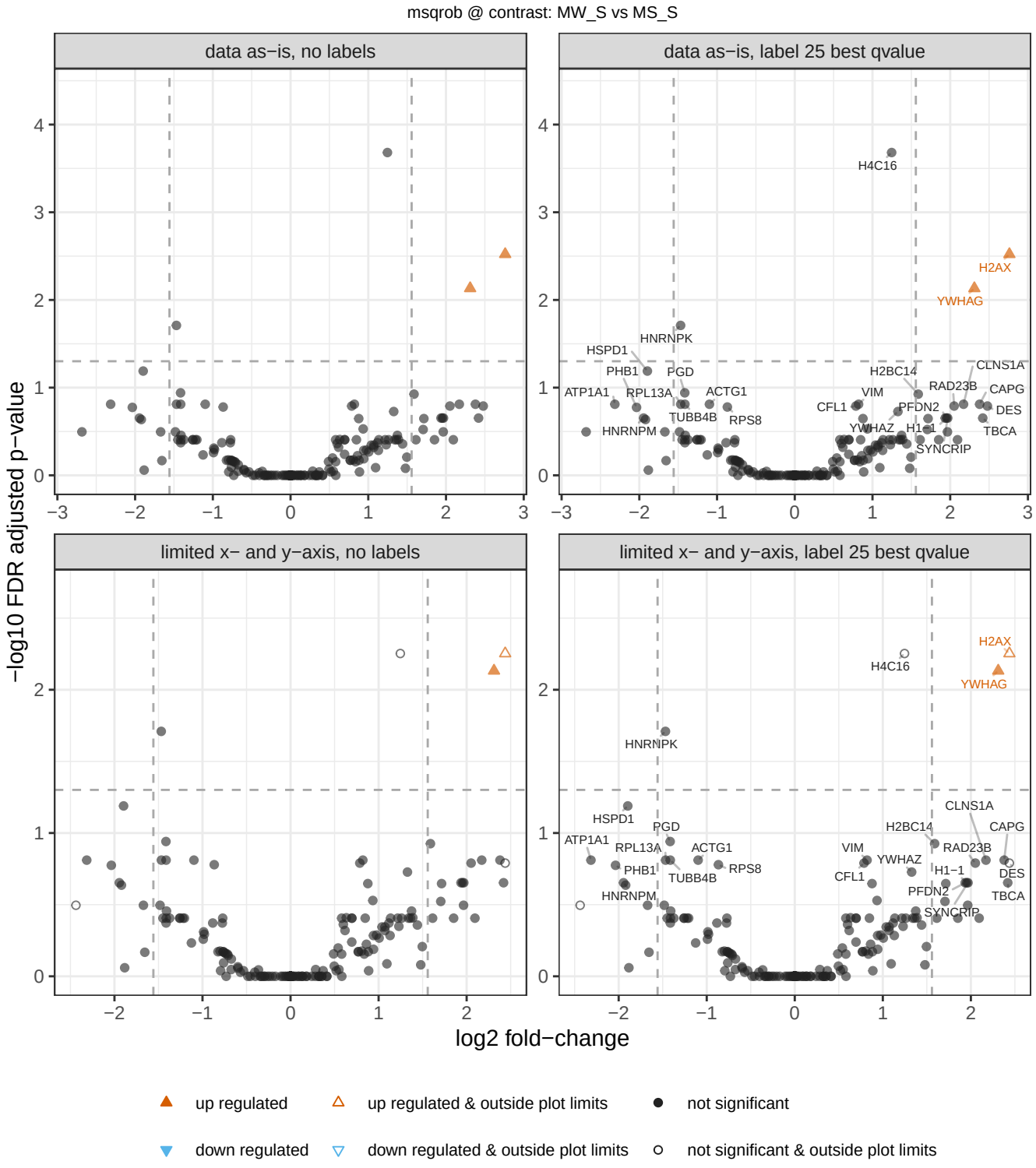
The plot title shows the statistical model and contrast (sample groups in the comparison). Left- and right-side figure panels on each row represent the same figure without and with labels for the 25 proteins with lowest p-value.

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: MW_S vs MS_S
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
RAD23B
TBCA
HSPA8
SYNCRIP
YWHAG
PHB1
H2BC14
ATP1A1
LDHA
HSPA5
LMNA
ERP29
NPM1
HSPD1
DES
PGAM2
MIF
CAPG
H2AX
TPI1
PPIA
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits

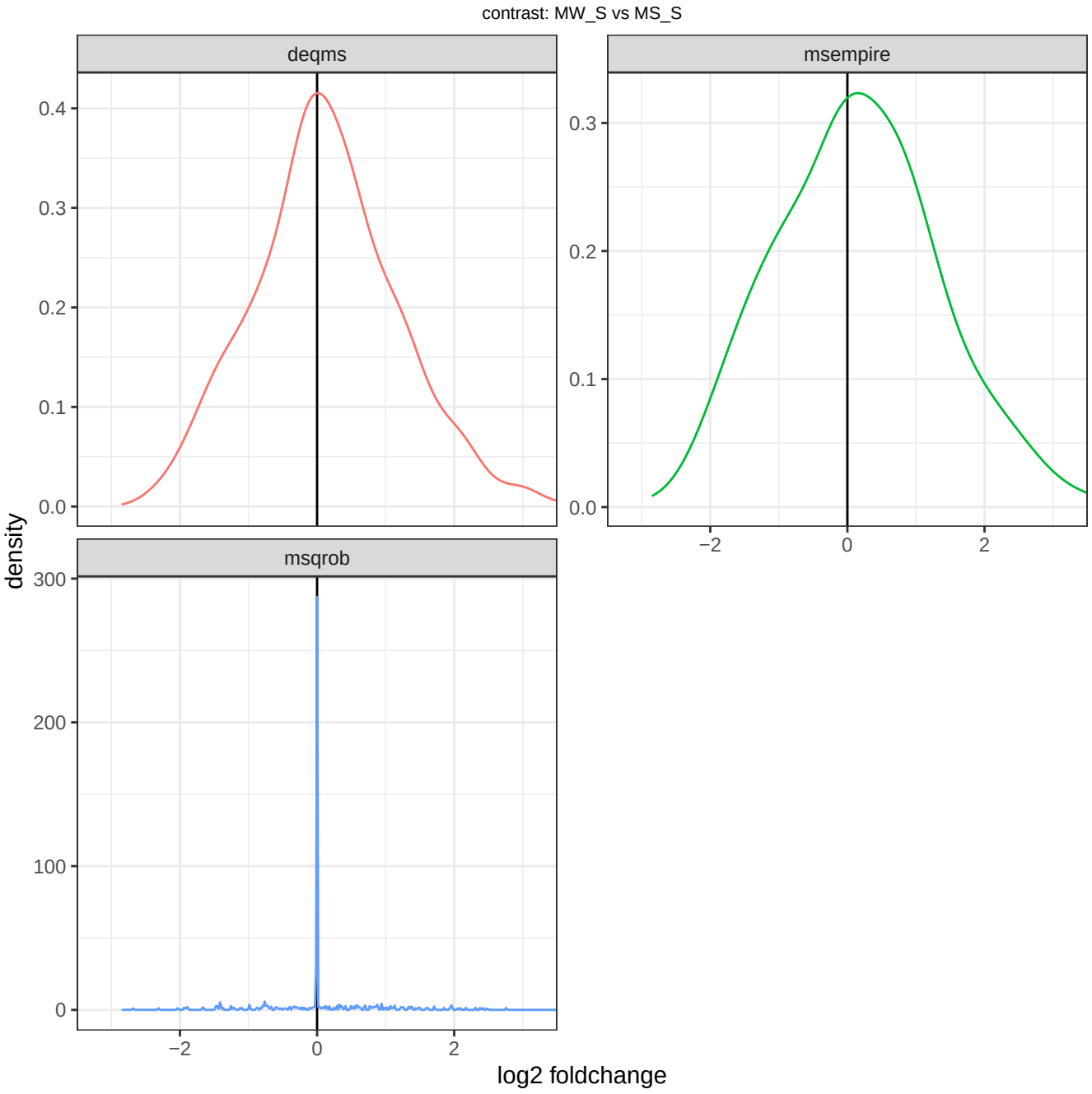

msempire @ contrast: MW_S vs MS_S
data as−is, no labels
data as−is, label 25 best qvalue
limited x− and y−axis, no labels
limited x− and y−axis, label 25 best qvalue
−log10 FDR adjusted p−value
log2 fold−change
BGN
DES
HSPA8
H2AX
LDHA
RPL4
TUBB5
HSPD1
HNRNPK
CLNS1A
LMNA
H1−2
YWHAG
H1−1
SYNCRIP
RPL13
PGAM2
TUBA4A
NCL
HNRNPU
FLNA
PPIA
RAN
RAD23B
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits


msqrob @ contrast: MW_S vs MS_S
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
H4C16
H2AX
YWHAG
HNRNPK
HSPD1
PHB1
PGD
ATP1A1
RPL13A
ACTG1
TUBB4B
RPS8
HNRNPM
H2BC14
CLNS1A
VIM
RAD23B
CAPG
CFL1
PFDN2
DES
YWHAZ
H1−1
TBCA
SYNCRIP
up regulated
up regulated & outside plot limits
not significant
down regulated
down regulated & outside plot limits
not significant & outside plot limits

