prolfqua: A Comprehensive R-package for Proteomics Differential Expression Analysis

Witold E. Wolski [†][‡][*] Paolo Nanni[*] Jonas Grossmann[*][‡] Maria d'Errico[*][‡]
Ralph Schlapbach[*] Christian Panse[*][‡]

2022-06-07 17:05:53

Abstract

Mass spectrometry is widely used for quantitative proteomics studies, relative protein quantification, and differential expression analysis of proteins. Nevertheless, there is a need for a flexible and easy-to-use application programming interface in R that transparently supports a variety of well principled statistical procedures. The *prolfqua* package can model simple experimental designs with a single explanatory variable and complex experiments with multiple factors and hypothesis testing. It integrates essential steps of the mass spectrometry-based differential expression analysis workflow: quality control, data normalization, protein aggregation, statistical modeling, hypothesis testing, and sample size estimation. The application programmer interface strives to be clear, predictable, discoverable, and consistent to make proteomics data analysis easy and exciting. Furthermore, the package implements benchmark functionality that can help to compare data acquisition, data preprocessing, or data modeling methods using a gold standard dataset. Finally, we show that the implemented methods allow sensitive and specific differential expression analysis. The *prolfqua* R package is available on GitHub https://github.com/fgcz/prolfqua, distributed under the MIT licence, and runs on all platforms supported by the R free software environment for statistical computing and graphics.

Contents

[*] Functional Genomics Center Zurich (FGCZ) - University of Zurich/ETH Zurich, Winterthurerstrasse 190, CH-8057 Zurich, Switzerland. [‡] Swiss Institute of Bioinformatics (SIB), Quartier Sorge - Batiment Amphipole, 1015 Lausanne, Switzerland.
[†] Correspondence: wew@fgcz.ethz.ch

1 Introduction

> To paraphrase provocatively, 'machine learning is statistics minus any checking of models and assumptions'.
>
> – Brian D. Ripley, useR! 2004, Vienna.

Proteins carry out most crucial functions and give structure to living cells. Hence, measuring changes in protein abundance is the subject of active research (Vidova and Spacil 2017). Bottom-up mass spectrometric methods, where proteins are specifically and reproducibly digested into protein fragments - peptides, are employed to identify and quantify proteins in complex samples containing hundreds to thousand of proteins (Bubis et al. 2017; Veiga Leprevost et al. 2020). The peptides are first separated by their chemical and physical properties using liquid chromatography (LC). Afterward, they are ionised, weighed, identified, and quantified using mass spectrometric techniques, e.g., electro-spray-ionization mass spectrometry (ESI-MS). Finally, peptide identification is achieved by fragmenting and matching the measured fragment masses to theoretical masses computed from known peptide sequences (Eng et al. 2015; Yu, Li, and Yu 2016; Kong et al. 2017). For quantification, intact peptide ions (Yu et al. 2020; Cox and Mann 2008) or products of peptide ion fragmentation (Röst et al. 2014; Demichev et al. 2020) are counted and aggregated to obtain peptide abundances. Finally, the identified and quantified peptides are assigned to proteins based on protein sequence information.

Proteomics quantification experiments enable monitoring relative abundances of thousands of proteins in biological samples. Most studies use parallel-group designs, where one or many treatment groups are compared to the control group (Leeuw et al. 2022; Laubscher et al. 2021). More recently, more complex experimental designs with an increasing number of samples are studied, e.g., factorial designs and longitudinal studies (time series), sometimes with repeated measurements on the same subject (Tan et al. 2022; Meier-Abt et al. 2021). The data can be modeled using linear fixed-, mixed-, or random-effects models (Bates et al. 2015). Based on these models, tests can be applied to examine whether specific factors and factor interactions are significant, e.g., it can be tested if differences in protein abundance between groups are statistically significant.

An important aspect of mass spectrometric data are missing peptide and protein quantifications. Rubin (1976) classified missing data problems into three categories: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). For instance, in data-dependent acquisition (DDA) MS, only a limited number of MS1 signals are selected for fragmentation and identified. Peptide quantification algorithms transfer identification information between MS1 features in different samples by aligning the data using retention time and mass information, thus reducing the amount of missing data. Nevertheless, highly abundant proteins can suppress the detection of other proteins in a sample, a MAR mechanism. Furthermore, a weak correlation between the number of missing measurements in a group and the abundance of the quantified protein can be observed, which is caused by the limit of detection (LOD), a MNAR mechanism (McGurk et al. 2020).

Several data analysis packages exist to model mass spectrometry protein quantification experiments, e.g., *limma* (Ritchie et al. 2015), *MSstats* (Choi et al. 2014), *PECA* (Suomi and Elo 2017), *msqrob2* (Goeminne, Gevaert, and Clement 2016) or *proDA* (Ahlmann-Eltze and Anders 2020), to mention some, all implemented in R. Each of them has some unique features, for example, *MSstats* infers the model and generates the model formula from the sample annotation, allowing users with limited statistical knowledge to perform differential expression analysis (DEA). At the same time, *limma* allows to specify a design matrix using a linear model formula and implements the empirical Bayes variance shrinkage method. The package *PECA*, performs a roll-up of peptide level differences and peptide level p-value estimates, obtained from *limma* or *PECA*, to protein level estimates. Furthermore, *msqrob2* implements two models: robust linear models fitted to protein intensities and robust ridge regression fitted to peptide intensities, and combines them with empirical Bayes variance shrinkage. The *proDA* package implements a linear probabilistic dropout model to account for missing data without imputation. *MSstats* handles missing data by feature filtering and using imputation. Other means of modelling missing observations are the Hurdle model discussed by Goeminne et al. (2020), while the R package *proDA* models missingness using probabilistic dropout models (Ahlmann-Eltze and Anders 2020).

Table 1: Rows - R package, Columns - types of models supported: pd - parallel design, rm - repeated measurements, fd - factorial design, int - interactions among factors, mem - mixed effect models, eb - empirical Bayes, md - missing data modelling. Y - yes, * - repeated measurements are modeled using a fixed effect. ? - the hurdle model was published but is not available in the msqrob2 package.

	pd	rm	mem	fd	int	eb	md
MSstats	Y	Y	Y	NA	NA	NA	NA
ROPECA	Y	Y*	NA	Y	Y	Y	NA
limma	Y	Y*	NA	Y	Y	Y	NA
MSqRob2_rlm	Y	Y*	NA	Y	Y	Y	?
proDA	Y	Y*	NA	Y	Y	Y	Y
prolfqua	Y	Y	Y	Y	Y	Y	Y

When analyzing parallel-group designs using a single explanatory variable, and contrasting groups, we can use all R packages; but, we can use only some of them if we want to model factorial designs or repeated measurements. Table 1 gives an overview of the models and features supported by these packages. We see that packages such as *limma* and *proDA* allow us to fit a comprehensive variety of models and test various hypotheses; however, in-depth knowledge of design matrix specification using the R formula interface is required (Law et al. 2020).

When developing the R package *prolfqua* we were inspired by the R package *caret* (Kuhn 2008), which enables us to call various machine learning (ML) methods, and makes selecting the best ML algorithm for your problem easy. We aimed for a similar R package for the differential expression analysis of proteomics data. However, after examining the R packages for modeling proteomics quantification data, we observed that model specification, input, and output formats differ widely. At the same time, they have in common that: they fit linear models either to peptide or protein intensities, determine differences among groups, and afterward apply empirical Bayes variance shrinkage. Therefore, the revised goal was to provide a modular object-oriented (OO) design, with R linear models as a core, where we can add features such as p-value aggregation, e.g., *PECA*, variance shrinkage, or modeling of missing observations.

Furthermore, the functionality of *prolfqua* also includes methods specific to proteomics data. For example, we implemented strategies to estimate protein intensities from peptide intensities: top N (Grossmann et al. 2010), Tukey's median polish (Tukey and others 1977), robust linear models (Goeminne et al. 2020). Furthermore, peptide or protein abundances can be scaled and transformed, using robust scaling, *quantile* normalization or *vsn* to remove systematic differences among samples and heteroscedasticity. In this respect, *prolfqua* can be compared with other R packages such as *DEP* (Zhang et al. 2018) or *POMA* (Castellano-Escuder, Andrés-Lacueva, and Sánchez-Pla 2021) which support the entire differential expression analysis pipeline.

We also implemented functionality and use the Ionstar (Shen et al. 2018) dataset to benchmark the modeling methods implemented within *prolfqua* and to compare our results with those of *MSstats* and *proDA*. Since group sizes are relatively small, typically with four or five subjects per group, the high power of the tests is a relevant criterion to assess the modeling method. The quantified proteins can be ranked using the estimated fold-change, t-statistics, or scaled p-value, and afterward subjected to gene set enrichment (GSEA) or over-representation analysis (Subramanian et al. 2005) to determine up or down-regulated groups of proteins. Furthermore, the statistical model must provide an unbiased estimate of the false discovery rate (FDR) to manage expectations when selecting protein lists for follow-up experiments. We will use the partial area under the receiver operator curve (ROC) to assess the power of the tests and compare the FDR with the false discovery proportion (FDP).

