

MS-DAP: Mass Spectrometry Downstream Analysis Pipeline

version: 1.0.7 <https://github.com/ftwkoopmans/msdap/>

Contents

1	Quality control	2
1.1	number of peptides and proteins	2
1.2	data completeness	5
1.3	abundance distributions	7
1.4	retention time	10
1.5	variation among replicates	18
1.6	PCA	27
2	Differential abundance analysis	32
2.1	Axillary_Buds_0h vs Axillary_Buds_2h	32
2.2	Apical_Buds_0h vs Apical_Buds_2h	37
2.3	Apical_Buds_0h vs Axillary_Buds_0h	42
3	Summary of differential testing	47
4	log	48
5	R command history	50
6	R session info	52

1 Quality control

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

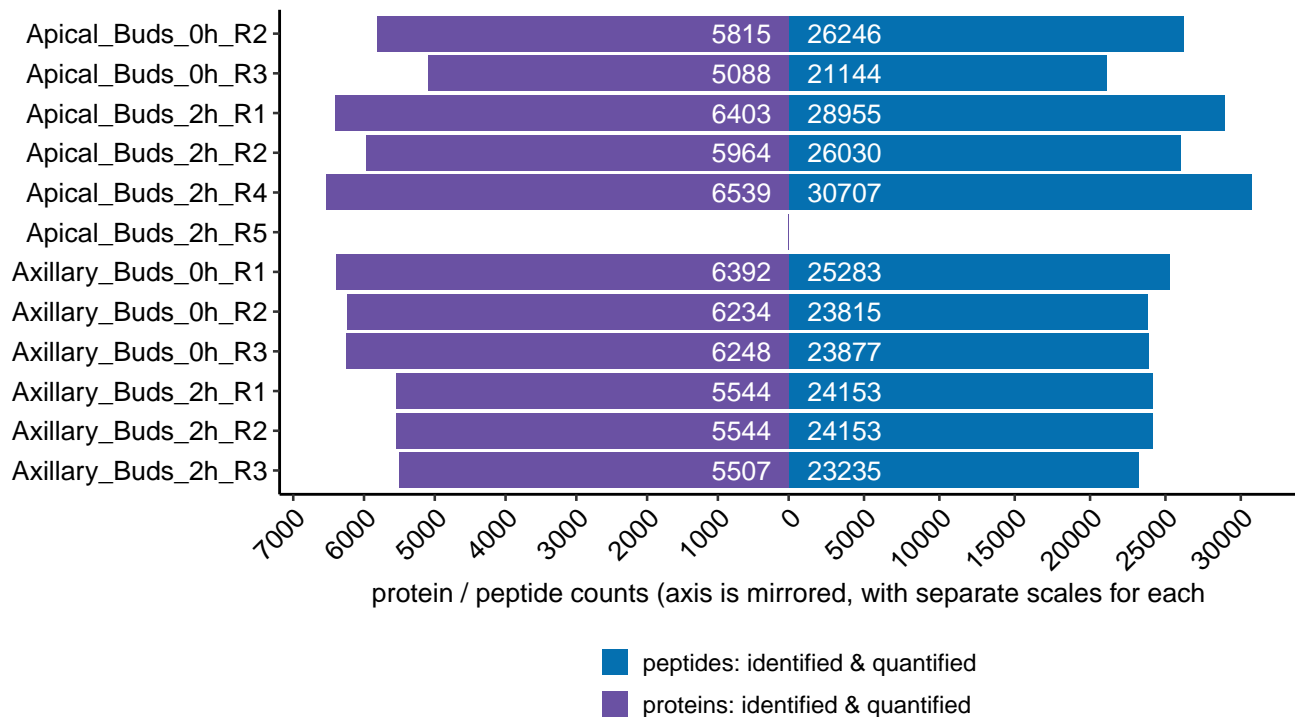
- number of peptides/proteins detected in each sample
- dataset completeness
- local effects in HPLC peptide retention time per sample
- reproducibility of peptide quantification among replicates
- PCA of all samples to visualize clustering

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

1.1 number of peptides and proteins

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

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

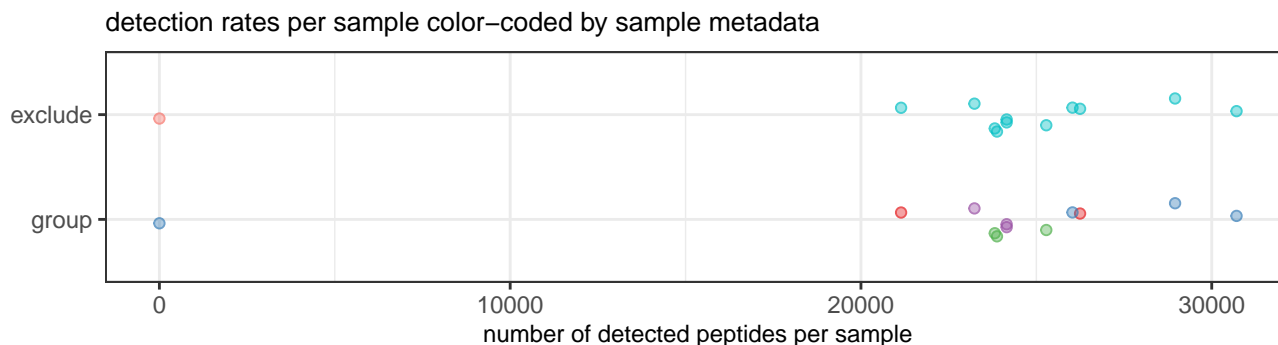


1.1.1 color-coding sample metadata

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

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

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



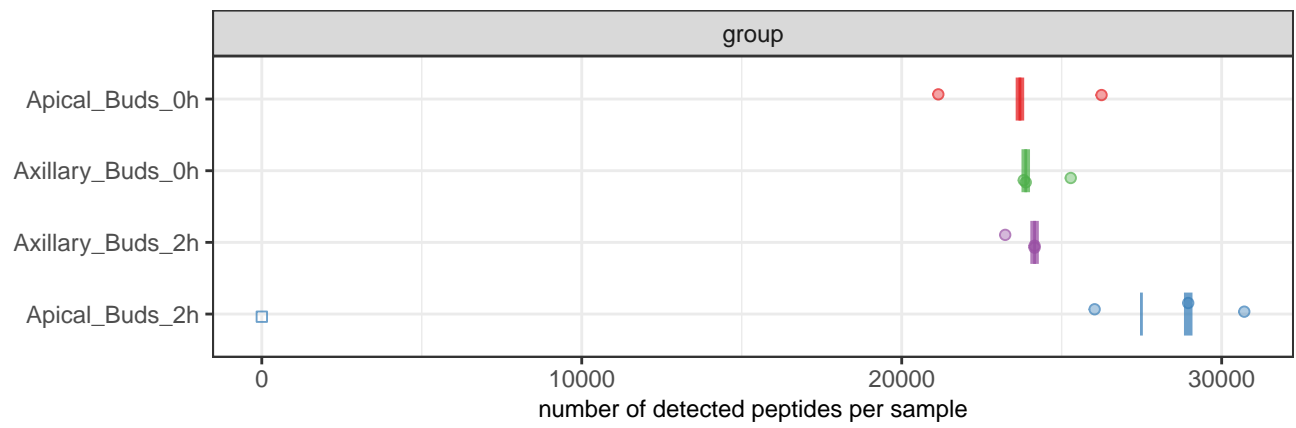
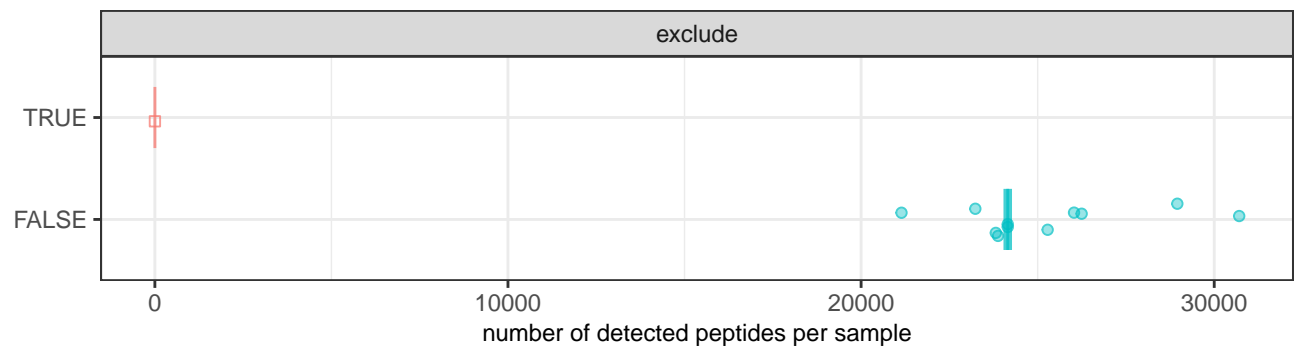
color-coding sample metadata, expanded

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

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

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

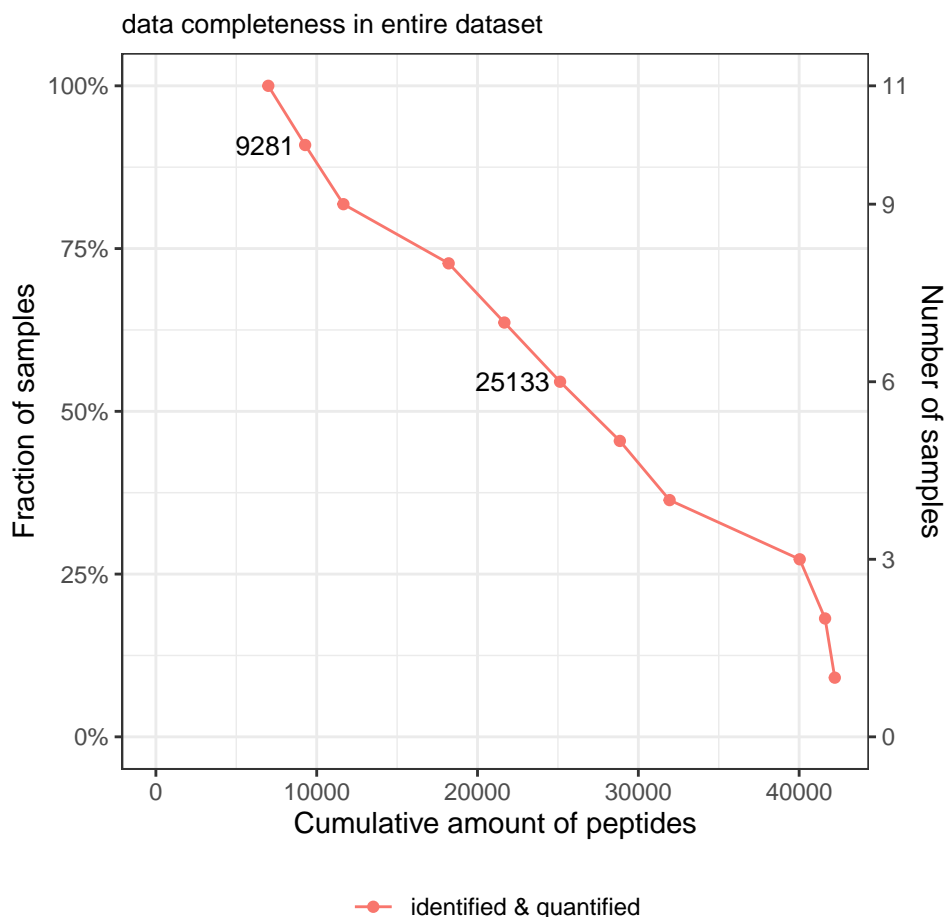
For example: the first plot shows color-coding by the ‘group’ property, so each row represents a sample group. If samples in a particular group systematically yield fewer peptides than another group, a clear pattern will be visible.



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

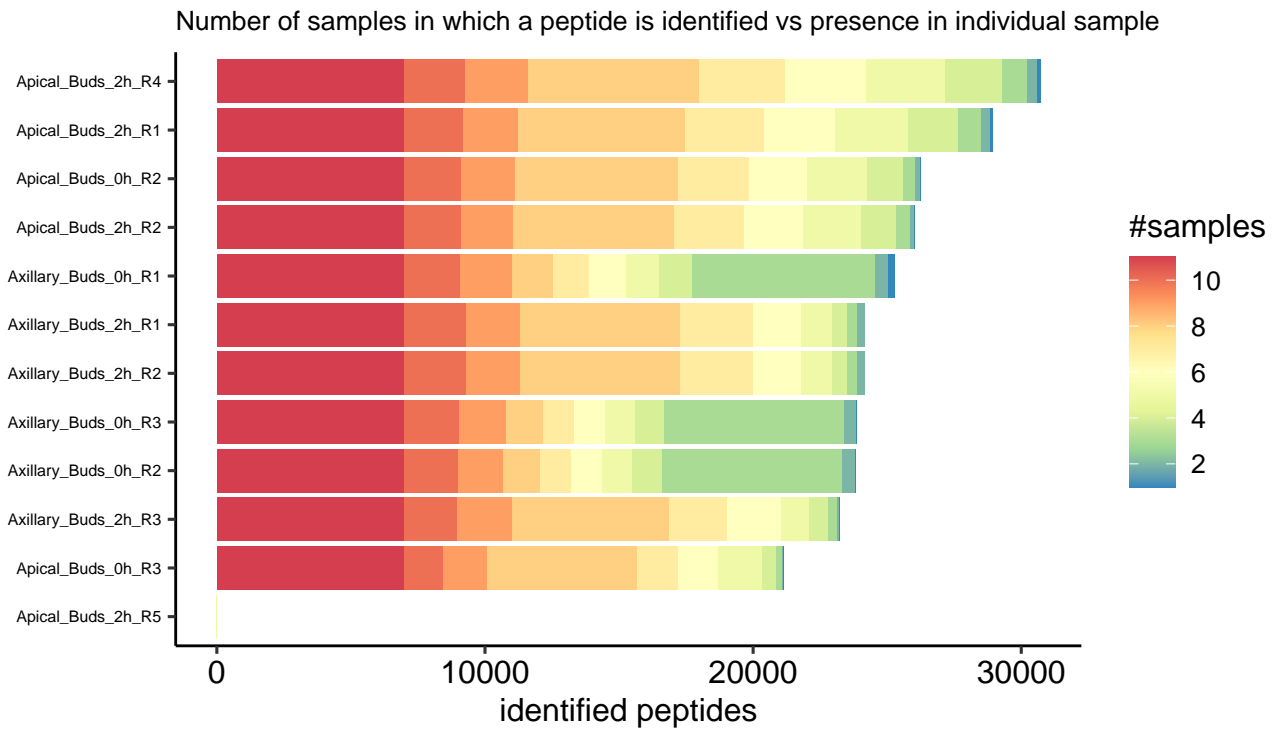


Samples flagged as 'exclude' (by user) are not taken into account in this figure. Exact values are shown for data points matching 90% and 50% of samples to convenience comparison between analyses (e.g. before/after configuring 'exclude' samples, or comparing between experiments of similar protocol).

1.2.2 peptide detection frequency

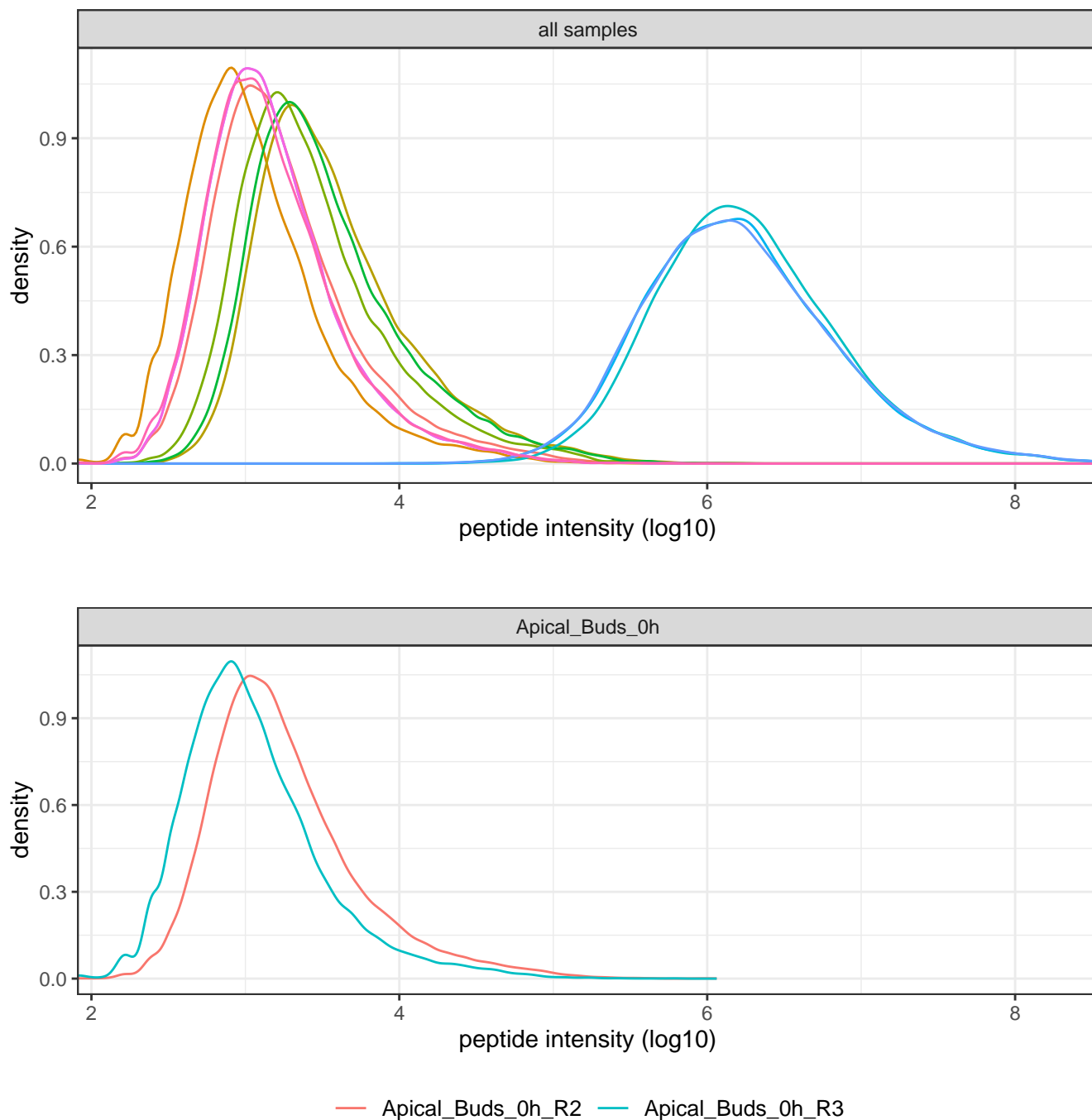
Each identified peptide in a sample is classified and color-coded by the number of other samples where the same peptide is present. Visualization of the amount of peptides that overlap with other samples in the dataset, from peptides identified in most samples (red) to one-hit-wonders (blue), helps identify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

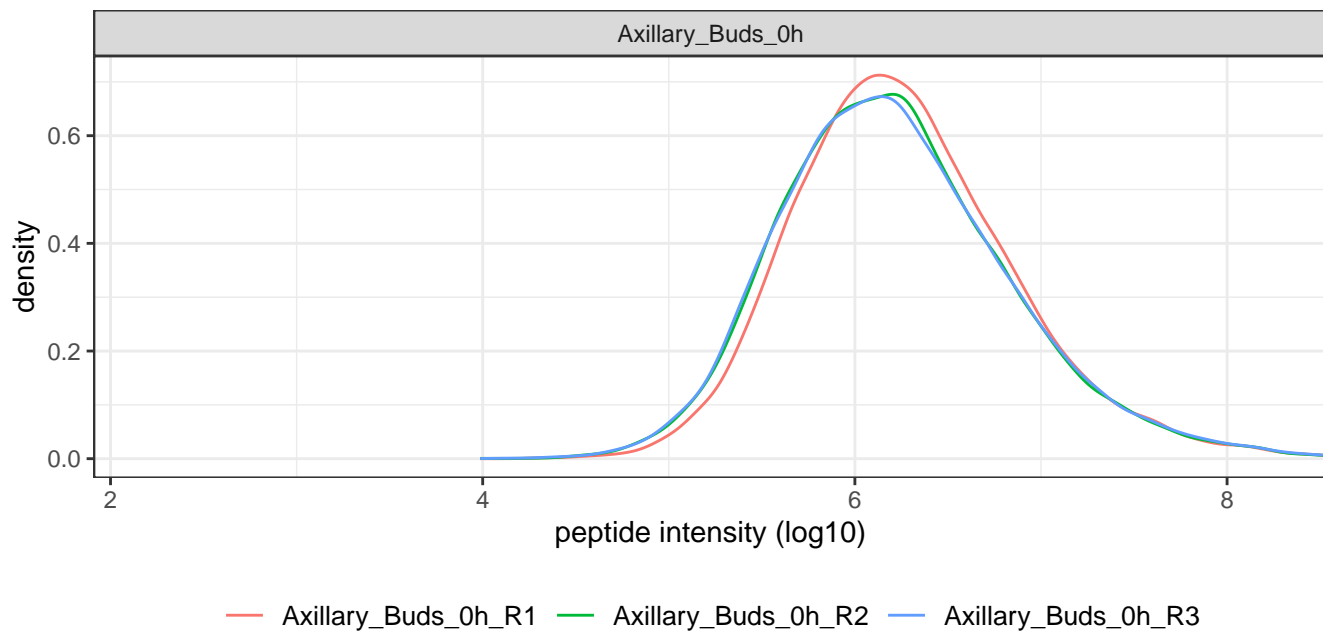
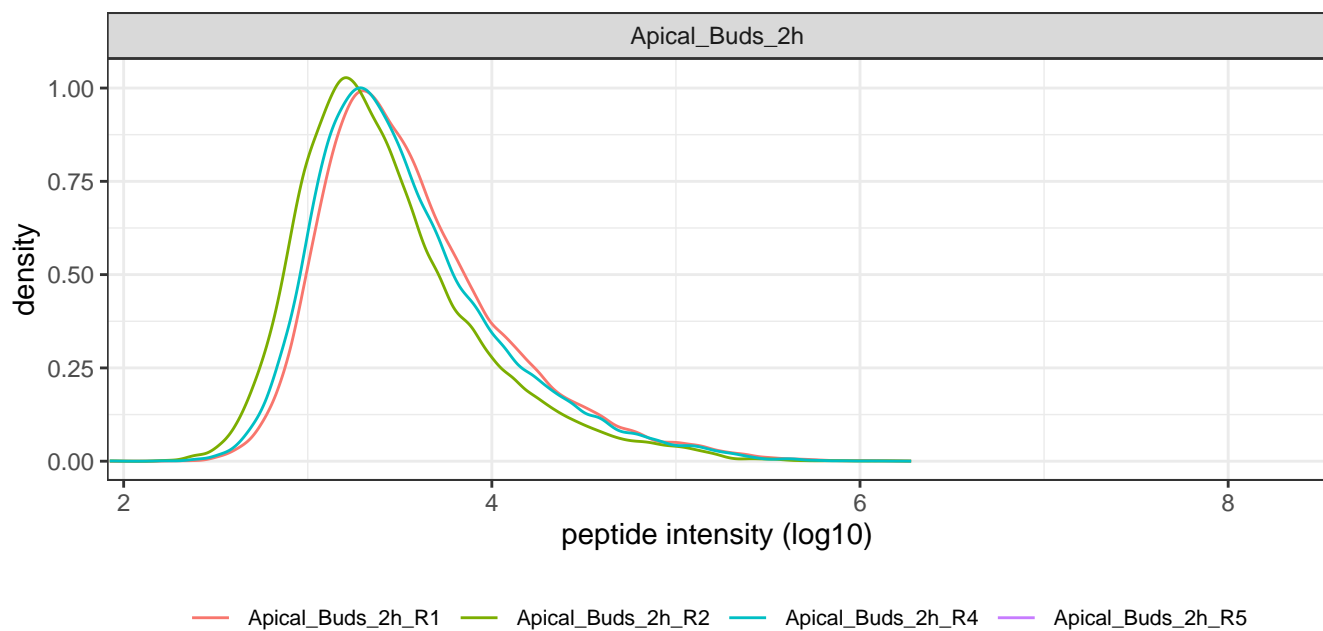
Optimally, the majority of peptides in each sample are red~orange with relatively few uniquely identified peptides (blue~green). Samples are sorted by the total amount of detected peptides.