2.7.2 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



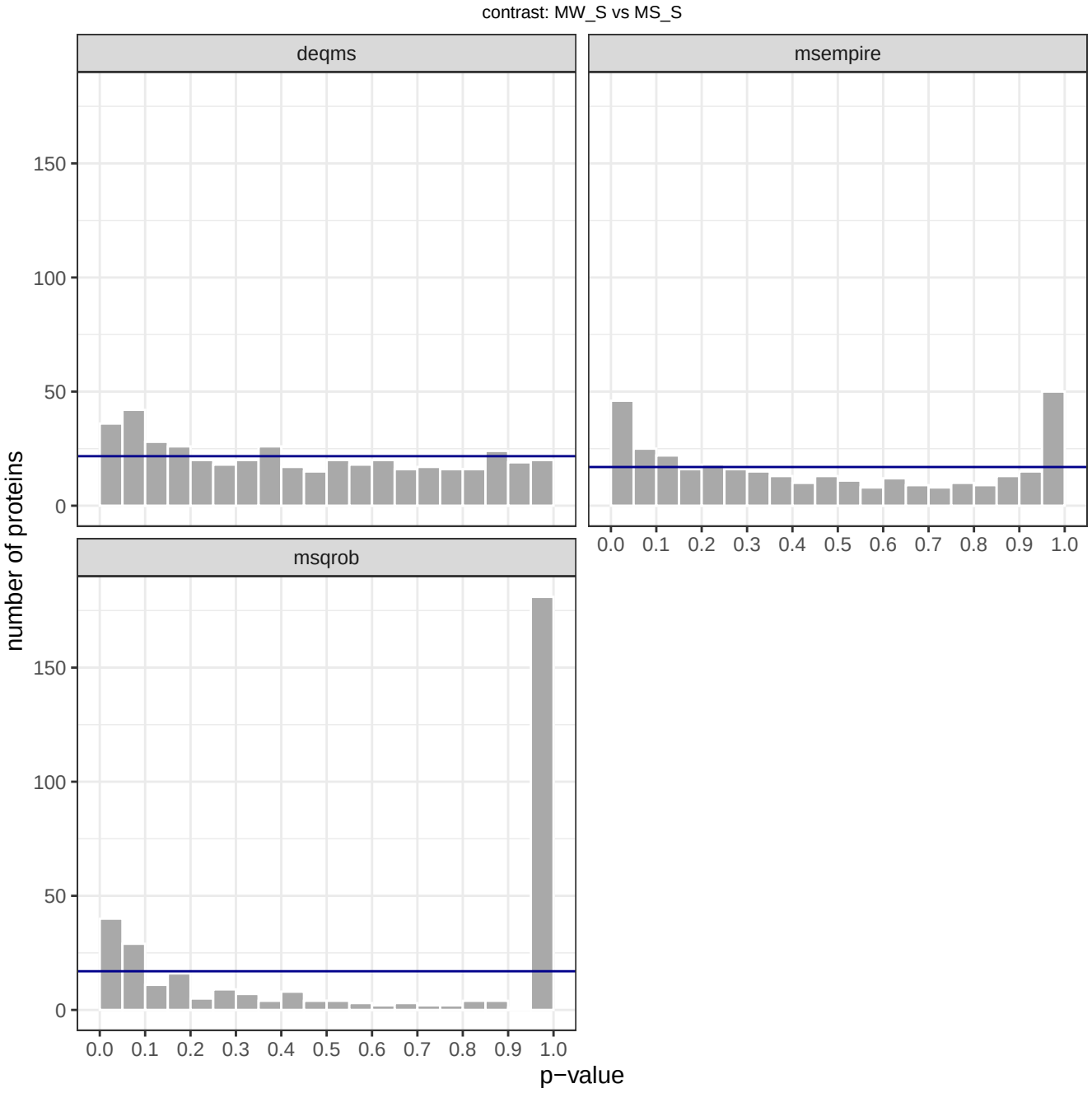
2.7.3 p-value distribution

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

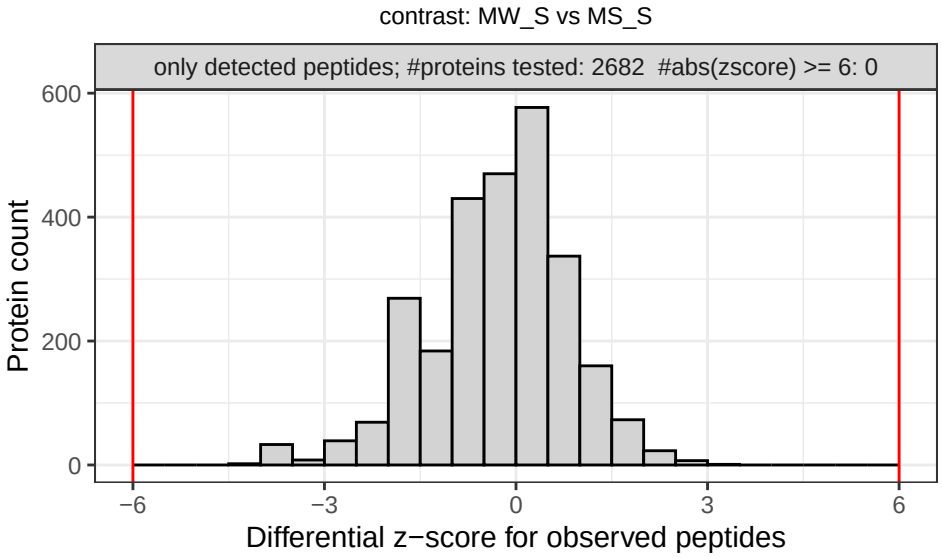


2.7.4 differential detect

Some proteins may not have peptides with sufficient data points over samples to be used for differential expression analysis (DEA), 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 study, those are interesting proteins. For this purpose, a basic metric for differential testing based on observed peptide counts is provided in MS-DAP as a situational tool. Importantly, this approach is less robust than DEA and the criteria to find a reliable set of differentially expressed proteins (by differential testing) might differ between real-world datasets.

As general guidelines for differential detection, the recommended default setting is to filter for proteins that were observed with at least 2 peptides in at least 3 replicates (or 50% of replicates, whichever number is greater). Use the plots of differential detection z-score histograms to observe the overall distribution and start your data exploration at proteins with the strongest z-scores to find desired z-score cutoffs (typically an absolute z-score of 5 or higher, but this is not set in stone for all datasets). This works best for DDA experiments, for DIA only the most extreme values are informative in many cases (e.g. proteins exclusively identified in condition A & found in nearly all replicates of condition A).

Below figure shows the distribution of these scores with thresholds at 6 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 (DEA): summary of significant proteins. The "signif" column indicates the number of significant hits at user-provided thresholds for adjusted p-value and foldchange. Columns "q<0.01" and "q<0.05" show the number of significant hits at 1% and 5% FDR cutoffs, respectively. The top10 proteingroups with strongest p-value per contrast and DEA algorithm is shown in the last column (note that these may include non-significant proteins, i.e. even if there are no significant hits the top10 is still shown).

contrast	algorithm	#proteins	signif	q<0.01	q<0.05	top10
MWd4_L vs MSd4_L	deqms	2912	0	0	0	pvalb, pdlim7, smtnl2, psmb2, coa6, akr1b1, tp53rkb, coq9, lmod1, sgtb
MWd4_L vs MSd4_L	msempire	2824	2	11	16	ahnak, hnrnpm, son, plec, myh3, sptbn1, hspa5, myh9, ckm, sptan1
MWd4_L vs MSd4_L	msqrob	2824	1	1	4	h4c16, myh10, taldo1, coa6, ywhae, txnl1, ckm, lpp, stip1, pgm1
MWd4_Exo vs MSd4_Exo	deqms	971	0	0	0	tpm2, tpd52l2, amn, hypk, atp5f1a, krt6a, uqcrc1, atp5me, itpa, tpm1
MWd4_Exo vs MSd4_Exo	msempire	909	58	68	114	aldoa, myh3, rps8, krt6a, flnc, krt10, hspg2, lmna, krt16, sdcbp
MWd4_Exo vs MSd4_Exo	msqrob	909	7	13	23	myh3, aldoa, sod1, rpl24, rps19, vcl, krt6a, tagln2, ncam1, plg
MWd4_S vs MSd4_S	deqms	2141	0	0	0	rpl32, anxa1, hnrnpm, cycs, mt2, pvalb, cxcl12, ckap4, col15a1, rpn1
MWd4_S vs MSd4_S	msempire	2093	33	42	81	hspg2, myh3, col18a1, ckap4, fstl1, col15a1, hspa5, ccn3, sparc, cycs
MWd4_S vs MSd4_S	msqrob	2093	5	5	23	cycs, aco2, anxa1, wdr1, myg1, lmnb2, mfap4, matn2, pdgfrl, cct6a
FW_L vs FS_L	deqms	2757	537	0	581	h2bc3, cdkn1c, eef2, rps27l, brd8, sptbn1, ethe1, stac3, eny2, hspa8
FW_L vs FS_L	msempire	2700	749	617	966	sptan1, ahnak, mdh2, p4hb, hspa5, tln1, des, tagln, eef2, tpm2
FW_L vs FS_L	msqrob	2700	544	267	666	plec, hspa8, sptbn1, sptan1, flnc, flna, tpm1, vcp, mdh2, eny2
FW_S vs FS_S	deqms	1738	0	0	0	mrps5, trim32, ckb, npc2, hnrnpm, atp5f1b, uqcrc1, cpb2, smoc2, copz1
FW_S vs FS_S	msempire	1672	70	75	150	col12a1, flnb, map4, actn4, myh9, ckb, des, prdx1, lamb1, hsp90ab1
FW_S vs FS_S	msqrob	1672	20	24	48	vcp, hsp90ab1, lamb1, ckb, cct6a, prdx1, cotl1, sae1, fmod, col18a1
MW_L vs MS_L	deqms	4485	0	0	0	synpo2, hbb-b2, pacsin3, itih2, xirp1, eno3, ttn, hspb2, cobll1, ankrd27
MW_L vs MS_L	msempire	4460	283	398	713	bin1, col6a3, myl1, myh3, fasn, eno3, des, tagln, fbln2, actn4
MW_L vs MS_L	msqrob	4460	115	148	331	psmc3, rpl28, fscn1, col6a1, itga5, anxa2, rpl18, ugp2, atp5me, rpl37a
MW_S vs MS_S	deqms	434	0	0	0	atp1a1, phb1, capg, h2bc14, des, h2ax, hspa8, h3f4, rad23b, erp29
MW_S vs MS_S	msempire	339	5	4	5	bgn, des, hspa8, h2ax, ldha, h1-2, rpl4, ywhag, tubb5, hnrnpk
MW_S vs MS_S	msqrob	339	2	3	4	h4c16, h2ax, ywhag, hnrnpk, hspd1, pgd, h2bc14, rpl13a, actg1, vim

Differential detection: summary of proteins with extreme differences in observed peptides. A simple metric to complement results from DEA, which is the main result, especially for proteins that lack data to perform DEA.

