Single Cell Proteomics Analysis Workflow

2024

This pipeline is prepared to be used with the output files from ProteomeDiscoverer. It requires two input files:

1) *""_InputFiles.txt*
2) *""_PSM.txt with the following columns only:* File.ID, Master.Protein.Accessions, Annotated.Sequence, Percolator.PEP, all Abundance TMTs.

*To analyse multiple experiments/searches at the same time, the initial files should contain:

1. Multiple results in one file (one under the other).
2. The two experiments should not have any *File.ID* in common. (ex. If F1 appears in both experiments, which is a likely situation, modify all files of experiment1 from letter *F* to letter *G*, so ending up with experiment1 *F1* and experiment2 *G1*; and so on).
3. Add a new column in the original *""_InputFiles.txt*, named "Experiment", with the name of the experiment in all rows.

The current script for SCP is written in R (version 4.1.2)[1]:

- *SCP_2024.R*

Preprocessing:

First of all, it will ask to select both PSMs and InputFiles, then a pop up will appear asking if the files have been already converted or if they are coming straight from PD. This steps will take a while depending on the size of the files, and it will produce two output files:

1) *""_InputFiles_forSCP.txt*
2) *""_PSM_forSCP.txt*

Always check for extra popups. (ex. During this process a pop up will ask about the TMT type: 16, 18 etc)

Choosing parameters:

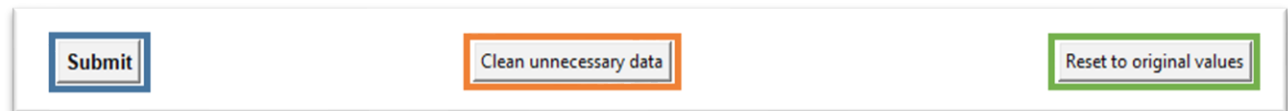A pop up will appear with a series of dynamic values for the user to decide:

1) Name of the SCP experiment: All output files will contain this name. (ex. *SCP_Experiment*)
2) Minimum amount of PSMs allowed by Sample.*
3) Lower and Upper threshold for Single Cell Ratio.*
4) Booster, name of the column of the Booster to use. *Abundance.126* by default, to use median of single cell ratios use: *Abundance.TotalMedianIntensity*, for any other specify the column name.
5) FDR threshold, usually 0.01. *
6) Median Relative Intensity: Intensity of the Reporter Ions. *
7) Median Coefficient of Variance, lower and upper threshold. *
8) Imputation type, either from normal distribution or using KNN. Norm dist by default.
9) Batch effect, use the group to use for batch effect correction:
   a. Channels: By TMT labels
   b. Samples: By files
   c. Both: Correct for both batch effect of TMT and Files.

d. If "Experiment" selected, a batch correction for experiment will also be performed.

10) Minimum number of unique peptides per protein.
11) How to filter for the allowed missingness % threshold, using the booster strategy (30% allowed on the booster channel) or selecting it manually during the analysis.
12) Make extra run make extra analysis, more results (always yes, unless in a rush).
13) To save the generated plots and graphics.
14) To save the tables and files generated.
15) If there are multiple experiments or only one.
16) Fasta file to be used, provide the whole directory as in the example. Description for gene name: For identification of the gene names (eg. MOUSE, HUMAN)
17) Submit button: Submit parameters. Clean unnecessary data: Only provide necessary data, usually used for first analysis. Reset original values: Reset to default.

*Parameters starred can remain empty for the analysis. During the following, some pop-ups will appear where the thresholds can be easily set.

_____

The analysis is based on the SCP package from Bioconductor (V. Christophe et al. 2022)[2].

PSMs:

The first step is to remove those samples with lower amounts of PSMs. If the parameters for PSMs are left empty, a new pop-up will appear to be able to decide the threshold:
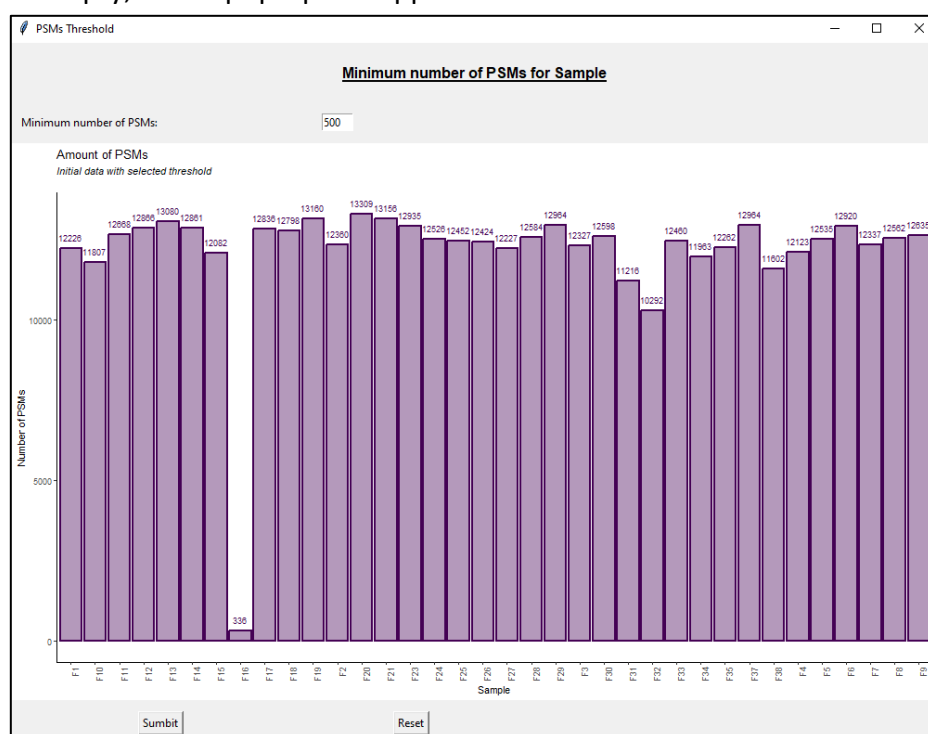
In this example we see that there's one file (F16) almost empty, so we can discard it by adding minimum number of PSMs higher than 336, for example 500. Once selected, the following output files will be saved, containing the amount of PSMs on each sample:

1) *1_Initial_PSM_x_Sample_Cutoff_* **SCP-Experiment**.pdf
2) *1_Initial_PSM_x_Sample_* **SCP-Experiment**.pdf

Distribution

During the whole analysis there will be several output files plotting the distribution of different levels of intensities according to both Files and TMT. Those files will be named with the following pattern:

1) *1_#_Distribution_***SCP-Experiment**.pdf

Being #: Initial & DivideByReference.