2 Methods

2.1 Implementation

We store all the data needed for analysis in a tidy table, i.e., every column is a variable, every row is an observation, and every cell is a single value (Wickham 2014). Using an R6 (Chang 2020) configuration object (Figure 1), we specify what variable is in which column, making it easy to integrate new inputs in *prolfqua* if provided as a tidy table. For example, to visualize tidy Spectronaut(Bruderer et al. 2015), DiaNN(Demichev et al. 2020), or Skyline(MacLean et al. 2010) outputs, or data in *MSstats*(Choi et al. 2014) format, only a few lines of code are needed to update the *prolfqua* `AnalysisTableConfiguration` configuration. The configuration encapsulates the differences in column names among the various input formats, and enables the use of the methods without additional arguments. We show an example code for creating an MSFragger(Yu et al. 2020) configuration in the Appendix. For popular software like MaxQuant(Cox and Mann 2008), or MSFragger, which stores the same variable, e.g., intensity, in multiple columns, one for each sample, we implemented methods that transform the data into tidy tables. Relying on the tidy data table enables us to interface with many data manipulation, visualization, and modeling methods, implemented in base R (R Core Team 2021) and the tidyverse (Wickham et al. 2019), easily. We use R6 classes to structure the functionality of the package (see Figure 1 and Figure 2). R6 classes are well supported by command-line completion features in RStudio (RStudio Team 2022), and help to implement argument free functions (Figure 6.

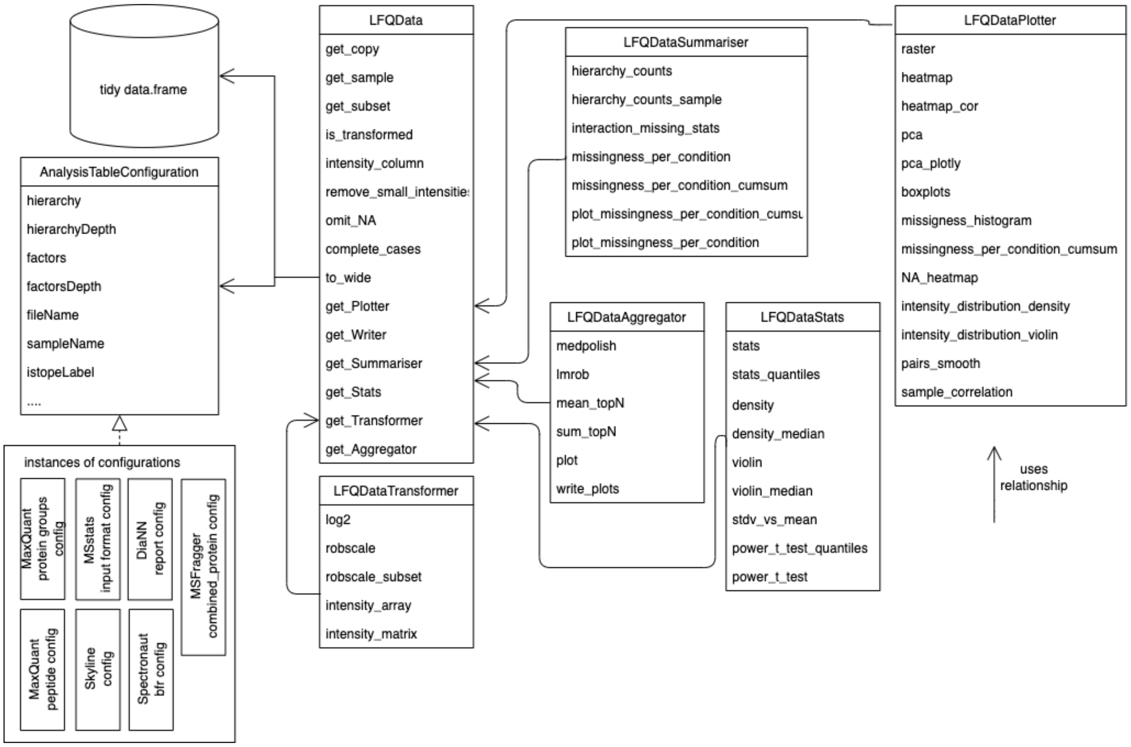


Figure 1: Class Diagram of classes representing the proteomics data. The LFQData class encapsulates the quantitative proteomics data stored in a tidy table. An instance of the AnalysisTableConfiguration class specifies a mapping of columns in the tidy table. The LFQDataPlotter class and other classes decorate the LFQData class with additional functionality. Similarly, the `LFQDataStats` and `LFQDataSummary` reference the `LFQData` class and group methods for variance and sample size estimation or summarizing peptide and protein counts.

In addition, we implement features specific to high throughput experiments, such as the experimental Bayes variance and p-value moderation, which utilizes the parallel structure of the protein measurements and the analysis (Ritchie et al. 2015). We also compute probabilities of differential protein regulation based on peptide level models (Suomi and Elo 2017). We used R6 classes to encapsulate the statistical modelling functionality in *prolfqua* (see Figure 2). We specify a contrast interface (`ContrastsInterface`). Several implementations allow to perform differential expression analysis given linear or mixed effect models (`Contrasts`), variance shrinkage (`ContrastsModerated`), or to impute contrasts in cases when observations are missing for an entire group of samples (`ContrastsSimpleImpute`). Further implementations of the interface encapsulate and integrate differential expression analysis results of external tools such as *proDA* or of SAINTexpress (Teo et al. 2014) used to analyze data from protein interaction studies.

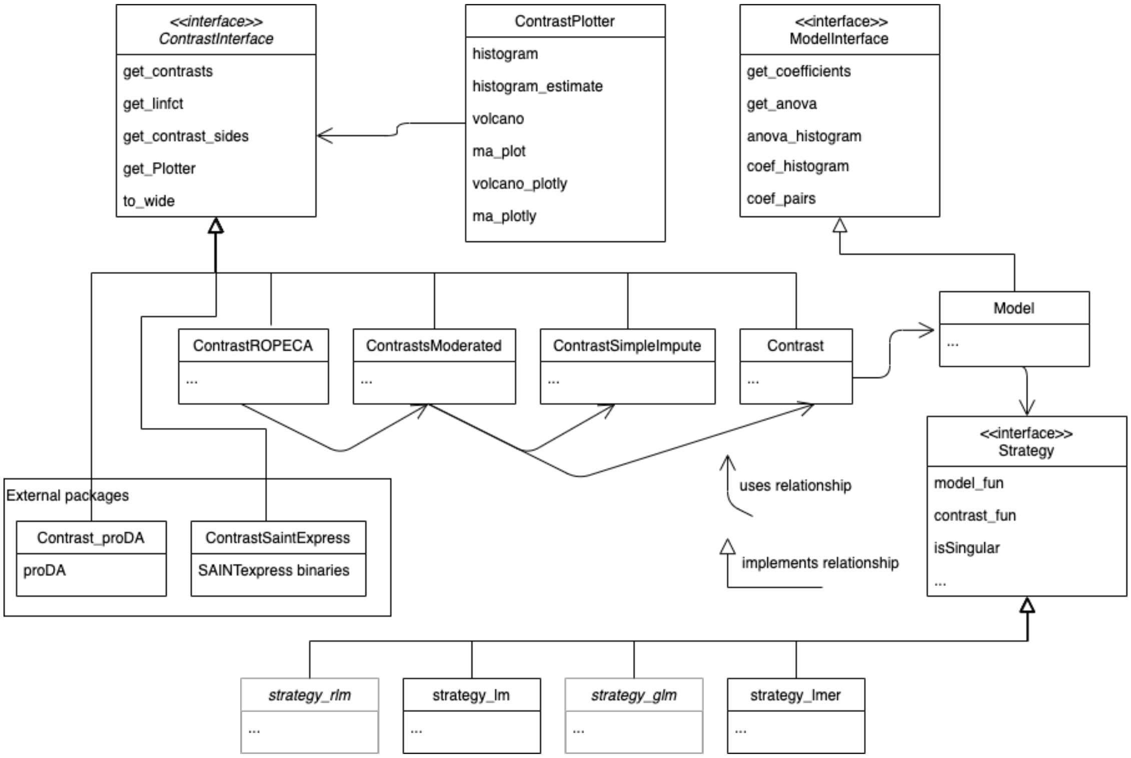


Figure 2: The UML (unified modeling language) diagram of modeling and contrast related classes. Different strategies, e.g., *lm*, *lmer*, and *glm* (Table 3), can fit various types of models. The model builder method fits the statistical model given the data and a strategy. The obtained model can then be used for the analysis of variance or to estimate contrasts. All classes estimating contrasts implement the *Contrasts* interface. Results of external tools, e.g., SAINTexpress, can be adapted by implementing the Contrasts Interface.

2.2 Dataset for benchmarking

To evaluate the performance of differential expression analysis, we use the IonStar benchmark dataset(Shen et al. 2018), available from the Proteomics Identifications Database (PRIDE) identifier PXD003881. $DH5\alpha$ *E. coli* lysate was spiked in human pancreatic cancer cells (Panc-1) lysate at five different levels: 3% *E. coli*, 4.5% *E. coli*, 6% *E. coli*, 7.5% *E. coli* and 9% *E. coli*. We annotated these dilutions from smallest to largest with the letters a, b, c, d, e. By comparing the various dilutions, we obtain different effect sizes, e.g., when comparing dilution e (9%) against dilution d (7.5%), the expected difference is 1.2 for E. coli proteins and 1 for human proteins. There are four technical replicates for each dilution, and hence 20 raw files in total. We processed the raw data using MaxQuant (Cox and Mann 2008) version Version 1.6.10.43, with MaxQuant default settings. The text files generated by MaxQuant are available in the *prolfquadata* R package (Wolski 2021). To compare the performance of the various methods implemented in *prolfqua* we use only the contrasts resulting in minor differences $\Delta = (1.20, 1.25, 1.30, 1.50)$, because for bigger differences all models perform similarly.