Differential detection scores based only on 'detected' peptides;

contrast	#proteins	#abs(zscore) >= 6	top10
MWd4_L vs MSd4_L	5123	3	erbin, mcm6, dpp3, decr2, abr, or4a80;..., arhgef26, ltbp4, cops9, prkcb
MWd4_Exo vs MSd4_Exo	2228	0	myh1, golgb1, pdlim4, arpp21, tnnt3, krt31, ndufa2, etfb, krt35, fubp3
MWd4_S vs MSd4_S	3156	437	atp5me, ddx21, cald1, basp1, nop58, rbm17, tomm34, eif4g3, nfyc, gga2
FW_L vs FS_L	5224	0	gnb1, prr32, nit2, fhl1, dnaja2, efemp2, aco1, c3, dnaaf5, kif1c
FW_S vs FS_S	3055	2	vdac1, ttn, klhl41, pycr1, atp5me, vdac2, msln, dhrs4, mgp, ssc5d
MW_L vs MS_L	5311	1612	ckm, sync, ttn, actn3, unc45b, nrap, mybpc1, lmod2, tppp3, speg
MW_S vs MS_S	2682	0	vdac1, dhx15, slc25a4, snrpe, tufm, pcyox1, zmynd11, fkbp8, ifnar2, banf1

4 log

[info] reading DIA-NN report...

[info] input file: report.tsv

[info] 42578/48197 precursors remain after selecting the 'best' precursor for each modified sequence

[info] 6401/6401 protein accessions and 6062/6062 protein groups were mapped to provided fasta file(s)

[info] numeric variable: condition

[info] contrast: MWd4_L vs MSd4_L # condition_variable: group

[info] numeric variable: condition

[info] contrast: MWd4_Exo vs MSd4_Exo # condition_variable: group

[info] numeric variable: condition

[info] contrast: MWd4_S vs MSd4_S # condition_variable: group

[info] numeric variable: condition

[info] contrast: FW_L vs FS_L # condition_variable: group

[info] numeric variable: condition

[info] contrast: FW_S vs FS_S # condition_variable: group

[info] numeric variable: condition

[info] contrast: MW_L vs MS_L # condition_variable: group

[info] numeric variable: condition

[info] contrast: MW_S vs MS_S # condition_variable: group

[info] using 7 threads for multiprocessing

[progress] caching filter data took 8 seconds

[info] filter dataset with settings: min_detect = 1; min_quant = 1; norm_algorithm = 'vsn&modebetween_protein'; rollup_algorithm = 'maxlfq'
2202/42578 peptides were retained after filtering over all groups
17163/42578 peptides were retained after filtering within contrast: MWd4_L vs MSd4_L
7351/42578 peptides were retained after filtering within contrast: MWd4_Exo vs MSd4_Exo
16772/42578 peptides were retained after filtering within contrast: MWd4_S vs MSd4_S
18456/42578 peptides were retained after filtering within contrast: FW_L vs FS_L
13506/42578 peptides were retained after filtering within contrast: FW_S vs FS_S
30414/42578 peptides were retained after filtering within contrast: MW_L vs MS_L
8784/42578 peptides were retained after filtering within contrast: MW_S vs MS_S
42578/42578 peptides were retained after filtering within each group independently ("by group")

[progress] peptide filtering and normalization took 25 seconds

[info] differential expression analysis for contrast: MWd4_L vs MSd4_L # condition_variable: group

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

[info] log2 foldchange threshold estimated by bootstrap analysis: 0.884

[warning] DEA requires at least 2 values per experimental condition; 823 / 3735 proteins were removed prior to applying DEqMS
(in typical MS-DAP workflows one would apply peptide-filtering upstream, e.g. with parameters in the analysis_quickstart() function)

[progress] DEqMS took 1 seconds

[warning] DEA requires at least 2 values per experimental condition; 6358 / 17163 proteins were removed prior to applying MS-EmpiRe
(in typical MS-DAP workflows one would apply peptide-filtering upstream, e.g. with parameters in the analysis_quickstart() function)

[progress] MS-EmpiRe took 15 seconds

[warning] DEA requires at least 2 values per experimental condition; 6358 / 17163 peptides were removed prior to applying msqrob
(in typical MS-DAP workflows one would apply peptide-filtering upstream, e.g. with parameters in the analysis_quickstart() function)

[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 2.7 minutes

[info] differential expression analysis for contrast: MWd4_Exo vs MSd4_Exo # condition_variable: group

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

[info] log2 foldchange threshold estimated by bootstrap analysis: 0.983

[warning] DEA requires at least 2 values per experimental condition; 608 / 1579 proteins were removed prior to applying DEqMS
(in typical MS-DAP workflows one would apply peptide-filtering upstream, e.g. with parameters in the analysis_quickstart() function)

[progress] DEqMS took 1 seconds

[warning] DEA requires at least 2 values per experimental condition; 3801 / 7351 proteins were removed prior to applying MS-EmpiRe
(in typical MS-DAP workflows one would apply peptide-filtering upstream, e.g. with parameters in the analysis_quickstart() function)

[progress] MS-EmpiRe took 7 seconds

[warning] DEA requires at least 2 values per experimental condition; 3801 / 7351 peptides were removed prior to applying msqrob
(in typical MS-DAP workflows one would apply peptide-filtering upstream, e.g. with parameters in the analysis_quickstart() function)

[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 44 seconds

[info] differential expression analysis for contrast: MWd4_S vs MSd4_S # condition_variable: group

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

[info] log2 foldchange threshold estimated by bootstrap analysis: 0.564

[warning] DEA requires at least 2 values per experimental condition; 473 / 2614 proteins were removed prior to applying DEqMS
(in typical MS-DAP workflows one would apply peptide-filtering upstream, e.g. with parameters in the analysis_quickstart() function)

[progress] DEqMS took 1 seconds

[warning] DEA requires at least 2 values per experimental condition; 4552 / 16772 proteins were removed prior to applying MS-EmpiRe
(in typical MS-DAP workflows one would apply peptide-filtering upstream, e.g. with parameters in the analysis_quickstart() function)

[progress] MS-EmpiRe took 1.1 minutes

[warning] DEA requires at least 2 values per experimental condition; 4552 / 16772 peptides were removed prior to applying msqrob
(in typical MS-DAP workflows one would apply peptide-filtering upstream, e.g. with parameters in the analysis_quickstart() function)

[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 1.8 minutes

[info] differential expression analysis for contrast: FW_L vs FS_L # condition_variable: group

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

[info] log2 foldchange threshold estimated by bootstrap analysis: 0.63

[warning] DEA requires at least 2 values per experimental condition; 1117 / 3874 proteins were removed prior to applying DEqMS
(in typical MS-DAP workflows one would apply peptide-filtering upstream, e.g. with parameters in the analysis_quickstart() function)

[progress] DEqMS took 1 seconds

[warning] DEA requires at least 2 values per experimental condition; 8024 / 18456 proteins were removed prior to applying MS-EmpiRe
(in typical MS-DAP workflows one would apply peptide-filtering upstream, e.g. with parameters in the analysis_quickstart() function)

[progress] MS-EmpiRe took 38 seconds

[warning] DEA requires at least 2 values per experimental condition; 8024 / 18456 peptides were removed prior to applying msqrob
(in typical MS-DAP workflows one would apply peptide-filtering upstream, e.g. with parameters in the analysis_quickstart() function)

[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 2.1 minutes

[info] differential expression analysis for contrast: FW_S vs FS_S # condition_variable: group

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

[info] log2 foldchange threshold estimated by bootstrap analysis: 1.095

[warning] DEA requires at least 2 values per experimental condition; 514 / 2252 proteins were removed prior to applying DEqMS
(in typical MS-DAP workflows one would apply peptide-filtering upstream, e.g. with parameters in the analysis_quickstart() function)

[progress] DEqMS took 1 seconds

[warning] DEA requires at least 2 values per experimental condition; 4640 / 13506 proteins were removed prior to applying MS-EmpiRe
(in typical MS-DAP workflows one would apply peptide-filtering upstream, e.g. with parameters in the analysis_quickstart() function)

[progress] MS-EmpiRe took 18 seconds

[warning] DEA requires at least 2 values per experimental condition; 4640 / 13506 peptides were removed prior to applying msqrob
(in typical MS-DAP workflows one would apply peptide-filtering upstream, e.g. with parameters in the analysis_quickstart() function)