Median SCR

Next step is to compute the Median Single Cell ratio. This allow us to see the ratios of intensities of each one of the PSM, comparing the single cells and the booster. If the parameters for Median SCR are left empty, a new pop-up will appear to be able to decide both thresholds:
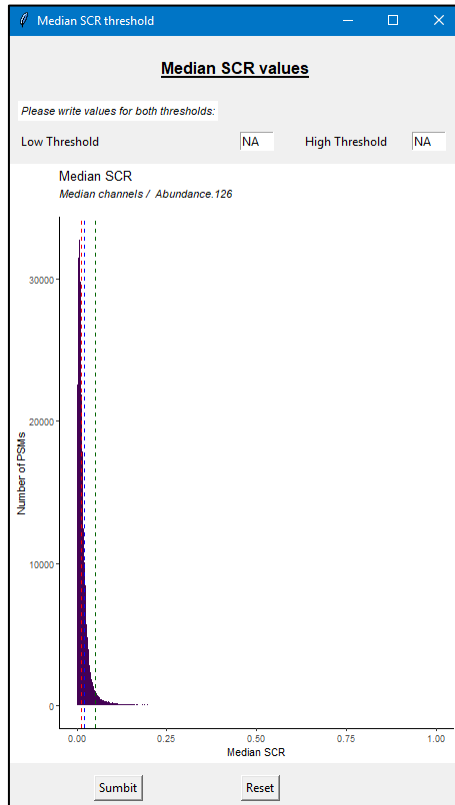
This data is computed with the following formula, for each PSM:

$$x = \frac{\text{Med(Intensity of SingleCell Channels)}}{\text{Booster intensity}}$$

This step allows to determine if all the samples have an equally distributed booster intensity, as well as, determine the thresholds for the reliable amount of Booster/SingleCell ratio. For the Low Threshold, use 0 since the data is not log transformed.

The output file has the following name:

1)   2_MedianSCR_**LowThreshold_HighThreshold_SCP-Experiment**.pdf

Only the data inside the thresholds will be used for the following analysis.

The colored lines represent: The expected ratio 1/20 (green), Mean (blue) & Median (red).

5

FDR

The FDR is computed as q-values from the posterior error probabilities (PEPs). Again if the parameters for FDR are left empty, a new pop-up will appear to be able to decide the threshold:
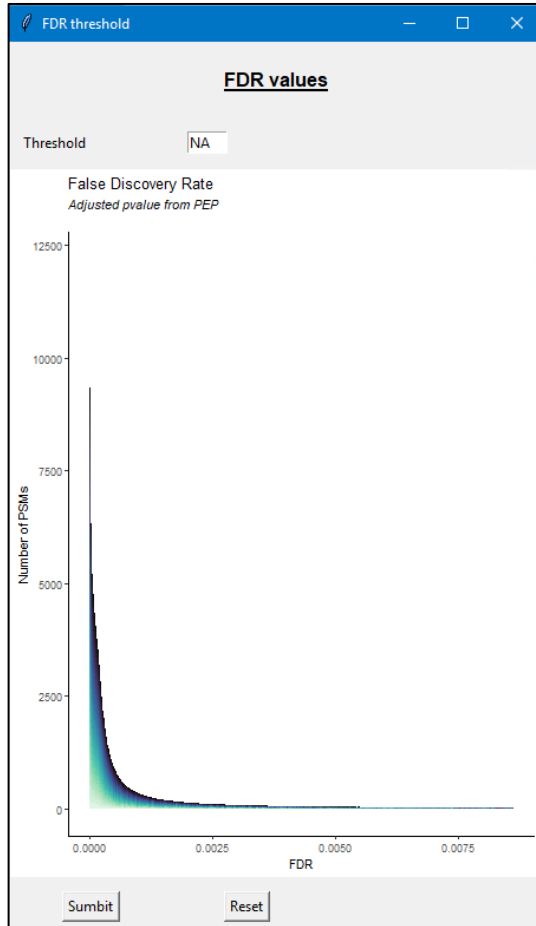
Usually the threshold selected is 0.01. The output file has the following name:

1)   3_FDR_***ThresholdSelected_SCP-Experiment***.pdf

Only the data below the threshold will be used for the following analysis.

Next, the intensities are normalized dividing them by the reference: *TotalMedianIntensity* by default.

Now the PSMs are grouped to Peptides. Using the median on the PSMs to determine the intensity of the peptide.

Peptides

Once the PSMs are added to Peptide level, a new output file will be saved with the amount of peptides by samples, named:

1) *1_Initial_Peptides_x_Sample_**SCP-Experiment**.pdf*

Containing a similar graphic that the one below, the horizontal line corresponds to the average number.
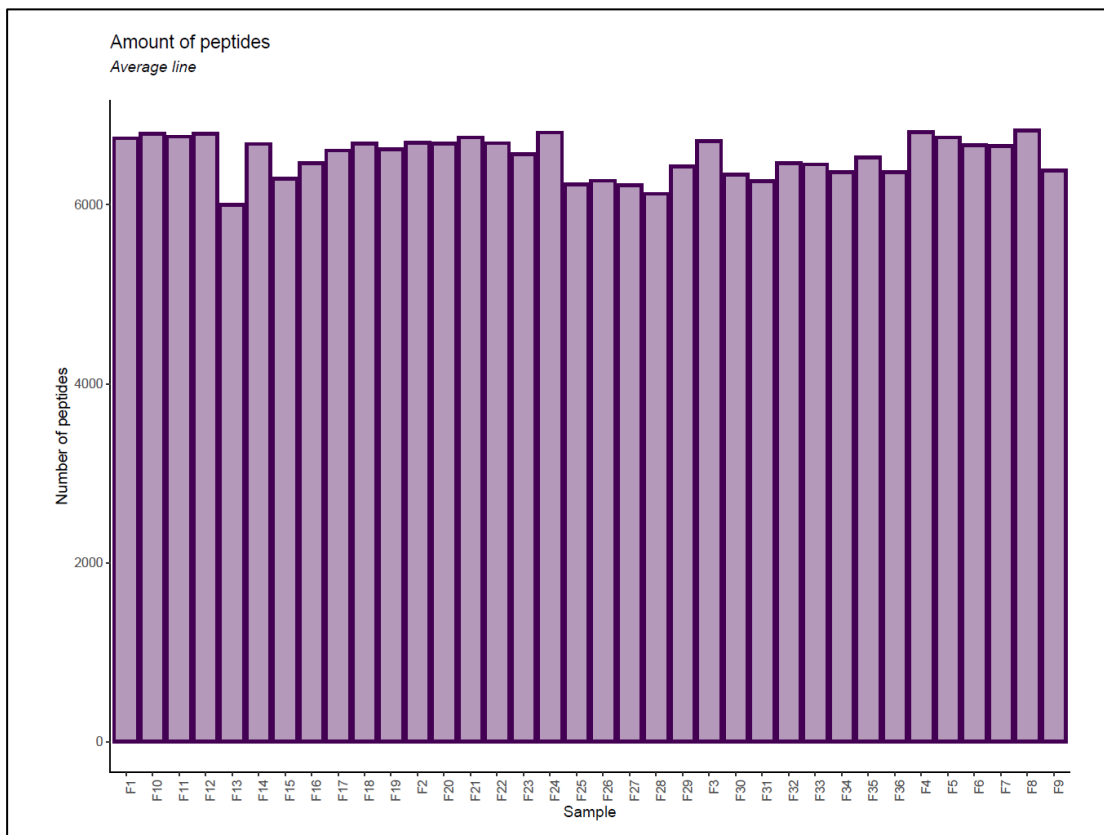
Median RI