Table 2: Confusion matrix, TP - true positive, FP - false positive, FN - false negative, TN - true negatives, P - all positive cases (all *E. coli* proteins), N - all negative cases (all *H. sapiens* proteins), m - all proteins.

Prediction \ Truth	E.coli	H.sapiens	Total
beta != 0	TP	FP	R
beta == 0	FN	TN	
Total	P	N	m

2.3 Data preprocessing for model comparison

The peptide abundances (from the MaxQuant *peptide.txt* file) were log_2 transformed and subsequently scaled, where median and the mean average deviation (mad) were obtained from the human proteins only. We removed *one hit wonders*, i.e., proteins with a single peptide assignment. Protein abundances are inferred from the peptide intensities using Tukey's median polish. Finally, we fitted the fixed effect linear models, the dropout model *proDA* to protein abundances, the mixed effect linear model, and the *PECA* model to peptide intensities.

2.4 Benchmark metrics

The IonStar dataset contains *H. sapiens* proteins with constant concentrations and *E. coli* proteins with varying concentrations. We know that for *H. sapiens* proteins, the difference β between two dilutions should be $\beta = 0$, while for *E. coli* proteins, we know that the difference between dilutions should be $\beta \neq 0$.

We can use various statistics to examine the alternative hypothesis $\beta \neq 0$: the contrast estimate, i.e. the $\log_2$ fold-change β, the t-statistic $\frac{\beta}{\sqrt{var(\beta)}}$, or the p-value and moderated p-value. For each statistic and each value of the statistics we then compute a confusion matrix (see Table 2). From the confusion matrix we obtain measures such as true positive rate (TPR), false positive rate (FPR), or false discovery proportion (FDP) which are given by:

with

$$\begin{aligned} TPR &= \frac{TP}{TP+FN} = \frac{TP}{P} && (1)\\ FPR &= \frac{FP}{FP+TN} = \frac{FP}{N} && (2)\\ FDP &= \frac{FP}{TP+FP} = \frac{FP}{R} && (3) \end{aligned}$$

By plotting the TPR versus the FPR we obtain the receiver operator characteristic curve (ROC curve). The area under the curve (AUC) or partial areas under the curve (pAUC), at various values of the FPR, are measures of performance derived from the ROC curve. By using these measures we can compare the performances of the statistics produced by the various methods examined.

In order to compute the confusion matrices based on the p-value we first need to rescale it (see Equation (11)). Thus, the p-value is a function of the t-statistics and the degrees of freedom.

2.5 Modelling

2.5.1 Robust scaling of the data

Välikangas, Suomi, and Elo (2016) discuss and benchmark various methods of peptide or protein intensity normalization, such as variance stabilizing normalization (Huber et al. 2002) or quantile normalization (Bolstad et al. 2003). In this work, we use a robust version of the z-score, where instead of the mean $\bar{x}$ we use the median, and instead of the standard deviation $\tilde{S}$ we use the median absolute deviation (mad):

$$z = \frac{x - \tilde{x}}{\tilde{S}} \tag{4}$$

Because we need to estimate the differences among groups on the original scale, we must multiply the z-score by the average variance of all the n samples in the experiment.

$$z' = z \cdot \frac{1}{n} \sum_{i=1}^{n} S_i \tag{5}$$

To apply this transformation, we need to estimate two parameters per sample, therefore it works for experiments with thousands of proteins and experiments where only a few hundred proteins per sample are measured. For the Ionstar dataset, we used the intensities of *H. sapiens* proteins, whose concentrations do not change, to determine $\bar{x}$ and S and then applied it to all the intensities (including *E. coli*) in the sample.

2.5.2 Estimating differences between groups

Given a linear model $y = \beta X$, we can compute the difference β_c between two groups by a linear combination c of linear model parameters β, where c is a column vector with as many elements as there are coefficients β in the linear model. If c has 0 for one or more of its rows, then the corresponding coefficient in β is not involved in determining the contrast (Irizarry and Love 2018).

The difference estimate β_c, is given by the dot product:

$$\hat{\beta}_c \quad = \quad c^T \beta \tag{6}$$

and the variance of β_c by:

$$\text{var}(\hat{\beta}_c) \quad = \quad \sqrt{c^T \sigma^2 (X^T X)^{-1} c} \tag{7}$$

with X being the design matrix. The degrees of freedom for the contrast are equal to the residual degrees of freedom of the linear model. For estimating contrasts from mixed effects models we used the function `contest` implemented in the R package *lmerTest* and used the Satterthwaite (Kuznetsova, Brockhoff, and Christensen 2017) method to estimate the denominator degrees of freedom. These methods are available in the class `Contrast` (see Figure 2)

2.5.3 Determining linear parameter combinations for treatment comparison

The package *prolfqua* provides a function to determine the vector of *parameter* weights c, from a linear models and a contrast specification string.

The linear model below explain the observed protein abundances using the explanatory variables `factor_1` and `factor_2` and the interaction among them `factor_1:factor_2`,

```
## Intensity ~ factor_1 + factor_2 + factor_1:factor_2
```

then the contrasts among `group_1` and `group_2` defined by `factor_1` can be specified using the string below.

```
c("contrast_name" = "factor_1group_1 - factor_1group_2")
```

Furthermore, contrasts for subgroups can be specified using the code below,

```
c("contrast_name" =
    "`factor_1group_1:factor_2group_a` - `factor_1group_1:factor_2group_b`")
```

where `factor_x` is the name of the explanatory variable, and `group_x` are group labels.

The following code shows an example where we specify two contrast: the first to compare Cells of type CMP/MEP with cells of type HSC, and the second to compare therapy NO with therapy NU for the celltype CMP/MEP (Meier-Abt et al. 2021). Finally, we compute the array of weights defining the contrast, used to multiply the model coefficient β.

```
Contrasts <-
    c("CMP/MEPvsHSC" = "`CelltypeCMP/MEP` - `CelltypeHSC`",
   "NOvsHU" = "`class_therapyc.NO:CelltypeCMP/MEP` - `class_therapyp.HU:CelltypeCMP/MEP`")

m <- prolfqua::prolfqua_data('data_basicModel_p1807')
linfct <- prolfqua::linfct_from_model(m,as_list = FALSE)

prolfqua::linfct_matrix_contrasts(linfct, Contrasts )
```

```
##              (Intercept) class_therapyc.NO class_therapyp.NO
## CMP/MEPvsHSC           0                 0                 0
## NOvsHU                 0                 1                 0
##              CelltypeHSC
## CMP/MEPvsHSC          -1
## NOvsHU                 0
```

2.5.4 Contrast estimation in presence of missing data.

Missing observations lead to different group sizes, which results in unbalanced designs. Linear and mixed effect models can handle unbalanced designs, as long as one observation per group is available, they will produce unbiased estimates, and no imputation is needed.

If there is no observation in a group, we assume that the protein is unobserved because the protein abundance is below the limit of detection (LOD) of the MS instrument. In this case, we will impute the mean using the protein abundance at the detection limit A_{LOD}. We estimate the abundance at the detection limit using the abundances of the proteins observed only in one sample of a group of samples. Then, we compute the median of the distribution and use it as the group mean if a protein is absent in a treatment group.

When computing differences Δ among two groups a and b, we will use the group mean $\bar{a}$ or $\bar{b}$ estimated from the data, or if no observations are present in a group, we use A_{LOD}, e.g., If there are no observations in group b then :

$$\Delta = \begin{cases} \bar{a} - A_{LOD} & \text{if } \ \bar{a} > A_{LOD} \\ 0 & \text{if } \ \bar{a} < A_{LOD}. \end{cases}$$

To estimate the variance, we assume that the variance of the protein is constant in all the groups and use the pooled variance based on all data:

$$s_p^2 = \frac{\sum_{i=1}^{k}(n_i - 1)s_i^2}{\sum_{i=1}^{k}(n_i - 1)} \tag{8}$$

with n_i the number of observations, and s_i the standard deviation in each group. The standard deviation for the t-statistics is then given by:

$$s = \sqrt{\frac{2n_g s_p^2}{n}}, \tag{9}$$

Where, n_g is the number of groups and n is the number of observations. If variance can not be estimated for a protein, we will use the median pooled variance of all the proteins.

This methods are implemented in the class `ContrastSimpleImpute` (see Figure 2).

2.5.5 p-value moderation

From the linear and the mixed effect models, we can obtain the residual standard deviation σ, and degrees of freedom df. Smyth (2004) discuss how, to use the σ and df of all models to estimate a prior σ and prior df, and posterior $\tilde{\sigma}$. These can be used to moderate the t-statistics by:

$$\tilde{t}_{pj} = \frac{t_{pj} s_p}{\tilde{s}_p}. \tag{10}$$

We implemented this method in the class `ContrastModerated` (Figure 2).

2.5.6 Summarizing peptide level differences and p-values on protein level

To summarize peptide level models to protein models, we apply the method suggested by Suomi and Elo (2017) that use the median scaled p-value of the peptide models and the cumulative distribution function of the Beta distribution (CDF) to determine a regulation probability of the protein.

To obtain the $\tilde{p}$ of a protein, we first rescaled the peptide p-values by taking the sign of the fold-change $\hat{\beta}$ into account, i.e.:

$$p_s = \begin{cases} 1 - p, & \text{if } \hat{\beta} > 0 \\ p - 1, & \text{otherwise} \end{cases} \tag{11}$$

Afterwards, the median scaled p-value $\tilde{p}_s$ is determined and using the transformation below, transformed back onto the original scale:

$$\tilde{p} = 1 - |\tilde{p}_s| \tag{12}$$

Because we use the median, with the i-th order statistic $i = \frac{n}{2} + 0.5$, we parametrize the CDF of the Beta distribution with $\gamma = i = \frac{n}{2} + 0.5$ and $\delta = n - i + 1 = n - (\frac{n}{2} + 0.5) + 1 = \frac{n}{2} + 0.5 = \gamma$. We implemented this method in the class `ContrastROPECA` (Figure 2).