[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 1.4 minutes

[info] differential expression analysis for contrast: MW_L vs MS_L # condition_variable: group

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

[info] log2 foldchange threshold estimated by bootstrap analysis: 0.365

[warning] DEA requires at least 2 values per experimental condition; 463 / 4948 proteins were removed prior to applying DEqMS
(in typical MS-DAP workflows one would apply peptide-filtering upstream, e.g. with parameters in the analysis_quickstart() function)

[progress] DEqMS took 1 seconds

[warning] DEA requires at least 2 values per experimental condition; 5055 / 30414 proteins were removed prior to applying MS-EmpiRe
(in typical MS-DAP workflows one would apply peptide-filtering upstream, e.g. with parameters in the analysis_quickstart() function)

[progress] MS-EmpiRe took 43 seconds

[warning] DEA requires at least 2 values per experimental condition; 5055 / 30414 peptides were removed prior to applying msqrob
(in typical MS-DAP workflows one would apply peptide-filtering upstream, e.g. with parameters in the analysis_quickstart() function)

[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 4.3 minutes

[info] differential expression analysis for contrast: MW_S vs MS_S # condition_variable: group

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

[info] log2 foldchange threshold estimated by bootstrap analysis: 1.558

[warning] DEA requires at least 2 values per experimental condition; 1287 / 1721 proteins were removed prior to applying DEqMS
(in typical MS-DAP workflows one would apply peptide-filtering upstream, e.g. with parameters in the analysis_quickstart() function)

[progress] DEqMS took 2 seconds

[warning] DEA requires at least 2 values per experimental condition; 8017 / 8784 proteins were removed prior to applying MS-EmpiRe
(in typical MS-DAP workflows one would apply peptide-filtering upstream, e.g. with parameters in the analysis_quickstart() function)

[progress] MS-EmpiRe took 4 seconds

[warning] DEA requires at least 2 values per experimental condition; 8017 / 8784 peptides were removed prior to applying msqrob
(in typical MS-DAP workflows one would apply peptide-filtering upstream, e.g. with parameters in the analysis_quickstart() function)

[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 15 seconds

[info] differential detection analysis: min_samples_observed=1 min_fraction_observed=0.00

[progress] creating PDF report…

[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 12 seconds

[progress] RT plots: creating plots took 4 seconds

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

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

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

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

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

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

[info] No data available for CoV leave-one-out computation in sample group 'MSd4_Exo', skipping plots

[info] No data available for CoV leave-one-out computation in sample group 'MSd4_L', skipping plots

[info] No data available for CoV leave-one-out computation in sample group 'MSd4_S', skipping plots

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

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

[info] No data available for CoV leave-one-out computation in sample group 'MWd4_Exo', skipping plots

[info] No data available for CoV leave-one-out computation in sample group 'MWd4_L', skipping plots

[info] No data available for CoV leave-one-out computation in sample group 'MWd4_S', skipping plots

[info] no CoV computation for sample group 'FS_L', require at least 3 replicates

[info] no CoV computation for sample group 'FS_S', require at least 3 replicates

[info] no CoV computation for sample group 'FW_L', require at least 3 replicates

[info] no CoV computation for sample group 'FW_S', require at least 3 replicates

[info] no CoV computation for sample group 'MS_L', require at least 3 replicates

[info] no CoV computation for sample group ‘MS_S’, require at least 3 replicates

[info] no CoV computation for sample group ‘MSd4_Exo’, require at least 3 replicates

[info] no CoV computation for sample group ‘MSd4_L’, require at least 3 replicates

[info] no CoV computation for sample group ‘MSd4_S’, require at least 3 replicates

[info] no CoV computation for sample group ‘MW_L’, require at least 3 replicates

[info] no CoV computation for sample group ‘MW_S’, require at least 3 replicates

[info] no CoV computation for sample group ‘MWd4_Exo’, require at least 3 replicates

[info] no CoV computation for sample group ‘MWd4_L’, require at least 3 replicates

[info] no CoV computation for sample group ‘MWd4_S’, require at least 3 replicates