The Median Relative reporter ion Intensity, checks the median intensity per cell, comparing it to the distribution of the channel, and allows to highlight low-quality cells. Again if the parameters for medianRI are left empty at the beginning of the analysis, a new pop-up will appear to be able to decide both thresholds:
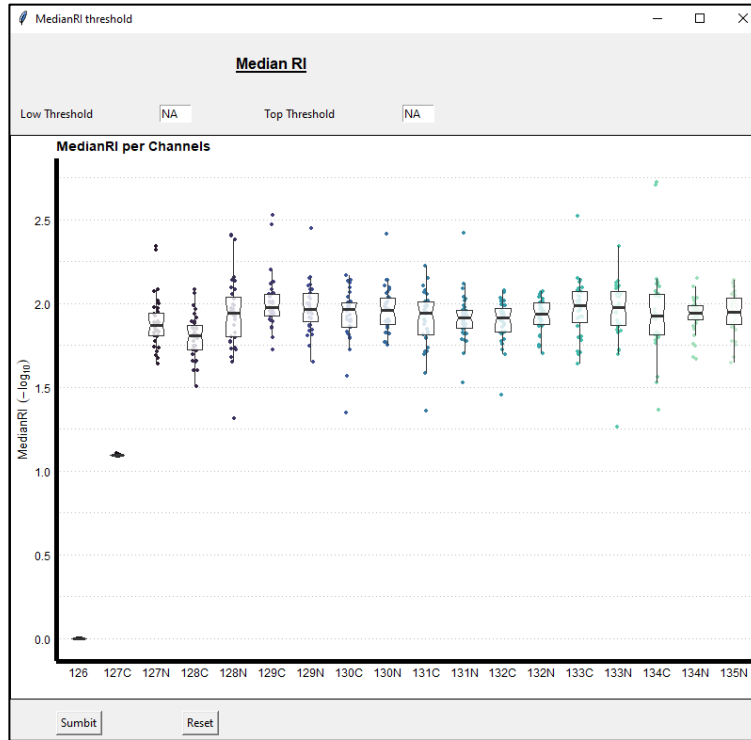
Notice the y axis is in -log10 scale. After selecting the Thresholds , a new output file will be saved with the same figure, named:

1) *4_MedianRI_Channels_**SCP-Experiment**.pdf*

The cells outside of the selected thresholds won't be used for the follow-up analysis.

MedianCV

The Median Coefficient of Variance $CV = \frac{\sigma}{Med}$ allows to determine the variability of the protein intensities of the cells in relation to their median. Again if the parameters for medianCV are left empty at the beginning of the analysis, a new pop-up will appear to be able to decide both thresholds:
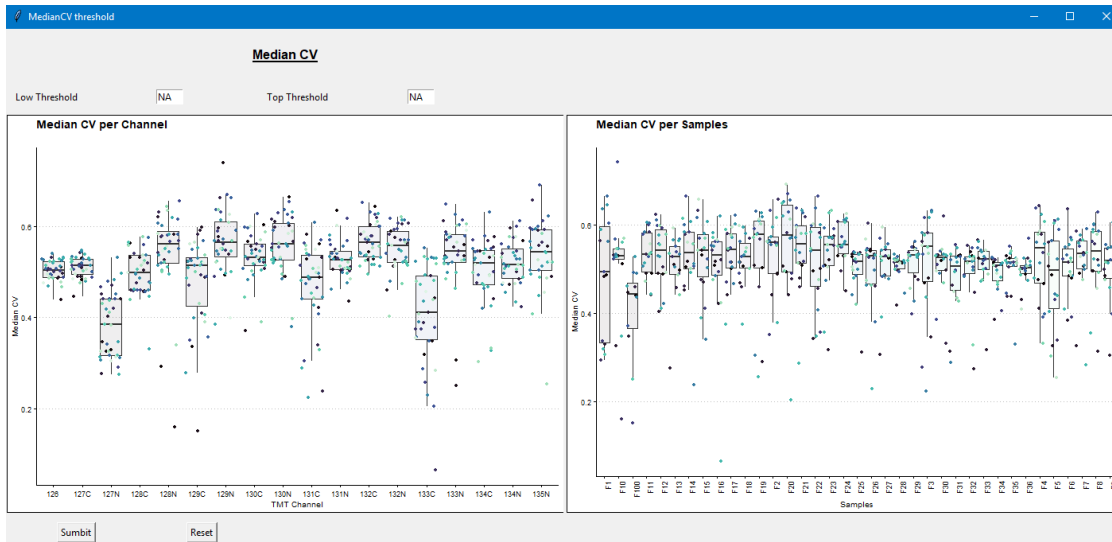
In this case, there's two ways of representing the data, organized by Channel or by Samples; if there are multiple experiments the data will also be represented by experiment. This step will generate 2 or 3 output files:

1) *4_MedianCV_**LowThreshold_TopThreshold_**Channels_**SCP-Experiment**.pdf*
2) *4_MedianCV_**LowThreshold_TopThreshold_**Samples_**SCP-Experiment**.pdf*
3) *4_MedianCV_**LowThreshold_TopThreshold_**Experiment_**SCP-Experiment**.pdf***

The cells outside of the selected thresholds won't be used for the follow-up analysis. The data will first be normalized and log scaled to proceed further on.