3 Results and Discussion.

3.1 Example analysis workflow

The code snippets in this section demonstrate how a differential expression analysis workflow can be implemented using the *prolfqua* R package. To speed up the computation of these examples, we use a random subset of the Ionstar dataset containing 163 proteins and 1258 peptides. Peptide abundances are $\log_2$ transformed and robust z-score scaled using the method `robscale`. Using the `LFQDataPlotter` class, we show the distribution of the normalized peptide abundances in Figure 3 Panel A. Afterwards, protein intensities are estimated from peptide intensities using Tukey's median polish. Figure 3 Panel B shows the peptide intensities and the estimated protein intensities. Next, we compute the standard deviation of all the proteins in each group and display their distribution using violin plots (Panel C). Finally, we create a boxplot (Panel D) showing the abundance of one protein.

```r
## load peptide level data
d <- prolfqua::prolfqua_data('data_ionstar')$filtered()

## create R6 obejct
lfqd <- prolfqua::LFQData$new(d$data, d$config)

##  transform intensities
t <- lfqd$get_Transformer()

lfqd <-  t$log2()$robscale()$lfq
lfqd$rename_response("peptide_abundance")

## infer protein intensities from peptide intensity
agr <- lfqd$get_Aggregator()
lfqp <- agr$medpolish()
lfqp$rename_response("protein_abundance")

## plot panels A-D
pl <- lfqd$get_Plotter()
panelA <- pl$intensity_distribution_density() +
  ggplot2::labs(tag = "A") + ggplot2::theme(legend.position = "none")
panelB <- agr$plot()$plots[[1]] + ggplot2::labs(tag = "B")
panelC <- lfqp$get_Stats()$violin() + ggplot2::labs(tag = "C")
pl <- lfqp$get_Plotter()
panelD <- pl$boxplots()$boxplot[[1]] + ggplot2::labs(tag = "D")
ggpubr::ggarrange(panelA, panelB, panelC, panelD)
```

The following code example illustrates how we compute differences among groups. First, the linear model and the differences are specified. Afterward, the model is fitted to the data using the `build_model` function. Next, we estimate the contrasts either from the linear model using the `Contrasts` class or directly from the data using the `ContrastsSimpleImpute` class. Afterward, we apply *t*-statistic moderation using the `ContrastModerated` class. Finally, the `addContrastResults` function merges both sets of contrast estimates, preferring the one obtained from the linear model if both are available. Then we create the plots shown in Figure 4. Panel A shows the distribution of the *p*-values, while Panel B shows the volcano plot for each comparison.

```r
# specify differences among groups
contrasts <- c(
  "dilution_(9/7.5)_1.2" =   "dilution.e - dilution.d",
  "dilution_(7.5/6)_1.25" =   "dilution.d - dilution.c"
)
## fit model
lmmodel <- paste(lfqp$intensity_column()," ~ dilution.")
modelFunction <- prolfqua::strategy_lm( lmmodel, model_name = "lm")
models <- prolfqua::build_model(lfqp, modelFunction)

## compute contrasts from linear model, moderate t-statistics and p-values
contr <- prolfqua::Contrasts$new(models, contrasts) |>
  prolfqua::ContrastsModerated$new()

# compute contrasts using imputation imputation and moderate
conI <- prolfqua::ContrastsSimpleImpute$new( lfqp, contrasts) |>
  prolfqua::ContrastsModerated$new()
```

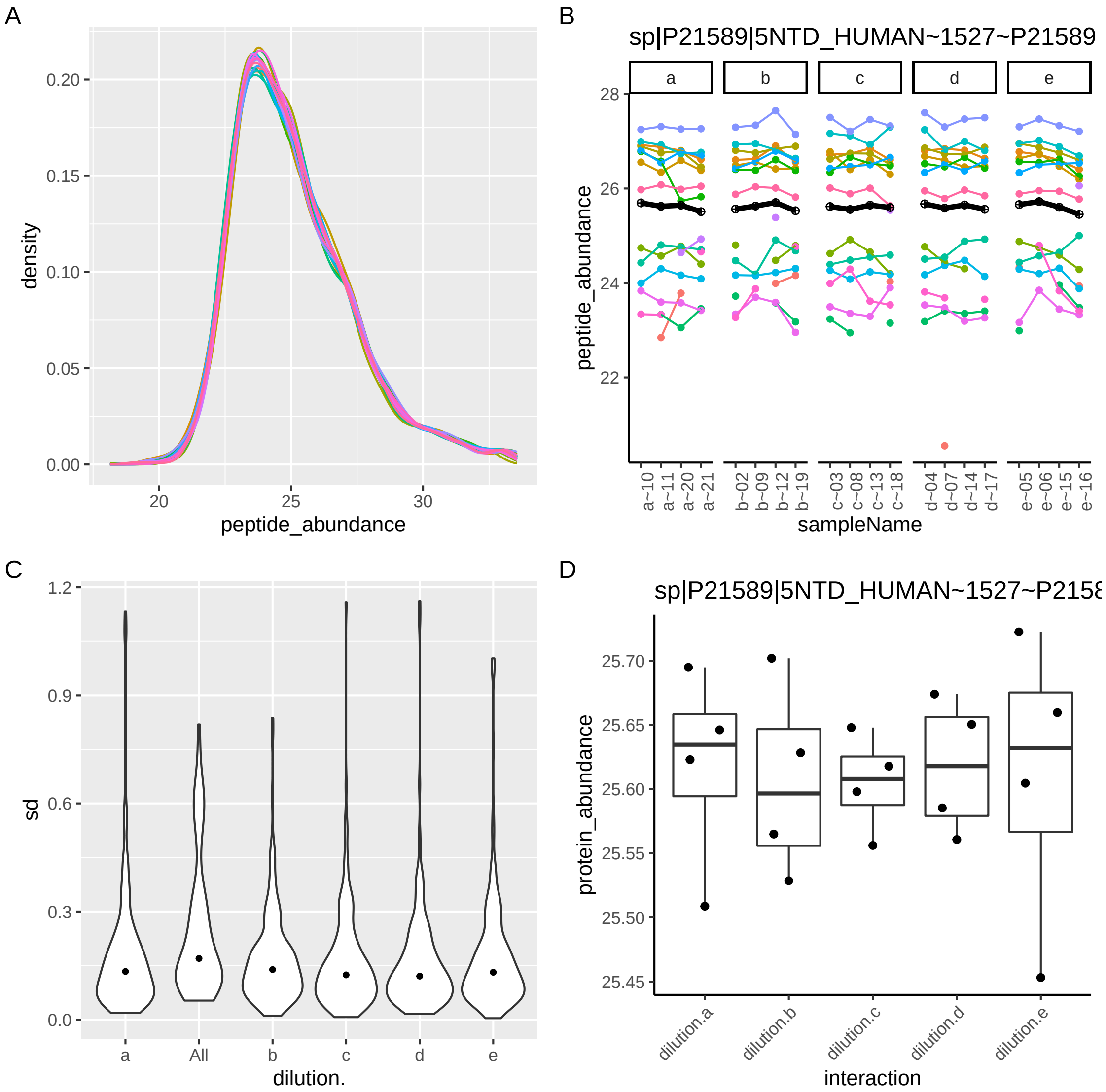


Figure 3: Panel A - Peptide intensity distributions for 20 samples. For each sample a line with a different colour is shown. Panel B - Peptide intensities for protein *5NTD* are shown using a line of different colour, and the protein intensity estimate is shown using a fat black line, Panel C - distribution of standard deviations of all proteins in each dilution group (a, b, c, d, e) and overall (all), Panel D - Distribution of protein intensities for protein *5NTD*.

```
# merge contrasts, to obtain differences for all proteins
contrasts <- prolfqua::addContrastResults(prefer = contr, add = conI)

## visualize results in panels
pl <- contrasts$merged$get_Plotter()
panelA <- pl$histogram()$p.value + ggplot2::labs(tag = "A")
panelB <- pl$volcano()$FDR +
   ggplot2::theme(legend.position = "bottom") +
   ggplot2::labs(tag = "B")


gridExtra::grid.arrange(panelA, panelB, ncol = 2)
```