[info] No data available for CoV computation, skipping plots

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).

```
)
rownames(result) <- result$protein
F_set <- rownames(result)[result$FW.S == TRUE]
M_set <- rownames(result)[result$MW.S == TRUE]
Md4_set <- rownames(result)[result$MWd4.S == TRUE]
if(FMMd4 == 0){
venn_data <- list(
"F" = F_set,
"M" = M_set,
"Md4" = Md4_set
)
}
else{
venn_data <- list(
"MW_vs_Md4W" = M_set,
"MS_vs_Md4S" = Md4_set
)
}
venn_diag <- ggvenn(venn_data, fill_color = c("skyblue", "pink", "lightgreen")) + ggtitle(paste0("Diagrama de Venn:",param)) + theme(plot
if (!dir.exists(direc)) {
dir.create(direc)
}
ggsave(paste0(direc, "/ven_", param, ".png"), plot = venn_diag, bg="white")
return(venn_data)
}
data_FS_down <- leer_archivos_param("../Extract stats/FWS", "down")
data_MS_down <- leer_archivos_param("../Extract stats/MWL", "down")
data_Md4S_down <- leer_archivos_param("../Extract stats/MWd4S", "down")
valores_Md4S_down <- diagrama_Venn_grupo(data_Md4S_down, "Md4S_down", "S_down")
valores_Md4S_down <- diagrama_Venn_grupo(data_Md4S_down, "Md4S_down", "S_down")
valores_MS_down <- diagrama_Venn_grupo(data_MS_down, "MS_down", "S_down")
valores_FS_down <- diagrama_Venn_grupo(data_FS_down, "FS_down", "S_down")
datos_DS <- diagrama_Venn_condicion("S_down", valores_Md4S_down, valores_MS_down, valores_FS_down, "Down_S_not_combined")
DS <- w_file(datos_DS, "Down_S_id_nc", id = 1)
DS <- w_file(datos_DS, "Down_S_nc", id = 0)
data_FS_up <- leer_archivos_param("../Extract stats/FWS", "up")
data_MS_up <- leer_archivos_param("../Extract stats/MWS", "up")
data_Md4S_up <- leer_archivos_param("../Extract stats/MWd4S", "up")
valores_Md4S_up <- diagrama_Venn_grupo(data_Md4S_up, "Md4S_up", "S_up")
valores_Md4S_up <- diagrama_Venn_grupo(data_Md4S_up, "Md4S_up", "S_up")
valores_MS_up <- diagrama_Venn_grupo(data_MS_up, "MS_up", "S_up")
valores_FS_up <- diagrama_Venn_grupo(data_FS_up, "FS_up", "S_up")
datos_uS <- diagrama_Venn_condicion("S_up", valores_Md4S_up, valores_MS_up, valores_FS_up, "Up_S_nc")
uS <- w_file(datos_uS, "Up_S_id_nc", id = 1)
uS <- w_file(datos_uS, "Up_S_nc", id = 0)
data_Md4E_up <- leer_archivos_param("../Extract stats/MWd4E", "up")
valores_Md4E_up <- diagrama_Venn_grupo(data_Md4E_up, "Md4E_up", "E_up")
datos_uE <- diagrama_Venn_condicion("E_up",valores_Md4E_up, NULL, NULL, "Up_E_nc")
uE <- w_file(datos_uE, "Up_E_id_nc", id = 1)
uE <- w_file(datos_uE, "Up_E_nc", id = 0)
data_Md4E_down <- leer_archivos_param("../Extract stats/MWd4E", "down")
valores_Md4E_down <- diagrama_Venn_grupo(data_Md4E_down, "Md4E_down", "E_down")
datos_DE <- diagrama_Venn_condicion("E_down", valores_Md4E_down, NULL, NULL, "Down_E_nc")
DE <- w_file(datos_DE, "Down_E_id_nc", id = 1)
DE <- w_file(datos_DE, "Down_E_nc", id = 0)
data_FL_down <- leer_archivos_param("../Extract stats/FWL", "down")
data_ML_down <- leer_archivos_param("../Extract stats/MWL", "down")
```

```
data_Md4L_down <- leer_archivos_param("../Extract stats/MWd4L", "down")
valores_Md4L_down <- diagrama_Venn_grupo(data_Md4L_down, "Md4L_down", "L_down")
valores_ML_down <- diagrama_Venn_grupo(data_ML_down, "ML_down", "L_down")
valores_FL_down <- diagrama_Venn_grupo(data_FL_down, "FL_down", "L_down")
valores_FL_down <- diagrama_Venn_grupo(data_FL_down, "FL_down", "L_down")
datos_DL <- diagrama_Venn_condicion("L_down", valores_Md4L_down, valores_ML_down, valores_FL_down, "Down_L_nc")
valores_FL_down <- diagrama_Venn_grupo(data_FL_down, "FL_down", "L_down")
datos_DL <- diagrama_Venn_condicion("L_down", valores_Md4L_down, valores_ML_down, valores_FL_down, "Down_L_nc")
DL <- w_file(datos_DL, "Down_L_id_nc", id = 1)
DL <- w_file(datos_DL, "Down_L_nc", id = 0)
data_FL_up <- leer_archivos_param("../Extract stats/FWL", "up")
data_ML_up <- leer_archivos_param("../Extract stats/MWL", "up")
data_Md4L_up <- leer_archivos_param("../Extract stats/MWd4L", "up")
valores_Md4L_up <- diagrama_Venn_grupo(data_Md4L_up, "Md4L_up", "L_up")
valores_FL_up <- diagrama_Venn_grupo(data_FL_up, "FL_up", "L_up")
datos_uL <- diagrama_Venn_condicion("L_up", valores_Md4L_up, valores_ML_up, valores_FL_up, "Up_L_nc")
uL <- w_file(datos_uL, "Up_L_id_nc", id = 1)
uL <- w_file(datos_uL, "Up_L_nc", id = 0)
valores_Md4L_down <- combined_Md4L$protein[combined_Md4L$logFC < -0.5 &  combined_Md4L$qvalue<0.05]
valores_ML_down <- combined_ML$protein[combined_ML$logFC < -0.5 & combined_ML$qvalue<0.05]
valores_FL_down <- combined_FL$protein[combined_FL$logFC < -0.5 & combined_FL$qvalue<0.05]
datos_DL <- diagrama_Venn_condicion("L_down", valores_Md4L_down, valores_ML_down, valores_FL_down, "Down_L")
DL <- w_file(datos_DL, "Down_L_id", id = 1)
DL <- w_file(datos_DL, "Down_L", id = 0)
valores_Md4L_up <- combined_Md4L$protein[combined_Md4L$logFC > 0.5 & combined_Md4L$qvalue<0.05]
valores_ML_up <- combined_ML$protein[combined_ML$logFC > 0.5 & combined_ML$qvalue<0.05]
valores_FL_up <- combined_FL$protein[combined_FL$logFC > 0.5 & combined_FL$qvalue<0.05]
datos_uL <- diagrama_Venn_condicion("L_up", valores_Md4L_up, valores_ML_up, valores_FL_up, "Up_L")
uL <- w_file(datos_uL, "Up_L", id = 0)
uL <- w_file(datos_uL, "Up_L_id", id = 1)
valores_Md4E_down <- combined_Md4E$protein[combined_Md4E$logFC < -0.5 & combined_Md4E$qvalue < 0.05]
datos_DE <- diagrama_Venn_condicion("E_down", valores_Md4E_down, NULL, NULL, "Down_E")
DE <- w_file(datos_DE, "Down_E_id", id = 1)
DE <- w_file(datos_DE, "Down_E", id = 0)
valores_Md4E_up <- combined_Md4E$protein[combined_Md4E$logFC > 0.5 & combined_Md4E$qvalue < 0.05]
datos_uE <- diagrama_Venn_condicion("E_up", valores_Md4E_up, NULL, NULL, "Up_E")
uE <- w_file(datos_uE, "Up_E_id", id = 1)
uE <- w_file(datos_uE, "Up_E", id = 0)
valores_Md4S_down <- combined_Md4S$protein[combined_Md4S$logFC < -0.5 & combined_Md4S$qvalue < 0.05]
valores_MS_down <- combined_MS$protein[combined_MS$logFC < -0.5 & combined_MS$qvalue < 0.05]
valores_FS_down <- combined_FS$protein[combined_FS$logFC < -0.5 & combined_FS$qvalue < 0.05]
datos_DS <- diagrama_Venn_condicion("S_down", valores_Md4S_down, valores_MS_down, valores_FS_down, "Down_S")
DS <- w_file(datos_DS, "Down_S_id", id = 1)
DS <- w_file(datos_DS, "Down_S", id = 0)
valores_Md4S_up <- combined_Md4S$protein[combined_Md4S$logFC > 0.5 & combined_Md4S$qvalue < 0.05]
valores_MS_up <- combined_MS$protein[combined_MS$logFC > 0.5 & combined_MS$qvalue < 0.05]
valores_FS_up <- combined_FS$protein[combined_FS$logFC > 0.5 & combined_FS$qvalue < 0.05]
datos_uS <- diagrama_Venn_condicion("S_up", valores_Md4S_up, valores_MS_up, valores_FS_up, "Up_S")
uS <- w_file(datos_uS, "Up_S_id", id = 1)
uS <- w_file(datos_uS, "Up_S", id = 0)
knitr::opts_chunk$set(echo = TRUE)
library(clusterProfiler)
library(org.Mm.eg.db)
library(ggplot2)
library(ggkegg)
library(enrichplot)
library(DESeq2)
library(DOSE)
library(plotly)
library(visNetwork)
library(tidygraph)
library(dplyr)
library(ggraph)
library(ggplot2)
library(rmarkdown)
library(htmltools)
library(UpSetR)
library(ggupset)
library(ReactomePA)
library(writexl)
library(openxlsx)
library(pathview)
rename_columns_to_first_word <- function(data) {
# Renombrar las columnas usando la primera palabra
data <- data %>%
```

```
rename_with(~ sapply(strsplit(.x, "_"), `[`, 1)) # Cambiar aquí el separador a "_"
return(data)
}
create_filtered_data <- function(data, param=NULL) {
# # Validar que el parámetro sea uno de los valores permitidos
# if (!param %in% c("ebayes", "deqms", "msempire", "msqrob")) {
#   stop("El parámetro debe ser uno de: 'ebayes', 'deqms', 'msempire', 'msqrob'")
# }
# Filtrar las columnas
# selected_columns <- data %>%
#   select(protein_id,gene_symbols_or_id, ENTREZID, contains(param))
selected_columns <- data
# Asignar a una variable dinámica
# assign(paste0("data_", param), selected_columns, envir = .GlobalEnv)
selected_columns <- rename_columns_to_first_word(selected_columns)
# Retornar el dataset filtrado
return(selected_columns)
}
get_data_by_param <- function(param) {
# Validar que el parámetro sea uno de los permitidos
if (!param %in% c("bayes", "deq", "empire", "rob")) {
stop("El parámetro debe ser uno de: 'bayes', 'deq', 'empire', 'rob'")
}
# Crear el nombre de la variable correspondiente
variable_name <- paste0(param, "_data")
# Verificar si la variable existe en el entorno global
if (!exists(variable_name, envir = .GlobalEnv)) {
stop(paste("La variable", variable_name, "no existe en el entorno global"))
}
# Retornar la variable correspondiente
return(get(variable_name, envir = .GlobalEnv))
}
capitalize_first <- function(text) {
paste0(toupper(substr(text, 1, 1)), tolower(substr(text, 2, nchar(text))))
}
generate_genes <- function(result, up, down,i){
downregulated_genes_symbol <- sapply(result$datos_down$gene[sapply(result$datos_down$gene, capitalize_first) %in% sapply(down[,i],capital
upregulated_genes_symbol <- sapply(result$datos_up$gene[sapply(result$datos_up$gene, capitalize_first) %in% sapply(up[,i], capitalize_fir
downregulated_genes_filtered <- result$data_down$ENTREZID[result$data_down$ENTREZID %in% down_entrez$ENTREZID]
upregulated_genes_filtered <- result$data_up$ENTREZID[result$data_up$ENTREZID %in% up_entrez$ENTREZID]
# upregulated_genes_filtered <- upregulated_genes_filtered[!is.na(upregulated_genes_filtered)]
# downregulated_genes_filtered <- downregulated_genes_filtered[!is.na(upregulated_genes_filtered)]
return(list(downregulated_genes_symbol = downregulated_genes_symbol, upregulated_genes_symbol = upregulated_genes_symbol, downregulated_
}
generate_data <- function(datos, i, p.value){
data_down <- datos[datos$logFC< -0.5 & datos$qvalue < p.value,]
data_up <- datos[datos$logFC > 0.5 & datos$qvalue < p.value,]
data_down <- data_down[order(data_down$pvalue), ]
data_up <- data_up[order(data_up$pvalue), ]
datos_down <- data_down[data_down$gene %in% sapply(genes_down[,i],capitalize_first),]
datos_up <- data_up[data_up$gene %in% sapply(genes_up[,i],capitalize_first),]
return(list(data_down = data_down, data_up = data_up, datos_down = datos_down, datos_up = datos_up))
}
write_dataframe <- function(dataframes, directory, param){
write.xlsx(dataframes$down, file = paste0(directory,"/",param,"_down.xlsx"))
write.xlsx(dataframes$up, file = paste0(directory,"/",param,"_up.xlsx"))
write.xlsx(dataframes$datos, file = paste0(directory,"/",param,".xlsx"))
}
for (archivo in archivos){
data <- read.xlsx(archivo)
sample <- gsub(".*([a-zA-Z])\\.xlsx$", "\\1", archivo)
print(sample)
genes_up <- sapply(read.xlsx(paste0("../Venn/Resultados/resultados_Up_", sample, ".xlsx")),capitalize_first)
genes_down <- sapply(read.xlsx(paste0("../Venn/Resultados/resultados_Down_", sample, ".xlsx")), capitalize_first)
grupo <- gsub(".*/.*_([a-zA-Z0-9]{2,})[a-zA-Z]*\\.xlsx$", "\\1", archivo)
print(grupo)
if (grupo == "ML" | grupo == "MS"){
exclusive_column <- 1
}
else if (grupo == "Md4L" | grupo == "Md4S" | grupo == "Md4E"){
exclusive_column <- 2
# grupo <- gsub("^MWd4$", "Md4", grupo)
}
else if (grupo == "FL" | grupo == "FS"){
exclusive_column <- 3
```

```
}
print(exclusive_column)
uniprot_ids <- unlist(strsplit(as.character(data$protein), split = " "))
ids_entrez <- bitr(uniprot_ids, fromType = "UNIPROT", toType = "ENTREZID", OrgDb = org.Mm.eg.db)
up_entrez <- sapply(as.character(data$protein[data$gene%in%
na.omit(genes_up[, exclusive_column])]), function(x) {
ids <- unlist(strsplit(x, split = " "))  # Dividir IDs separados por ";"
matched_ids <- ids_entrez$ENTREZID[ids_entrez$UNIPROT %in% ids]  # Buscar IDs coincidentes en ids_entrez
if (length(matched_ids) > 0) {
paste(unique(matched_ids), collapse = ";")  # Concatenar IDs únicos si hay coincidencias
} else {
NA  # Devolver NA si no hay coincidencias
}
})
up_entrez <- as.data.frame(up_entrez)
if (dim(up_entrez)[1] > 0){
colnames(up_entrez) <- "ENTREZID"
}
down_entrez <- sapply(as.character(data$protein[data$gene %in%
na.omit(genes_down[, exclusive_column])]), function(x) {
ids <- unlist(strsplit(x, split = " "))  # Dividir IDs separados por ";"