Now the Peptides are grouped into Proteins. Using the median on the peptides to determine the intensity of the protein.

Number of peptides

Filtering for number of unique peptides allows us to use only those proteins with a higher confidence of identification. If we have left the value empty at the parameters for minimum number of unique peptides, a new pop-up will appear to select the threshold:
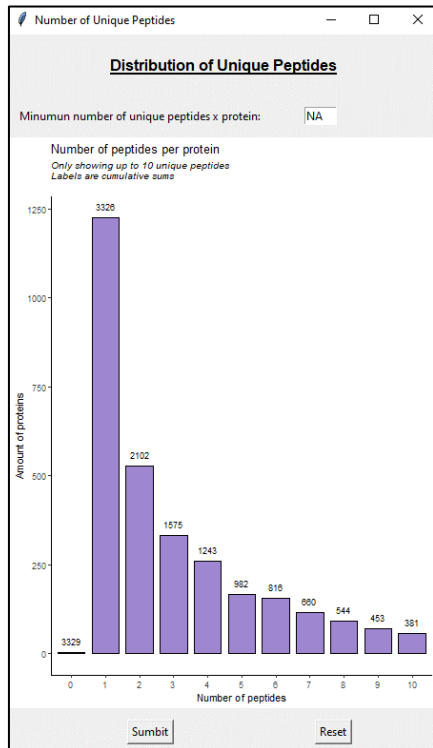
We can see in the x axis the amount of unique peptides, and on the y axis the corresponding amount of proteins. The labels are the cumulative sums of the amount of proteins. This allows us to filter for proteins containing, for example, at least one unique peptide; which in this case would leave us with 3326 amount of proteins. This step will generate 1 output file:

1) *5_UniquePeptides_ **SCP-Experiment**.pdf*

The proteins lower than the selection threshold won't be used for the follow-up analysis.

Proteins

Once the peptides are added to protein level, a new output file will be saved with the amount of peptides by samples, named:

1) *1_Initial_Proteins_x_Sample_**SCP-Experiment**.pdf*

Containing a similar graphic that the one below, the horizontal line corresponds to the average number.
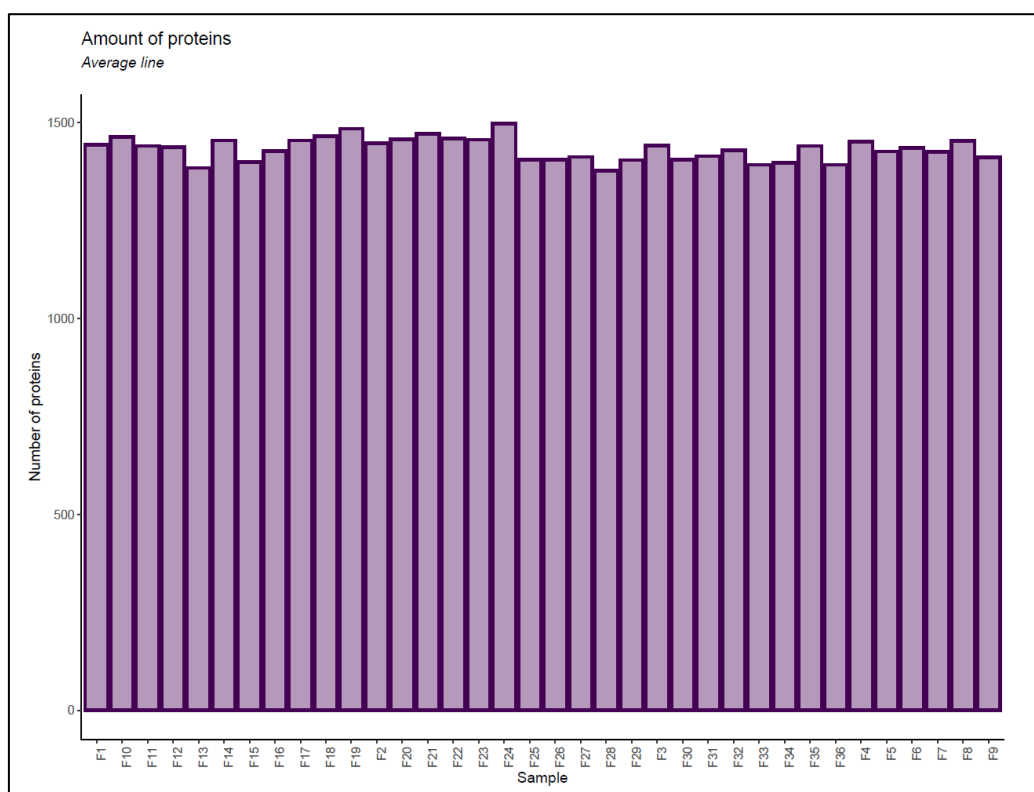
QC metrics