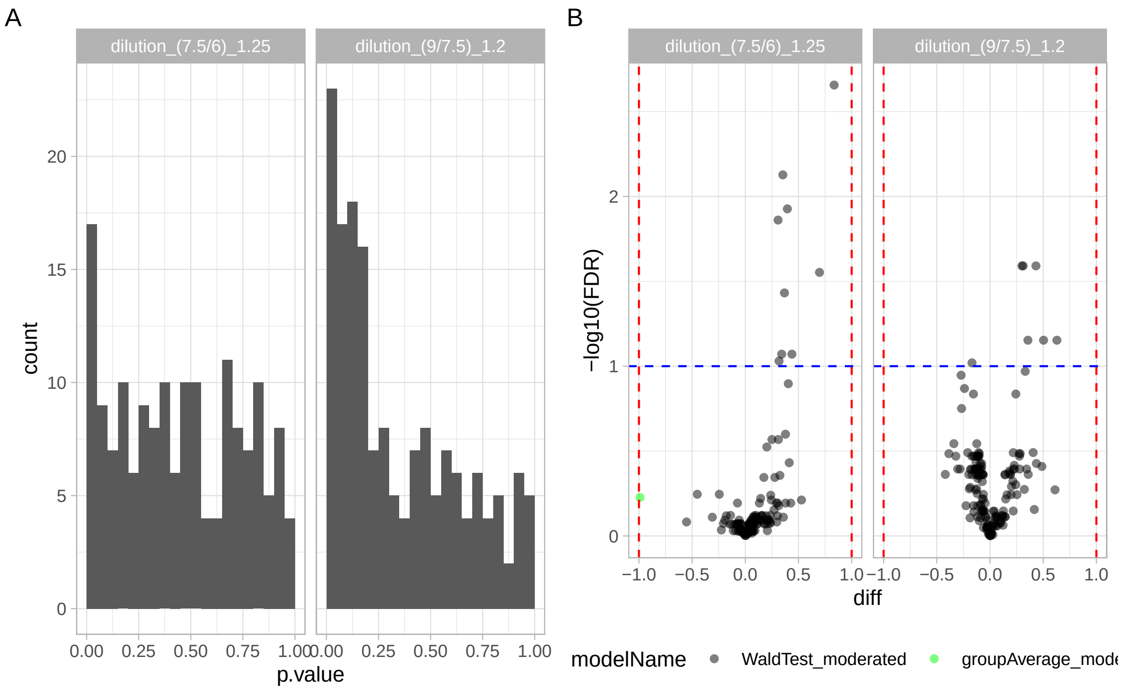


Figure 4: Panel A - Histogram showing the distribution of p-values for 163 proteins. Panel B - Volcano plot showing $-\log_{10}$ transformed FDR as function of the difference between groups for 163 proteins.

R linear model and linear mixed effect models allow modeling parallel designs, repeated measurements, factorial designs, and many more. Models in *prolfqua* are specified using R's linear model and mixed model formula interface. Therefore, knowledge of the R regression model infrastructure (Faraway 2016; Venables and Ripley 2002) is advantageous when using our package. Furthermore, this glass box approach should make it easy to reimplement an analysis performed with *prolfqua* using base R or other programming languages by reading the analysis script. However, acknowledging the R formula interface's complexity to non-statisticians and the popularity of *MSstats*, we provide the functionality to suggest a model formula from the sample annotation provided in tabular form similarly to *MSstats*.

Using the tidy table to model the data ensures interoperability with other proteomics-related packages that manage their data with tidy-tables, e.g., *protti* (Quast, Schuster, and Picotti 2022). The use of R6 classes, which encapsulate the configuration and the data, allows for writing very concise code (no function arguments needed). Auto-completion support in the Rstudio editor makes it easy for novices to explore *prolfqua*s functionality (Figure 6). To simplify integration of *prolfqua* with Bioconductor based workflows there is a method to convert the LFQData class into a *SummarizedExperiment*.

To ease the usage barriers of the R package to users not being familiar with statistics and R programming, we integrated some workflows into our data management platform b-fabric (Türker et al. 2010). This integration enables our users to select the input and basic settings in a graphical user interface. Then, as an output, the user receives a report including input files, the R markdown file, and R scripts necessary to replicate the analysis using their in-house R installation. The b-fabric system runs a computing infrastructure controlled by a local resource management system that supports cloud-bursting (Aleksiev et al. 2013). In this way, b-fabric helps scientists to meet requirements from funding agencies, journals, and academic institutions to publish data according to the FAIR (Findable, Accessible, Interoperable and Reusable) (Wilkinson et al. 2016) data principles.

3.2 Benchmarking modelling approaches

The Benchmark functionality of *prolfqua* includes receiver operator curves (ROC) and computes partial areas under those curves (pAUC) and other scores, e.g., the false discovery proportion FDP. We use those scores, i.e. the $pAUC$ at 10 FDR and the FDP, to examine how well the methods implemented in *prolfqua* model quantitative mass spectrometric high throughput data and compare them with results produced by *MSstats* and *proDA*. Table 4 summarizes all modelling methods we evaluated.

When comparing differential expression analysis performance, a relevant parameter is the number of proteins for which a method estimated differences (see Figure 5 Panel A), which indicates how robust the procedure works in the presence of missing observations. For each protein, we tried to determine four differences ($\Delta = (1.20, 1.25, 1.30, 1.50)$). Given 4046 proteins with at least two peptides, there are in total 16184 possible differences. However, some methods can not estimate all of them. The set of proteins with effect size estimates might differ for each method, biasing direct comparison of scores such as pAUC. For instance, we observe that *MSstats* estimates 16058 group differences while the mixed effect models estimates the fewest with 15940. Hence, to conclude that one method shows a better performance, it does not suffice if the pAUC is greater, but the number of proteins with differential expression results needs to be equal or larger.

Figure 5 Panel B shows how various estimates obtained from the models, i.e., the difference between groups, t-statistics, and scaled p-values allow identifying true positives (TP) given a false positive rate (FPR) of 10, by displaying the partial area under the ROC. Ordering proteins by the t-statistic or p-value leads to a higher $pAUC_{10}$ than when ordering by the estimated difference among groups. Suppose an accurate estimate of the difference among groups is essential. In that case, the linear models fitted to protein intensities, calculated using Tukey's median polish, perform better than those directly modeling peptide intensities, e.g., `ropeca` or `prot_mixed_effect_moderated` (see Table 4 Abundance column). We speculate that outliers at the peptide level do not affect the protein estimates when using Tukey's median polish method, a non-parametric method to infer protein abundances. This hypothesis could be examined, by including other forms of protein intensity inferences implemented in `prolfqua`, e.g. `top-N` or `rlm`, into the benchmark.

There are only minor differences in the $pAUC_{10}$ between the t-statistics or the scaled p-value (see Figure 5 Panel A). Interestingly, the $pAUC$ increases when using p-values instead of the t-statistics for linear models, while it decreases for mixed effect models. The reason is an erroneous denominator degrees of freedom estimation for many proteins in the case of the mixed effect models.

We also benchmark if the FDR obtained from a model is an unbiased estimate of the false discovery proportion FDP. Figure 5 Panel C draws on the horizontal axis the FDR determined from the model, and on the vertical axis, the FDP obtained from the confusion matrix. Most lines are below the diagonal, indicating that the FDR estimates are modestly conservative for this benchmark dataset. In the case of *MSstats*, we observe a high proportion of false discoveries for small FDR values. In the case of *PECA*, the FDR estimates, obtained by Benjamini-Hochberg correcting the regulation probabilities, strongly overestimate the FDP.

Using a benchmark dataset whose ground truth is known (see Methods), we assessed the performance of different modeling approaches implemented in *prolfqua* (Tables 3 and 4), *MSstats* and *proDA*. Table 4 summarizes which methods we have evaluated, which MaxQuant results we used, and if the models are fitted to peptide or protein intensities. We make the R-markdown files to replicate the benchmarking available at *prolfquabenchamrk* and at BenchmarkingIonstarData.

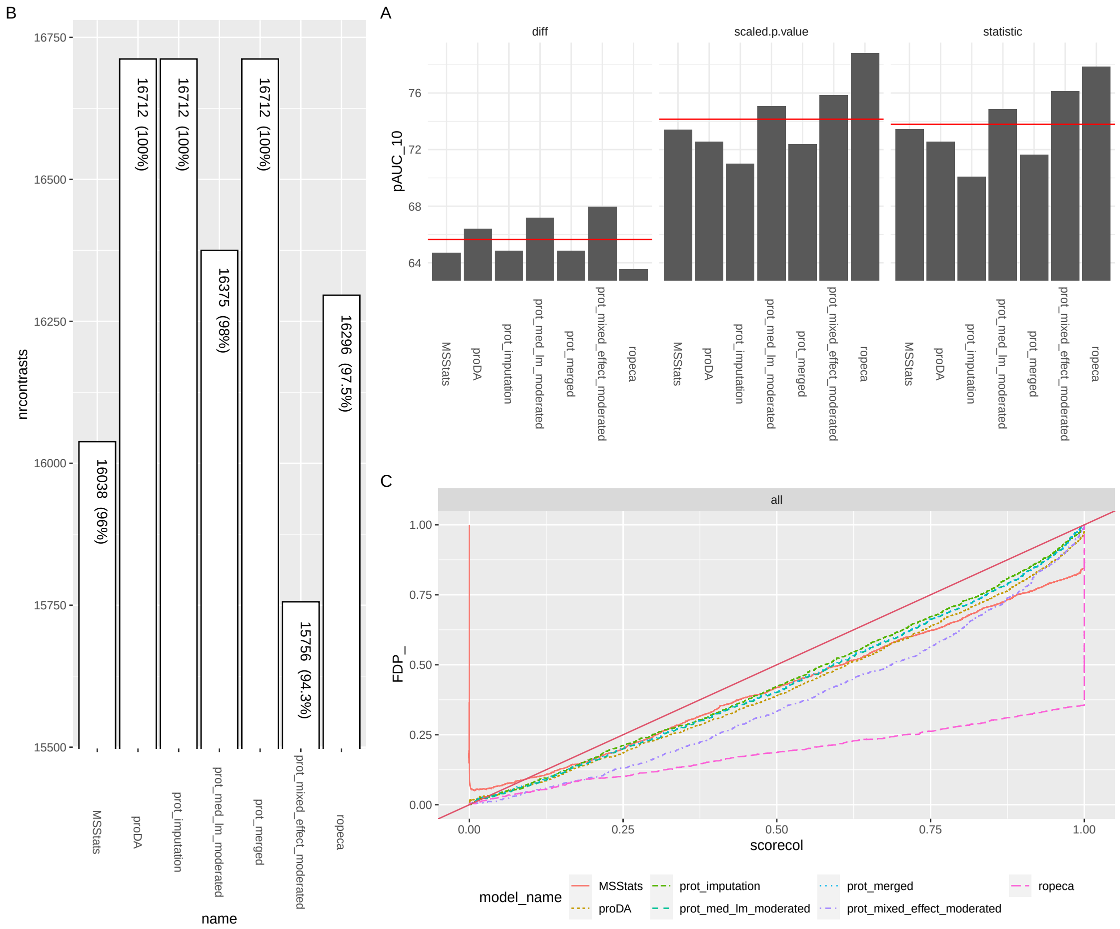


Figure 5: Panel A - Number of estimated contrasts for each modeling method (higher is better). Panel B - Partial area under the ROC curve at 10% FPR ($pAUC_10$) for all contrasts and three different statistics: the difference among groups, the scaled p-value (sign(diff) · p.value) and the t-statistics (higher is better). The red line indicates the average area under the curve of all methods. Panel C - Plots the false discovery proportion (FDP) as a function of the FDR. Ideally, the FDR should be equal to the FDP. Therefore larger distances from the diagonal are worse.