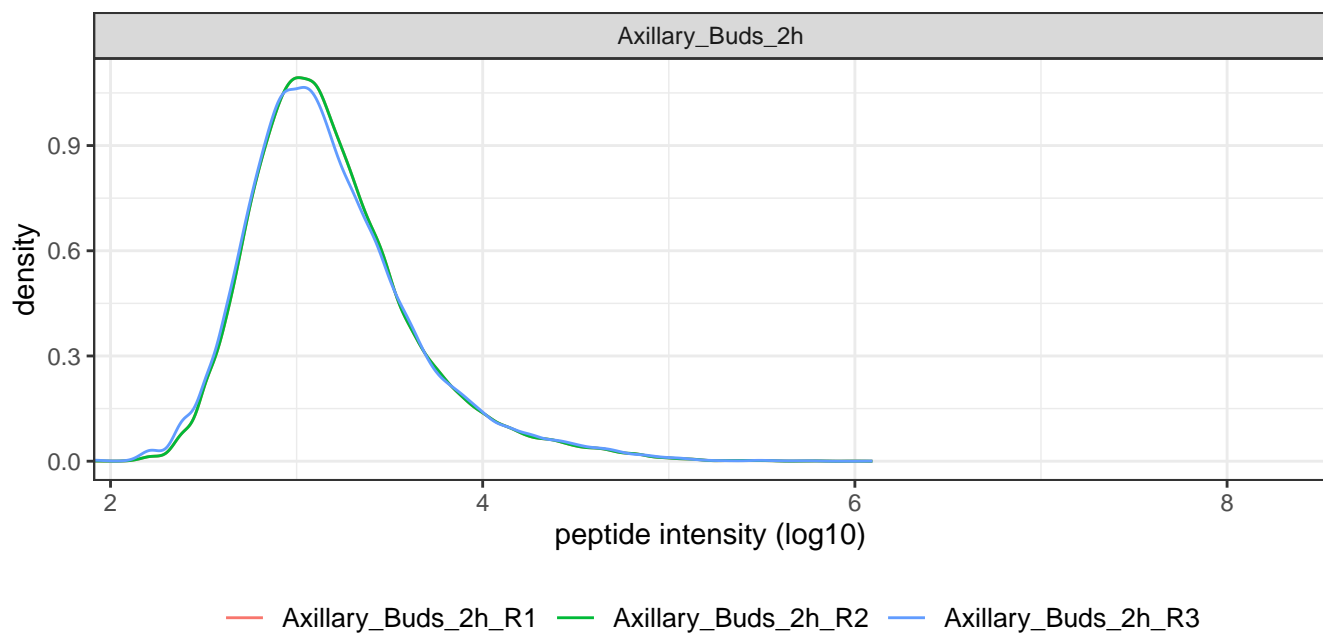


1.3 abundance distributions

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.





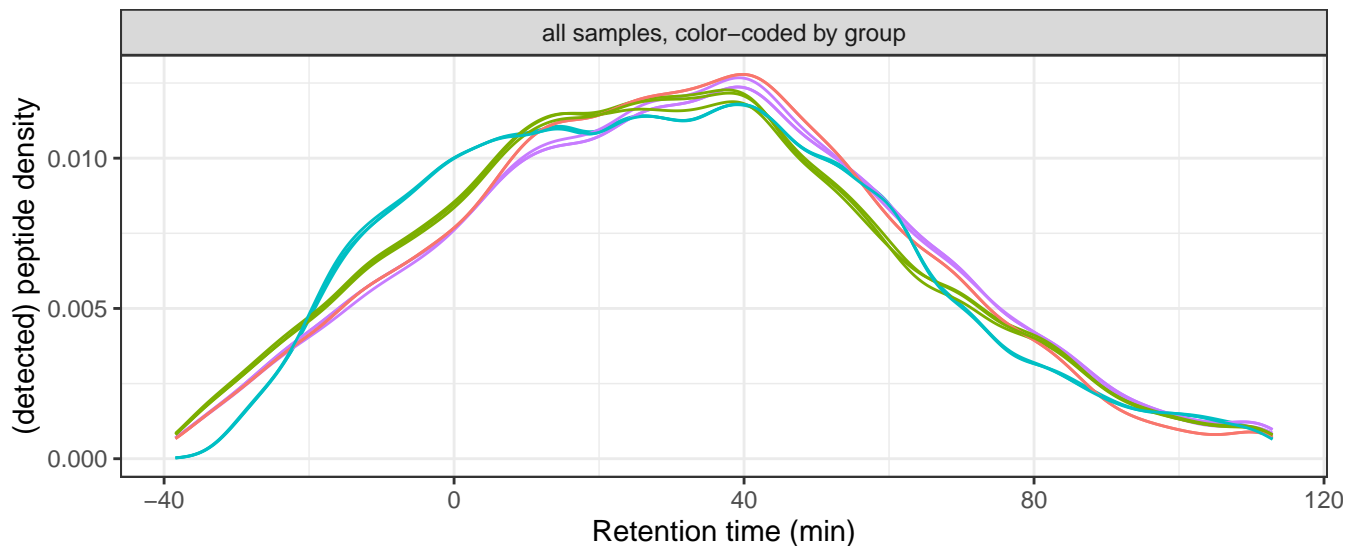
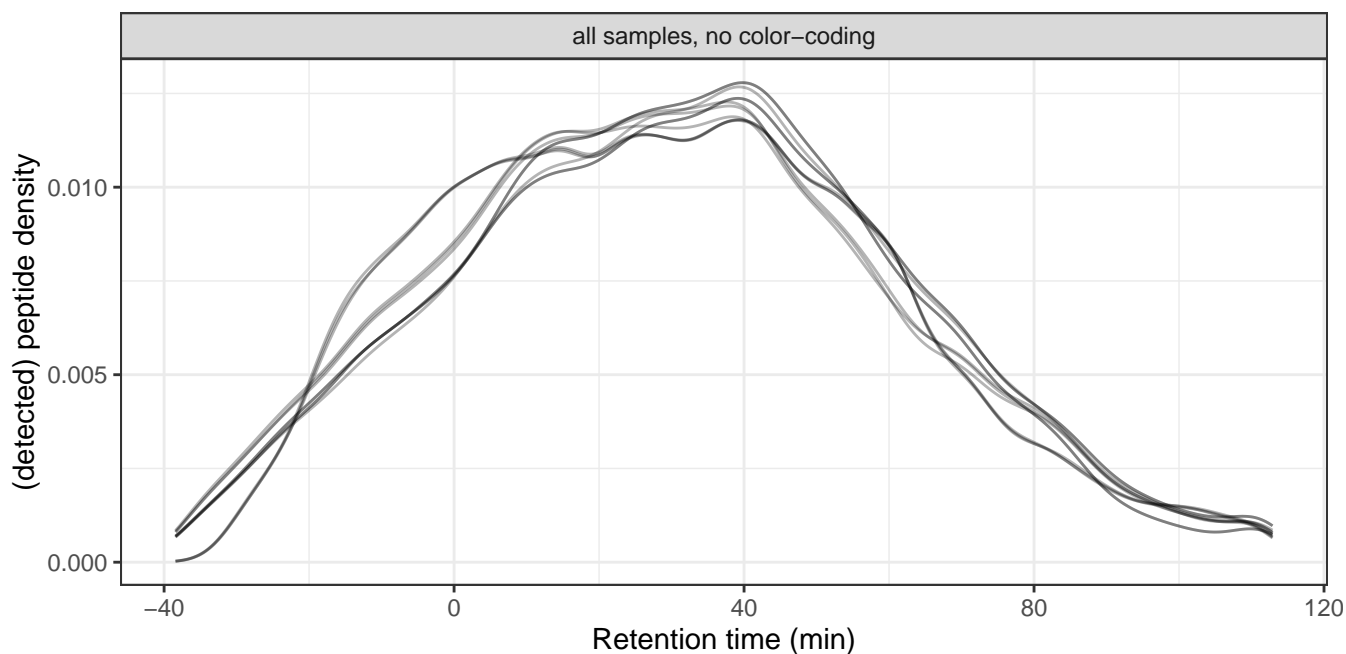


1.4 retention time

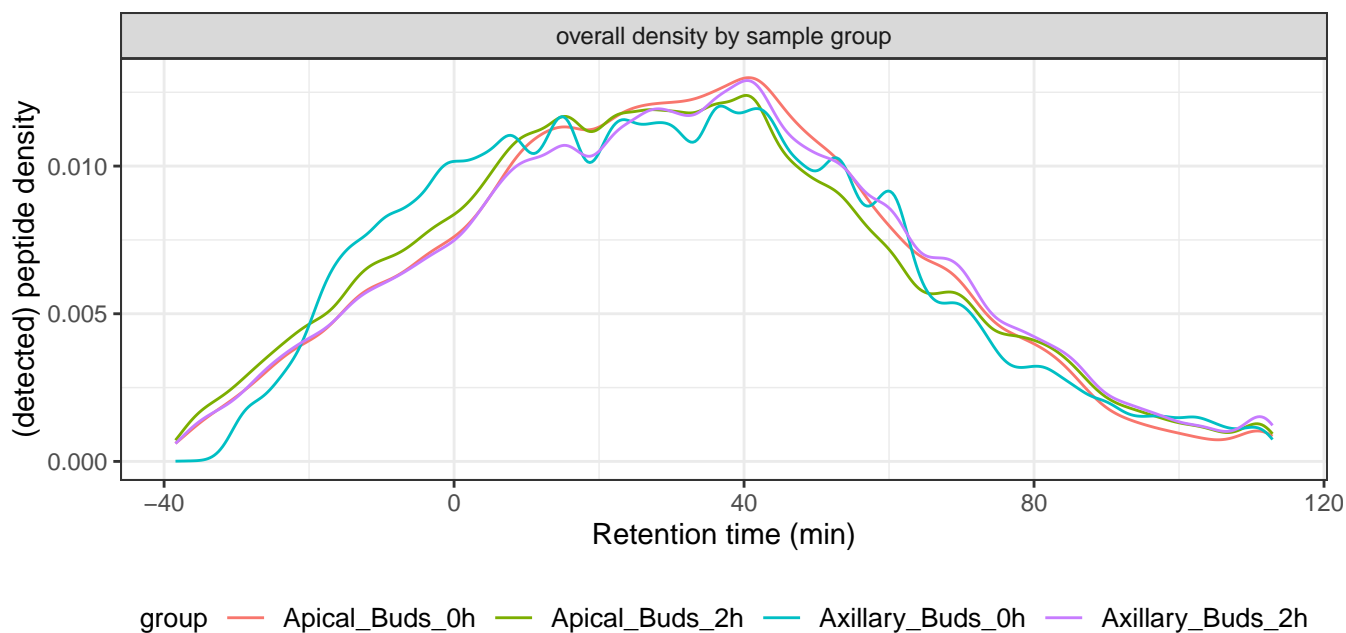
The figures in this section allow you to identify potential problems during HPLC elution, such as a temporarily blocking column, failing ionization spray or decreasing sensitivity over time. For each sample, all peptides that are also observed in a replicate (such that there is a point of reference available)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



group — Apical_Buds_0h — Apical_Buds_2h — Axillary_Buds_0h — Axillary_Buds_2h

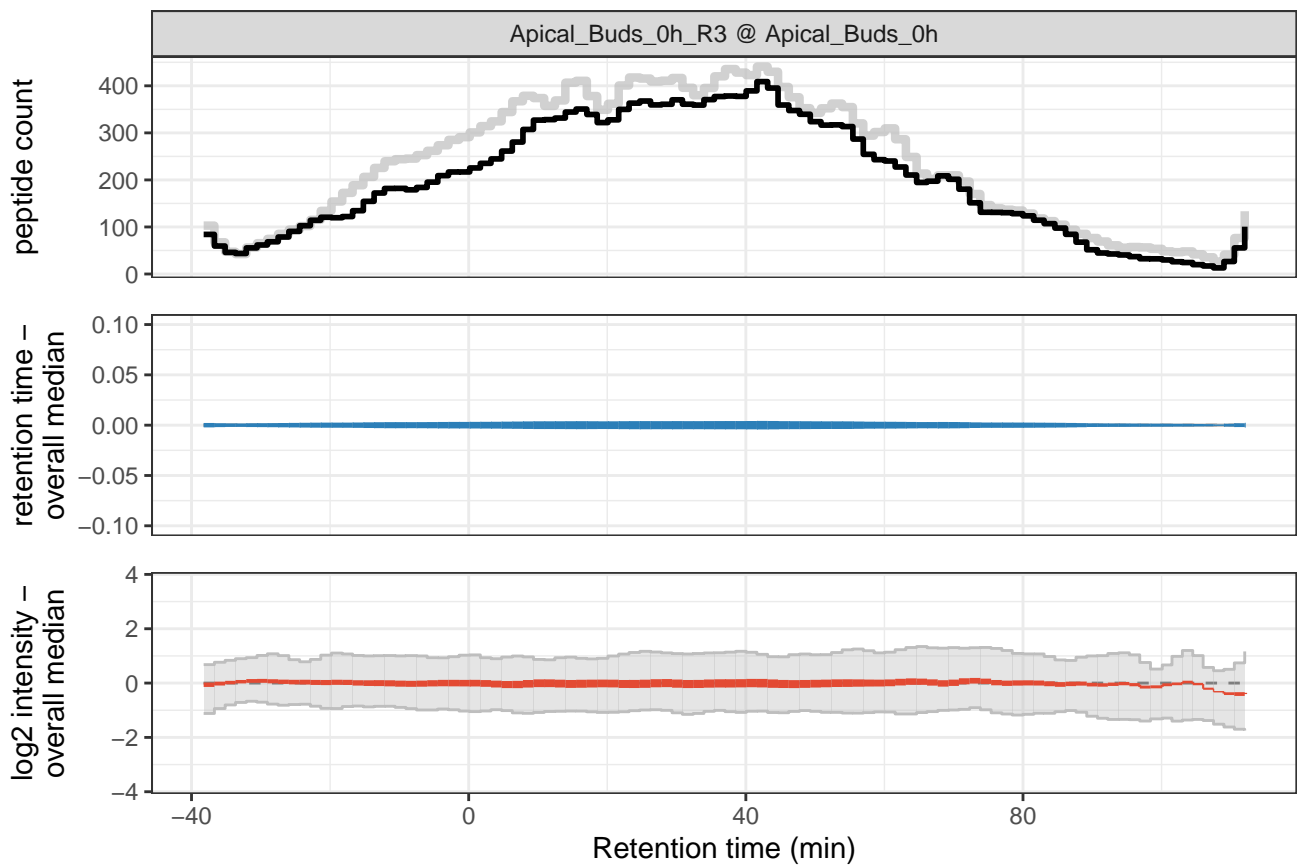
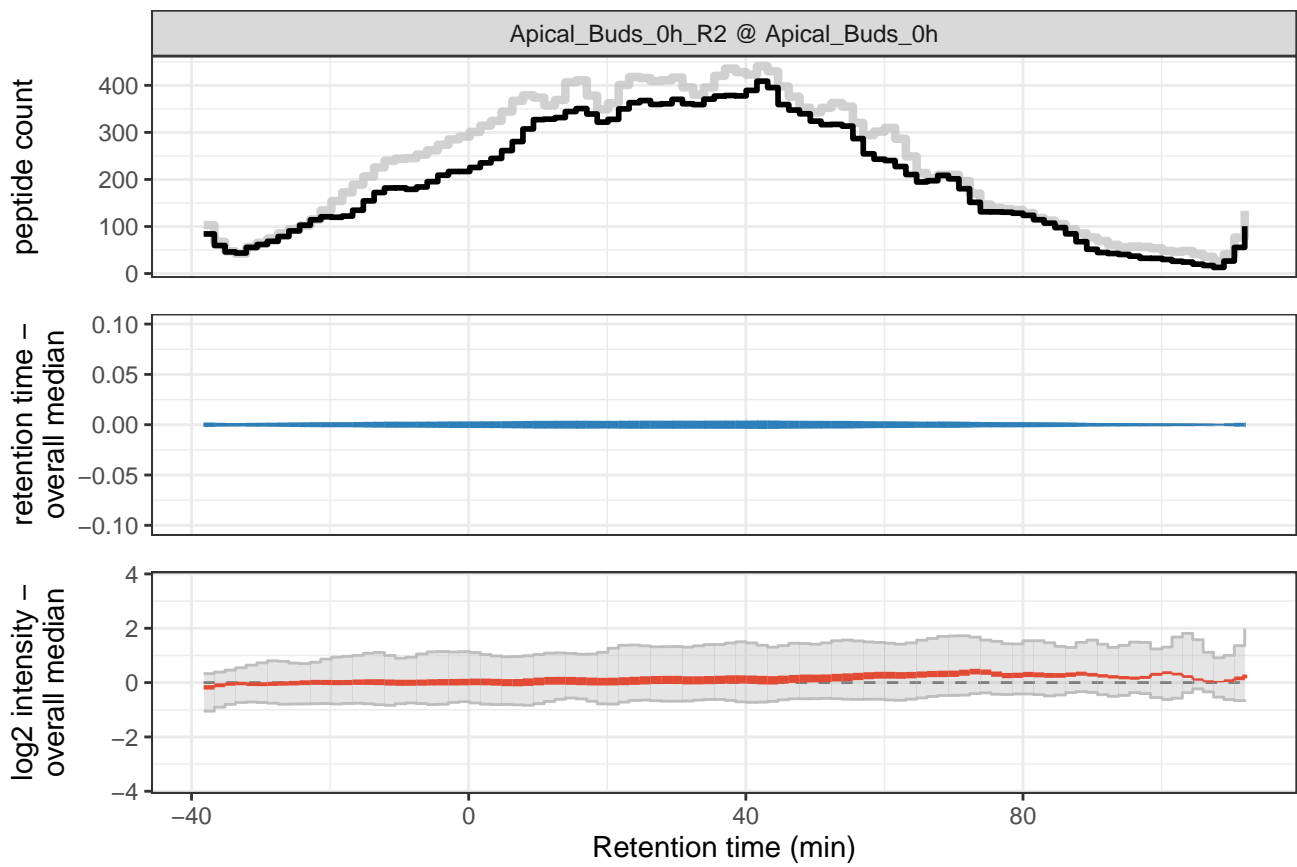


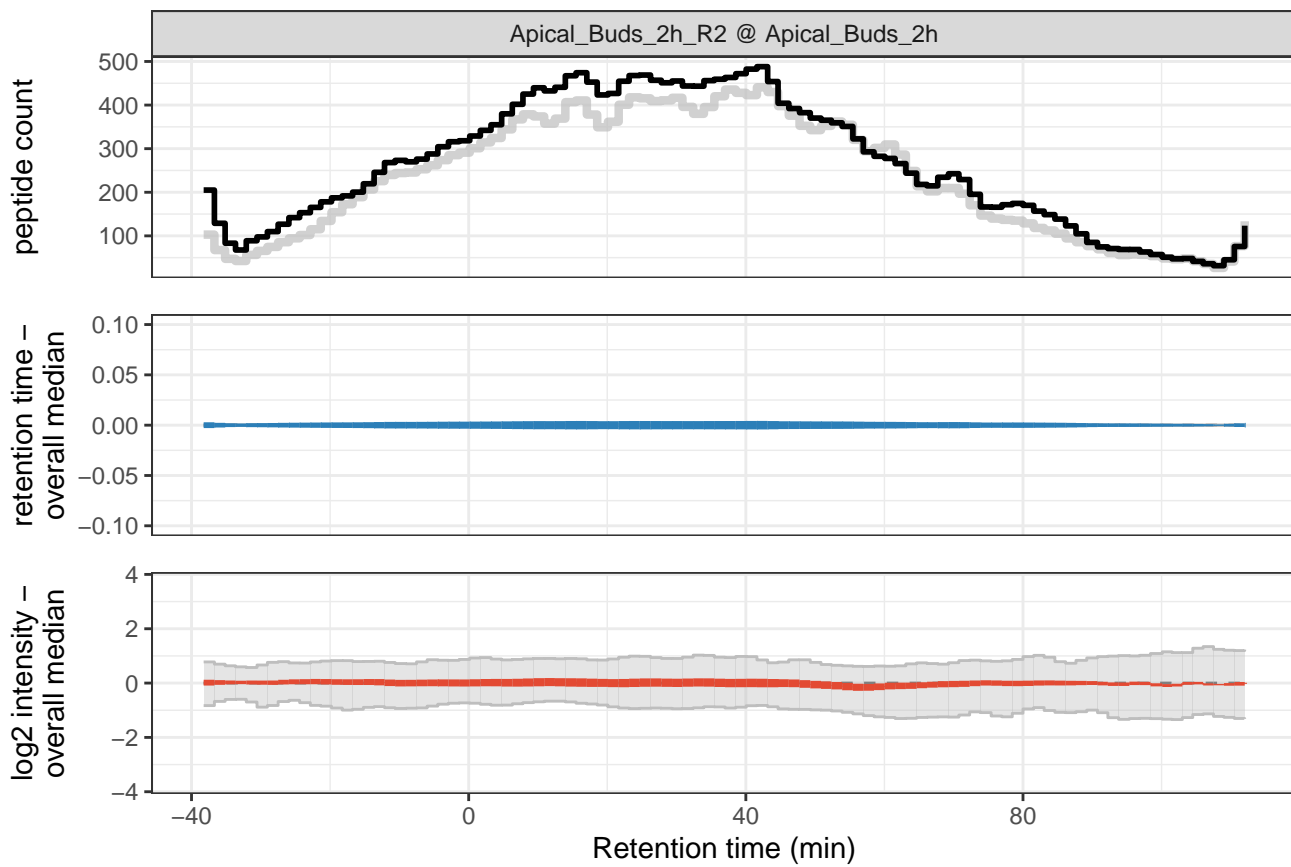
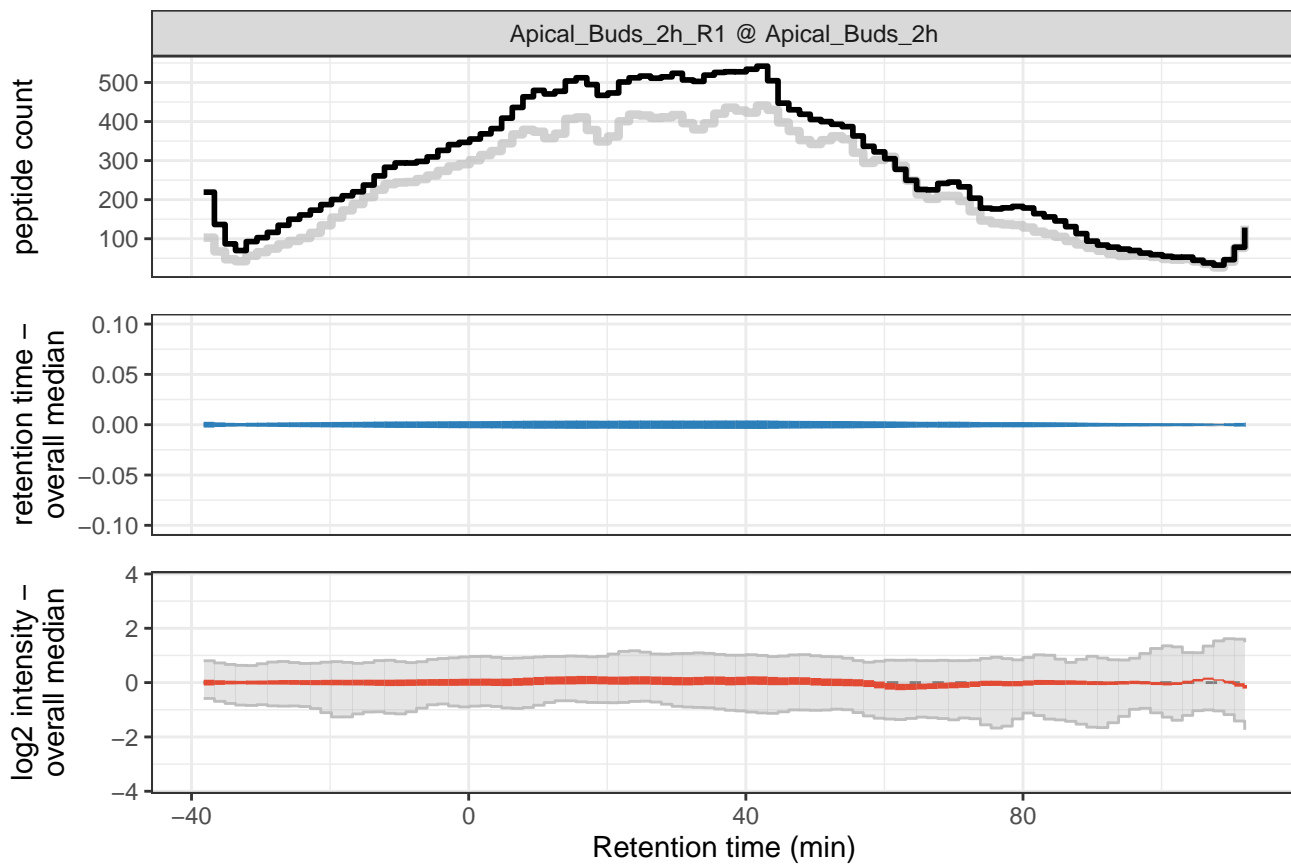
1.4.2 retention time local effects