The QC metrics is used to calculate the number of identified proteins and total abundance per sample, among others and can be used to identify and remove low quality cells that may be outliers or technical artifacts, keeping only high quality cells to avoid bias in the downstream analysis. A new pop-up will appear to be able to decide on thresholds:
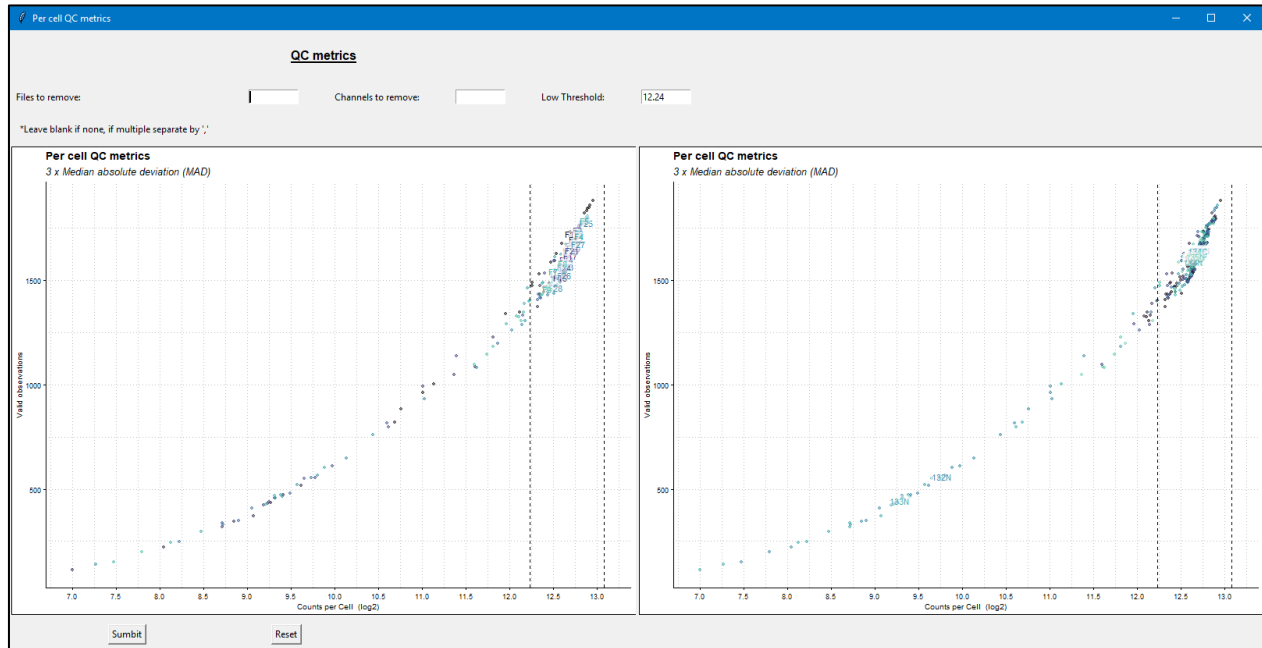
There's three ways of filtering the data, one or multiple can be used:

1) Using File name/TMT: Some files/TMT are low quality and can be removed. Eg. TMT133N and TMT132N in this case contain 10 cells instead of 1. It's easy to spot, while on the MedianCV this difference was not visible.

2) According to the MAD (Median absolute deviation): Set to 3 x from the median as default, but it can be modified in Low Threshold. (This is quite similar to MedianCV, the MAD measures absolute variability, the MCV measures relative variability).

After selecting the Threshold and/or Files, a new output file will be saved with the same figure, named:

1) *4_QCmetrics_**SCP-Experiment**.pdf*

The cells outside of the selected threshold & files won't be used for the follow-up analysis. The data will first be normalized and log scaled to proceed further on.

Now the Peptides are grouped into Proteins. Using the median on the peptides to determine the intensity of the protein.

Missingness %

For each protein, a computation of the percentage of missingness is performed. To be able to determine the percentage of allowed missingness, there's first a computation of the booster and then one of the single cells:
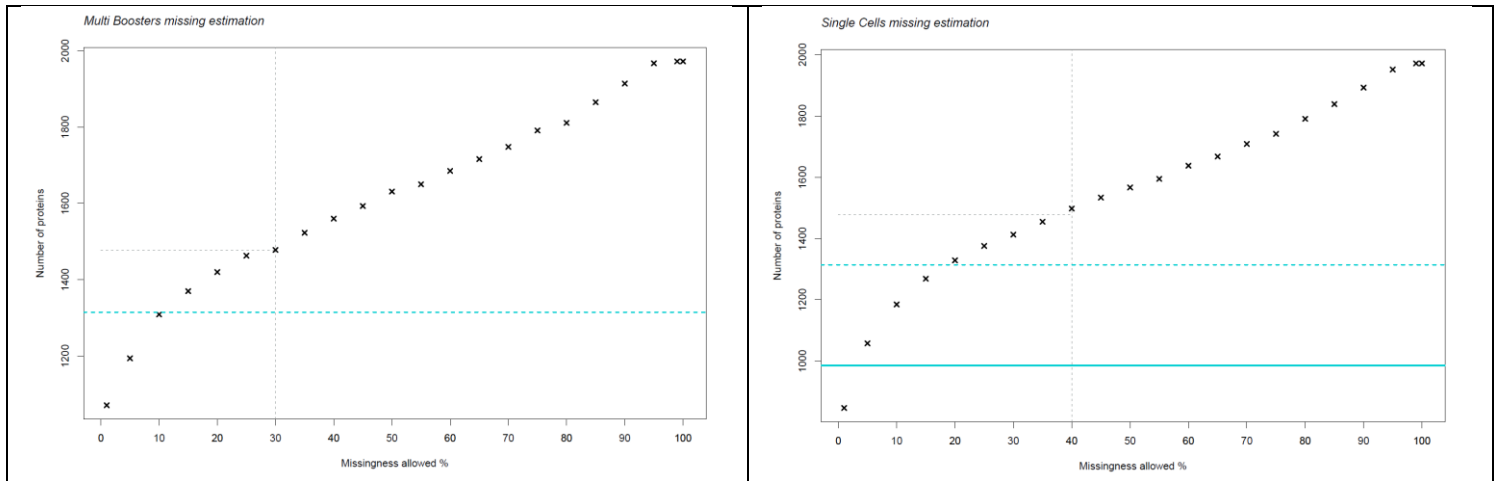
The graphics represent the amount of proteins remaining according to the missing proteins percentage. The number of proteins remaining with a 30% missingness in the booster channels, in this case corresponds to a 40% missingness in the single cells. These two graphics are saved with the filenames:

1) 5_Missingness_Boosters_**SCP-Experiment**.pdf
2) 5_Missingness_SC_**SCP-Experiment**.pdf

If when determining the parameters, the type of missingness filtering is set to booster, the threshold will be automatically selected from it. On the other hand, if the type of filtering is set to manual, a pop up will appear questioning the percentage of allowed missingness:

Imputation

Once removed the proteins with a higher amount of missingness, the imputation will be performed; this one will be done either by a random sample from the normal distribution or using KNN, according to the determined parameters.