Table 3: Prolfqua functions to fit various models.

prolfqua functions	model
strategy__lm, Contrasts	linear modelling of peptide or protein abundances and group difference estimation
strategy__lmer, Contrasts	mixed effect modelling of peptide or protein abundances and group differences estimation
ContrastsSimpleImpute	group difference estimation when no observations in one of the groups
ContrastsROPECA	estimating group differences for proteins by summarizing peptide differences
ContrastsModerated	empirical Bayes variance shrinkage for group difference estimates (limma)
runSaint, ContrastsSaintExpress	protein interaction scoring (SAINTExpress)
strategy__proDA*, Contrasts__proDA*	adapter to the probabilistic dropout model implemented in proDA

[a] in development

Table 4: All benchmarked models. Description - prolfqua function names, Abudances - indicates if model is fitted to peptide or protein abundances, Input File - name of MaxQuant file used as input.

Label	Description	Abundance	Input File
MSstats	preprocess with default parameters		evidence.txt
proDA	probabilistic dropout model	protein	peptide.txt
prot_imputation	ContrastsSimpleImpute, ContrastsModerated	protein	peptide.txt
prot_med_lm_moderated	strategy_lm, Contrasts, ContrastsModerated	protein	peptide.txt
prot_merged	addContrastResults(prefer = prot_med_lm_moderated , add = prot_imputation) *	protein	peptide.txt
prot_mixed_effect_moderated	strategy_lmer, Contrasts, ContrastsModerated	peptide	peptide.txt
ropeca	strategy_lm, Contrasts, ContrastsModerated, ContrastsROPECA	peptide	peptide.txt

[a] Merges results of the 'prot_med_lm_moderated' and 'prot_imputation' modeling pipeline, preferring those of 'prot_med_lm_moderated' if available.

Since only technical replicates are available for each dilution, essential sources of variation present in every experiment, such as the biochemical and biological, are not measured. Therefore, the dataset captures only the variance from the chromatography, electro-spray, and mass spectrometric measurement method. Thus, while we can extrapolate some of the results obtained to more realistic datasets, we need to be careful not to over-interpret the results. Specifically, the observed variances will be higher in more realistic experiments, and the power will be lower for the same sample sizes. Furthermore, the proportion of missing observations in real-life datasets might also be higher.

We can conclude that if we want to sort the proteins according to the likelihood of being differentially regulated to perform gene set enrichment analysis (Subramanian et al. 2005), the t-statistic is better suited than the fold-change estimate. Modeling the degrees of freedom when computing the p-values might improve the inference. However, this improvement is minuscule (see Figure 5 panel B). There is no such improvement for the mixed effect model, most likely because the degrees of freedom are erroneously estimated for many models. Furthermore, for the fixed effect linear model, the empirical Bayes variance shrinkage, as suggested by Smyth (2004), consistently improves the ranking of proteins compared with the unmoderated estimates (not shown) and fails to do so for mixed effect models.

Computing the statistics at the peptide level, e.g., the t-statistics or p-value, then summarizing these statistics using their median produces the highest AUC scores among all the tested models (see Figure 5 Panel A left). Furthermore, by using the Beta distribution to model the number of peptides observed, we can further improve the $pAUC$ scores (see Figure 5 Panel A center). However, the properties of Beta-based probabilities are not well understood; for instance, the p-values are not uniformly distributed under the null hypothesis (not shown). Furthermore, the FDR estimates obtained when correcting for multiple testing with the Benjamini-Hochberg method are biased and overestimate the false discovery proportion (see Figure 5 Panel C). Therefore, we can not recommend this method if an unbiased estimate of FDR is essential, which is frequently the case. In addition, peptides are stronger affected by missing values, reducing the number of contrasts we could estimate for the dataset using this method (see Figure 5 Panel C).

The probabilistic dropout analysis implemented in the *proDA* produces inferences comparable to those of other methods (Figure 5 Panel A). Because of the robustness of the dropout model to missing observations we obtain difference estimates for all proteins and contrasts (Figure 5 Panel B). Moreover, the estimated differences are unbiased and show high diagnostic accuracy (Figure 5 Panel A). Furthermore, the performance of the scaled p-values or the t-statistics is comparable with that of the linear model with variance moderation (`prot_med_lm_moderated`). Therefore, we are planning to integrate the *proDA* package as an additional modeling option into *prolfqua* (see Figure 2).

The R-package *proDA* and *prolfqua* model the missing data directly, while MSstats imputes the data using an accelerated failure model. Despite imputation, *MSstats* does not estimate group differences for more proteins and does not achieve a higher $pAUC$ score than *prolfqua*. Furthermore, Figure 5 Panel C shows that when using *MSstats* the proportion of false discoveries might be very high even when filtering for a low FDR.

We focused our benchmark on comparing the statistical modeling methods using three different scores, while we fixed the pre-processing steps. However, there are other equally or even more important parameters of a protein quantification pipeline (Fröhlich et al. 2022). One of them is the normalization of the intensities within the samples to remove systematic differences (Pursiheimo et al. 2015). The method used to infer protein intensities from peptide intensities is an additional important factor (Grossmann et al. 2010). For instance, the original *proDA* publication uses MaxLFQ (Cox et al. 2014) protein estimates. However, when using MaxLFQ intensities reported by MaxQuant, the $pAUC_{10}$ is significantly lower ($pAUC_{10}$(t-statistics) $= 66$) compared with results obtained when protein abundances are estimated from peptide abundances using Tukey's median polish ($pAUC_{10}$(t-statistics) $= 73$). Last but not least, the software (Cox and Mann 2008; Yu et al. 2020), identifying proteins and generating the quantification values can also significantly contribute to the performance of the entire pipeline, altering the number of identified proteins and the sensitivity and specificity of the differential expression analysis.

4 Conclusion

prolfqua allows for considerable flexibility to model quantitative proteomics experiments. Various types of models are available (see Figure 2 and Table 3), and the contrast specification is explicit and consistent for all models. The modular design of *prolfqua*, allows for adding new features, e.g., generalized linear models (*glm*'s) to model the presence or absence information of a protein, or robust linear models (*rlm*'s), in the future. R's formula interface for linear models is flexible, widely used, and well documented (Faraway 2016). We use the formula interface to specify the models, making it easy to reproduce an analysis performed with *prolfqua* in other statistical methods programming languages. However, the developed framework is flexible enough to in the future integrate other modeling methods, e.g., the probabilistic dropout model (Ahlmann-Eltze and Anders 2020) or accurate variance estimation (Zhu et al. 2020). Hence, *prolfqua* enables you to call various methods and makes selecting the best differential expression analysis algorithm for your problem easy.

When comparing statistical modeling methods for the differential expression analysis, we assessed performance measures such as the number of estimated contrasts, the $pAUC$, and if the FDR is an unbiased estimate of the FDR. It is relevant that an analysis pipeline shows good performance in all these categories. Leveraging these computer experiments, we can provide the following advice: i) estimate protein abundances from peptide abundances using a robust or nonparametric regression method; ii) use linear models because they show good performance in all categories; iii) if the measurements are correlated, as for technical replicates, mixed effect models might work if the sample sizes are large; if not, aggregate the replicates and fit a linear model instead; iv) if you use fixed-effect linear models, apply variance moderation to improve the t-statistics and p-value estimates; v) If you want to sort your protein lists to perform gene set enrichment analysis, use the t-statistic instead of the difference; vi) do not impute missing observation but statistically model missingness to estimate parameters, i.e., group differences. Finally, the differential expression analysis result obtained with prolfqua are comparable to or better than when using other differential expression analysis tools.

In summary, *prolfqua* is an easy-to-use and feature-rich R package to analyze quantitative mass spectrometric data with simple or complex experimental designs. It also can generate conclusive reports and to benchmark MS software and statistical methods. Furthermore, with minimal adaptations, this R package can analyze different quantitative proteomics data (e.g., labeling-based TMT-, PRM- DIA-data). We provide documentation, in vignette format, at the website https://github.com/fgcz/prolfqua/. This document was created using Rmarkdown. All the code needed to replicate the document or the benchmark results is available at: https://github.com/wolski/prolfquabenchmark.

Acknowledgements

The authors thank the technology platform fund (TPF) of the University of Zurich and all FGCZ proteomics colleagues for fruitful discussions.