matched_ids <- ids_entrez$ENTREZID[ids_entrez$UNIPROT %in% ids]  # Buscar IDs coincidentes en ids_entrez
if (length(matched_ids) > 0) {
paste(unique(matched_ids), collapse = ";")  # Concatenar IDs únicos si hay coincidencias
} else {
NA  # Devolver NA si no hay coincidencias
}
})
down_entrez <- as.data.frame(down_entrez)
if (dim(down_entrez)[1] > 0){
colnames(down_entrez) <- "ENTREZID"
}
data$ENTREZID <- sapply(as.character(data$protein), function(x) {
ids <- unlist(strsplit(x, split = " "))
matched_ids <- ids_entrez$ENTREZID[ids_entrez$UNIPROT %in% ids]
if (length(matched_ids) > 0) {
paste(unique(matched_ids), collapse = ";")
} else {
NA
}
})
data$gene<- sapply(data$gene, capitalize_first)
final_data <- create_filtered_data(data)
resultado <- generate_data(final_data, exclusive_column, 1)
genes <- generate_genes(resultado, genes_up,genes_down, exclusive_column)
directory <- paste0("combined/",grupo, sample)
name <- paste0(grupo, sample)
kegg_result <- kegg_function(genes)
process_kegg(kegg_result, "final", directory)
go_result <- go_function(genes)
process_go(go_result, "final", directory)
reactome_result <- reactome_function(genes)
process_reactome(reactome_result, "final", directory)
get_pvalues_df(resultado, "final")
dataframes_final <- create_dataframes(resultado,"final")
# create_pathways(dataframes_rob,"rob")
write_dataframe(dataframes_final, directory, name)
}
for (archivo in archivos){
data <- read.xlsx(archivo)
sample <- gsub(".*([a-zA-Z])\\.xlsx$", "\\1", archivo)
print(sample)
genes_up <- sapply(read.xlsx(paste0("../Venn/Resultados/resultados_Up_", sample, ".xlsx")),capitalize_first)
genes_down <- sapply(read.xlsx(paste0("../Venn/Resultados/resultados_Down_", sample, ".xlsx")), capitalize_first)
grupo <- gsub(".*/.*_([a-zA-Z0-9]{2,})[a-zA-Z]*\\.xlsx$", "\\1", archivo)
print(grupo)
if (grupo == "ML" | grupo == "MS"){
exclusive_column <- 1
}
else if (grupo == "Md4L" | grupo == "Md4S" | grupo == "Md4E"){
exclusive_column <- 2
# grupo <- gsub("^MWd4$", "Md4", grupo)
}
else if (grupo == "FL" | grupo == "FS"){
exclusive_column <- 3
```

```
}
print(exclusive_column)
uniprot_ids <- unlist(strsplit(as.character(data$protein), split = " "))
ids_entrez <- bitr(uniprot_ids, fromType = "UNIPROT", toType = "ENTREZID", OrgDb = org.Mm.eg.db)
up_entrez <- sapply(as.character(data$protein[data$gene%in%
na.omit(genes_up[, exclusive_column])]), function(x) {
ids <- unlist(strsplit(x, split = " "))  # Dividir IDs separados por ";"
matched_ids <- ids_entrez$ENTREZID[ids_entrez$UNIPROT %in% ids]  # Buscar IDs coincidentes en ids_entrez
if (length(matched_ids) > 0) {
paste(unique(matched_ids), collapse = ";")  # Concatenar IDs únicos si hay coincidencias
} else {
NA  # Devolver NA si no hay coincidencias
}
})
up_entrez <- as.data.frame(up_entrez)
if (dim(up_entrez)[1] > 0){
colnames(up_entrez) <- "ENTREZID"
}
down_entrez <- sapply(as.character(data$protein[data$gene %in%
na.omit(genes_down[, exclusive_column])]), function(x) {
ids <- unlist(strsplit(x, split = " "))  # Dividir IDs separados por ";"
matched_ids <- ids_entrez$ENTREZID[ids_entrez$UNIPROT %in% ids]  # Buscar IDs coincidentes en ids_entrez
if (length(matched_ids) > 0) {
paste(unique(matched_ids), collapse = ";")  # Concatenar IDs únicos si hay coincidencias
} else {
NA  # Devolver NA si no hay coincidencias
}
})
down_entrez <- as.data.frame(down_entrez)
if (dim(down_entrez)[1] > 0){
colnames(down_entrez) <- "ENTREZID"
}
data$ENTREZID <- sapply(as.character(data$protein), function(x) {
ids <- unlist(strsplit(x, split = " "))
matched_ids <- ids_entrez$ENTREZID[ids_entrez$UNIPROT %in% ids]
if (length(matched_ids) > 0) {
paste(unique(matched_ids), collapse = ";")
} else {
NA
}
})
data$gene<- sapply(data$gene, capitalize_first)
final_data <- create_filtered_data(data)
resultado <- generate_data(final_data, exclusive_column, 1)
genes <- generate_genes(resultado, genes_up,genes_down, exclusive_column)
directory <- paste0("combined/",grupo, sample)
name <- paste0(grupo, sample)
kegg_result <- kegg_function(genes)
process_kegg(kegg_result, "final", directory)
go_result <- go_function(genes)
process_go(go_result, "final", directory)
reactome_result <- reactome_function(genes)
process_reactome(reactome_result, "final", directory)
get_pvalues_df(resultado, "final")
dataframes_final <- create_dataframes(resultado,"final")
# create_pathways(dataframes_rob,"rob")
write_dataframe(dataframes_final, directory, name)
}
for (archivo in archivos){
grupo <- gsub(".*/.*_([a-zA-Z0-9]{2,})[a-zA-Z]*\\.xlsx$", "\\1", archivo)
sample <- gsub(".*([a-zA-Z])\\.xlsx$", "\\1", archivo)
directory <- paste0(grupo,sample)
datos <- read_excel(paste0("combined/", directory,"/", directory,".xlsx"), sheet = "Datos")
go <- read_excel(paste0("combined/", directory,"/", directory,".xlsx"), sheet = "Go")
# Filtrar funciones GO con p.adjust < 0.05
go_filtered <- go %>% filter(p.adjust < 0.05)
# Separar los genes en la columna geneID en filas individuales
go_filtered <- go_filtered %>%
separate_rows(geneID, sep = "/")  # Cambia el separador si es diferente
# Asegurarse de que ambas columnas sean del mismo tipo
datos$gene <- trimws(as.character(datos$gene))
go_filtered$geneID <- trimws(as.character(go_filtered$geneID))
# Unir los datos después de separar los genes
merged_data <- merge(datos, go_filtered, by.x = "gene", by.y = "geneID", all.x = FALSE)
# Filtrar valores únicos para logFC
```

```
unique_logFC <- merged_data %>%
distinct(gene, .keep_all = TRUE)  # Quedarse solo con una fila por gen
# Ordenar los genes por logFC
unique_logFC <- unique_logFC %>%
arrange(logFC) %>%
mutate(gene = factor(gene, levels = unique(gene)))
# Gráfico principal: logFC de las proteínas (valores únicos)
logfc_plot <- ggplot(unique_logFC, aes(x = gene, y = logFC)) +
geom_bar(stat = "identity", fill = "skyblue") +
theme_minimal() +
theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
labs(title = "logFC de Proteínas (Único y Ordenado por logFC)", x = "Gen", y = "logFC")
# Gráfico suplementario: Funciones GO asociadas a las proteínas
go_plot <- ggplot(merged_data, aes(x = gene, y = ID, fill = p.adjust)) +
geom_tile() +
scale_fill_gradient(low = "lightgreen", high = "red", name = "p.adjust") +
theme_minimal() +
theme(axis.text.x = element_text(angle = 90, hjust = 1),
axis.text.y = element_text(size = 8)) +
labs(title = "Funciones GO asociadas (p.adjust < 0.05)", x = "Gen", y = "Funciones GO")
# Combinar los gráficos
combined_plot <- logfc_plot / go_plot
# Mostrar los gráficos combinados
print(combined_plot)
}
library(patchwork)
for (archivo in archivos){
grupo <- gsub(".*/.*_([a-zA-Z0-9]{2,})[a-zA-Z]*\\.xlsx$", "\\1", archivo)
sample <- gsub(".*([a-zA-Z])\\.xlsx$", "\\1", archivo)
directory <- paste0(grupo,sample)
datos <- read_excel(paste0("combined/", directory,"/", directory,".xlsx"), sheet = "Datos")
go <- read_excel(paste0("combined/", directory,"/", directory,".xlsx"), sheet = "Go")
# Filtrar funciones GO con p.adjust < 0.05
go_filtered <- go %>% filter(p.adjust < 0.05)
# Separar los genes en la columma geneID en filas individuales
go_filtered <- go_filtered %>%
separate_rows(geneID, sep = "/")  # Cambia el separador si es diferente
# Asegurarse de que ambas columnas sean del mismo tipo
datos$gene <- trimws(as.character(datos$gene))
go_filtered$geneID <- trimws(as.character(go_filtered$geneID))
# Unir los datos después de separar los genes
merged_data <- merge(datos, go_filtered, by.x = "gene", by.y = "geneID", all.x = FALSE)
# Filtrar valores únicos para logFC
unique_logFC <- merged_data %>%
distinct(gene, .keep_all = TRUE)  # Quedarse solo con una fila por gen
# Ordenar los genes por logFC
unique_logFC <- unique_logFC %>%
arrange(logFC) %>%
mutate(gene = factor(gene, levels = unique(gene)))
# Gráfico principal: logFC de las proteínas (valores únicos)
logfc_plot <- ggplot(unique_logFC, aes(x = gene, y = logFC)) +
geom_bar(stat = "identity", fill = "skyblue") +
theme_minimal() +
theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
labs(title = "logFC de Proteínas (Único y Ordenado por logFC)", x = "Gen", y = "logFC")
# Gráfico suplementario: Funciones GO asociadas a las proteínas
go_plot <- ggplot(merged_data, aes(x = gene, y = ID, fill = p.adjust)) +
geom_tile() +
scale_fill_gradient(low = "lightgreen", high = "red", name = "p.adjust") +
theme_minimal() +
theme(axis.text.x = element_text(angle = 90, hjust = 1),
axis.text.y = element_text(size = 8)) +
labs(title = "Funciones GO asociadas (p.adjust < 0.05)", x = "Gen", y = "Funciones GO")
# Combinar los gráficos
combined_plot <- logfc_plot / go_plot
# Mostrar los gráficos combinados
print(combined_plot)
}
library(tidyverse)
for (archivo in archivos){
grupo <- gsub(".*/.*_([a-zA-Z0-9]{2,})[a-zA-Z]*\\.xlsx$", "\\1", archivo)
sample <- gsub(".*([a-zA-Z])\\.xlsx$", "\\1", archivo)
directory <- paste0(grupo,sample)
datos <- read_excel(paste0("combined/", directory,"/", directory,".xlsx"), sheet = "Datos")
go <- read_excel(paste0("combined/", directory,"/", directory,".xlsx"), sheet = "Go")
```

```
# Filtrar funciones GO con p.adjust < 0.05
go_filtered <- go %>% filter(p.adjust < 0.05)
# Separar los genes en la columna geneID en filas individuales
go_filtered <- go_filtered %>%
separate_rows(geneID, sep = "/")  # Cambia el separador si es diferente
# Asegurarse de que ambas columnas sean del mismo tipo
datos$gene <- trimws(as.character(datos$gene))
go_filtered$geneID <- trimws(as.character(go_filtered$geneID))
# Unir los datos después de separar los genes
merged_data <- merge(datos, go_filtered, by.x = "gene", by.y = "geneID", all.x = FALSE)
# Filtrar valores únicos para logFC
unique_logFC <- merged_data %>%
distinct(gene, .keep_all = TRUE)  # Quedarse solo con una fila por gen
# Ordenar los genes por logFC
unique_logFC <- unique_logFC %>%
arrange(logFC) %>%
mutate(gene = factor(gene, levels = unique(gene)))
# Gráfico principal: logFC de las proteínas (valores únicos)
logfc_plot <- ggplot(unique_logFC, aes(x = gene, y = logFC)) +
geom_bar(stat = "identity", fill = "skyblue") +
theme_minimal() +
theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
labs(title = "logFC de Proteínas (Único y Ordenado por logFC)", x = "Gen", y = "logFC")
# Gráfico suplementario: Funciones GO asociadas a las proteínas
go_plot <- ggplot(merged_data, aes(x = gene, y = ID, fill = p.adjust)) +
geom_tile() +
scale_fill_gradient(low = "lightgreen", high = "red", name = "p.adjust") +
theme_minimal() +
theme(axis.text.x = element_text(angle = 90, hjust = 1),
axis.text.y = element_text(size = 8)) +
labs(title = "Funciones GO asociadas (p.adjust < 0.05)", x = "Gen", y = "Funciones GO")
# Combinar los gráficos
combined_plot <- logfc_plot / go_plot
# Mostrar los gráficos combinados
print(combined_plot)
}
knitr::opts_chunk$set(echo = TRUE)
library(dplyr)
library(msdap)
library(limma)
library(Biostrings)
library(writexl)
library(openxlsx)
library(readxl)
dataset <- import_dataset_diann(filename = "report.tsv")
dataset <- import_fasta(dataset, files = "archivo_combinado.fasta")
dataset <- import_sample_metadata(dataset, filename = "sample_metadata.xlsx")
dataset = setup_contrasts(dataset,
contrast_list = list(c("MWd4_L","MSd4_L"),
c("MWd4_Exo","MSd4_Exo"),
c("MWd4_S","MSd4_S"),
c("FW_L","FS_L"),
c("FW_S","FS_S"),
c("MW_L","MS_L"),
c("MW_S","MS_S")
))
dataset = analysis_quickstart(dataset,
filter_min_detect = 1,
filter_min_quant = 1,
filter_fraction_detect = 0.0,
filter_fraction_quant = 0.0,
filter_min_peptide_per_prot = 1,
filter_by_contrast = TRUE,
norm_algorithm = c(
"vsn",
"modebetween_protein"
),
dea_algorithm = c("deqms",
"msempire", "msqrob"
),
dea_qvalue_threshold = 0.05,
dea_log2foldchange_threshold = NA,
diffdetect_min_peptides_observed = 1,
diffdetect_min_samples_observed = 1,
diffdetect_min_fraction_observed = 0,
```

```
output_qc_report = TRUE,
output_abundance_tables = TRUE,
output_dir = "msdap_results",
output_within_timestamped_subdirectory = TRUE
)
```