This step will generate one file:

1) *6_Imputation_* *_SCP-Experiment.pdf*

Being *: KNN / Rnorm

The file contains several graphics describing the data and the imputation performed.
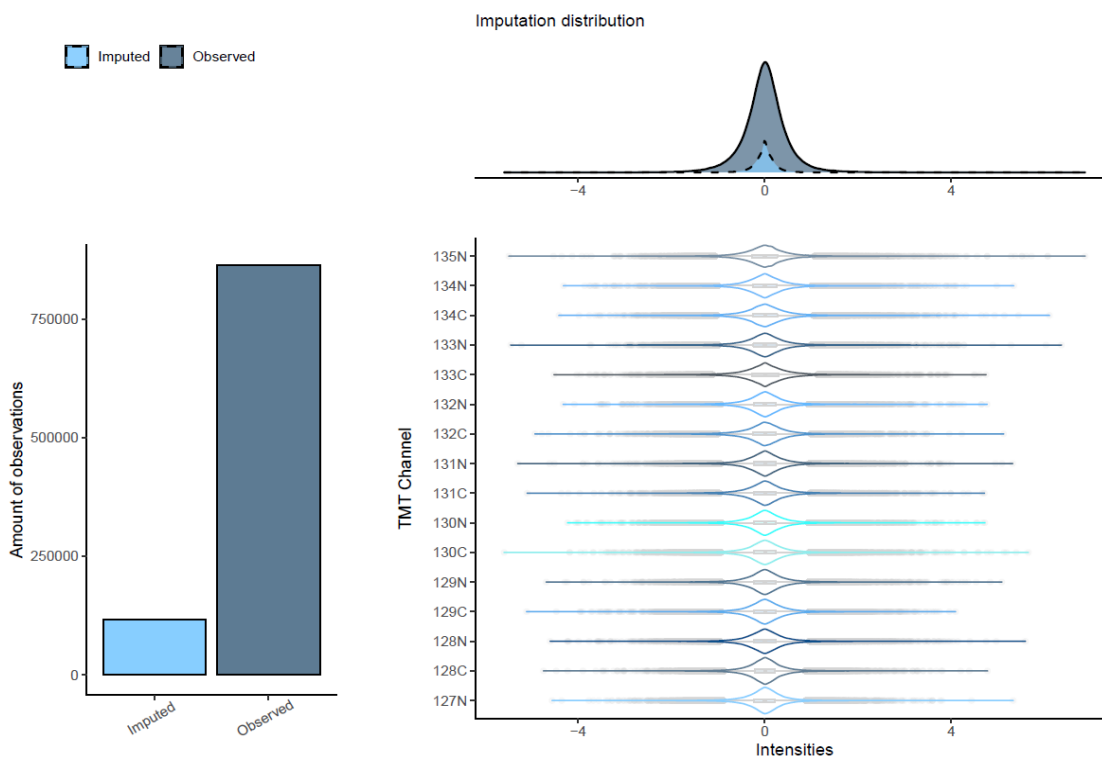
Batch correction

A batch correction is performed according to the chosen group at the first parameters setup. This batch correction is performed using the function *removeBatchEffect*, from the *limma* package *[limma version 3.52.2]*.

This step will generate two or three files, depending if there are multiple experiment:

1) *7_Batch_correction_Channels_**SCP-Experiment**.pdf*
2) *7_Batch_correction_Samples_**SCP-Experiment**.pdf*
3) *7_Batch_correction_Experiment_**SCP-Experiment**.pdf***
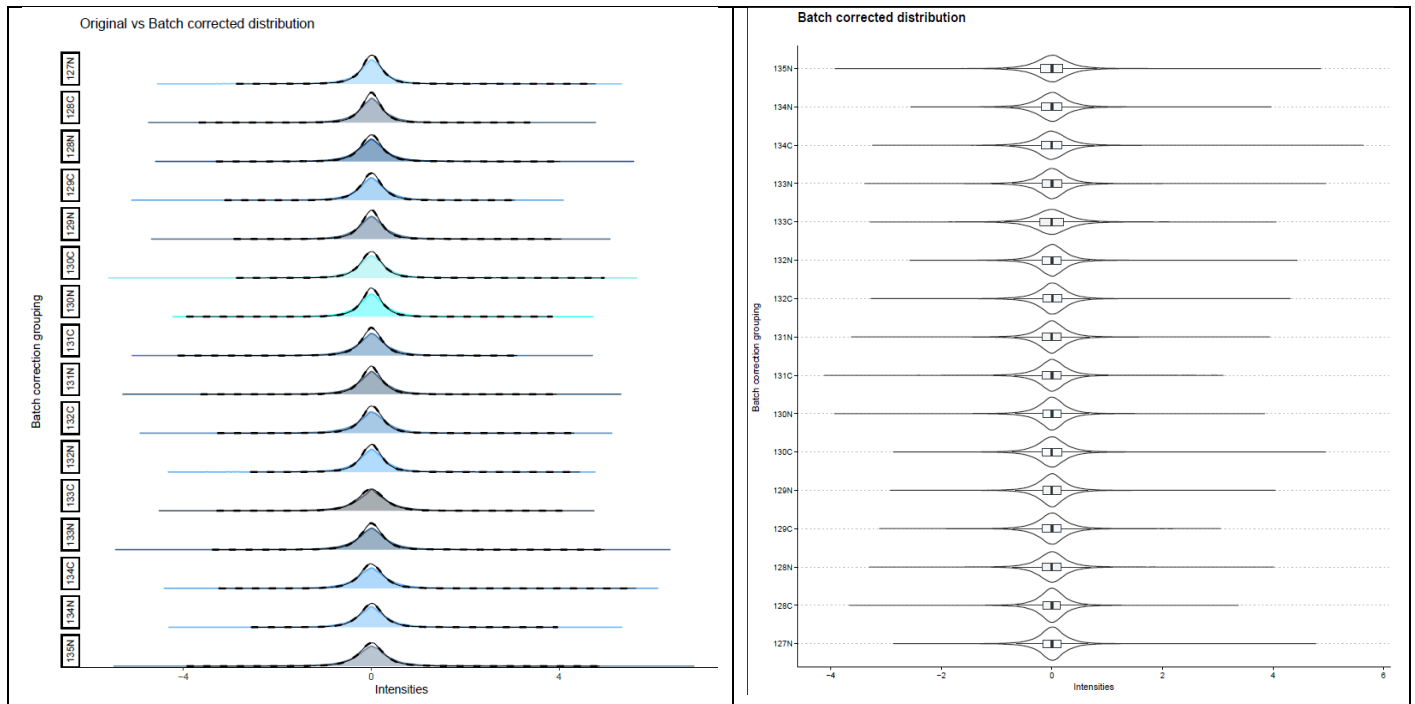
This is the last step of the data analysis, and a report with the parameters used for the experiment is reported. This file is named:

1) *ParametersData_**SCP-Experiment**.txt*

This file contains a list of the parameters used for the analysis, with their values and description. For easier re-construction of the analysis.