Abbreviations

Abbreviations	Explaination
AUC	Area Under the Curve
CDF	Cumulative Distribution Function
ESI-MS	Electro-Spray-Ionization Mass Spectrometry
FDP	False Discovery Proportion
FDR	False Discovery Rate
LC	Liquid Chromatography
LC-MS	Liquid Chromatography followed by Mass Spectrometry
LOD	Limit Of Detection
MAR	Missing At Random
MCAR	Missing Completely At Random
MS	mass spectrometry
OO	Object-Oriented
UML	Unified Modeling Language

References

Ahlmann-Eltze, Constantin, and Simon Anders. 2020. "ProDA: Probabilistic Dropout Analysis for Identifying Differentially Abundant Proteins in Label-Free Mass Spectrometry." *bioRxiv*. https://doi.org/10.1101/661496.

Aleksiev, Tyanko, Simon Barkow-Oesterreicher, Peter Kunszt, Sergio Maffioletti, Riccardo Murri, and Christian Panse. 2013. "VM-MAD: A Cloud/Cluster Software for Service-Oriented Academic Environments." In *Lecture Notes in Computer Science*, 447–61. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-38750-0_34.

Bates, Douglas, Martin Mächler, Ben Bolker, and Steve Walker. 2015. "Fitting Linear Mixed-Effects Models Using Lme4." *Journal of Statistical Software, Articles* 67 (1): 1–48. https://doi.org/10.18637/jss.v067.i01.

Bolstad, Benjamin M, Rafael A Irizarry, Magnus Åstrand, and Terence P. Speed. 2003. "A Comparison of Normalization Methods for High Density Oligonucleotide Array Data Based on Variance and Bias." *Bioinformatics* 19 (2): 185–93.

Bruderer, Roland, Oliver M Bernhardt, Tejas Gandhi, Saša M Miladinović, Lin-Yang Cheng, Simon Messner, Tobias Ehrenberger, et al. 2015. "Extending the Limits of Quantitative Proteome Profiling with Data-Independent Acquisition and Application to Acetaminophen-Treated Three-Dimensional Liver Microtissues." *Molecular & Cellular Proteomics* 14 (5): 1400–1410.

Bubis, Julia A., Lev I. Levitsky, Mark V. Ivanov, Irina A. Tarasova, and Mikhail V. Gorshkov. 2017. "Comparative Evaluation of Label-Free Quantification Methods for Shotgun Proteomics." *Rapid Communications in Mass Spectrometry* 31 (7): 606–12. https://doi.org/10.1002/rcm.7829.

Castellano-Escuder, Pol, Cristina Andrés-Lacueva, and Alex Sánchez-Pla. 2021. *POMA: User-Friendly Workflow for Metabolomics and Proteomics Data Analysis*. https://github.com/pcastellanoescuder/POMA.

Chang, Winston. 2020. *R6: Encapsulated Classes with Reference Semantics*. https://CRAN.R-project.org/package=R6.

Choi, Meena, Ching-Yun Chang, Timothy Clough, Daniel Broudy, Trevor Killeen, Brendan MacLean, and Olga Vitek. 2014. "MSstats: An R Package for Statistical Analysis of Quantitative Mass Spectrometry-Based Proteomic Experiments." *Bioinformatics* 30 (17): 2524–6.

Cox, Jürgen, Marco Y Hein, Christian A Luber, Igor Paron, Nagarjuna Nagaraj, and Matthias Mann. 2014. "Accurate Proteome-Wide Label-Free Quantification by Delayed Normalization and Maximal Peptide Ratio Extraction, Termed Maxlfq." *Molecular & Cellular Proteomics* 13 (9): 2513–26.

Cox, Jürgen, and Matthias Mann. 2008. "MaxQuant Enables High Peptide Identification Rates, Individualized P.p.b.-range Mass Accuracies and Proteome-Wide Protein Quantification." *Nature Biotechnology* 26 (12): 1367–72. https://doi.org/10.1038/nbt.1511.

Demichev, Vadim, Christoph B Messner, Spyros I Vernardis, Kathryn S Lilley, and Markus Ralser. 2020. "DIA-Nn: Neural Networks and Interference Correction Enable Deep Proteome Coverage in High Throughput." *Nature Methods* 17 (1): 41–44.

Eng, Jimmy K, Michael R Hoopmann, Tahmina A Jahan, Jarrett D Egertson, William S Noble, and Michael J MacCoss. 2015. "A Deeper Look into Comet—Implementation and Features." *Journal of the American Society for Mass Spectrometry* 26 (11): 1865–74.

Faraway, Julian J. 2016. *Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models*. Chapman; Hall/CRC.

Fröhlich, Klemens, Eva Brombacher, Matthias Fahrner, Daniel Vogele, Lucas Kook, Niko Pinter, Peter Bronsert, et al. 2022. "Benchmarking of Analysis Strategies for Data-Independent Acquisition Proteomics Using a Large-Scale Dataset Comprising Inter-Patient Heterogeneity." *Nature Communications* 13 (1): 1–13.

Goeminne, Ludger JE, Kris Gevaert, and Lieven Clement. 2016. "Peptide-Level Robust Ridge Regression Improves Estimation, Sensitivity, and Specificity in Data-Dependent Quantitative Label-Free Shotgun Proteomics." *Molecular & Cellular Proteomics* 15 (2): 657–68.

Goeminne, Ludger JE, Adriaan Sticker, Lennart Martens, Kris Gevaert, and Lieven Clement. 2020. "MSqRob Takes the Missing Hurdle: Uniting Intensity-and Count-Based Proteomics." *Analytical Chemistry* 92 (9): 6278–87.

Grossmann, Jonas, Bernd Roschitzki, Christian Panse, Claudia Fortes, Simon Barkow-Oesterreicher, Dorothea Rutishauser, and Ralph Schlapbach. 2010. "Implementation and Evaluation of Relative and Absolute Quantification in Shotgun Proteomics with Label-Free Methods." *Journal of Proteomics* 73 (9): 1740–6. https://doi.org/10.1016/j.jprot.2010.05.011.

Huber, Wolfgang, Anja Von Heydebreck, Holger Sültmann, Annemarie Poustka, and Martin Vingron. 2002. "Variance Stabilization Applied to Microarray Data Calibration and to the Quantification of Differential Expression." *Bioinformatics* 18 (suppl_1): S96–S104.

Irizarry, Rafael, and Michael Love. 2018. "PH525x Series - Biomedical Data Science." 2018. http://genomicsclass.github.io/book/pages/interactions_and_contrasts.html.

Kong, Andy T, Felipe V Leprevost, Dmitry M Avtonomov, Dattatreya Mellacheruvu, and Alexey I Nesvizhskii. 2017. "MSFragger: Ultrafast and Comprehensive Peptide Identification in Mass Spectrometry–Based Proteomics." *Nature Methods* 14 (5): 513–20.

Kuhn, Max. 2008. "Building Predictive Models in R Using the Caret Package." *Journal of Statistical Software, Articles* 28 (5): 1–26. https://doi.org/10.18637/jss.v028.i05.

Kuznetsova, Alexandra, Per Brockhoff, and Rune Christensen. 2017. "LmerTest Package: Tests in Linear Mixed Effects Models." *Journal of Statistical Software, Articles* 82 (13): 1–26. https://doi.org/10.18637/jss.v082.i13.

Laubscher, Dominik, Berkley E Gryder, Benjamin D Sunkel, Thorkell Andresson, Marco Wachtel, Sudipto Das, Bernd Roschitzki, et al. 2021. "BAF Complexes Drive Proliferation and Block Myogenic Differentiation in Fusion-Positive Rhabdomyosarcoma." *Nature Communications* 12 (1): 1–16.

Law, Charity W, Kathleen Zeglinski, Xueyi Dong, Monther Alhamdoosh, Gordon K Smyth, and Matthew E Ritchie. 2020. "A Guide to Creating Design Matrices for Gene Expression Experiments." *F1000Research* 9.

Leeuw, Sherida M de, Aron WT Kirschner, Karina Lindner, Ruslan Rust, Vanessa Budny, Witold E Wolski, Anne-Claude Gavin, Roger M Nitsch, and Christian Tackenberg. 2022. "APOE2, E3, and E4 Differentially Modulate Cellular Homeostasis, Cholesterol Metabolism, and Inflammatory Response in Isogenic iPSC-Derived Astrocytes." *Stem Cell Reports* 17 (1): 110–26.

MacLean, Brendan, Daniela M Tomazela, Nicholas Shulman, Matthew Chambers, Gregory L Finney, Barbara Frewen, Randall Kern, David L Tabb, Daniel C Liebler, and Michael J MacCoss. 2010. "Skyline: An Open Source Document Editor for Creating and Analyzing Targeted Proteomics Experiments." *Bioinformatics* 26 (7): 966–68.

McGurk, Kathryn A, Arianna Dagliati, Davide Chiasserini, Dave Lee, Darren Plant, Ivona Baricevic-Jones, Janet Kelsall, et al. 2020. "The Use of Missing Values in Proteomic Data-Independent Acquisition Mass Spectrometry to Enable Disease Activity Discrimination." *Bioinformatics* 36 (7): 2217–23.

Meier-Abt, Fabienne, Witold E Wolski, Ge Tan, Sandra Kummer, Sabine Amon, Markus G Manz, Ruedi Aebersold, and Alexandre Theocharides. 2021. "Reduced Cxcl4/Pf4 Expression as a Driver of Increased Human Hematopoietic Stem and Progenitor Cell Proliferation in Polycythemia Vera." *Blood Cancer Journal* 11 (2): 1–6.