Could not pretty-print your R code. Perhaps there are syntax errors in your R history? If so, either clear your R history or restart RStudio to amend.

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 4.3.3 (2024-02-29 ucrt)
os	Windows 10 x64 (build 19045)
system	x86_64, mingw32
ui	RStudio
language	(EN)
collate	Spanish_Spain.utf8
ctype	Spanish_Spain.utf8
tz	Europe/Madrid
date	2024-12-19
rstudio	2023.09.0+463 Desert Sunflower (desktop)
pandoc	3.1.12.2 @ C:/PROGRA~1/Pandoc/ (via rmarkdown)

System

package	loadedversion	source
BiocGenerics	0.48.1	Bioconductor
Biostrings	2.70.3	Bioconductor 3.18 (R 4.3.3)
dplyr	1.1.4	CRAN (R 4.3.3)
GenomeInfoDb	1.38.8	Bioconductor 3.18 (R 4.3.3)
ggplot2	3.5.1	CRAN (R 4.3.3)
IRanges	2.36.0	Bioconductor
limma	3.58.1	Bioconductor
msdap	1.2.1	Github (ftwkoopmans/msdap@fa8e9cbd66f7dafca6b124ea68286de90da7d239)
openxlsx	4.2.7.1	CRAN (R 4.3.3)
readxl	1.4.3	CRAN (R 4.3.3)
rlang	1.1.1	CRAN (R 4.3.1)
S4Vectors	0.40.2	Bioconductor
tibble	3.2.1	CRAN (R 4.3.3)
tidyr	1.3.1	CRAN (R 4.3.3)
writexl	1.5.1	CRAN (R 4.3.3)
XVector	0.42.0	Bioconductor

Attached packages

package	loadedversion	source
abind	1.4-8	CRAN (R 4.3.3)
affy	1.80.0	Bioconductor
affyio	1.72.0	Bioconductor
AnnotationDbi	1.64.1	Bioconductor
aod	1.3.3	CRAN (R 4.3.3)
ape	5.8	CRAN (R 4.3.3)
aplot	0.2.3	CRAN (R 4.3.3)
archive	1.1.10	CRAN (R 4.3.3)
askpass	1.2.1	CRAN (R 4.3.3)
backports	1.5.0	CRAN (R 4.3.3)
Biobase	2.62.0	Bioconductor
BiocManager	1.30.25	CRAN (R 4.3.3)
BiocParallel	1.36.0	Bioconductor
bit	4.5.0.1	CRAN (R 4.3.3)
bit64	4.5.2	CRAN (R 4.3.3)
bitops	1.0-9	CRAN (R 4.3.3)
blob	1.2.4	CRAN (R 4.3.3)
boot	1.3-31	CRAN (R 4.3.3)
broom	1.0.7	CRAN (R 4.3.3)
cachem	1.1.0	CRAN (R 4.3.3)
car	3.1-3	CRAN (R 4.3.3)
carData	3.0-5	CRAN (R 4.3.3)
caTools	1.18.3	CRAN (R 4.3.3)
cellranger	1.1.0	CRAN (R 4.3.3)
cli	3.6.1	CRAN (R 4.3.1)
clue	0.3-66	CRAN (R 4.3.3)
cluster	2.1.8	CRAN (R 4.3.3)
clusterProfiler	4.10.1	Bioconductor 3.18 (R 4.3.3)
codetools	0.2-20	CRAN (R 4.3.3)
colorspace	2.1-1	CRAN (R 4.3.3)
corpcor	1.6.10	CRAN (R 4.3.1)
cowplot	1.1.3	CRAN (R 4.3.3)
crayon	1.5.3	CRAN (R 4.3.3)
data.table	1.16.4	CRAN (R 4.3.3)
DBI	1.2.3	CRAN (R 4.3.3)
DEqMS	1.20.0	Bioconductor
devtools	2.4.5	CRAN (R 4.3.3)
diann	1.0.1	Github (vdemichev/diann-rpackage@af538f6e2cd5ab715e1381632e17cb8f234ebf53)
digest	0.6.33	CRAN (R 4.3.1)
doParallel	1.0.17	CRAN (R 4.3.3)
doRNG	1.8.6	CRAN (R 4.3.3)
DOSE	3.28.2	Bioconductor
ellipsis	0.3.2	CRAN (R 4.3.3)
enrichplot	1.22.0	Bioconductor
EnvStats	3.0.0	CRAN (R 4.3.3)
evaluate	1.0.1	CRAN (R 4.3.3)
fANCOVA	0.6-1	CRAN (R 4.3.1)
fansi	1.0.6	CRAN (R 4.3.3)
farver	2.1.2	CRAN (R 4.3.3)
fastmap	1.2.0	CRAN (R 4.3.3)