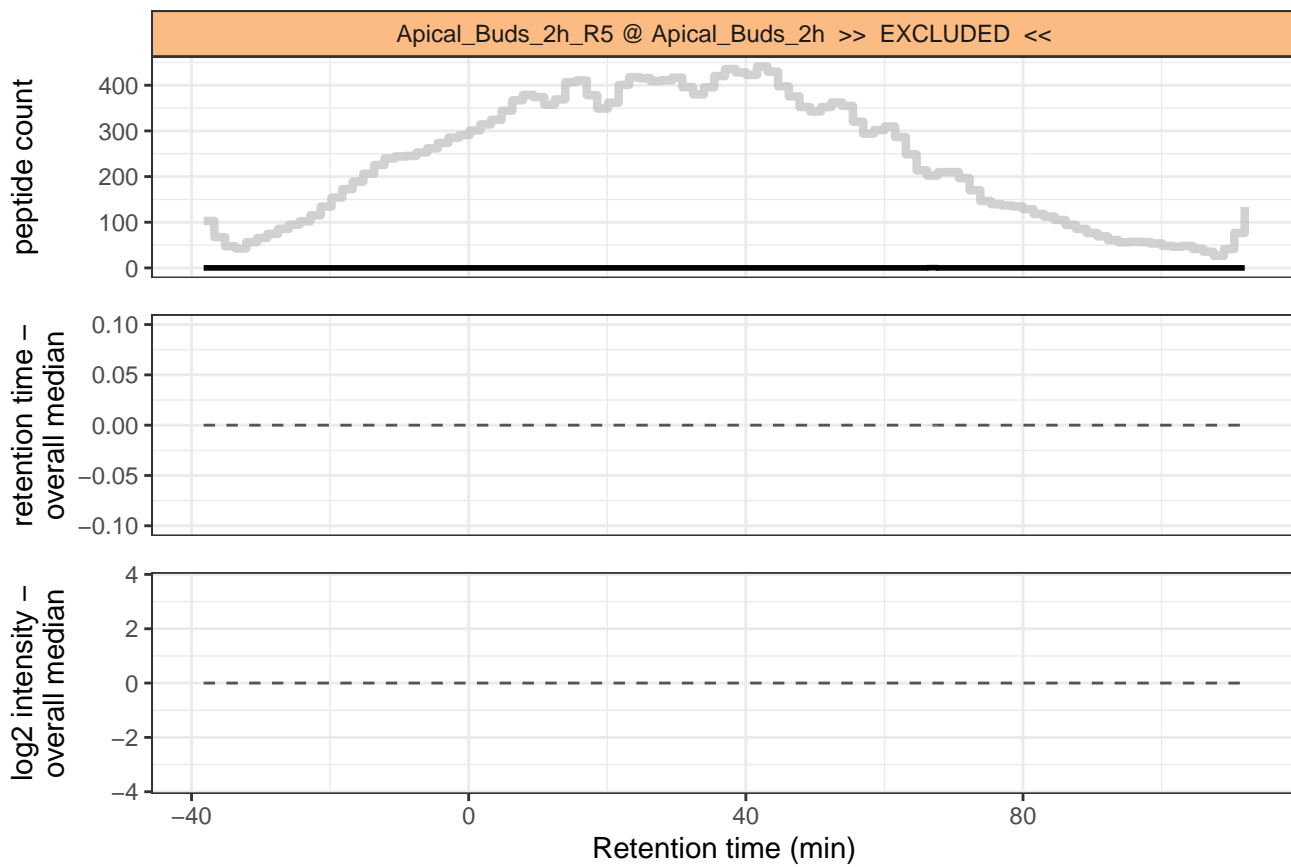
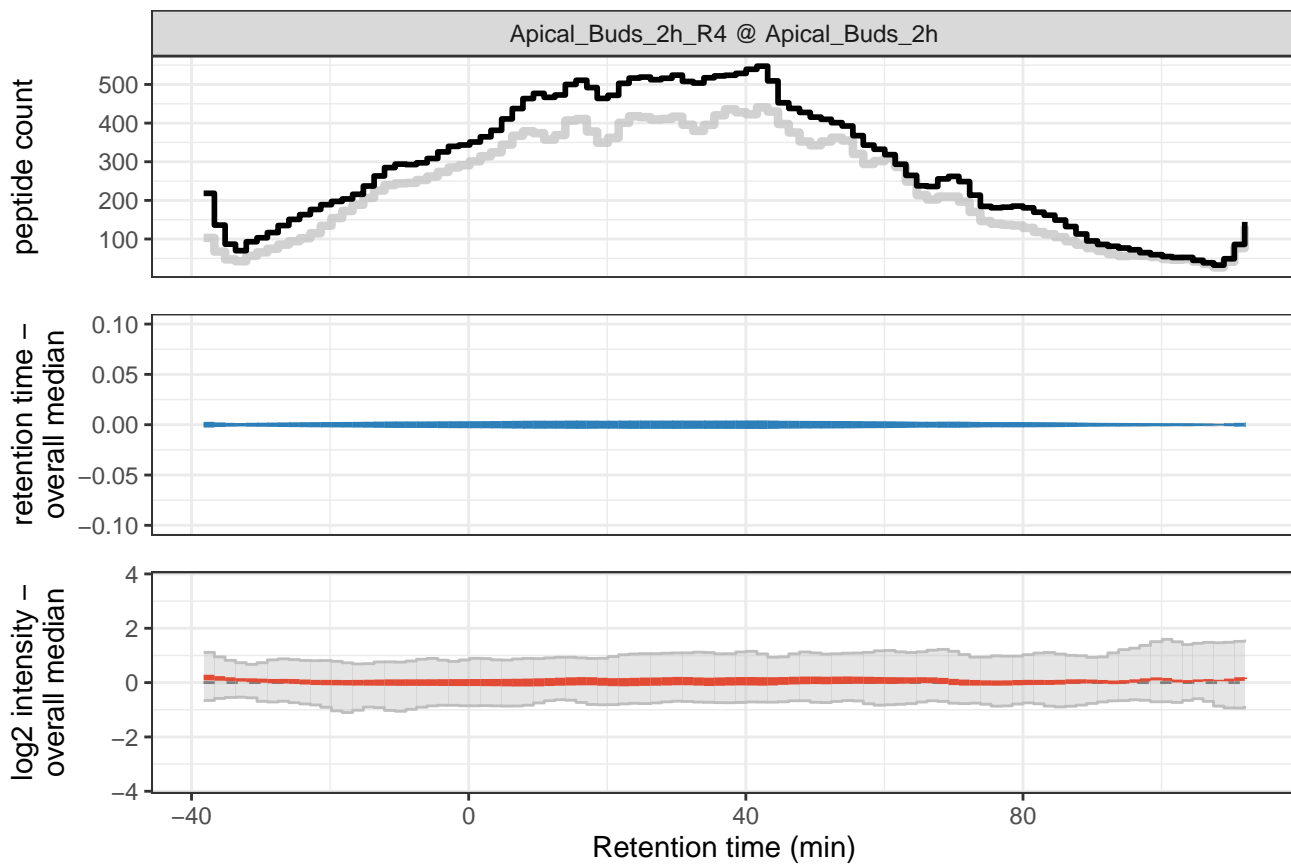
To investigate how each measurement differs from others, we visualize each sample as a 3 panel figure. First, the data is binned across the retention time dimension (x-axis). If a samples was marked as 'exclude' in the provided sample metadata, this is indicated in the plot title.

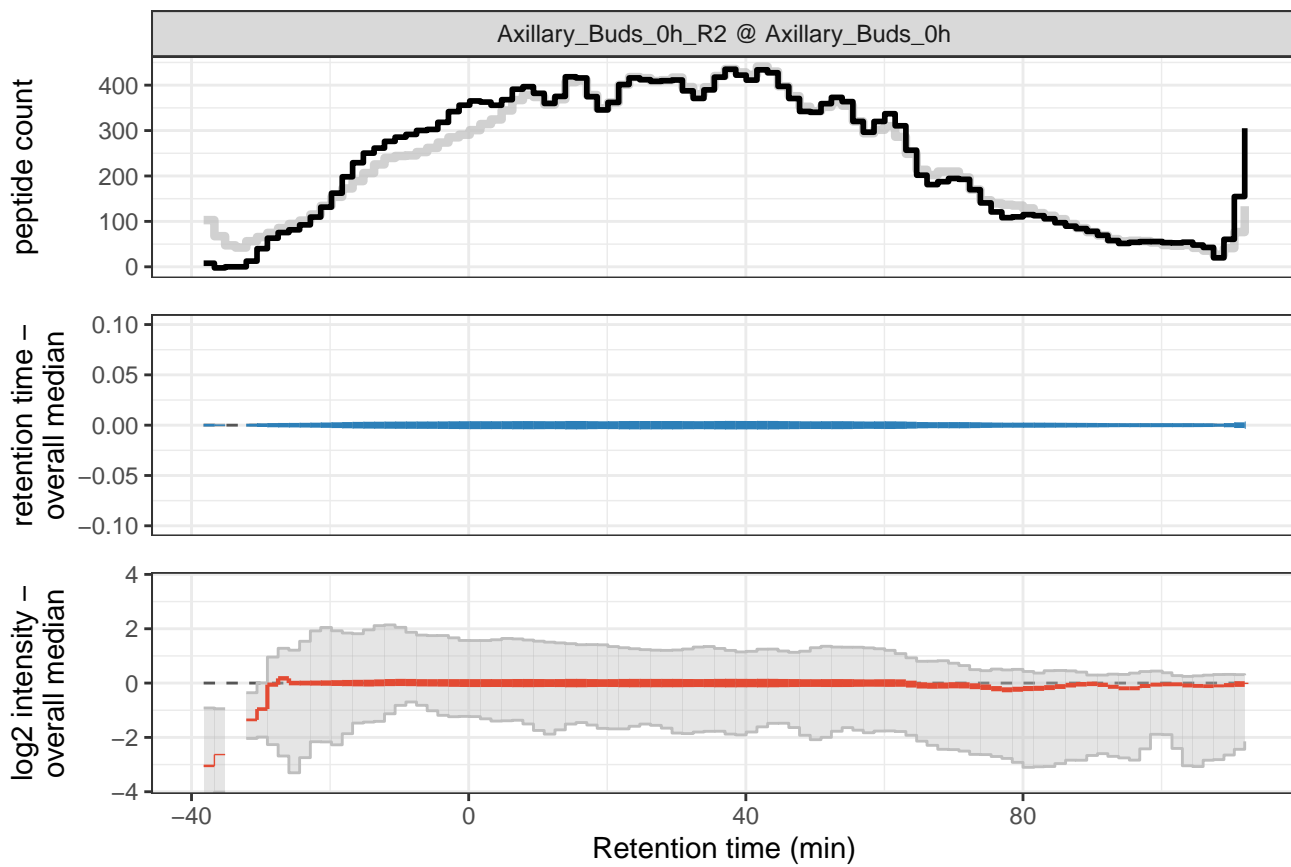
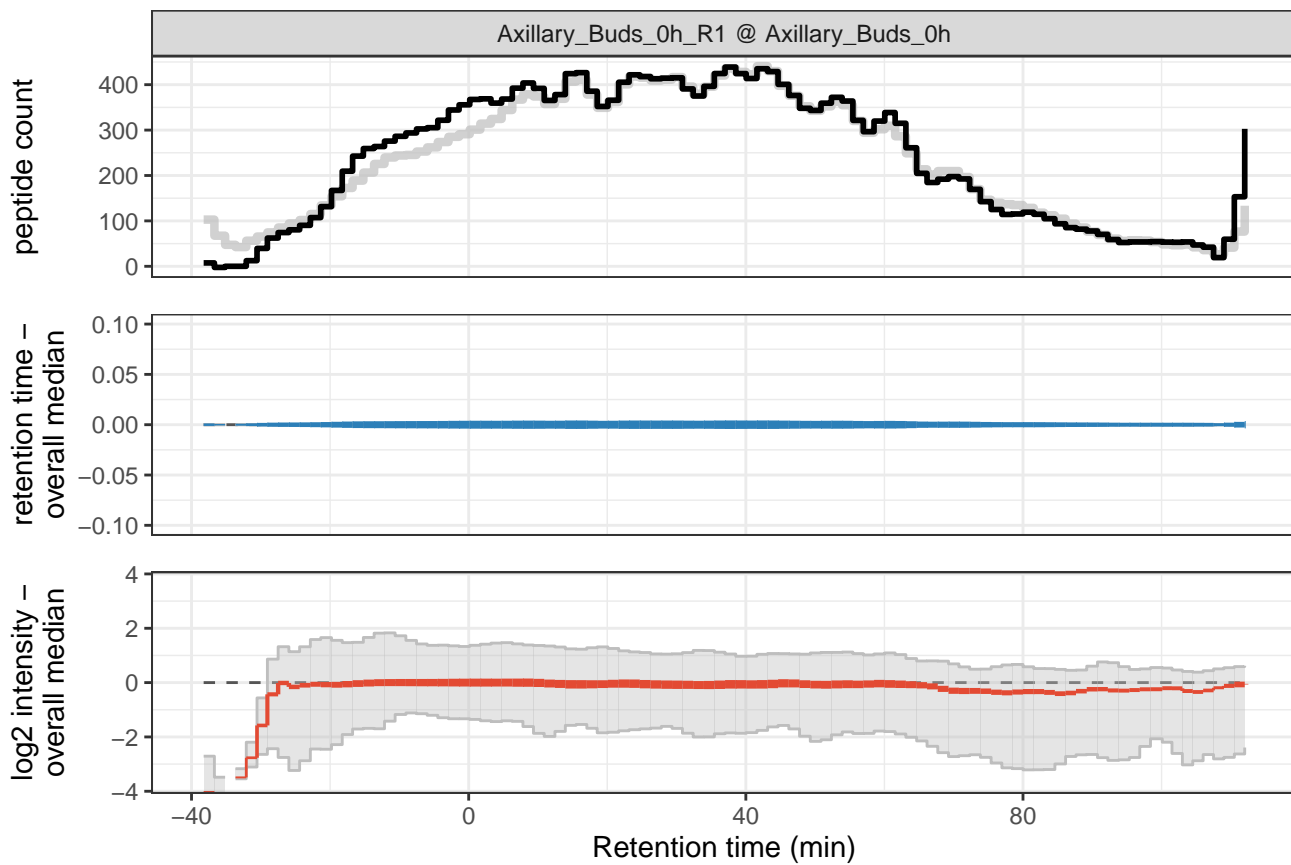
The top panel shows the number of peptides in the input data, e.g. as recognized by the software that generated input for this pipeline, over time (black line). For reference, the grey line shows the median amount over all samples (note; if this is the exact same in all samples, the grey line may not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

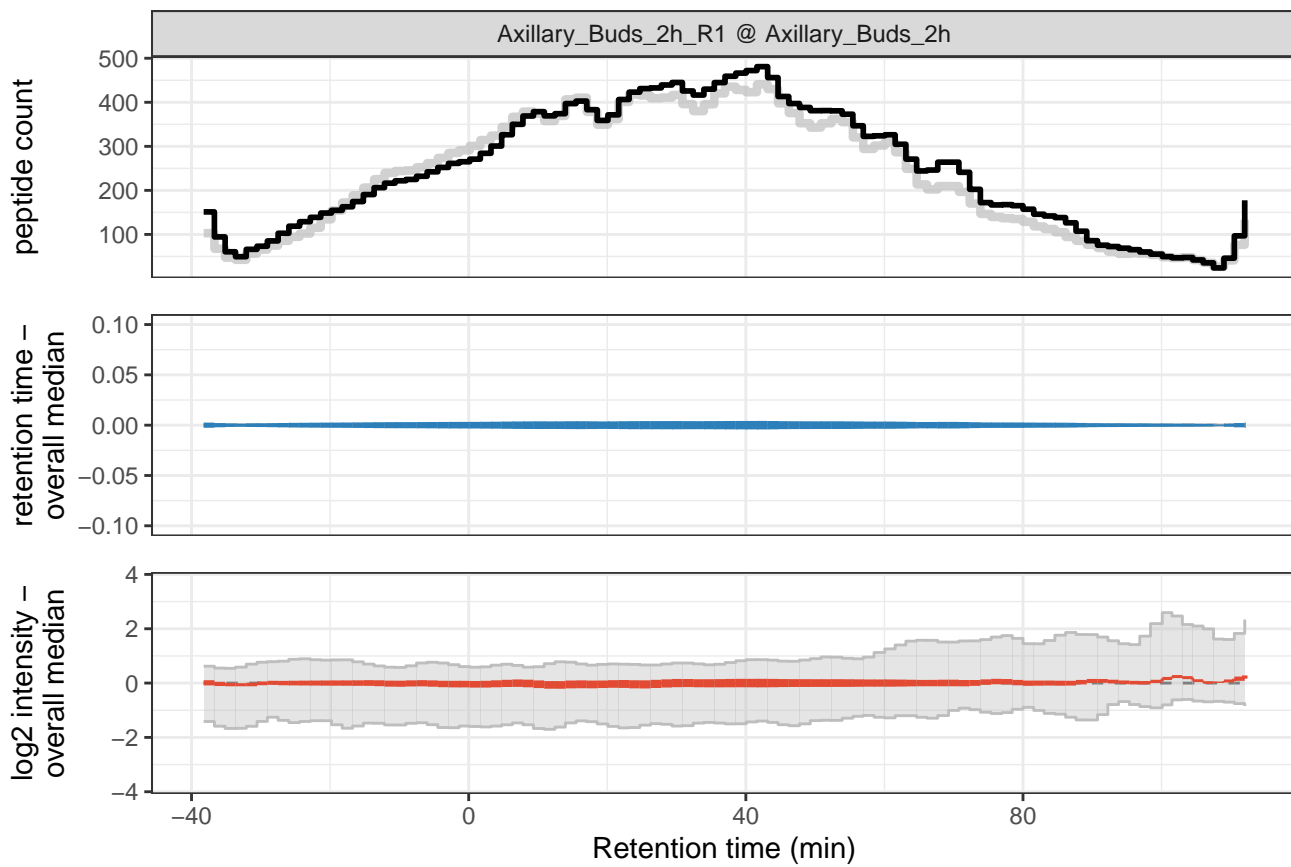
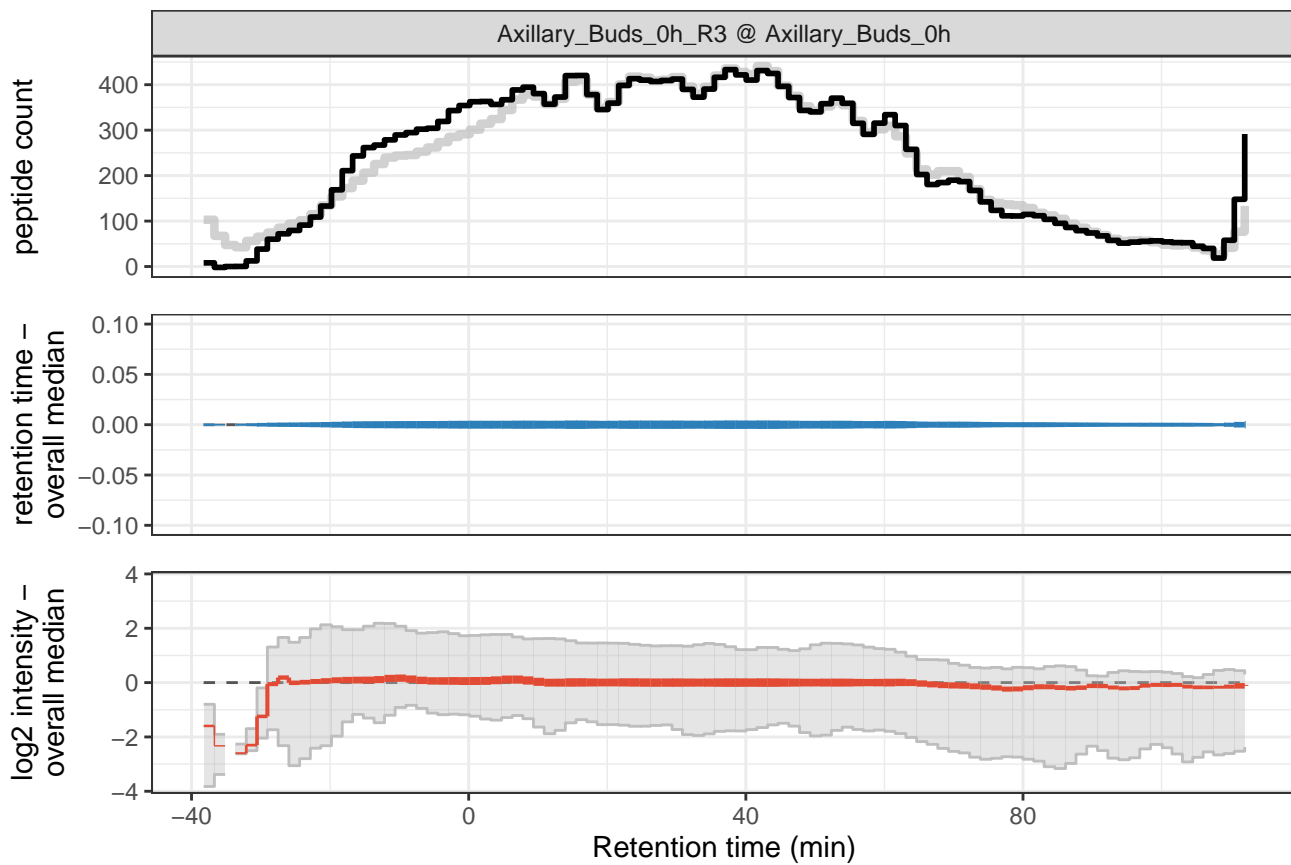
The middle panel indicates whether peptide retention times deviate from their median over all samples (blue line). The grey area depicts the 5% and 95% quantiles, respectively. The line width corresponds to the number of peptides eluting at that time (data from first panel). Analogously, the bottom panel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