Gene names

Finally, Protein accession numbers are translated into gene names. And the files will be stores as:

1) *Proteins_ **SCP-Experiment** _FilteredImputed.txt*
2) *Proteins_ **SCP-Experiment** _FilteredImputed_GeneNames.txt*

DIMENSIONALITY REDUCTION:

After the batch correction, all of the analysis steps are concluded, and the dimensionality reduction is performed.

This step can be done independently, if so, at the beginning it will ask for the *Proteins_ **SCP-Experiment** _FilteredImputed_GeneNames.txt* file & the *""_InputFiles_forSCP.txt.*

PCA

First of all, it will has for the amount of proteins to use for the dimensionality reductions. For that it will organize the proteins by its variance among the cells:

By default, all the proteins are selected.

Then, PCA analysis will start. This step will generate several files and plots.
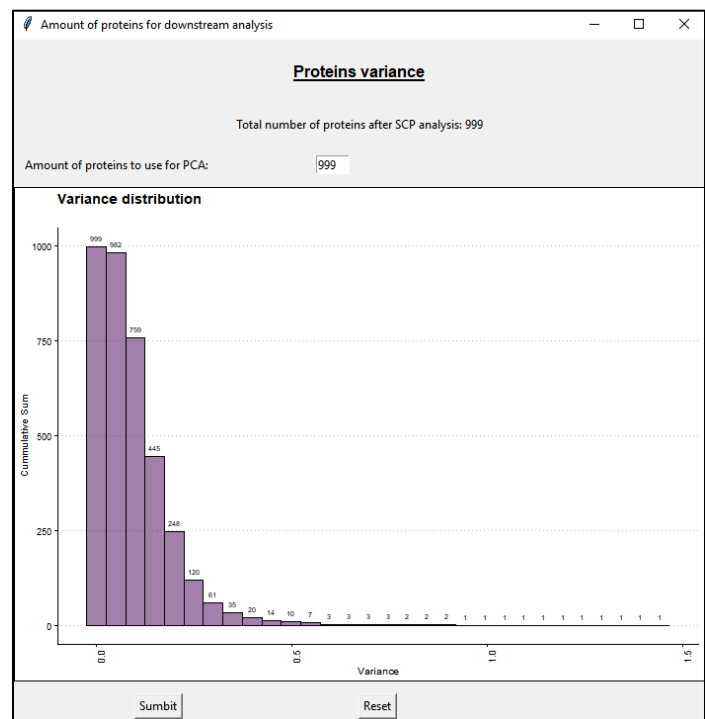
UMAP

To perform the UMAP, first it will compute several UMAPS with the range of parameters selected, and then it will ask to manually check them and finally determine which is the best combination. The parameters to determine are: The number of dimensions of the PCA and the neighbours to use.
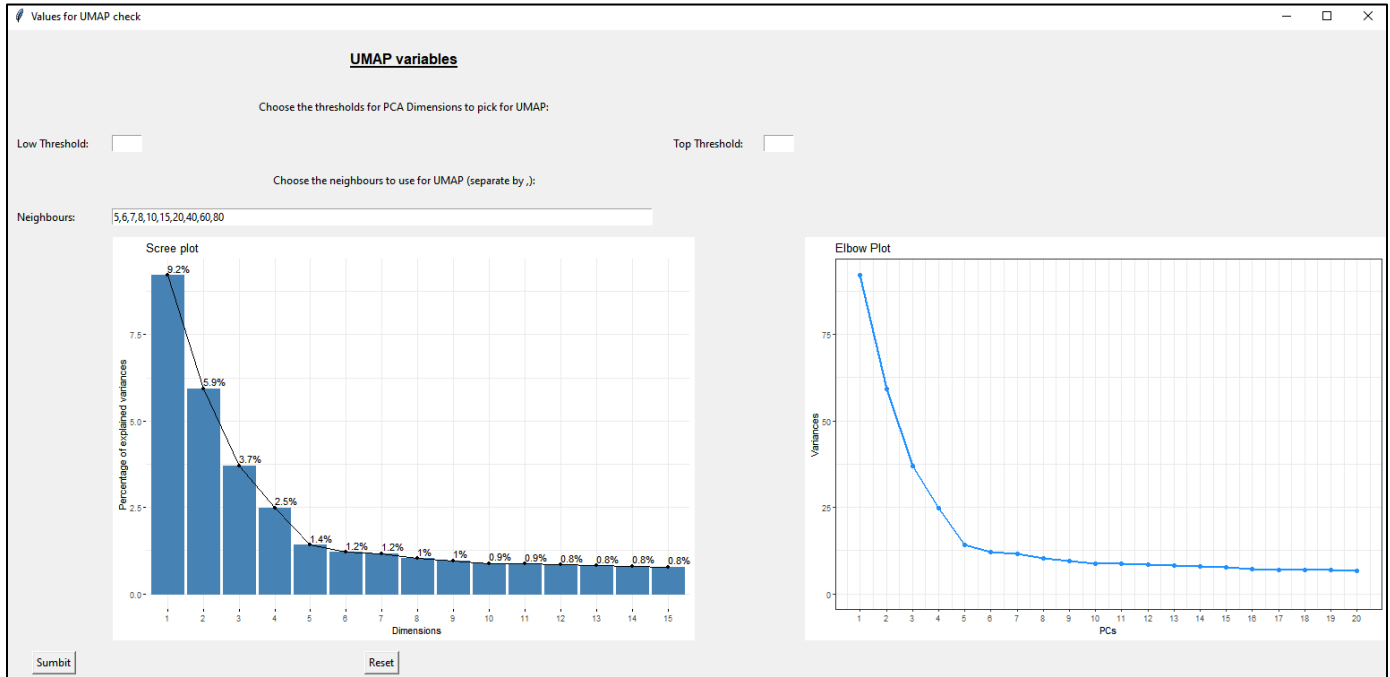
There are two plots to be able to visualize the variance explained by the dimensions, which can be used to select the lower and upper threshold of dimensions to check. As well as the amount of neighbours, (by default there's some already determined).

This will take a couple of minutes depend of the amount of UMAPs to compute. After this step it will ask to carefully look at the different UMAP option and select the final number of dimensions and the final number of neighbours:

After that, the analysis will be done and the final UMAP plot will be saved.