package	loadedversion	source
fastmatch	1.1-4	CRAN (R 4.3.1)
fgsea	1.28.0	Bioconductor
foreach	1.5.2	CRAN (R 4.3.3)
formatR	1.14	CRAN (R 4.3.3)
Formula	1.2-5	CRAN (R 4.3.1)
fs	1.6.5	CRAN (R 4.3.3)
generics	0.1.3	CRAN (R 4.3.3)
GenomeInfoDbData	1.2.11	Bioconductor
ggforce	0.4.2	CRAN (R 4.3.3)
ggfun	0.1.8	CRAN (R 4.3.3)
ggplotify	0.1.2	CRAN (R 4.3.3)
ggpubr	0.6.0	CRAN (R 4.3.3)
ggraph	2.2.1	CRAN (R 4.3.3)
ggrepel	0.9.6	CRAN (R 4.3.3)
ggsignif	0.6.4	CRAN (R 4.3.3)
ggtree	3.10.1	Bioconductor 3.18 (R 4.3.2)
glue	1.8.0	CRAN (R 4.3.3)
GO.db	3.18.0	Bioconductor
GOSemSim	2.28.1	Bioconductor 3.18 (R 4.3.2)
gplots	3.2.0	CRAN (R 4.3.3)
graphlayouts	1.2.1	CRAN (R 4.3.3)
gridExtra	2.3	CRAN (R 4.3.3)
gridGraphics	0.5-1	CRAN (R 4.3.3)
gson	0.1.0	CRAN (R 4.3.3)
gtable	0.3.6	CRAN (R 4.3.3)
gtools	3.9.5	CRAN (R 4.3.3)
HDO.db	0.99.1	Bioconductor
hms	1.1.3	CRAN (R 4.3.3)
htmltools	0.5.8.1	CRAN (R 4.3.3)
htmlwidgets	1.6.4	CRAN (R 4.3.3)
httpuv	1.6.15	CRAN (R 4.3.3)
httr	1.4.7	CRAN (R 4.3.3)
igraph	2.1.2	CRAN (R 4.3.3)
impute	1.76.0	Bioconductor
iq	1.10.1	CRAN (R 4.3.3)
iterators	1.0.14	CRAN (R 4.3.3)
itertools	0.1-3	CRAN (R 4.3.3)
jsonlite	1.8.9	CRAN (R 4.3.3)
KEGGREST	1.42.0	Bioconductor
KernSmooth	2.23-24	CRAN (R 4.3.3)
knitr	1.49	CRAN (R 4.3.3)
labeling	0.4.3	CRAN (R 4.3.1)
later	1.4.1	CRAN (R 4.3.3)
lattice	0.22-6	CRAN (R 4.3.3)
lazyeval	0.2.2	CRAN (R 4.3.3)
lifecycle	1.0.4	CRAN (R 4.3.3)
lme4	1.1-35.5	CRAN (R 4.3.3)
lmerTest	3.1-3	CRAN (R 4.3.3)
magrittr	2.0.3	CRAN (R 4.3.3)
MALDIquant	1.22.3	CRAN (R 4.3.3)

package	loadedversion	source
MASS	7.3-60.0.1	CRAN (R 4.3.3)
Matrix	1.6-4	CRAN (R 4.3.2)
matrixStats	1.4.1	CRAN (R 4.3.3)
memoise	2.0.1	CRAN (R 4.3.3)
mime	0.12	CRAN (R 4.3.1)
miniUI	0.1.1.1	CRAN (R 4.3.3)
minqa	1.2.8	CRAN (R 4.3.3)
missForest	1.5	CRAN (R 4.3.3)
MsCoreUtils	1.14.1	Bioconductor
msEmpiRe	0.1.0	Github (zimmerlab/MS-EmpiRe@8a85757c8d604014130ee9d379aa8cfeb05e3855)
MSnbase	2.28.1	Bioconductor
munsell	0.5.1	CRAN (R 4.3.3)
mvtnorm	1.3-2	CRAN (R 4.3.3)
mzID	1.40.0	Bioconductor
mzR	2.36.0	Bioconductor
ncdf4	1.23	CRAN (R 4.3.3)
nlme	3.1-166	CRAN (R 4.3.3)
nloptr	2.1.1	CRAN (R 4.3.3)
numDeriv	2016.8-1.1	CRAN (R 4.3.1)
openssl	2.2.2	CRAN (R 4.3.3)
patchwork	1.3.0	CRAN (R 4.3.3)
pbkrtest	0.5.3	CRAN (R 4.3.3)
pcaMethods	1.94.0	Bioconductor
pdftools	3.4.1	CRAN (R 4.3.3)
pillar	1.9.0	CRAN (R 4.3.3)
pkgbuild	1.4.5	CRAN (R 4.3.3)
pkgconfig	2.0.3	CRAN (R 4.3.3)
pkgload	1.4.0	CRAN (R 4.3.3)
plyr	1.8.9	CRAN (R 4.3.3)
png	0.1-8	CRAN (R 4.3.1)
polyclip	1.10-7	CRAN (R 4.3.3)
preprocessCore	1.64.0	Bioconductor
pROC	1.18.5	CRAN (R 4.3.3)
profvis	0.4.0	CRAN (R 4.3.3)
promises	1.3.2	CRAN (R 4.3.3)
ProtGenerics	1.34.0	Bioconductor
purrr	1.0.2	CRAN (R 4.3.3)
qpdf	1.3.4	CRAN (R 4.3.3)
qvalue	2.34.0	Bioconductor
R.cache	0.16.0	CRAN (R 4.3.3)
R.methodsS3	1.8.2	CRAN (R 4.3.3)
R.oo	1.27.0	CRAN (R 4.3.3)
R.utils	2.12.3	CRAN (R 4.3.3)
R6	2.5.1	CRAN (R 4.3.3)
randomForest	4.7-1.2	CRAN (R 4.3.3)
rbibutils	2.3	CRAN (R 4.3.3)
RColorBrewer	1.1-3	CRAN (R 4.3.1)
Rcpp	1.0.13-1	CRAN (R 4.3.3)
RcppEigen	0.3.4.0.2	CRAN (R 4.3.3)
RCurl	1.98-1.16	CRAN (R 4.3.3)

package	loadedversion	source
Rdpack	2.6.2	CRAN (R 4.3.3)
readr	2.1.5	CRAN (R 4.3.3)
remaCor	0.0.18	CRAN (R 4.3.3)
remotes	2.5.0	CRAN (R 4.3.3)
reshape2	1.4.4	CRAN (R 4.3.3)
RhpcBLASctl	0.23-42	CRAN (R 4.3.1)
rmarkdown	2.29	CRAN (R 4.3.3)
rngtools	1.5.2	CRAN (R 4.3.3)
RSQLite	2.3.9	CRAN (R 4.3.3)
rstatix	0.7.2	CRAN (R 4.3.3)
rstudioapi	0.17.1	CRAN (R 4.3.3)
scales	1.3.0	CRAN (R 4.3.3)
scatterpie	0.2.4	CRAN (R 4.3.3)
sessioninfo	1.2.2	CRAN (R 4.3.3)
shadowtext	0.1.4	CRAN (R 4.3.3)
shiny	1.9.1	CRAN (R 4.3.3)
statmod	1.5.0	CRAN (R 4.3.3)
stringi	1.8.4	CRAN (R 4.3.3)
stringr	1.5.1	CRAN (R 4.3.3)
styler	1.10.3	CRAN (R 4.3.3)
tidygraph	1.3.1	CRAN (R 4.3.3)
tidyselect	1.2.1	CRAN (R 4.3.3)
tidytree	0.4.6	CRAN (R 4.3.3)
tinytex	0.54	CRAN (R 4.3.3)
treeio	1.26.0	Bioconductor
tweenr	2.0.3	CRAN (R 4.3.3)
tzdb	0.4.0	CRAN (R 4.3.3)
urlchecker	1.0.1	CRAN (R 4.3.3)
usethis	3.1.0	CRAN (R 4.3.3)
utf8	1.2.4	CRAN (R 4.3.3)
variancePartition	1.32.5	Bioconductor 3.18 (R 4.3.2)
vctrs	0.6.5	CRAN (R 4.3.3)
viridis	0.6.5	CRAN (R 4.3.3)
viridisLite	0.4.2	CRAN (R 4.3.3)
vroom	1.6.5	CRAN (R 4.3.3)
vsn	3.70.0	Bioconductor
withr	3.0.2	CRAN (R 4.3.3)
xfun	0.49	CRAN (R 4.3.3)
XML	3.99-0.17	CRAN (R 4.3.3)
xtable	1.8-4	CRAN (R 4.3.3)
yaml	2.3.10	CRAN (R 4.3.3)
yulab.utils	0.1.8	CRAN (R 4.3.3)
zip	2.3.1	CRAN (R 4.3.3)
zlibbioc	1.48.2	Bioconductor 3.18 (R 4.3.3)

Packages that are not attached