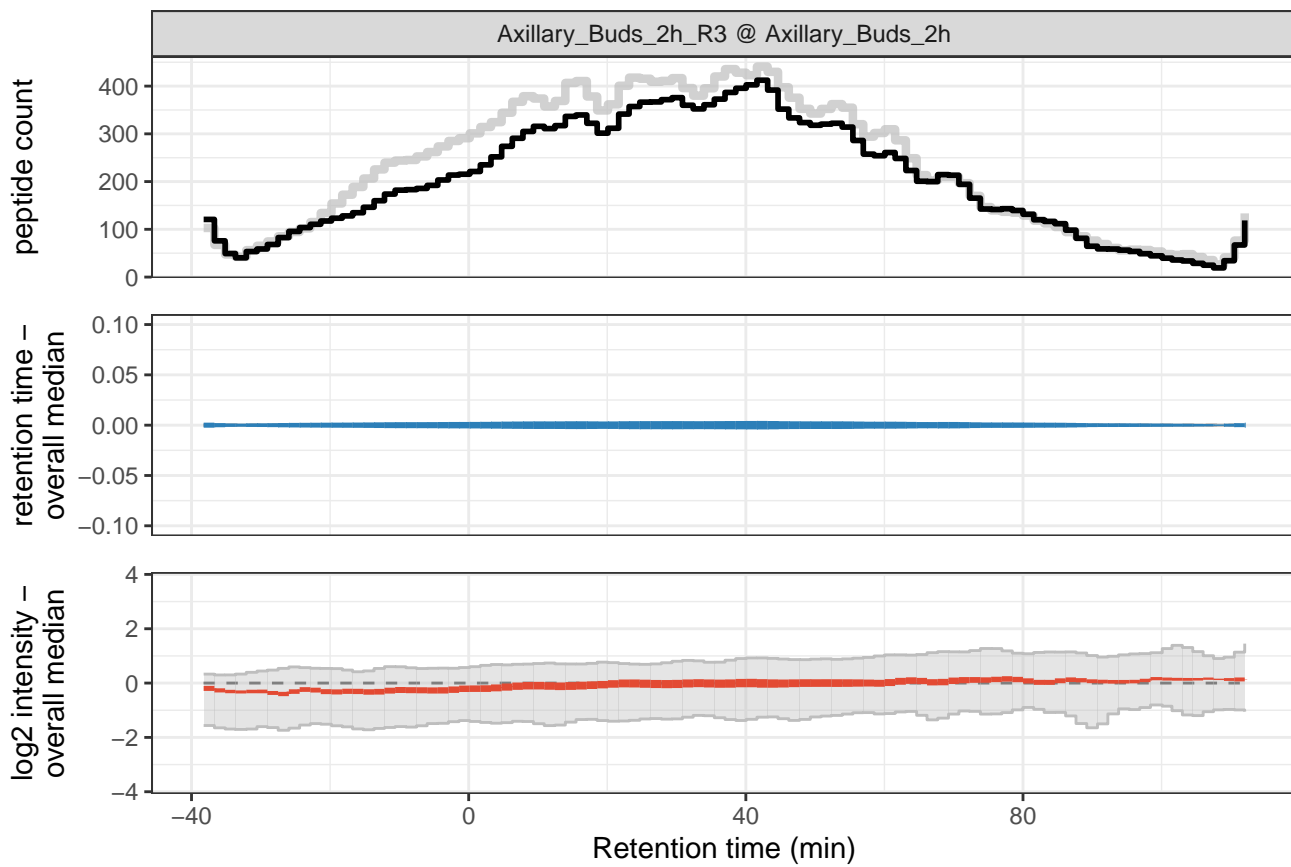
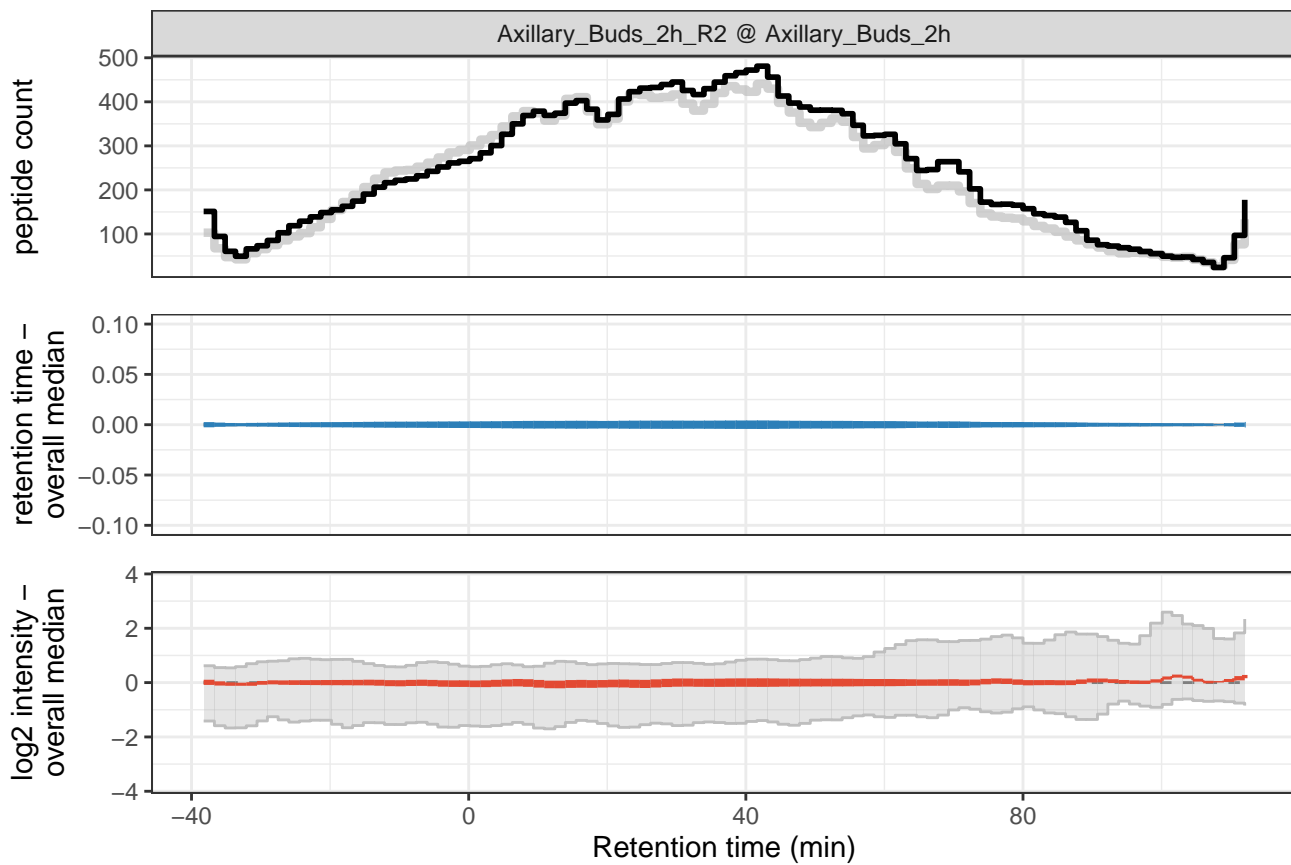












1.5 variation among replicates

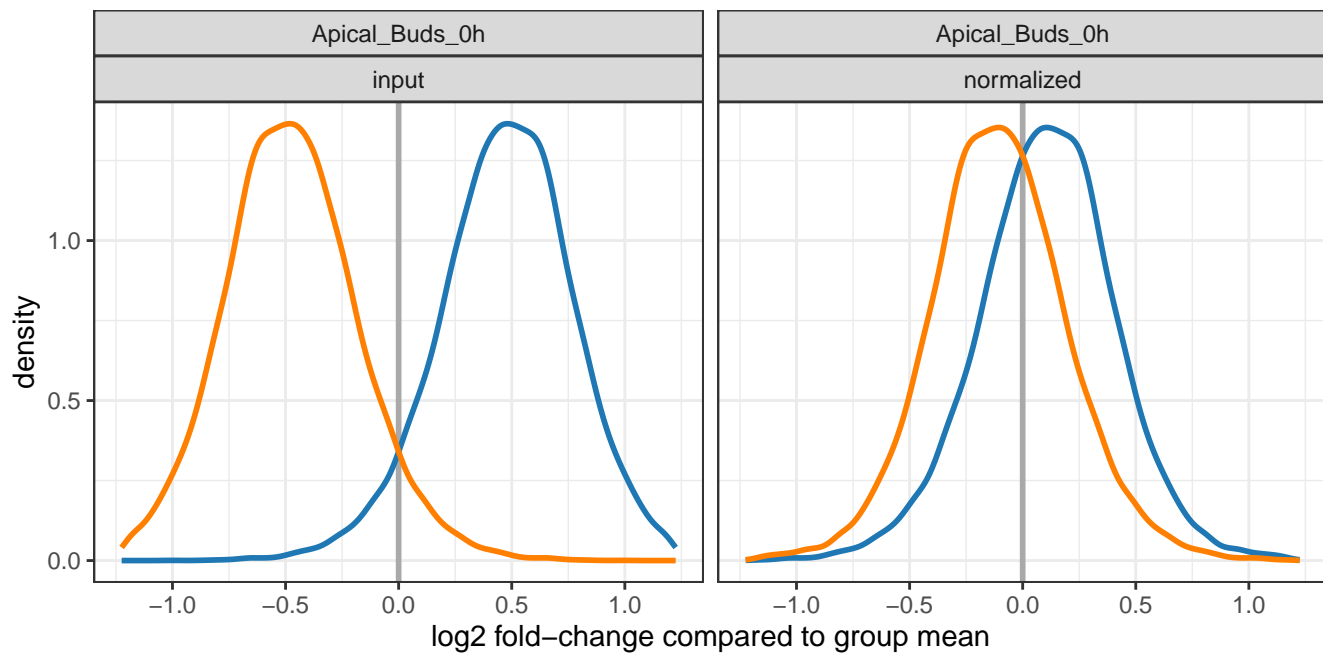
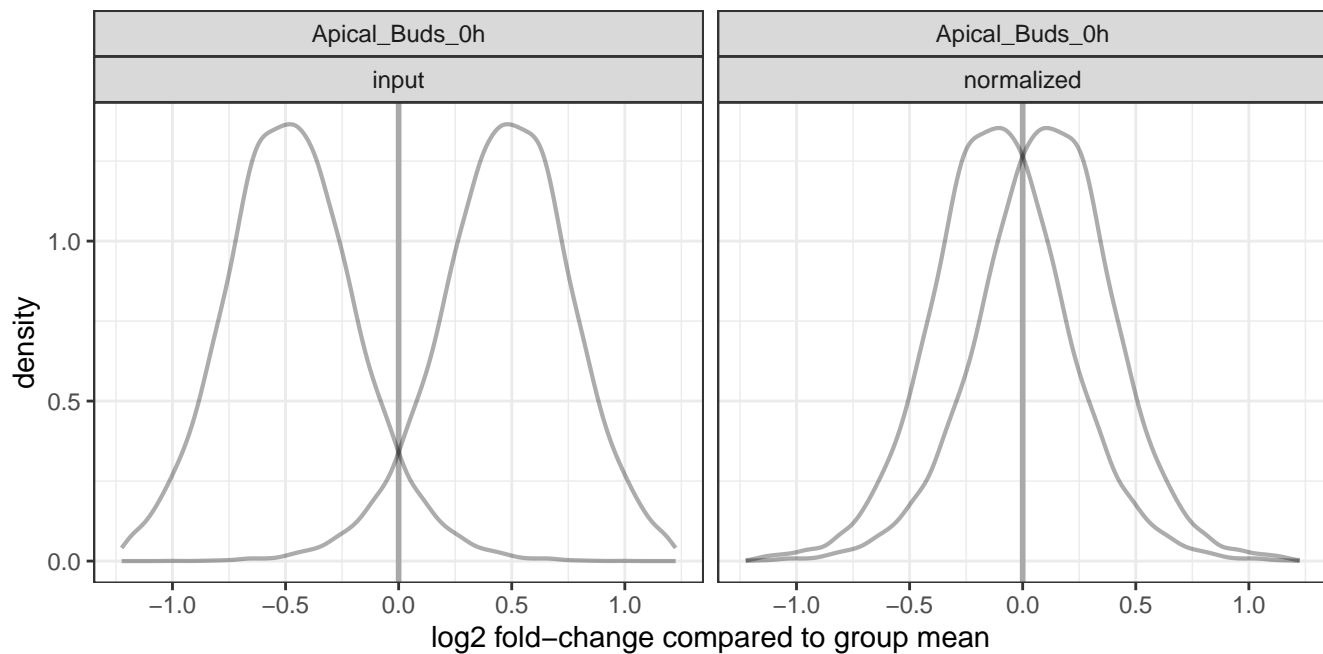
The reproducibility of replicate measurements is expressed in three different analyses. First, the difference between peptide intensities in each sample are compared to the mean value among all replicates (foldchange distributions). Next, the Coefficient of Variation (CoV) is used as a metric for reproducibility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

1.5.1 within-group foldchange distributions

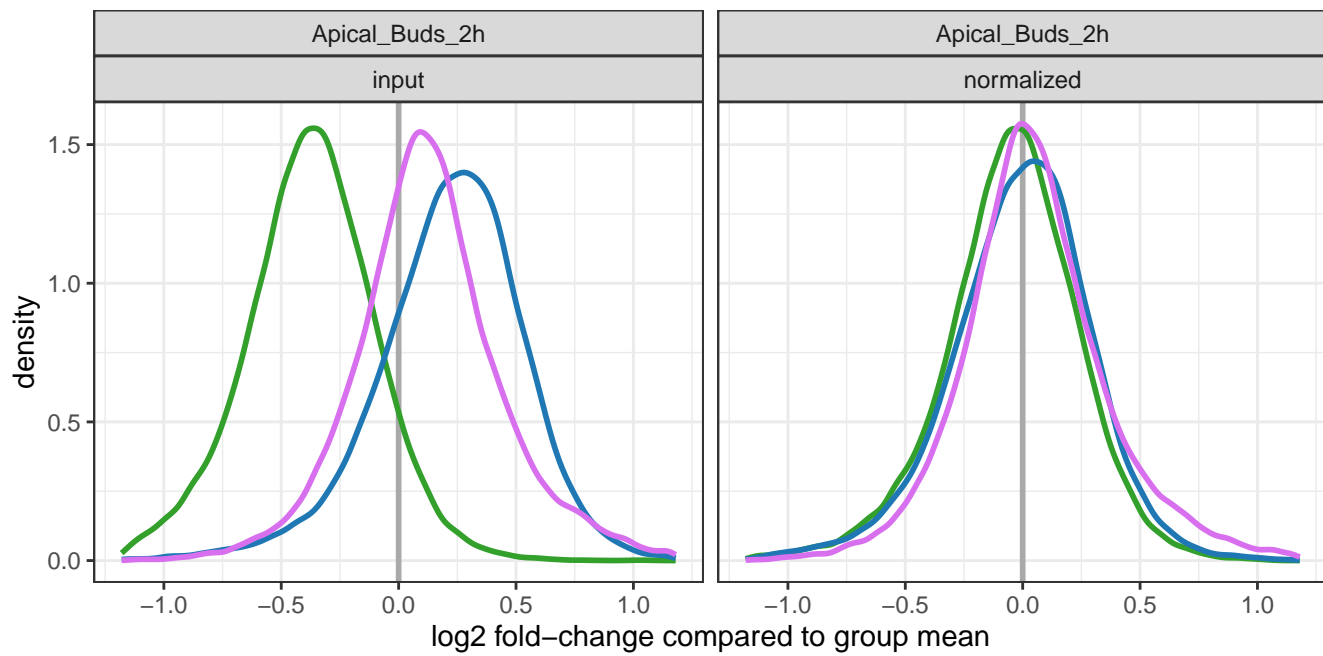
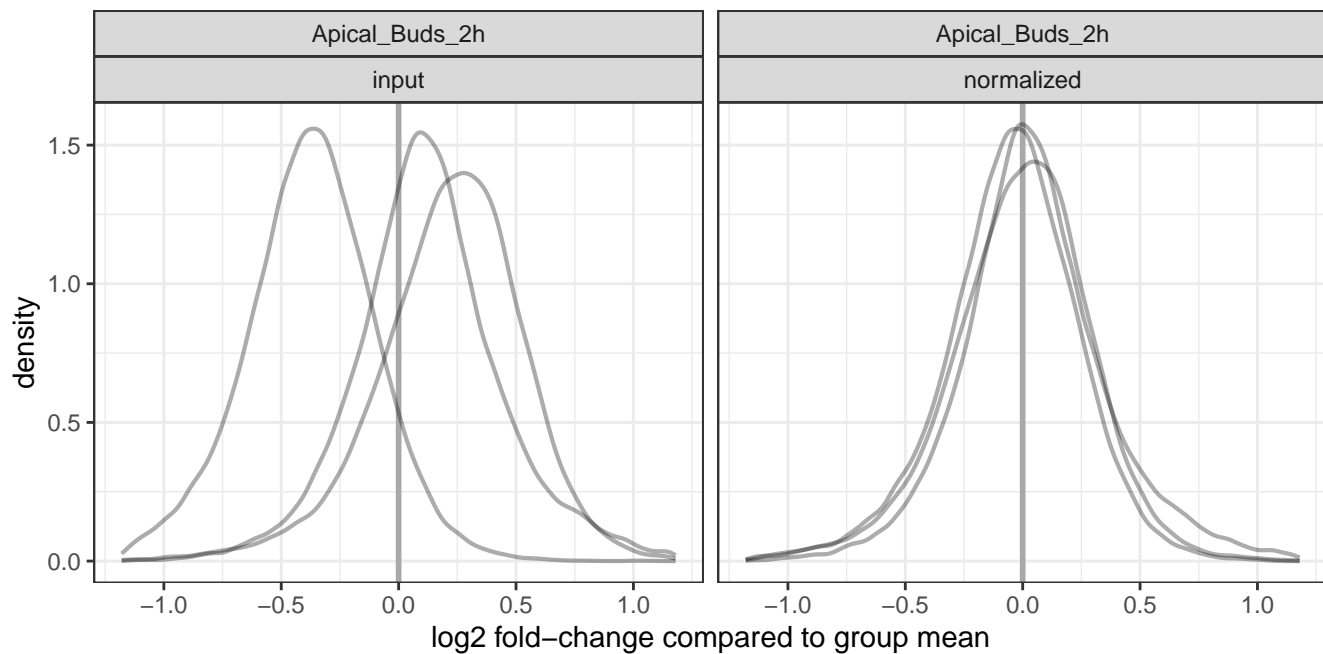
The foldchange of all peptides in a sample is compared to their respective mean value over all samples in the group. This visualizes how strongly each sample deviates from other samples in the same group which helps identify outlier samples. The same data was used as detailed in the “retention time”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

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.

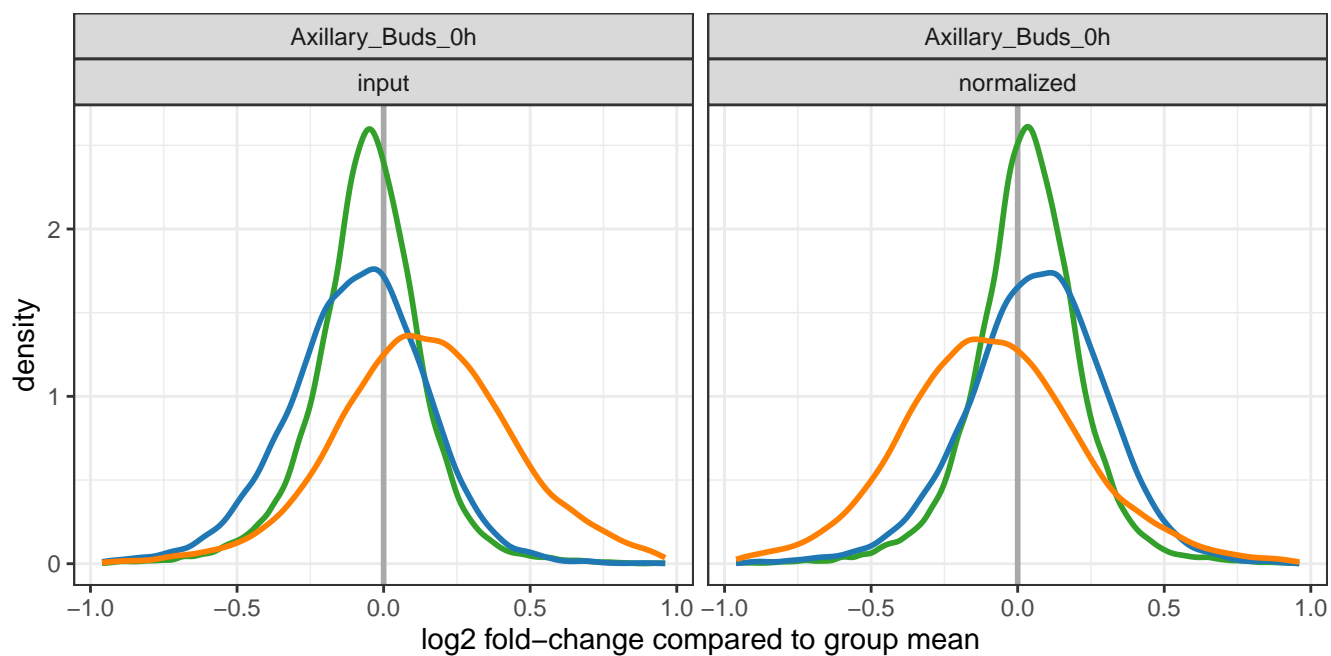
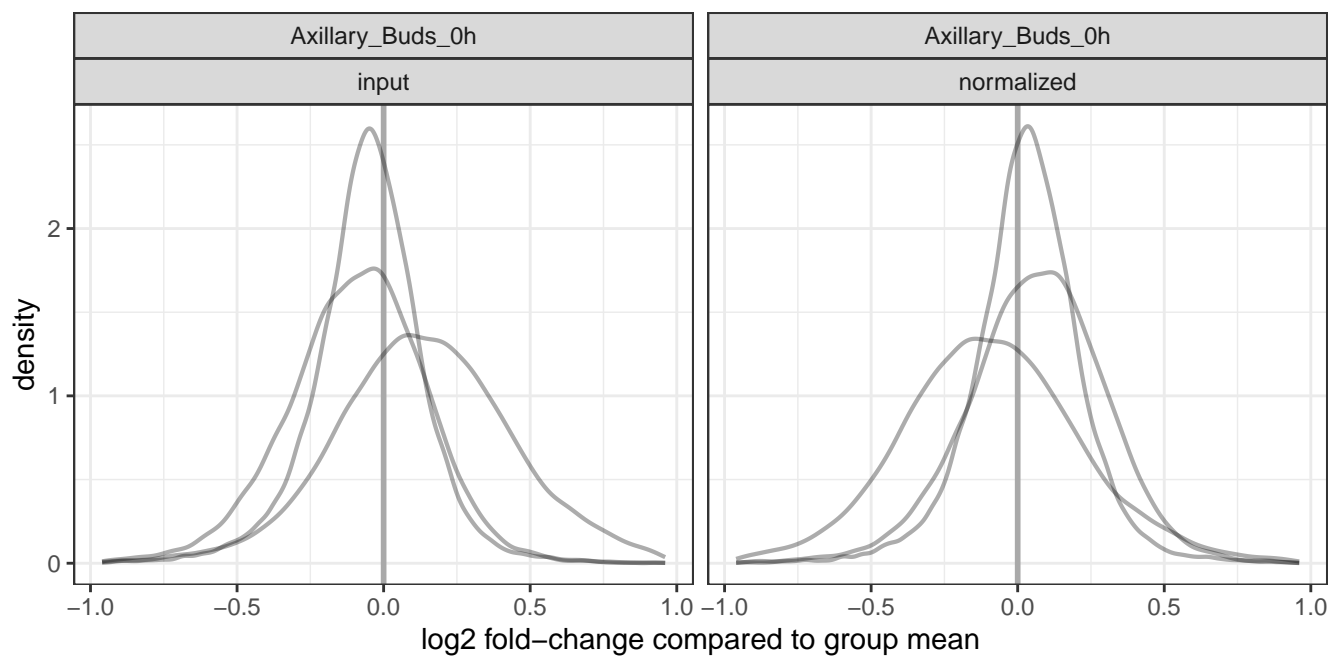
The ‘input’ panel is based on the peptide intensities as-is (i.e. the user-provided input data from upstream software), the ‘normalized’ panel shows the exact same samples and peptides after normalization (as specified by user). Samples marked as ‘exclude’ in the provided sample metadata table are visualized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

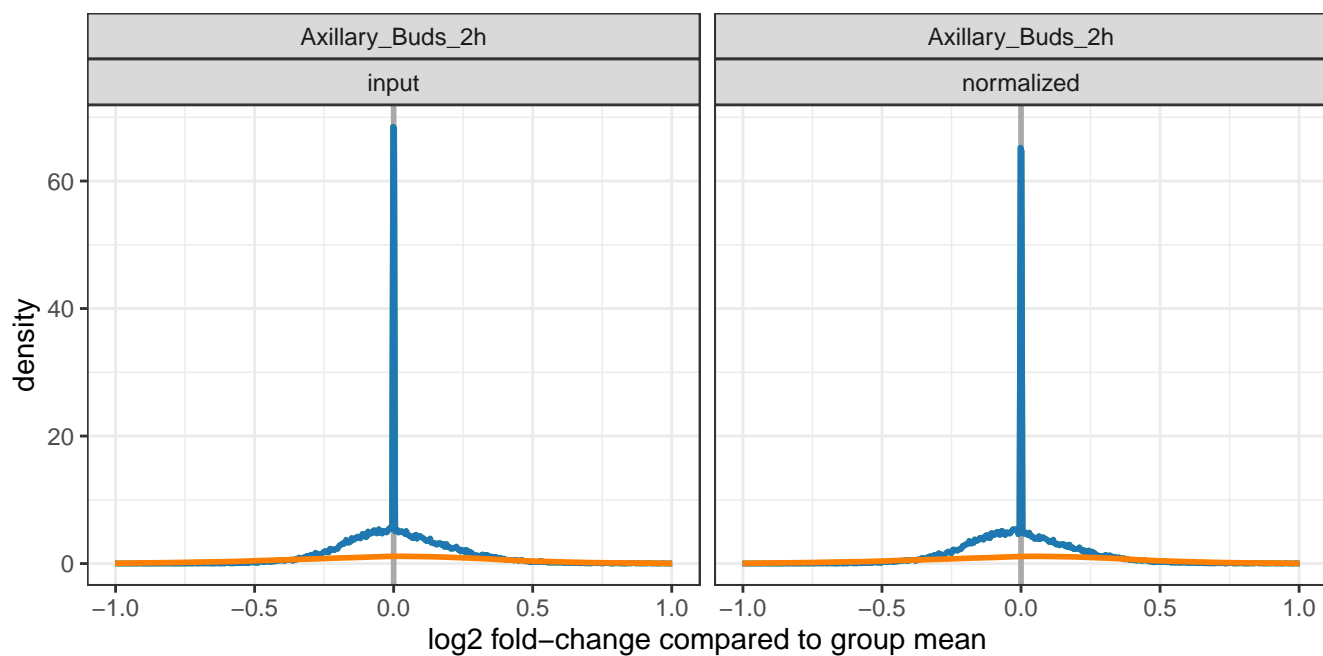
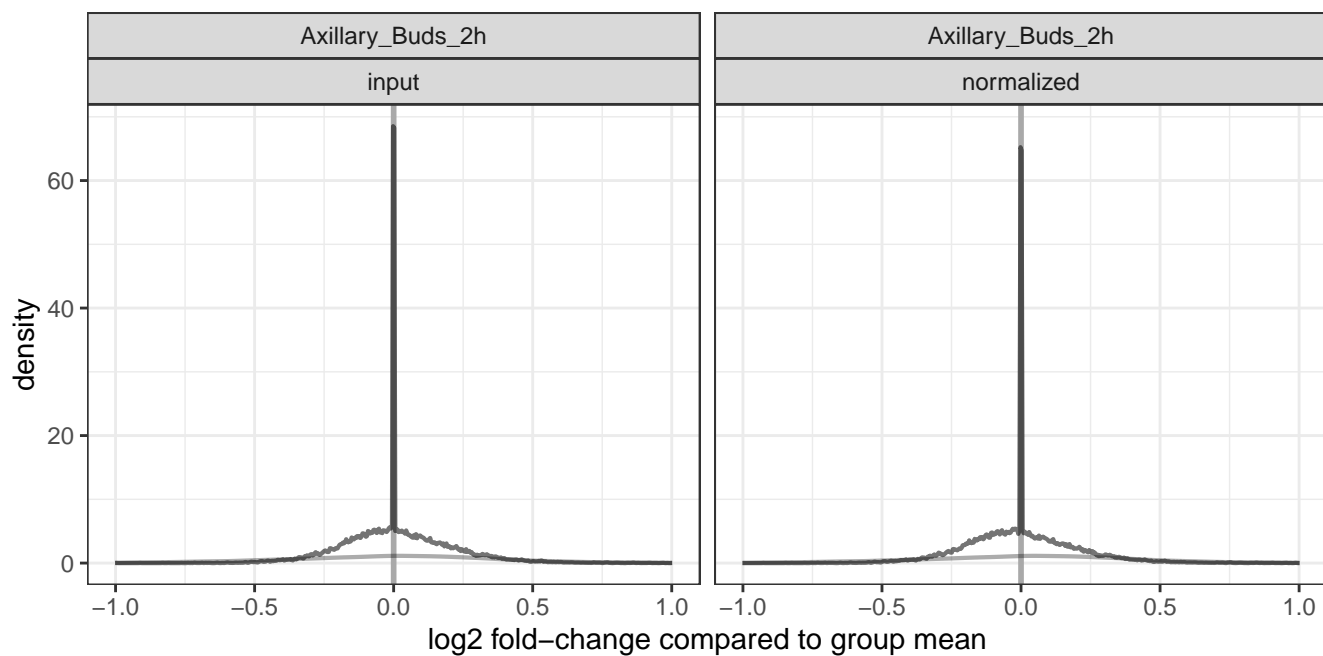