Supplementary data and references:

1) R session Information:

Session info ....

```
R version 4.3.3 (2024-02-29 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 11 x64 (build 22631)
RStudio 2024.09.0
```

Packages Versions ....

attached base packages:
```
 [1] tcltk    stats4    stats    graphics    grDevices    utils    datasets    methods    base
```

other attached packages:
```
 [1] ggthemes_5.1.0             reshape2_1.4.4              DEP_1.24.0
 [4] plotly_4.10.4             ggrepel_0.9.5              proDA_1.16.0
 [7] tidyr_1.3.1               magrittr_2.0.3             RColorBrewer_1.1-3
[10] ggpubr_0.6.0             seqinr_4.2-36             viridis_0.6.5
[13] viridisLite_0.4.2        ggbreak_0.1.2             scater_1.30.1
[16] dplyr_1.1.4              svDialogs_1.1.0           expss_0.11.6
[19] maditr_0.8.4            randomcoloR_1.1.0.1        GGally_2.2.1
[22] ggplot2_3.5.0           scuttle_1.12.0            SingleCellExperiment_1.24.0
[25] limma_3.58.1            scp_1.12.0                QFeatures_1.12.0
[28] MultiAssayExperiment_1.28.0 SummarizedExperiment_1.32.0 Biobase_2.62.0
[31] GenomicRanges_1.54.1    GenomeInfoDb_1.38.8        IRanges_2.36.0
[34] S4Vectors_0.40.2        BiocGenerics_0.48.1        MatrixGenerics_1.14.0
[37] matrixStats_1.2.0       devtools_2.4.5            usethis_2.2.3
```

---

2) Vanderaa Christophe and Laurent Gatto. The current state of single-cell proteomics data analysis. arXiv:2210.01020; DOI: https://doi.org/10.48550/arXiv.2210.01020 (2022).

3) All packages:

Packages --

loaded via a namespace (and not attached):
  [1] later_1.3.2   norm_1.0-11.1   bitops_1.0-7
  [4] ggplotify_0.1.2   tibble_3.2.1   preprocessCore_1.64.0
  [7] XML_3.99-0.16.1   lifecycle_1.0.4   rstatix_0.7.2
 [10] doParallel_1.0.17   lattice_0.22-5   MASS_7.3-60.0.1
 [13] backports_1.4.1   remotes_2.5.0   httpuv_1.6.14
 [16] sessioninfo_1.2.2   pkgbuild_1.4.4   cowplot_1.1.3
 [19] MsCoreUtils_1.14.1   ade4_1.7-22   abind_1.4-5
 [22] pkgload_1.3.4   zlibbioc_1.48.2   Rtsne_0.17
 [25] purrr_1.0.2   AnnotationFilter_1.26.0   RCurl_1.98-1.14
 [28] yulab.utils_0.1.4   sandwich_3.1-0   circlize_0.4.16
 [31] GenomeInfoDbData_1.2.11   irlba_2.3.5.1   MSnbase_2.28.1
 [34] svGUI_1.0.1   DelayedMatrixStats_1.24.0 ncdf4_1.22
 [37] codetools_0.2-19   DelayedArray_0.28.0   DT_0.32
 [40] gmm_1.8   shape_1.4.6.1   tidyselect_1.2.1
 [43] aplot_0.2.2   farver_2.1.1   ScaledMatrix_1.10.0
 [46] jsonlite_1.8.8   GetoptLong_1.0.5   BiocNeighbors_1.20.2
 [49] ellipsis_0.3.2   iterators_1.0.14   foreach_1.5.2
 [52] tools_4.3.3   Rcpp_1.0.12   glue_1.7.0
 [55] BiocBaseUtils_1.4.0   gridExtra_2.3   SparseArray_1.2.4
 [58] xfun_0.42   shinydashboard_0.7.2   withr_3.0.1
 [61] BiocManager_1.30.25   fastmap_1.1.1   fansi_1.0.6
 [64] digest_0.6.35   rsvd_1.0.5   R6_2.5.1
 [67] mime_0.12   gridGraphics_0.5-1   imputeLCMD_2.1
 [70] colorspace_2.1-0   utf8_1.2.4   generics_0.1.3
 [73] data.table_1.15.2   httr_1.4.7   htmlwidgets_1.6.4
 [76] S4Arrays_1.2.1   ggstats_0.5.1   pkgconfig_2.0.3
 [79] gtable_0.3.4   impute_1.76.0   ComplexHeatmap_2.18.0
 [82] XVector_0.42.0   htmltools_0.5.7   carData_3.0-5
 [85] profvis_0.3.8   MALDIquant_1.22.2   ProtGenerics_1.34.0
 [88] clue_0.3-65   scales_1.3.0   tmvtnorm_1.6
 [91] png_0.1-8   ggfun_0.1.4   knitr_1.45

```
 [94] rstudioapi_0.15.0          tzdb_0.4.0              rjson_0.2.21
 [97] checkmate_2.3.1            curl_5.2.1              zoo_1.8-12
[100] GlobalOptions_0.1.2        cachem_1.0.8            stringr_1.5.1
[103] parallel_4.3.3             miniUI_0.1.1.1          vipor_0.4.7
[106] mzID_1.40.0                vsn_3.70.0              pillar_1.9.0
[109] grid_4.3.3                 vctrs_0.6.5             pcaMethods_1.94.0
[112] urlchecker_1.0.1           promises_1.2.1          BiocSingular_1.18.0
[115] car_3.1-2                  beachmat_2.18.1         xtable_1.8-4
[118] cluster_2.1.6              beeswarm_0.4.0          htmlTable_2.4.2
[121] readr_2.1.5                mvtnorm_1.2-4           cli_3.6.2
[124] compiler_4.3.3            rlang_1.1.3              crayon_1.5.3
[127] ggsignif_0.6.4            labeling_0.4.3          affy_1.80.0
[130] plyr_1.8.9                fs_1.6.3                ggbeeswarm_0.7.2
[133] stringi_1.8.3            BiocParallel_1.36.0     assertthat_0.2.1
[136] munsell_0.5.0            lazyeval_0.2.2          V8_4.4.2
[139] Matrix_1.6-5             hms_1.1.3               patchwork_1.2.0
[142] sparseMatrixStats_1.14.0 statmod_1.5.0           shiny_1.8.0
[145] mzR_2.36.0               igraph_2.0.3            broom_1.0.5
[148] memoise_2.0.1            affyio_1.72.0
```