— Apical_Buds_0h_R3 — Apical_Buds_0h_R2



Apical_Buds_2h_R5 Apical_Buds_2h_R1
Apical_Buds_2h_R4 Apical_Buds_2h_R2





— Axillary_Buds_2h_R3 — Axillary_Buds_2h_R1
— Axillary_Buds_2h_R2

top10 outliers per sample group

Group	Sample	Standard Deviation
Apical_Buds_0h	Apical_Buds_0h_R2	0.290
Apical_Buds_0h	Apical_Buds_0h_R3	0.290

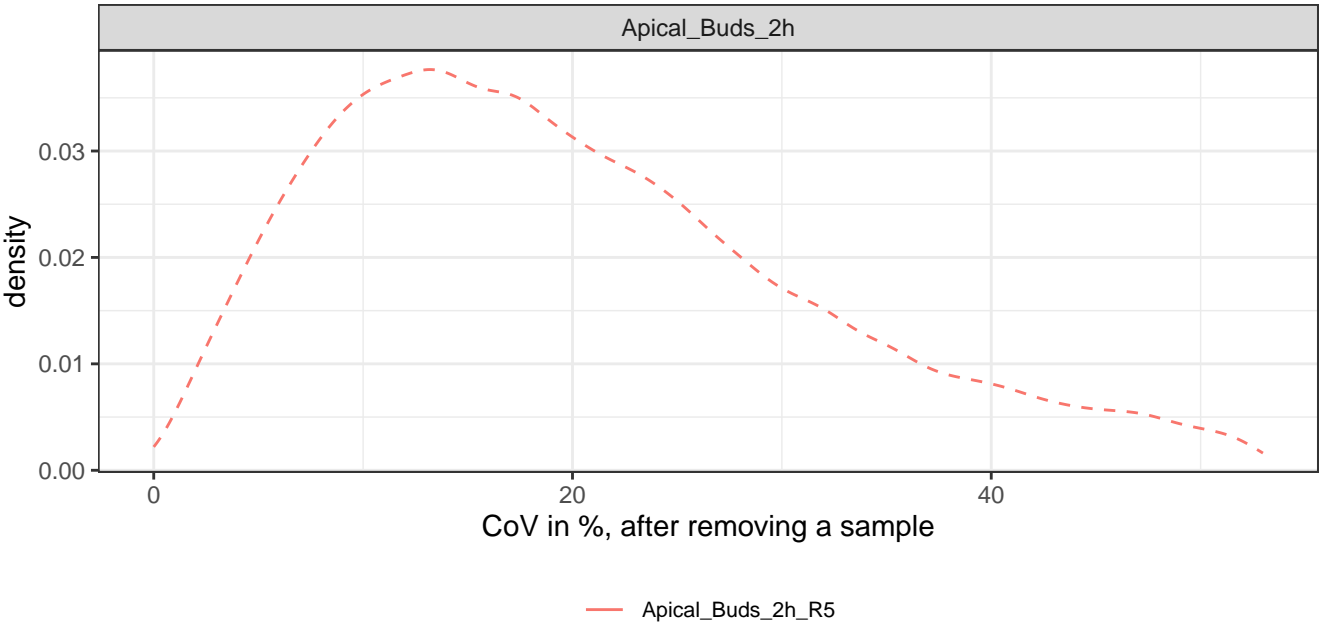
Group	Sample	Standard Deviation
Apical_Buds_2h	Apical_Buds_2h_R4	0.312
Apical_Buds_2h	Apical_Buds_2h_R1	0.290
Apical_Buds_2h	Apical_Buds_2h_R2	0.278
Apical_Buds_2h	Apical_Buds_2h_R5	NA

Group	Sample	Standard Deviation
Axillary_Buds_0h	Axillary_Buds_0h_R1	0.308
Axillary_Buds_0h	Axillary_Buds_0h_R3	0.229
Axillary_Buds_0h	Axillary_Buds_0h_R2	0.178

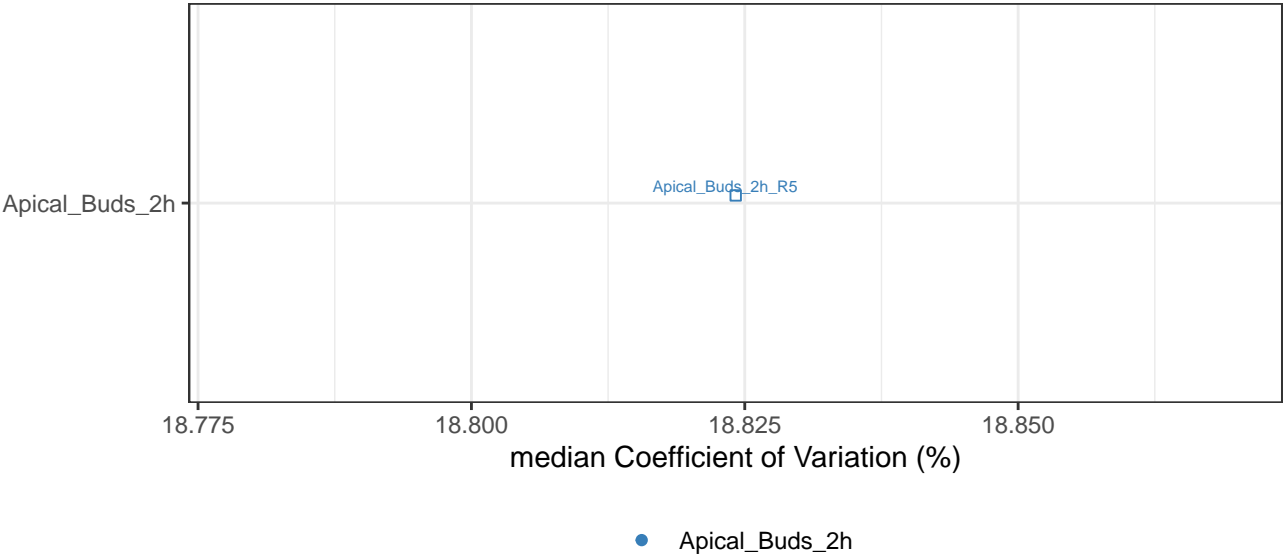
Group	Sample	Standard Deviation
Axillary_Buds_2h	Axillary_Buds_2h_R3	0.386
Axillary_Buds_2h	Axillary_Buds_2h_R1	0.174
Axillary_Buds_2h	Axillary_Buds_2h_R2	0.174

1.5.2 CoV, leave-one-out

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



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



shortname	group	exclude	median CoV
Apical_Buds_2h_R5	Apical_Buds_2h	TRUE	18.8

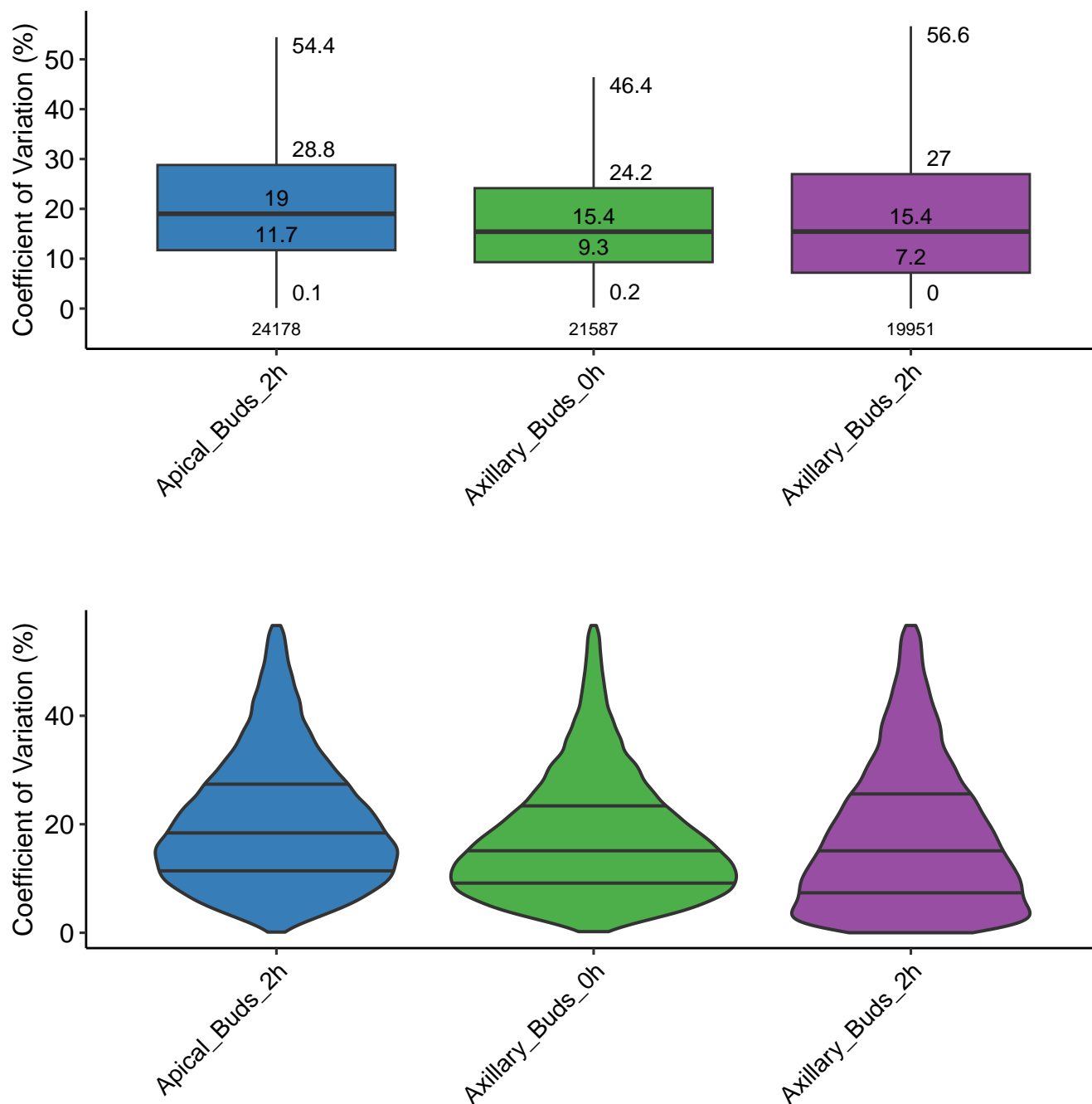
Leave-one-out impact on within-group CoV (%)

1.5.3 Coefficient of Variation

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

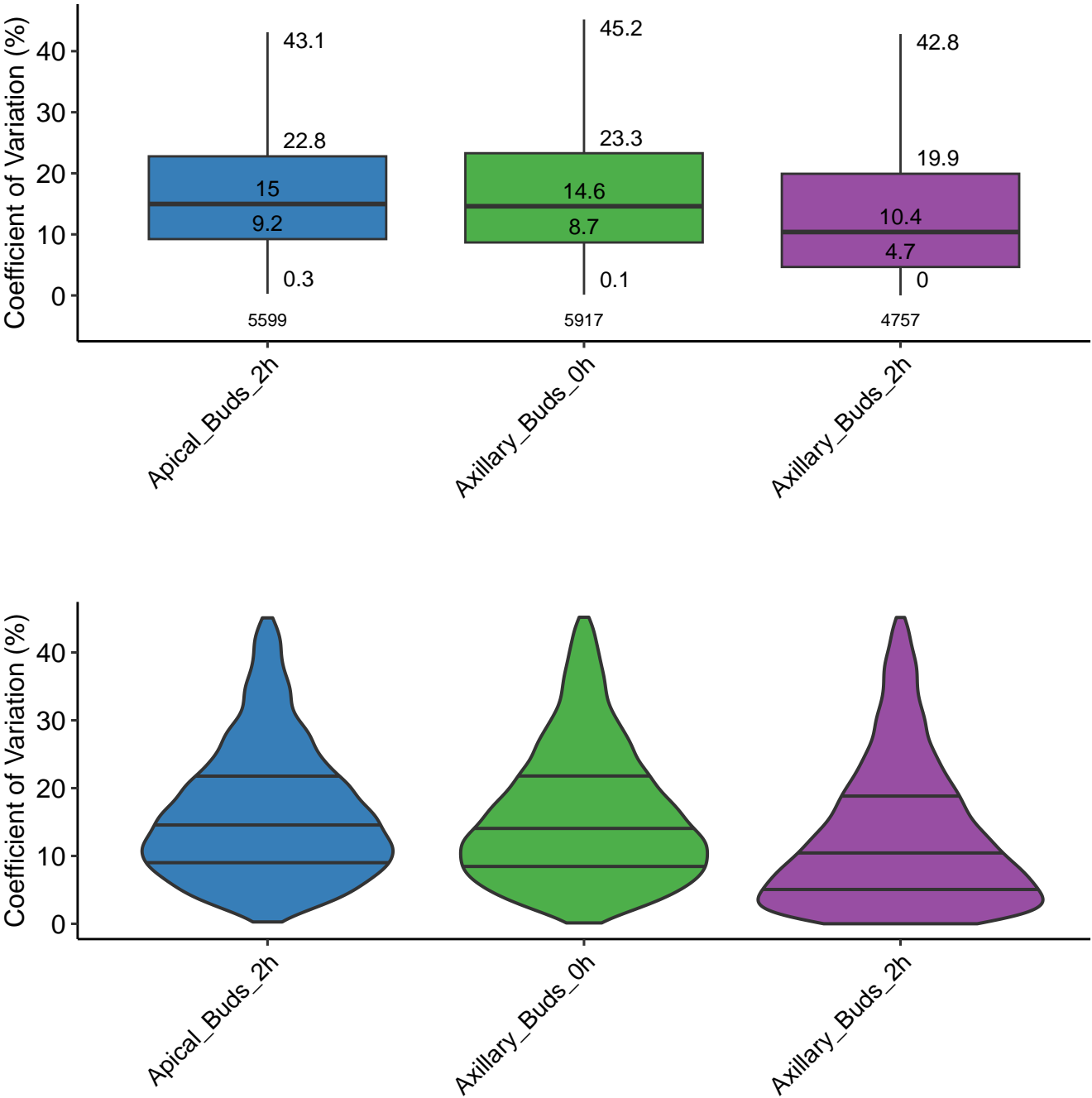
Only samples that are NOT marked 'exclude' in the provided sample metadata and are in a sample group among at least 3 replicates are used for these figures. The user-specified filtering rules (eg; filter_min_detect, filter_min_peptide_per_prot, etc.) were applied within each sample group independently and remaining peptides were subsequently normalized using the "vsn&modebetween_protein" algorithm (an important parameter to keep in mind when comparing across datasets/analyses). Only peptides with at least 3 data points across replicate samples are used for each CoV computation.

Peptide-level CoV:



Protein-level CoV:

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



1.6 PCA

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

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

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

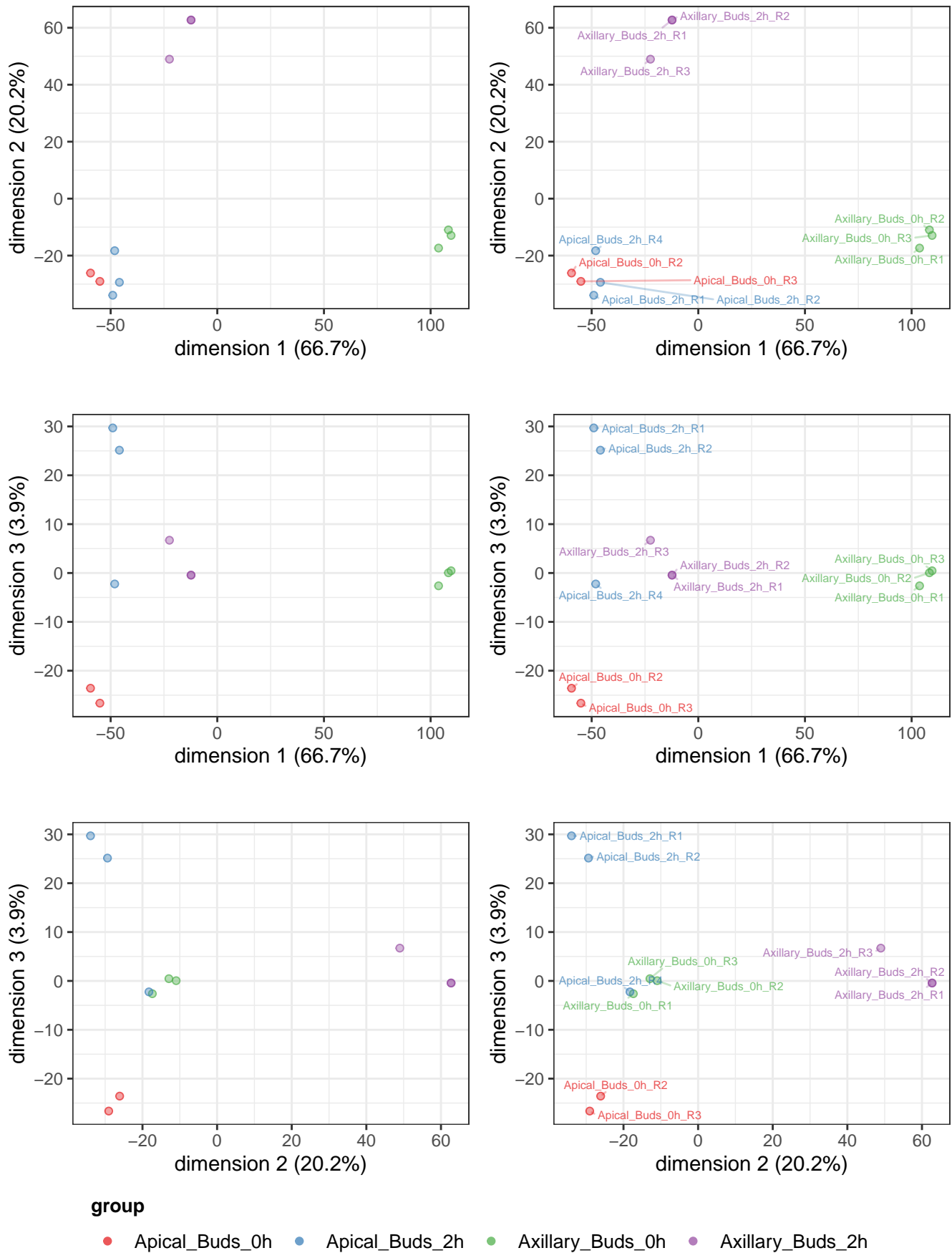
Rationale behind data filter

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

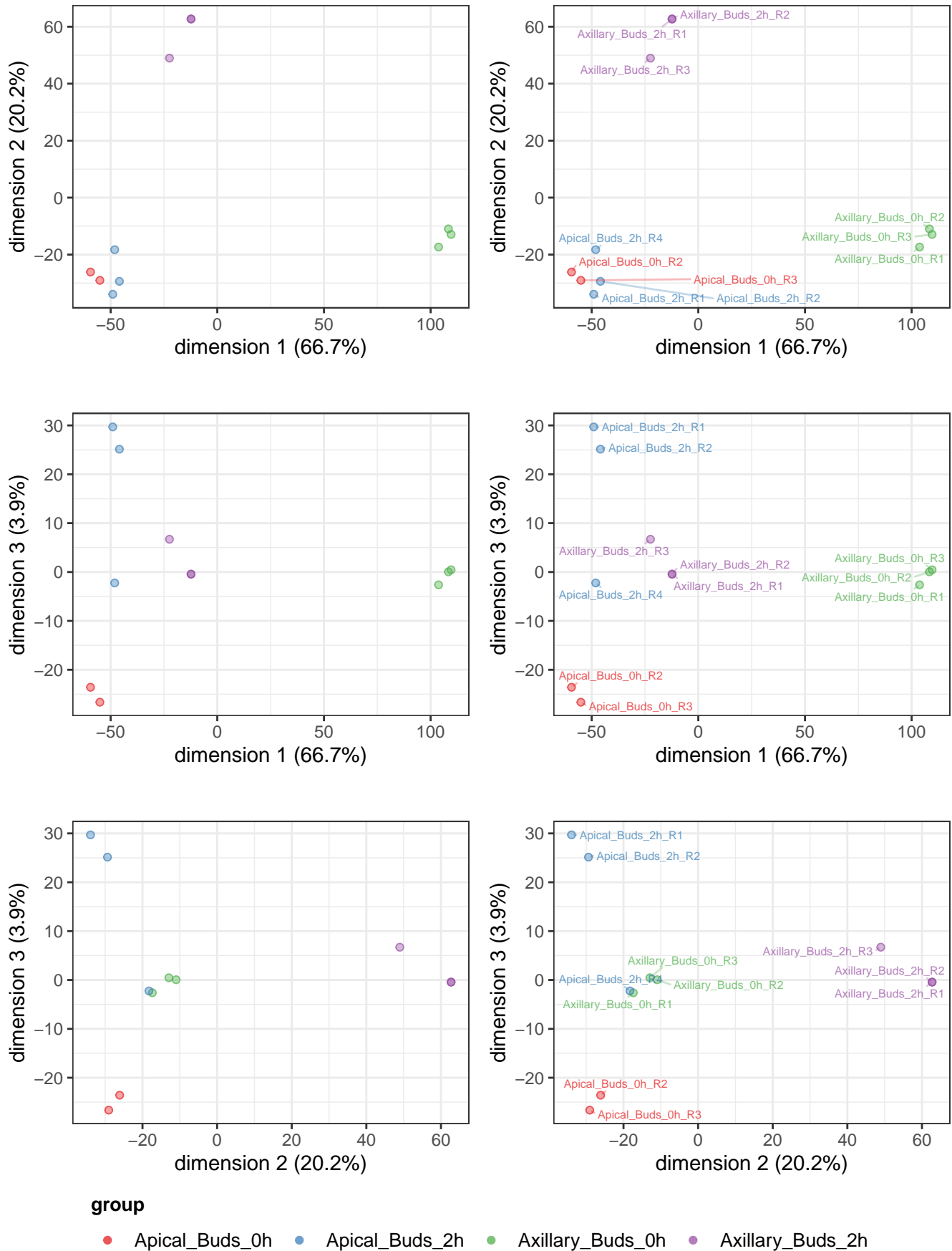
Figure legends

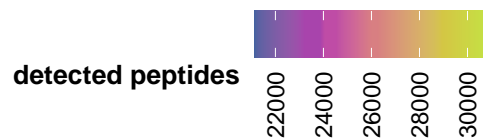
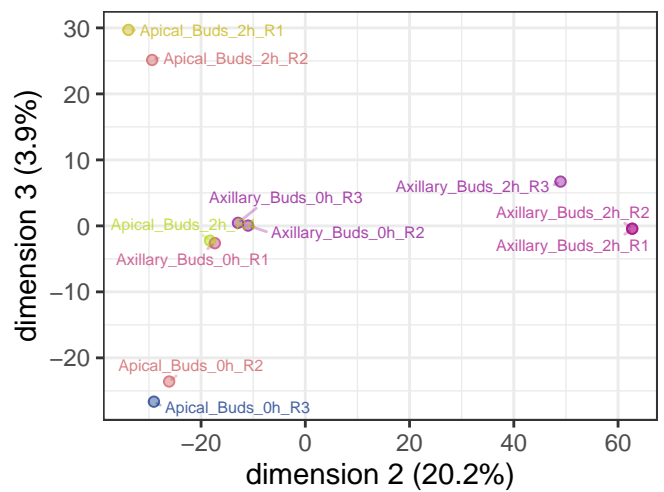
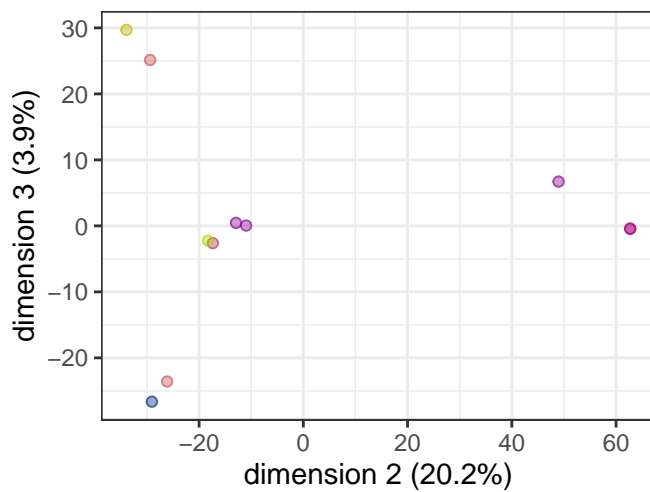
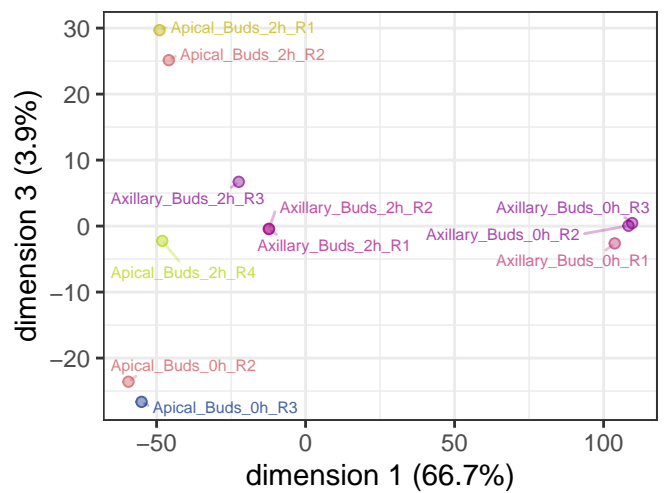
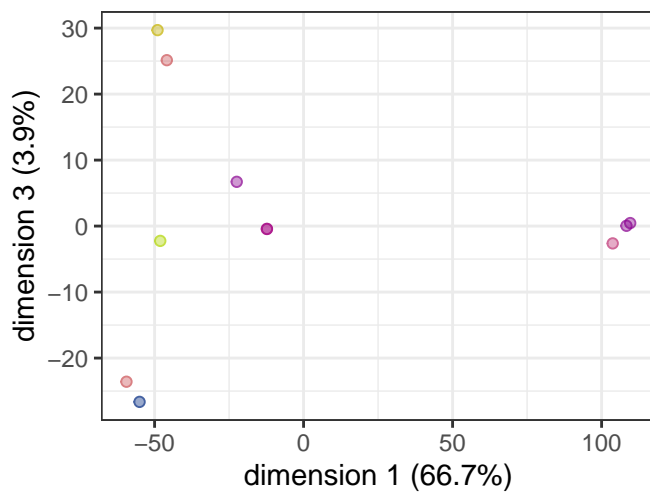
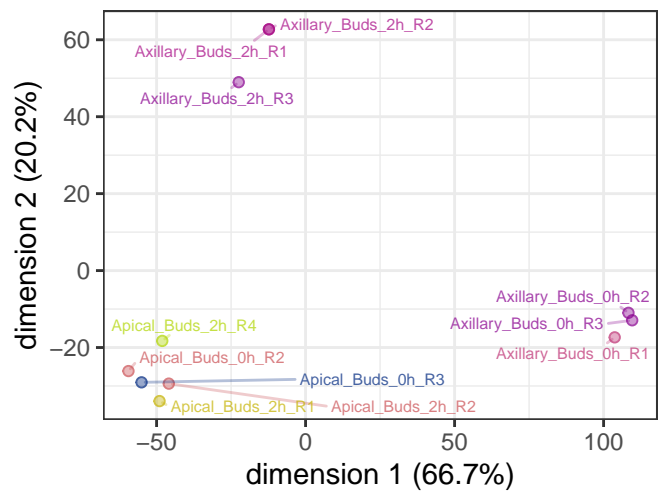
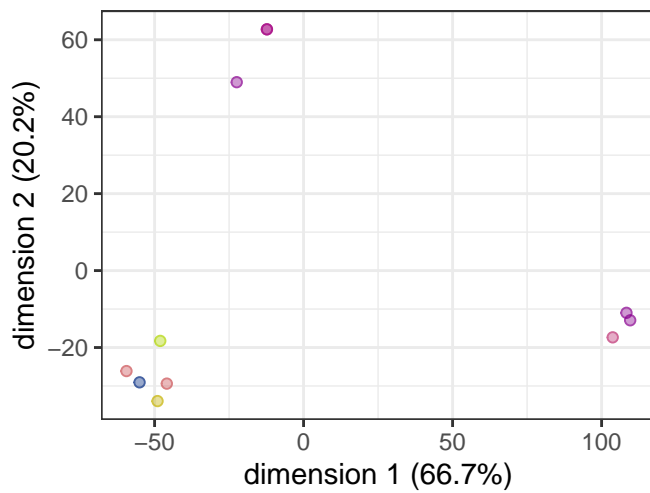
The first 3 principle components compared visually (1 *vs* 2, 1 *vs* 3, 2 *vs* 3) on the rows. Left- and right-side panels on each row represent the same figure without and with sample labels. The principle components are shown on the axis labels together with their respective percentage of variance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

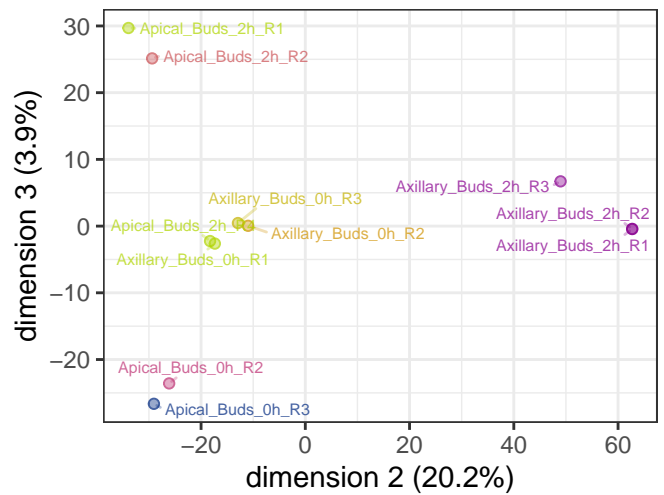
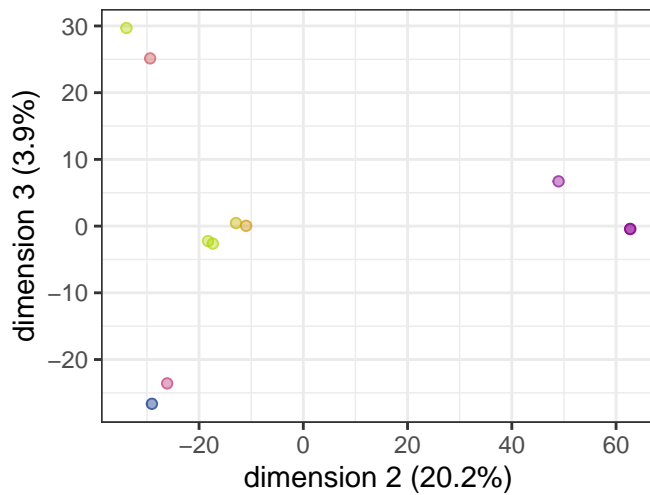
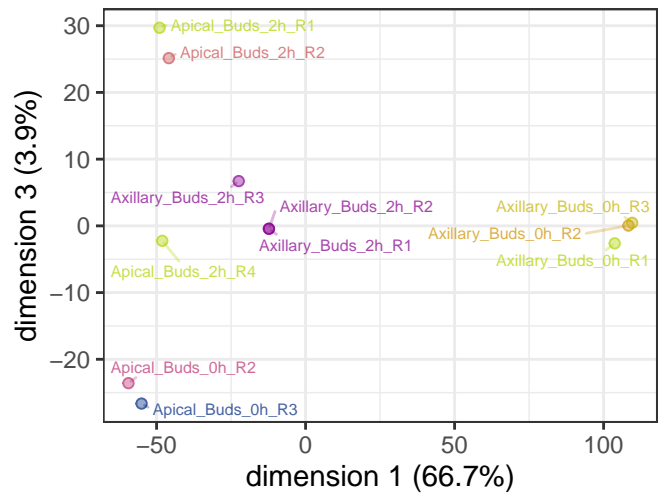
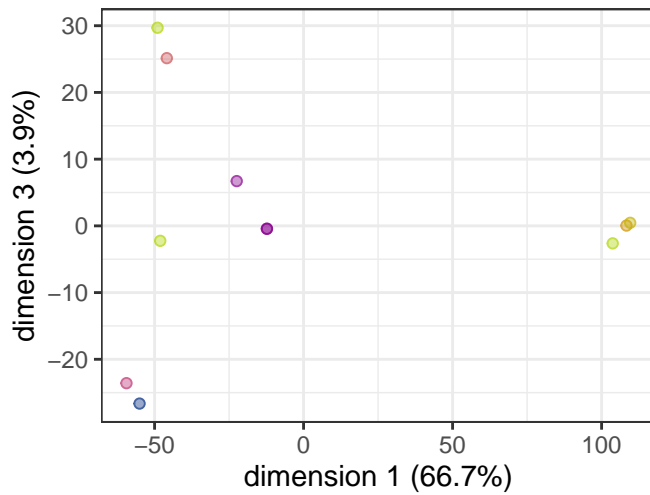
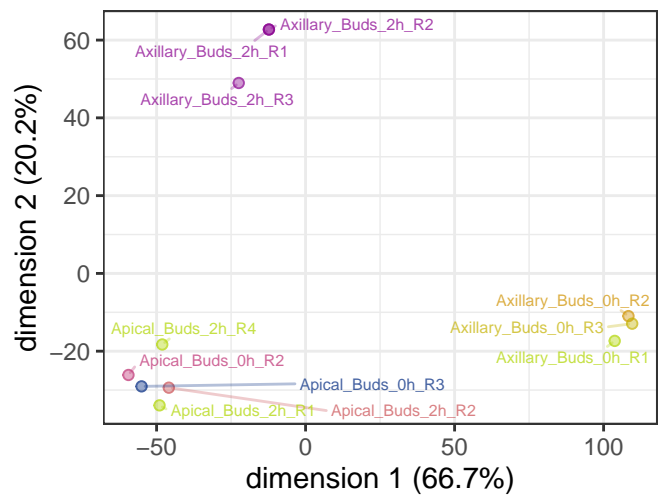
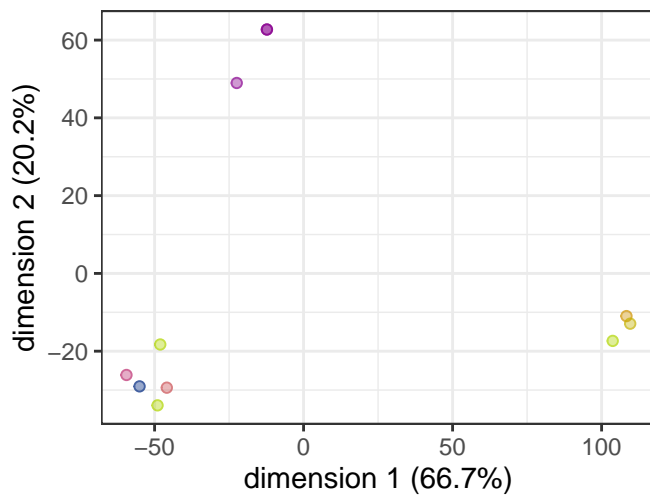
PCA of all samples, including those flagged as 'exclude', using 8487 peptides



PCA only on samples not flagged as 'exclude', using 8487 peptides







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 subsequently the resulting data matrix is used for all downstream statistical analyses. In the example above where peptide p is present in a subset of sample groups, p would either be left out (not present in majority of samples in entire dataset) or erroneously used when applying t-statistics to groups B and C (since p is not present in group C, it may differentially detected but there are no features available for quantitative analysis)

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 Axillary_Buds_0h vs Axillary_Buds_2h

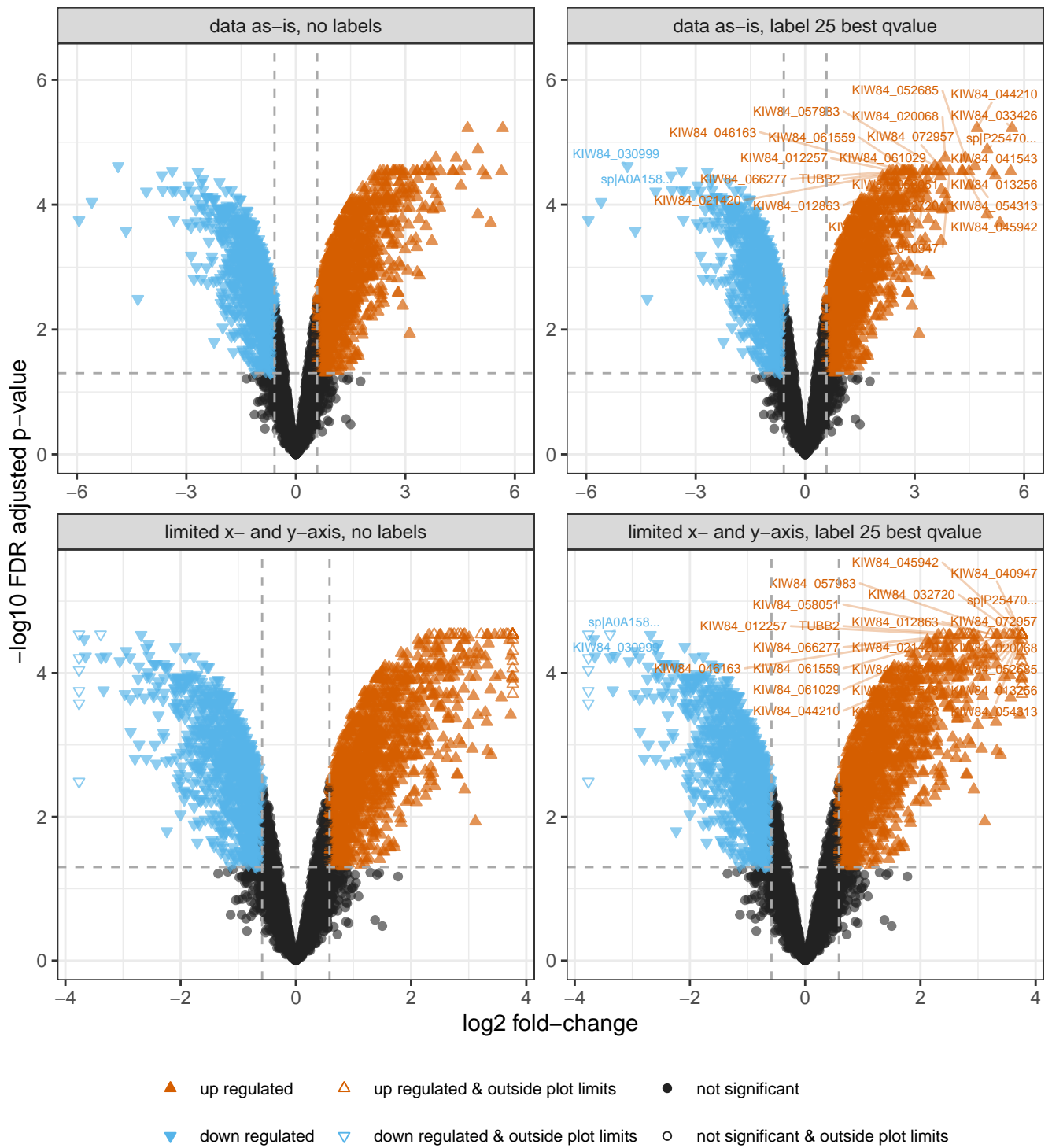
- **user setting:** using ‘filter by contrast’ peptide filtering approach
- 12927 peptides in 4060 proteins remain in the current contrast after peptide filters and are used for the statistical analysis in this section
- qvalue threshold: 0.05
- log2 foldchange threshold: 0.5849625

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.

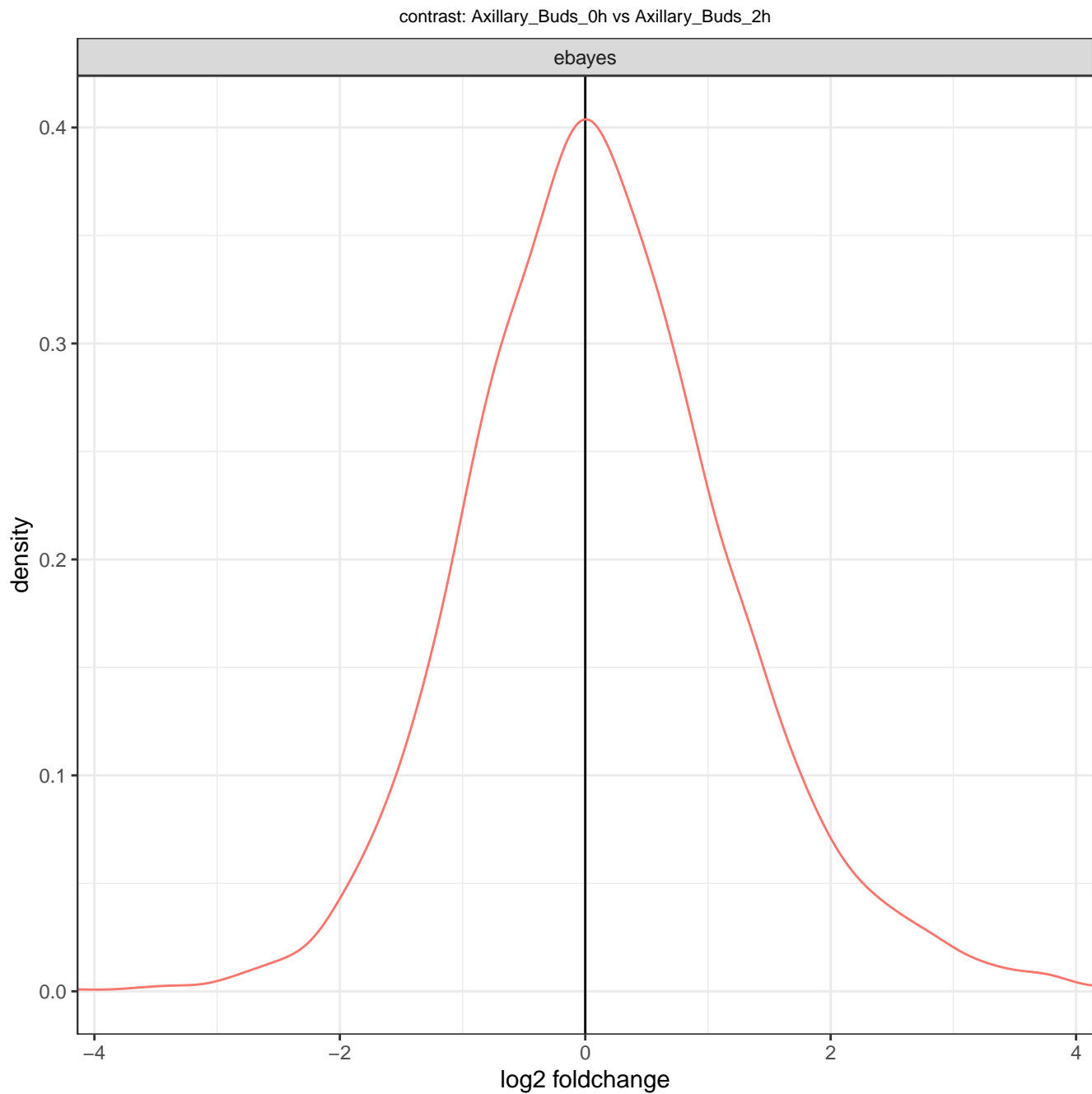
Bottom figure panels have limited x- and y-axis. For datasets with a small number of strong outliers in p-value or fold-change, which may have a profound effect on the plot scales, this allows inspection of the remainder of the volcano plot without disproportionate influence by ‘extreme’ values.

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 *).



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.

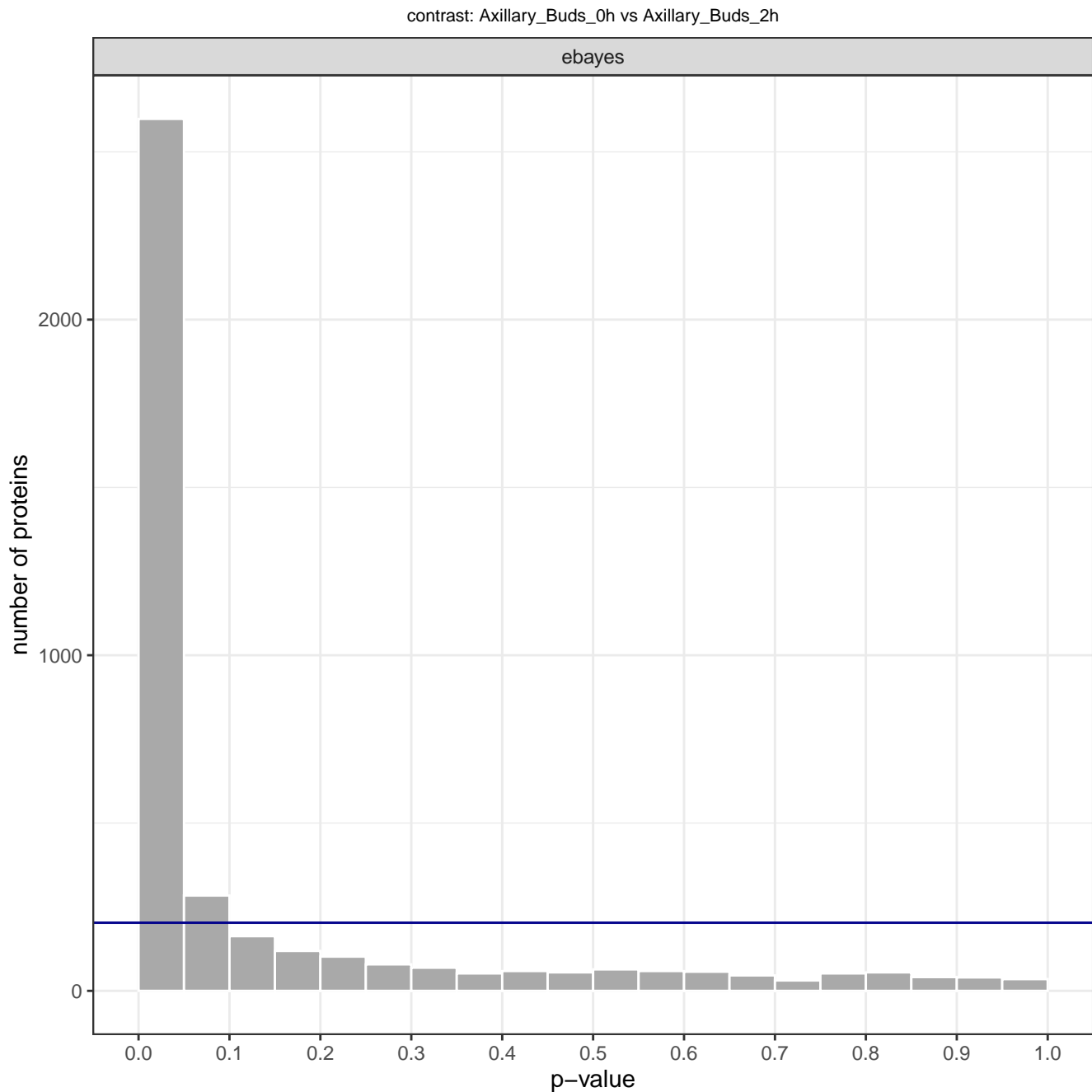


2.1.3 p-value distribution

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/>

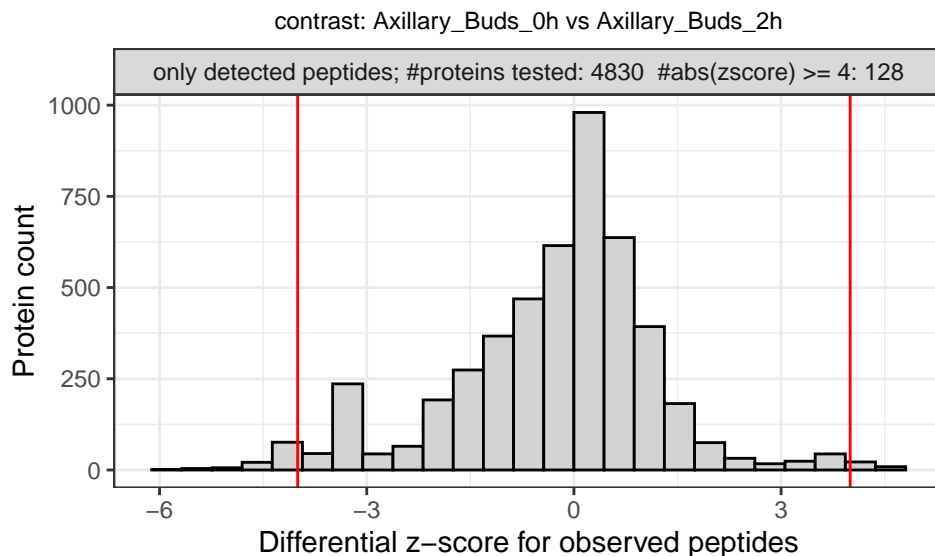


2.1.4 differential detect

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 4~5, 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 4 std. Both the z-scores and the counts these are based upon are available in the statistical result Excel table.



2.2 Apical_Buds_0h vs Apical_Buds_2h

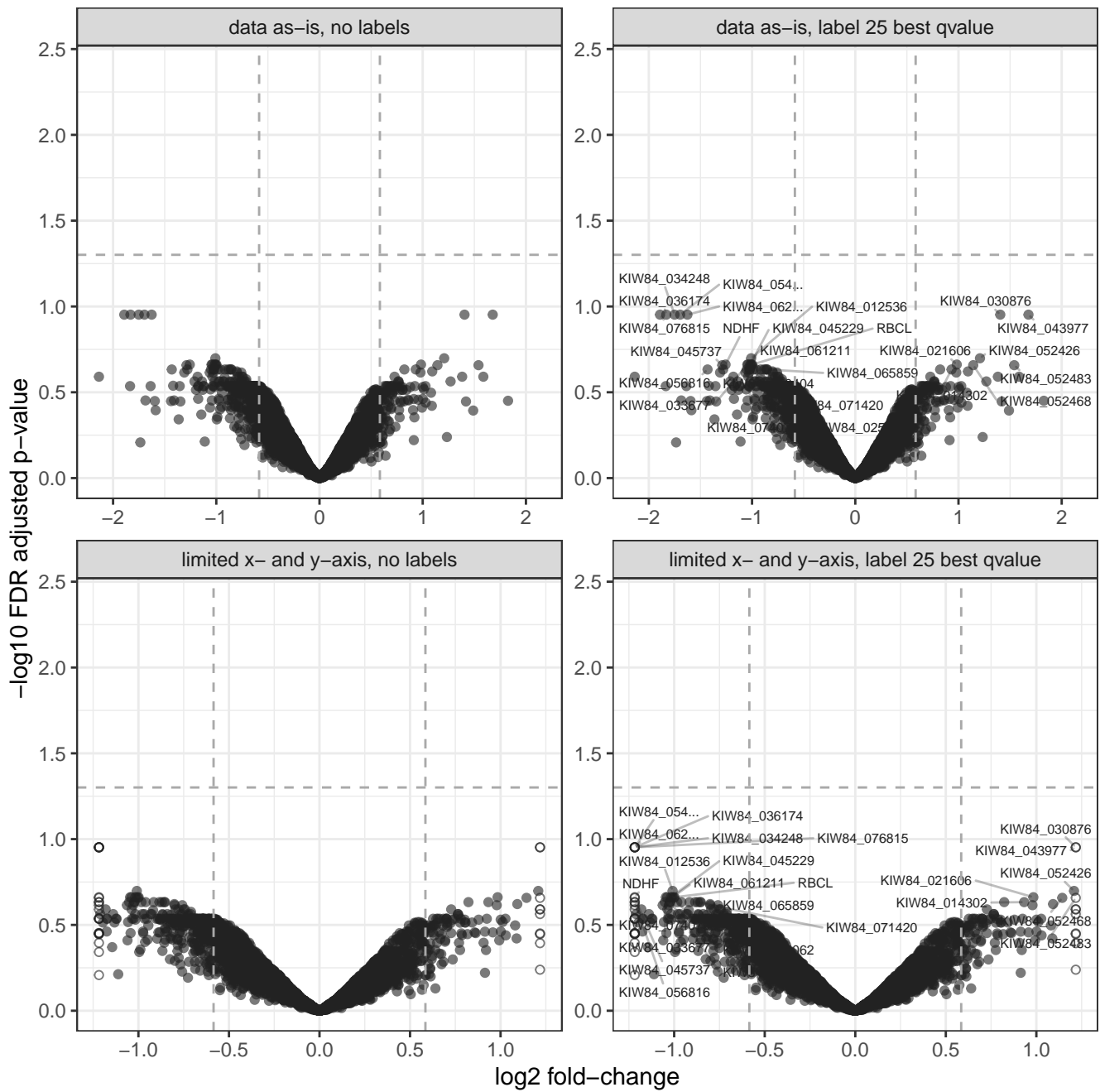
- **user setting:** using ‘filter by contrast’ peptide filtering approach
- 18609 peptides in 4572 proteins remain in the current contrast after peptide filters and are used for the statistical analysis in this section
- qvalue threshold: 0.05
- log2 foldchange threshold: 0.5849625

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.

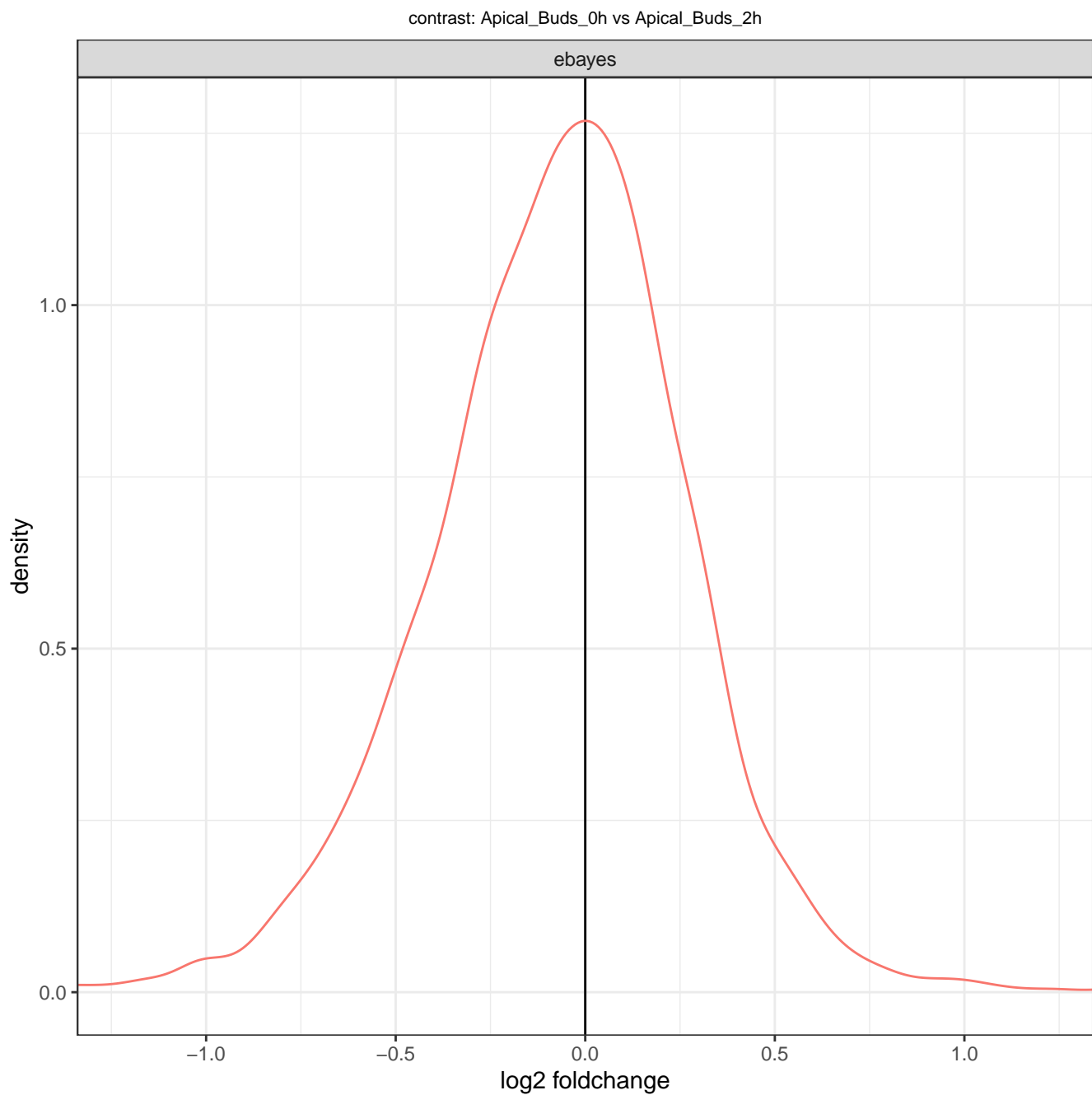
Bottom figure panels have limited x- and y-axis. For datasets with a small number of strong outliers in p-value or fold-change, which may have a profound effect on the plot scales, this allows inspection of the remainder of the volcano plot without disproportionate influence by ‘extreme’ values.

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 *).



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.

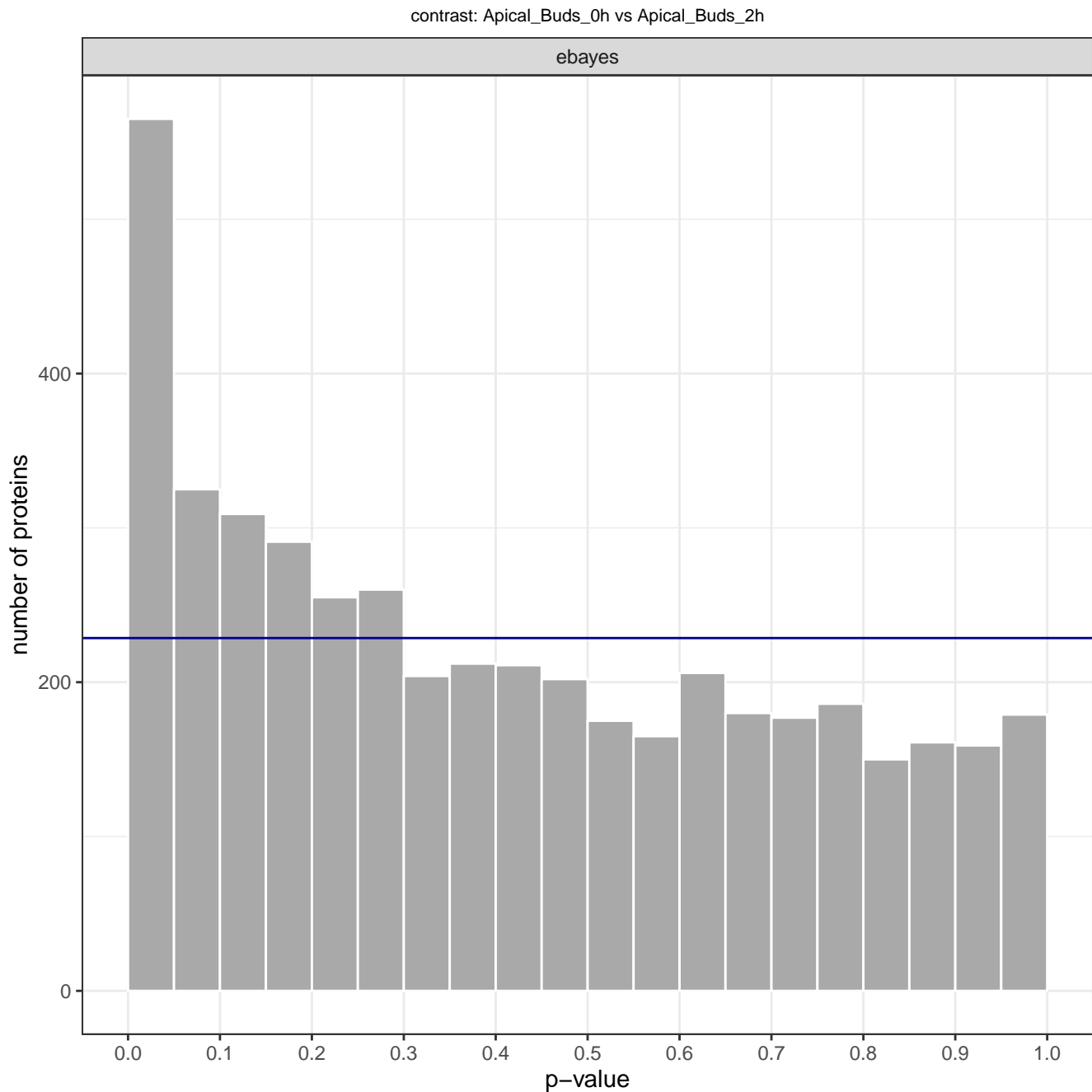


2.2.3 p-value distribution

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/>

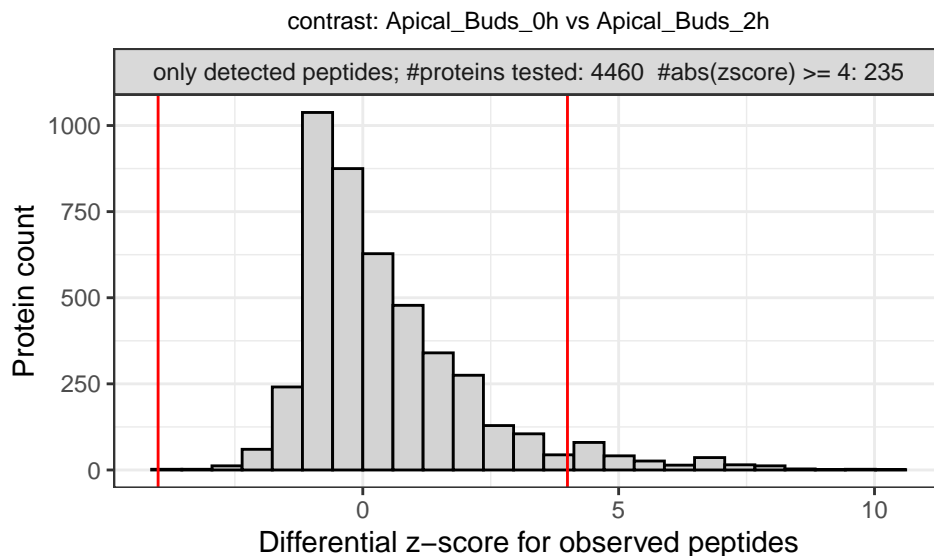


2.2.4 differential detect

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 4~5, 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 4 std. Both the z-scores and the counts these are based upon are available in the statistical result Excel table.



2.3 Apical_Buds_0h vs Axillary_Buds_0h

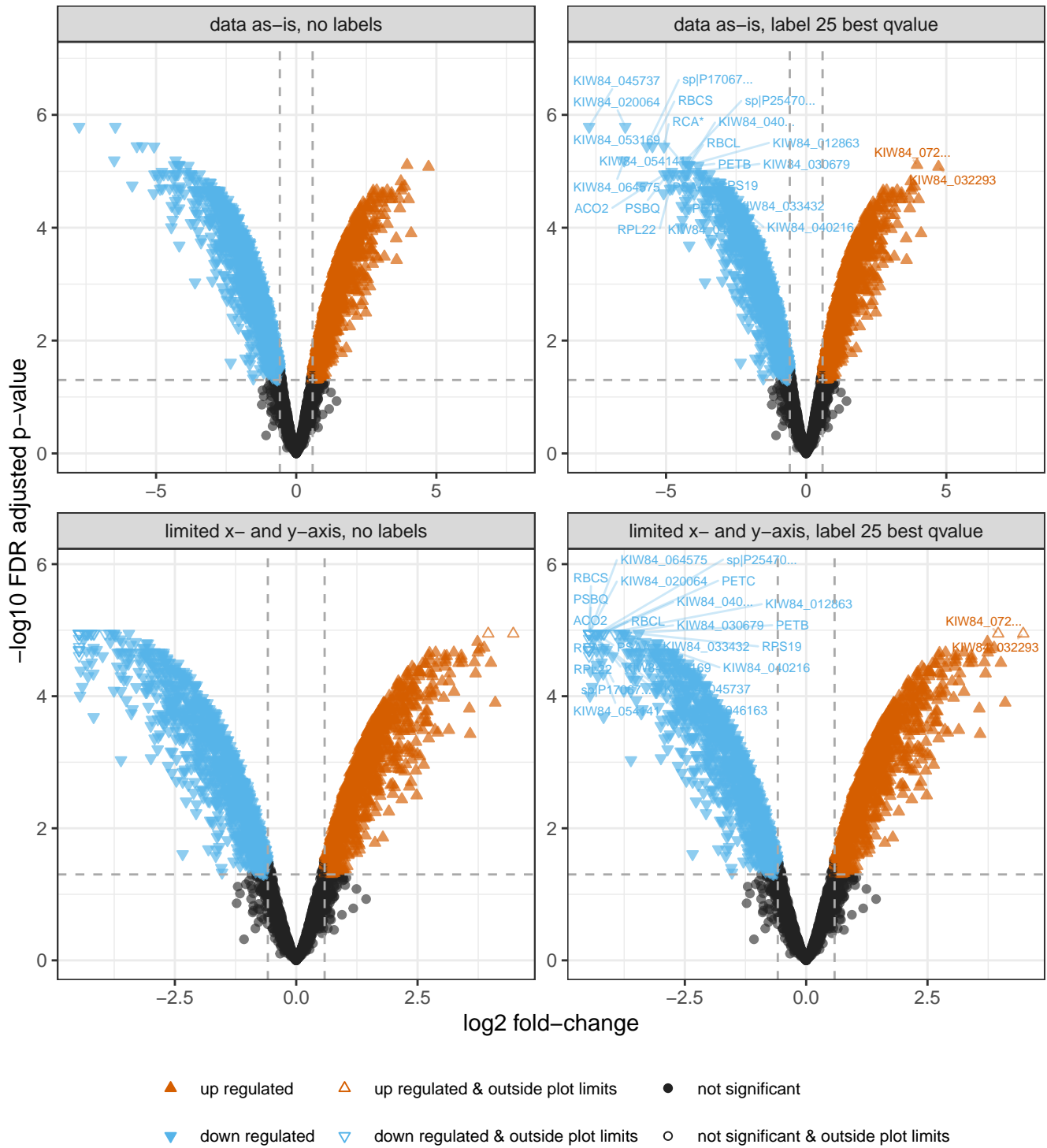
- **user setting:** using ‘filter by contrast’ peptide filtering approach
- 9908 peptides in 3344 proteins remain in the current contrast after peptide filters and are used for the statistical analysis in this section
- qvalue threshold: 0.05
- log2 foldchange threshold: 0.5849625

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.

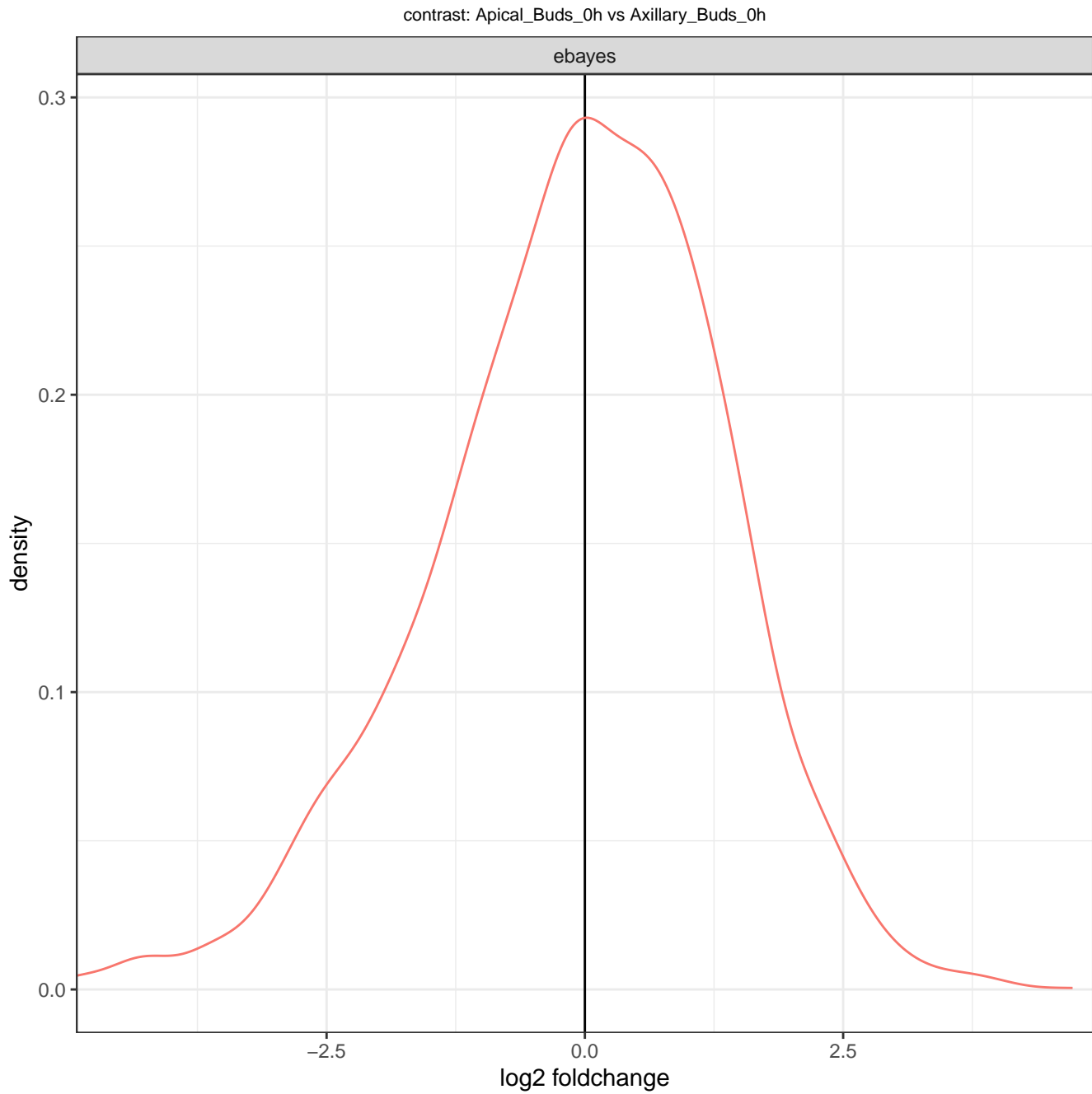
Bottom figure panels have limited x- and y-axis. For datasets with a small number of strong outliers in p-value or fold-change, which may have a profound effect on the plot scales, this allows inspection of the remainder of the volcano plot without disproportionate influence by ‘extreme’ values.

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 *).



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.

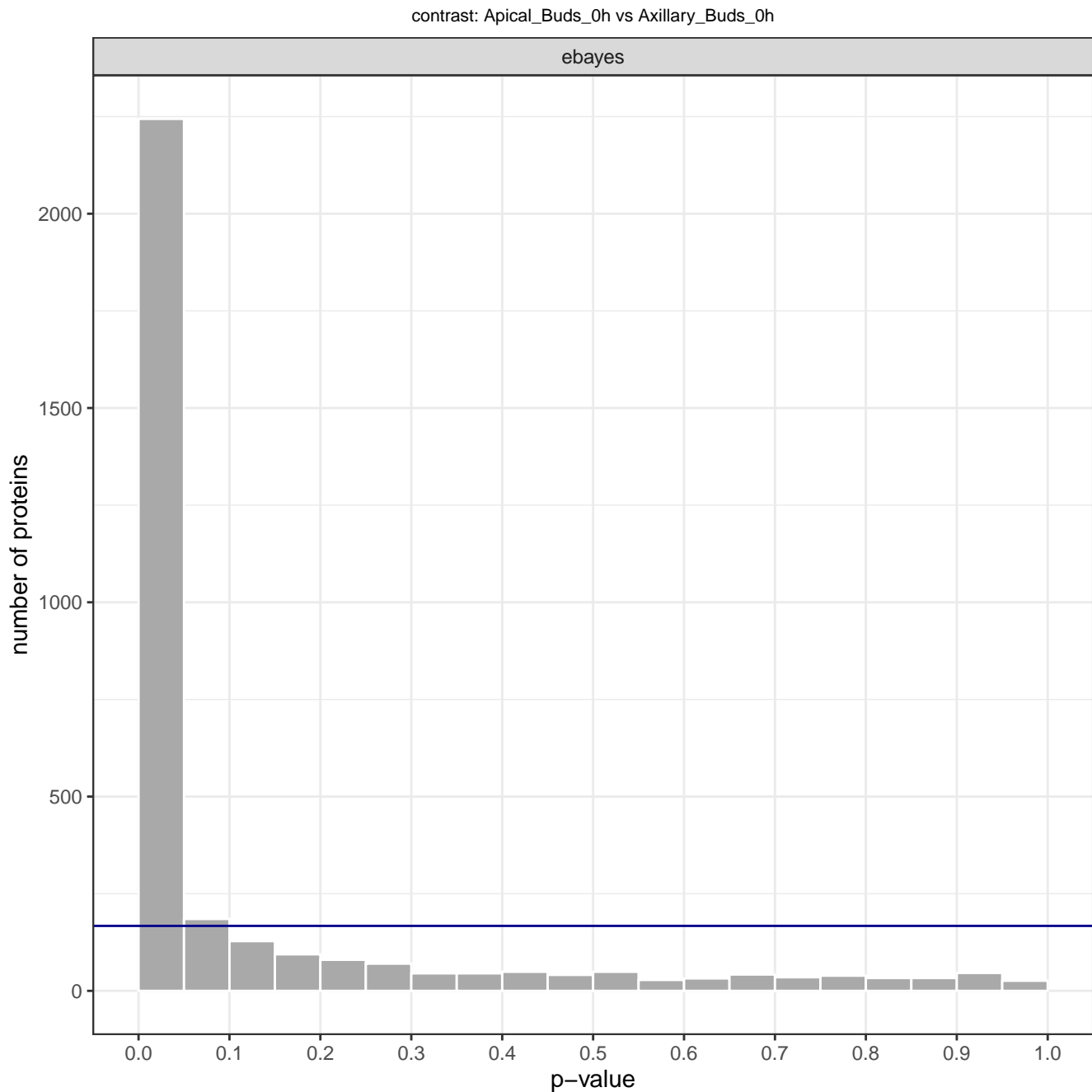


2.3.3 p-value distribution

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/>

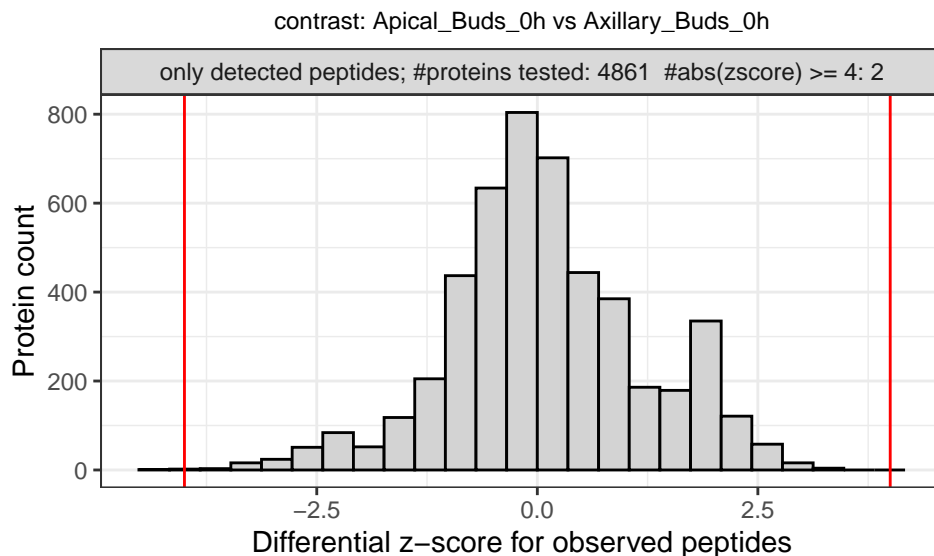


2.3.4 differential detect

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 4~5, 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 4 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
Axillary_Buds_0h vs lary_Buds_2h	Axil-	ebayes	4060	2115	1776	2435	kiw84_0..., kiw84_0..., sp p254..., kiw84_0..., kiw84_0..., kiw84_0..., kiw84_0..., kiw84_0..., kiw84_0...
Apical_Buds_0h vs cal_Buds_2h	Api-	ebayes	4572	0	0	0	kiw84_0..., kiw84_0..., kiw84_0..., kiw84_0..., kiw84_0..., kiw84_0..., kiw84_0..., kiw84_0..., kiw84_0..., ndhf
Apical_Buds_0h vs lary_Buds_0h	Axil-	ebayes	3344	2074	1603	2114	kiw84_0..., kiw84_0..., sp p170..., rbc, rca;kiw..., sp p254..., kiw84_0..., kiw84_0..., kiw84_0..., kiw84_0...

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

Differential detection scores based only on ‘detected’ peptides;

contrast		#proteins	#abs(zscore) >= 4	top10
Axillary_Buds_0h vs lary_Buds_2h	Axil-	4830	128	kiw84_0..., hsp70, kiw84_0..., kiw84_0..., hsp22, kiw84_0..., kiw84_0..., kiw84_0..., as1, kiw84_0...
Apical_Buds_0h vs cal_Buds_2h	Api-	4460	235	kiw84_0..., kiw84_0..., kiw84_0..., kiw84_0..., kiw84_0..., kiw84_0..., kiw84_0..., kiw84_0..., kiw84_0...
Apical_Buds_0h vs lary_Buds_0h	Axil-	4861	2	kiw84_0..., kiw84_0..., kiw84_0..., ndhh, kiw84_0..., kiw84_0..., thi1, kiw84_0..., kiw84_0..., kiw84_0...

4 log

[info] reading DIA-NN report...

[info] input file: Pisum_2023_2024_report.tsv

[info] 11 unique target (plain)sequences ambiguously mapped to multiple proteins and thus removed. Examples; AIADGSLLDFLR, DDLFNINA-GIVK, FKDVDESSFK, GGLDFTK, HQATQIVSMR

[info] 42216/45086 precursors remain after selecting the 'best' precursor for each modified sequence

[info] 11529/11529 protein accessions and 8059/8059 protein groups were mapped to provided fasta file(s)

[info] Sample metadata file "Pisum_metadata.xlsx" has been created. Please open it with Excel/LibreOffice and enter experiment metadata (this file contains documentation at the "instructions" tab)

[info] contrast: Axillary_Buds_0h vs Axillary_Buds_2h

[info] contrast: Apical_Buds_0h vs Apical_Buds_2h

[info] contrast: Apical_Buds_0h vs Axillary_Buds_0h

[progress] caching filter data took 3 seconds

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

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

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

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

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

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

[info] filter dataset with settings: min_detect = 2; fraction_detect = 0.5; min_quant = 2; fraction_quant = 0.5; norm_algorithm = 'vsn&modebetween_protein'; rollup_algorithm = 'maxlfq'

8487/42216 peptides were retained after filtering over all groups

12927/42216 peptides were retained after filtering within contrast: Axillary_Buds_0h vs Axillary_Buds_2h

18609/42216 peptides were retained after filtering within contrast: Apical_Buds_0h vs Apical_Buds_2h

9908/42216 peptides were retained after filtering within contrast: Apical_Buds_0h vs Axillary_Buds_0h

41235/42216 peptides were retained after filtering within each group independently ("by group")

[progress] peptide filtering and normalization took 10 seconds

[info] differential expression analysis for contrast: Axillary_Buds_0h vs Axillary_Buds_2h

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

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

[progress] eBayes took 1 seconds

[info] differential expression analysis for contrast: Apical_Buds_0h vs Apical_Buds_2h

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

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

[progress] eBayes took 1 seconds

[info] differential expression analysis for contrast: Apical_Buds_0h vs Axillary_Buds_0h

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

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

[progress] eBayes took 1 seconds

[info] differential detection analysis: min_samples_observed=2 min_fraction_observed=0.50

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

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

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

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

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

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

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

[progress] creating PDF report...

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

[progress] report: constructing plots specific for each contrast

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

[progress] RT plots: preparing data took 4 seconds

[progress] RT plots: creating plots took 1 seconds

[info] No data available for CoV leave-one-out computation in sample group 'Apical_Buds_0h', skipping plots

[info] No data available for CoV leave-one-out computation in sample group 'Axillary_Buds_0h', skipping plots

[info] No data available for CoV leave-one-out computation in sample group 'Axillary_Buds_2h', skipping plots

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

[info] no CoV computation for sample group 'Apical_Buds_0h', require at least 3 replicates

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

[info] no CoV computation for sample group 'Apical_Buds_0h', require at least 3 replicates

5 R command history

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

Using this feature

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

```
dataset = import_dataset_diann("Pisum_2023_2024_report.tsv")
dataset = import_fasta(dataset,
  files = "C:/Nitesh/Work/Akila/SUB968_Pisum/Pisum_2023_2024/uniprotkb_taxonomy_id_3888_2024_03_25.fasta"
)
write_template_for_sample_metadata(
  dataset,
  "Pisum_metadata.xlsx"
)
metadata = read.xlsx("Pisum_metadata.xlsx")
metadata = metadata[order(metadata$sample_id), ]
samplernames = read.xlsx("Sample names_Pisum_2023_2024.xlsx")
samplernames$File.name = gsub(
  ".mzML|\\s",
  "", samplernames$File.name
)
samplernames = samplernames[order(samplernames$File.name), ]
metadata$shortname = samplernames$Concatenation[samplernames$File.name %in%
  metadata$sample_id]
metadata$group = sub(
  "[^_]*$",
  "", metadata$shortname
)
metadata$exclude = grepl(
  "sample5",
  metadata$sample_id
)
metadata = metadata[
  ,
  c(1:4)
]
write.xlsx(
  metadata,
  "Pisum_metadata.xlsx"
)
dataset = import_sample_metadata(dataset,
  filename = "Pisum_metadata.xlsx"
)
dataset = setup_contrasts(dataset,
  contrast_list = list(
    c(
      "Axillary_Buds_0h",
      "Axillary_Buds_2h"
    ),
    c(
      "Apical_Buds_0h",
      "Apical_Buds_2h"
    ),
    c(
      "Apical_Buds_0h",
      "Axillary_Buds_0h"
    )
  )
)
dataset = analysis_quickstart(dataset,
  filter_min_detect = 2,
  filter_min_quant = 2,
  filter_fraction_detect = 0.5,
```

```

filter_fraction_quant = 0.5,
filter_by_contrast = TRUE,
norm_algorithm = c(
  "vsn",
  "modebetween_protein"
),
rollup_algorithm = "maxlfq",
dea_algorithm = c("ebayes"),
dea_qvalue_threshold = 0.05,
dea_log2foldchange_threshold = log2(1.5),
diffdetect_min_samples_observed = 2,
output_qc_report = TRUE,
output_abundance_tables = TRUE,
output_dir = "Pisum_MS-DAP_results",
output_within_timestamped_subdirectory = TRUE
)

```

6 R session info

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

setting	value
version	R version 4.4.0 (2024-04-24 ucrt)
os	Windows 11 x64 (build 22631)
system	x86_64, mingw32
ui	Rgui
language	(EN)
collate	English_United States.utf8
ctype	English_United States.utf8
tz	Asia/Calcutta
date	2024-05-06
pandoc	3.1.12.1 @ C:/Users/nites/AppData/Local/Pandoc/ (via rmarkdown)

System

package	loadedversion	source
Biobase	2.64.0	Bioconductor 3.19 (R 4.4.0)
BiocGenerics	0.50.0	Bioconductor 3.19 (R 4.4.0)
BiocManager	1.30.23	CRAN (R 4.4.0)
ComplexHeatmap	2.20.0	Bioconductor 3.19 (R 4.4.0)
devtools	2.4.5	CRAN (R 4.4.0)
dplyr	1.1.4	CRAN (R 4.4.0)
forcats	1.0.0	CRAN (R 4.4.0)
ggplot2	3.5.1	CRAN (R 4.4.0)
ggrepel	0.9.5	CRAN (R 4.4.0)
ggVennDiagram	1.5.2	CRAN (R 4.4.0)
limma	3.60.0	Bioconductor 3.19 (R 4.4.0)
lubridate	1.9.3	CRAN (R 4.4.0)
msdap	1.0.7	Github (ftwkoopmans/msdap@f1b12faf6934637e1d653c58cbb5f01982952a93)
MSnbase	2.30.1	Bioconductor 3.19 (R 4.4.0)
mzR	2.38.0	Bioconductor 3.19 (R 4.4.0)
openxlsx	4.2.5.2	CRAN (R 4.4.0)
ProtGenerics	1.36.0	Bioconductor 3.19 (R 4.4.0)
purrr	1.0.2	CRAN (R 4.4.0)
Rcpp	1.0.12	CRAN (R 4.4.0)
readr	2.1.5	CRAN (R 4.4.0)
rlang	1.1.3	CRAN (R 4.4.0)
S4Vectors	0.42.0	Bioconductor 3.19 (R 4.4.0)
stringr	1.5.1	CRAN (R 4.4.0)
tibble	3.2.1	CRAN (R 4.4.0)
tidyr	1.3.1	CRAN (R 4.4.0)
tidyverse	2.0.0	CRAN (R 4.4.0)
tinytex	0.50	CRAN (R 4.4.0)
usethis	2.2.3	CRAN (R 4.4.0)

Attached packages

package	loadedversion	source
abind	1.4-5	CRAN (R 4.4.0)
affy	1.82.0	Bioconductor 3.19 (R 4.4.0)
affyio	1.74.0	Bioconductor 3.19 (R 4.4.0)
AnnotationFilter	1.28.0	Bioconductor 3.19 (R 4.4.0)
aod	1.3.3	CRAN (R 4.4.0)
archive	1.1.8	CRAN (R 4.4.0)
askpass	1.2.0	CRAN (R 4.4.0)
backports	1.4.1	CRAN (R 4.4.0)
BiocParallel	1.38.0	Bioconductor 3.19 (R 4.4.0)
bit	4.0.5	CRAN (R 4.4.0)
bit64	4.0.5	CRAN (R 4.4.0)
bitops	1.0-7	CRAN (R 4.4.0)
blob	1.2.4	CRAN (R 4.4.0)
boot	1.3-30	CRAN (R 4.4.0)
broom	1.0.5	CRAN (R 4.4.0)
cachem	1.0.8	CRAN (R 4.4.0)
car	3.1-2	CRAN (R 4.4.0)
carData	3.0-5	CRAN (R 4.4.0)
caTools	1.18.2	CRAN (R 4.4.0)
circlize	0.4.16	CRAN (R 4.4.0)
cli	3.6.2	CRAN (R 4.4.0)
clue	0.3-65	CRAN (R 4.4.0)
cluster	2.1.6	CRAN (R 4.4.0)
codetools	0.2-20	CRAN (R 4.4.0)
colorspace	2.1-0	CRAN (R 4.4.0)
corpcor	1.6.10	CRAN (R 4.4.0)
cowplot	1.1.3	CRAN (R 4.4.0)
crayon	1.5.2	CRAN (R 4.4.0)
data.table	1.15.4	CRAN (R 4.4.0)
DBI	1.2.2	CRAN (R 4.4.0)
DelayedArray	0.30.0	Bioconductor 3.19 (R 4.4.0)
DEqMS	1.22.0	bioc_xgit (@bc7668015a94ed8a948600a50a607ea2e4bf45d2)
diann	1.0.1	Github (vdemichev/diann-rpackage@af538f6e2cd5ab715e1381632e17cb8f234ebf53)
digest	0.6.35	CRAN (R 4.4.0)
doParallel	1.0.17	CRAN (R 4.4.0)
doRNG	1.8.6	CRAN (R 4.4.0)
ellipsis	0.3.2	CRAN (R 4.4.0)
EnvStats	2.8.1	CRAN (R 4.4.0)
evaluate	0.23	CRAN (R 4.4.0)
fANCOVA	0.6-1	CRAN (R 4.4.0)
fansi	1.0.6	CRAN (R 4.4.0)
farver	2.1.1	CRAN (R 4.4.0)
fastmap	1.1.1	CRAN (R 4.4.0)
foreach	1.5.2	CRAN (R 4.4.0)
formatR	1.14	CRAN (R 4.4.0)
fs	1.6.4	CRAN (R 4.4.0)
generics	0.1.3	CRAN (R 4.4.0)
GenomeInfoDb	1.40.0	Bioconductor 3.19 (R 4.4.0)
GenomeInfoDbData	1.2.12	Bioconductor
GenomicRanges	1.56.0	Bioconductor 3.19 (R 4.4.0)

package	loadedversion	source
GetoptLong	1.0.5	CRAN (R 4.4.0)
ggpubr	0.6.0	CRAN (R 4.4.0)
ggsignif	0.6.4	CRAN (R 4.4.0)
GlobalOptions	0.1.2	CRAN (R 4.4.0)
glue	1.7.0	CRAN (R 4.4.0)
gplots	3.1.3.1	CRAN (R 4.4.0)
gridExtra	2.3	CRAN (R 4.4.0)
gtable	0.3.5	CRAN (R 4.4.0)
gtools	3.9.5	CRAN (R 4.4.0)
hms	1.1.3	CRAN (R 4.4.0)
htmltools	0.5.8.1	CRAN (R 4.4.0)
htmlwidgets	1.6.4	CRAN (R 4.4.0)
httpuv	1.6.15	CRAN (R 4.4.0)
httr	1.4.7	CRAN (R 4.4.0)
igraph	2.0.3	CRAN (R 4.4.0)
impute	1.78.0	Bioconductor 3.19 (R 4.4.0)
iq	1.9.12	CRAN (R 4.4.0)
IRanges	2.38.0	Bioconductor 3.19 (R 4.4.0)
iterators	1.0.14	CRAN (R 4.4.0)
itertools	0.1-3	CRAN (R 4.4.0)
jsonlite	1.8.8	CRAN (R 4.4.0)
KernSmooth	2.23-22	CRAN (R 4.4.0)
knitr	1.46	CRAN (R 4.4.0)
labeling	0.4.3	CRAN (R 4.4.0)
later	1.3.2	CRAN (R 4.4.0)
lattice	0.22-6	CRAN (R 4.4.0)
lazyeval	0.2.2	CRAN (R 4.4.0)
lifecycle	1.0.4	CRAN (R 4.4.0)
lme4	1.1-35.3	CRAN (R 4.4.0)
lmerTest	3.1-3	CRAN (R 4.4.0)
magrittr	2.0.3	CRAN (R 4.4.0)
MALDIquant	1.22.2	CRAN (R 4.4.0)
MASS	7.3-60.2	local
Matrix	1.7-0	CRAN (R 4.4.0)
MatrixGenerics	1.16.0	Bioconductor 3.19 (R 4.4.0)
matrixStats	1.3.0	CRAN (R 4.4.0)
memoise	2.0.1	CRAN (R 4.4.0)
mime	0.12	CRAN (R 4.4.0)
miniUI	0.1.1.1	CRAN (R 4.4.0)
minqa	1.2.6	CRAN (R 4.4.0)
missForest	1.5	CRAN (R 4.4.0)
MsCoreUtils	1.16.0	Bioconductor 3.19 (R 4.4.0)
msEmpiRe	0.1.0	Github (zimmerlab/MS-EmpiRe@8a85757c8d604014130ee9d379aa8cfeb05e3855)
MultiAssayExperiment	1.30.1	Bioconductor 3.19 (R 4.4.0)
munsell	0.5.1	CRAN (R 4.4.0)
mvtnorm	1.2-4	CRAN (R 4.4.0)
mzID	1.42.0	Bioconductor 3.19 (R 4.4.0)
ncdf4	1.22	CRAN (R 4.4.0)
nlme	3.1-164	CRAN (R 4.4.0)
nloptr	2.0.3	CRAN (R 4.4.0)

package	loadedversion	source
numDeriv	2016.8-1.1	CRAN (R 4.4.0)
openssl	2.1.2	CRAN (R 4.4.0)
patchwork	1.2.0	CRAN (R 4.4.0)
pbkrtest	0.5.2	CRAN (R 4.4.0)
pcaMethods	1.96.0	Bioconductor 3.19 (R 4.4.0)
pdftools	3.4.0	CRAN (R 4.4.0)
pillar	1.9.0	CRAN (R 4.4.0)
pkgbuild	1.4.4	CRAN (R 4.4.0)
pkgconfig	2.0.3	CRAN (R 4.4.0)
pkgload	1.3.4	CRAN (R 4.4.0)
plyr	1.8.9	CRAN (R 4.4.0)
png	0.1-8	CRAN (R 4.4.0)
preprocessCore	1.66.0	Bioconductor 3.19 (R 4.4.0)
pROC	1.18.5	CRAN (R 4.4.0)
profvis	0.3.8	CRAN (R 4.4.0)
promises	1.3.0	CRAN (R 4.4.0)
PSMatch	1.8.0	Bioconductor 3.19 (R 4.4.0)
QFeatures	1.14.0	Bioconductor 3.19 (R 4.4.0)
qpdf	1.3.3	CRAN (R 4.4.0)
R.cache	0.16.0	CRAN (R 4.4.0)
R.methodsS3	1.8.2	CRAN (R 4.4.0)
R.oo	1.26.0	CRAN (R 4.4.0)
R.utils	2.12.3	CRAN (R 4.4.0)
R6	2.5.1	CRAN (R 4.4.0)
randomForest	4.7-1.1	CRAN (R 4.4.0)
rbibutils	2.2.16	CRAN (R 4.4.0)
RColorBrewer	1.1-3	CRAN (R 4.4.0)
RcppEigen	0.3.4.0.0	CRAN (R 4.4.0)
Rdpack	2.6	CRAN (R 4.4.0)
remaCor	0.0.18	CRAN (R 4.4.0)
remotes	2.5.0	CRAN (R 4.4.0)
reshape2	1.4.4	CRAN (R 4.4.0)
RhpcBLASctl	0.23-42	CRAN (R 4.4.0)
rjson	0.2.21	CRAN (R 4.4.0)
rmarkdown	2.26	CRAN (R 4.4.0)
rngtools	1.5.2	CRAN (R 4.4.0)
RSQLite	2.3.6	CRAN (R 4.4.0)
rstatix	0.7.2	CRAN (R 4.4.0)
S4Arrays	1.4.0	Bioconductor 3.19 (R 4.4.0)
scales	1.3.0	CRAN (R 4.4.0)
sessioninfo	1.2.2	CRAN (R 4.4.0)
shape	1.4.6.1	CRAN (R 4.4.0)
shiny	1.8.1.1	CRAN (R 4.4.0)
SparseArray	1.4.0	Bioconductor 3.19 (R 4.4.0)
statmod	1.5.0	CRAN (R 4.4.0)
stringi	1.8.3	CRAN (R 4.4.0)
styler	1.10.3	CRAN (R 4.4.0)
SummarizedExperiment	1.34.0	Bioconductor 3.19 (R 4.4.0)
tidyselect	1.2.1	CRAN (R 4.4.0)
timechange	0.3.0	CRAN (R 4.4.0)

package	loadedversion	source
tzdb	0.4.0	CRAN (R 4.4.0)
UCSC.utils	1.0.0	Bioconductor 3.19 (R 4.4.0)
urlchecker	1.0.1	CRAN (R 4.4.0)
utf8	1.2.4	CRAN (R 4.4.0)
variancePartition	1.34.0	bioc_xgit (@b4e63a4ddb7699fe5ff11c90e42c29e061db6a)
vctrs	0.6.5	CRAN (R 4.4.0)
viridis	0.6.5	CRAN (R 4.4.0)
viridisLite	0.4.2	CRAN (R 4.4.0)
vroom	1.6.5	CRAN (R 4.4.0)
vsn	3.72.0	Bioconductor 3.19 (R 4.4.0)
withr	3.0.0	CRAN (R 4.4.0)
xfun	0.43	CRAN (R 4.4.0)
XML	3.99-0.16.1	CRAN (R 4.4.0)
xtable	1.8-4	CRAN (R 4.4.0)
XVector	0.44.0	Bioconductor 3.19 (R 4.4.0)
yaml	2.3.8	CRAN (R 4.4.0)
zip	2.3.1	CRAN (R 4.4.0)
zlibbioc	1.50.0	Bioconductor 3.19 (R 4.4.0)

Packages that are not attached