Pursiheimo, Anna, Anni P Vehmas, Saira Afzal, Tomi Suomi, Thaman Chand, Leena Strauss, Matti Poutanen, Anne Rokka, Garry L Corthals, and Laura L Elo. 2015. "Optimization of Statistical Methods Impact on Quantitative Proteomics Data." *Journal of Proteome Research* 14 (10): 4118–26.

Quast, Jan-Philipp, Dina Schuster, and Paola Picotti. 2022. "Protti: An R Package for Comprehensive Data Analysis of Peptide-and Protein-Centric Bottom-up Proteomics Data." *Bioinformatics Advances* 2 (1): vbab041.

R Core Team. 2021. "R: A Language and Environment for Statistical Computing." Vienna, Austria: R Foundation for Statistical Computing. 2021. https://www.R-project.org/.

Ritchie, Matthew E, Belinda Phipson, Di Wu, Yifang Hu, Charity W Law, Wei Shi, and Gordon K Smyth. 2015. "Limma Powers Differential Expression Analyses for Rna-Sequencing and Microarray Studies." *Nucleic Acids Research* 43 (7): e47–e47.

Röst, Hannes L, George Rosenberger, Pedro Navarro, Ludovic Gillet, Saša M Miladinović, Olga T Schubert, Witold Wolski, et al. 2014. "OpenSWATH Enables Automated, Targeted Analysis of Data-Independent Acquisition Ms Data." *Nature Biotechnology* 32 (3): 219–23.

RStudio Team. 2022. *RStudio: Integrated Development Environment for R.* Boston, MA: RStudio, Inc. http://www.rstudio.com/.

Rubin, Donald B. 1976. "Inference and Missing Data." *Biometrika* 63 (3): 581–92.

Shen, Xiaomeng, Shichen Shen, Jun Li, Qiang Hu, Lei Nie, Chengjian Tu, Xue Wang, et al. 2018. "Ion-Star Enables High-Precision, Low-Missing-Data Proteomics Quantification in Large Biological Cohorts." *Proceedings of the National Academy of Sciences* 115 (21): E4767–E4776.

Smyth, Gordon K. 2004. "Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments." *Statistical Applications in Genetics and Molecular Biology* 3 (1): 1–25.

Subramanian, Aravind, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, et al. 2005. "Gene Set Enrichment Analysis: A Knowledge-Based Approach for Interpreting Genome-Wide Expression Profiles." *Proceedings of the National Academy of Sciences* 102 (43): 15545–50.

Suomi, Tomi, and Laura L Elo. 2017. "Enhanced Differential Expression Statistics for Data-Independent Acquisition Proteomics." *Scientific Reports* 7: 5869.

Tan, Ge, Witold E Wolski, Sandra Kummer, Mara Hofstetter, Alexandre PA Theocharides, Markus G Manz, Ruedi Aebersold, and Fabienne Meier-Abt. 2022. "Proteomic Identification of Proliferation and Progression Markers in Human Polycythemia Vera Stem and Progenitor Cells." *Blood Advances*.

Teo, Guoci, Guomin Liu, Jianping Zhang, Alexey I Nesvizhskii, Anne-Claude Gingras, and Hyungwon Choi. 2014. "SAINTexpress: Improvements and Additional Features in Significance Analysis of Interactome Software." *Journal of Proteomics* 100: 37–43.

Tukey, John W, and others. 1977. *Exploratory Data Analysis*. Vol. 2. Reading, MA.

Türker, Can, Fuat Akal, Dieter Joho, Christian Panse, Simon Barkow-Oesterreicher, Hubert Rehrauer, and Ralph Schlapbach. 2010. "B-Fabric: The Swiss Army Knife for Life Sciences." In *Proceedings of the 13th International Conference on Extending Database Technology - EDBT 10*. ACM Press. https://doi.org/10.1145/1739041.1739135.

Välikangas, Tommi, Tomi Suomi, and Laura L Elo. 2016. "A Systematic Evaluation of Normalization Methods in Quantitative Label-Free Proteomics." *Briefings in Bioinformatics* 19 (1): 1–11.

Veiga Leprevost, Felipe da, Sarah E Haynes, Dmitry M Avtonomov, Hui-Yin Chang, Avinash K Shanmugam, Dattatreya Mellacheruvu, Andy T Kong, and Alexey I Nesvizhskii. 2020. "Philosopher: A Versatile Toolkit for Shotgun Proteomics Data Analysis." *Nature Methods* 17 (9): 869–70.

Venables, W. N., and B. D. Ripley. 2002. *Modern Applied Statistics with S*. Fourth. New York: Springer. https://www.stats.ox.ac.uk/pub/MASS4/.

Vidova, Veronika, and Zdenek Spacil. 2017. "A Review on Mass Spectrometry-Based Quantitative Proteomics: Targeted and Data Independent Acquisition." *Analytica Chimica Acta* 964: 7–23.

Wickham, Hadley. 2014. "Tidy Data." *The Journal of Statistical Software* 59 (10). https://doi.org/10.18637/jss.v059.i10.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

Wilkinson, Mark D., Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, et al. 2016. "The FAIR Guiding Principles for Scientific Data Management and Stewardship." *Scientific Data* 3 (1). https://doi.org/10.1038/sdata.2016.18.

Wolski, Witold. 2021. "Prolfquadata." 2021. https://gitlab.bfabric.org/wolski/prolfquadata.

Yu, Fengchao, Sarah E Haynes, Guo Ci Teo, Dmitry M Avtonomov, Daniel A Polasky, and Alexey I Nesvizhskii. 2020. "Fast Quantitative Analysis of timsTOF Pasef Data with Msfragger and Ionquant." *BioRxiv*.

Yu, Fengchao, Ning Li, and Weichuan Yu. 2016. "PIPI: PTM-Invariant Peptide Identification Using Coding Method." *Journal of Proteome Research* 15 (12): 4423–35.

Zhang, Xiaofei, Arne H Smits, Gabrielle BA van Tilburg, Huib Ovaa, Wolfgang Huber, and Michiel Vermeulen. 2018. "Proteome-Wide Identification of Ubiquitin Interactions Using Ubia-Ms." *Nature Protocols* 13: 530–50.

Zhu, Yafeng, Lukas M Orre, Yan Zhou Tran, Georgios Mermelekas, Henrik J Johansson, Alina Malyutina, Simon Anders, and Janne Lehtiö. 2020. "DEqMS: A Method for Accurate Variance Estimation in Differential Protein Expression Analysis." *Molecular & Cellular Proteomics* 19 (6): 1047–57.

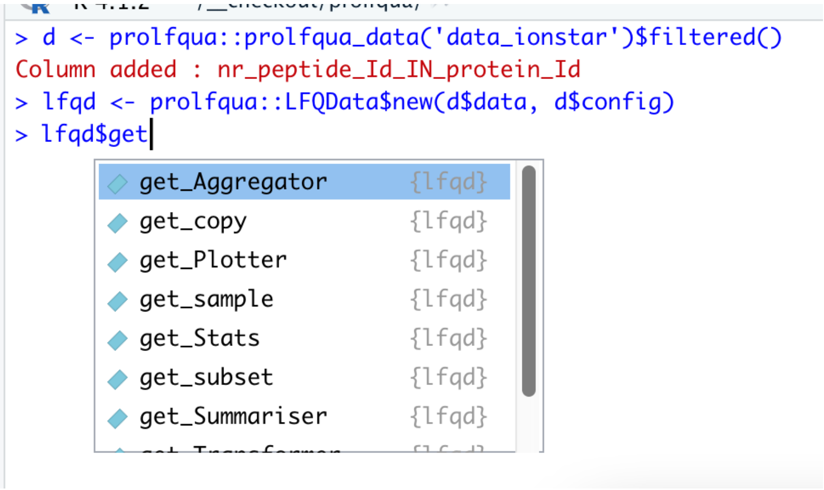
Appendix

Creating a prolfqua configuration

The following code demonstrates how we use *prolfqua* to analyze protein intensities reported in the MSFragger `combined_protein.tsv` file. First, we create a tidy table containing the protein abundances by reading the `combined_protein.tsv` file using `tidy_MSFragger_combined_protein`. Then, we read the sample annotation from the file `annotation.xlsx` file. Next, we create an `AnalysisTableAnnotation` R6 object. Bottom-up proteomics data is hierarchical, i.e., a protein has peptides, peptides might be modified, etc. Therefore, the `AnalysisTableAnnotation` has a `hierarchy` field storing a list with an entry for each hierarchy level. Since `combined_portein.tsv` only holds protein level data, the hierarchy list has one element, and we use it to specify which column contains the protein identifiers. We also need to define which column contains the protein abundances we want to use for the data analysis. Finally, we have to specify which columns contain the explanatory variables of the analysis. The `AnalysisTableAnnotation` has the field `factors`, a list with as many entries as explanatory variables. Here we include two explanatory variables, the dilution, specified in the column 'sample', and 'run' stored in the column 'run_ID', representing the order of the measurement.

```
datadir <- file.path(find.package("prolfquadata") , "quantdata")
inputFragfile <-  file.path(datadir, "MSFragger_IonStar2018_PXD003881.zip")
inputAnnotation <- file.path(datadir, "annotation_Ionstar2018_PXD003881.xlsx")
# read input annotation
annotation <- readxl::read_xlsx(inputAnnotation)

protein <- tibble::as_tibble(
    read.csv(unz(inputFragfile,"IonstarWithMSFragger/combined_protein.tsv"),
             header = TRUE, sep = "\t", stringsAsFactors = FALSE))

# read combined_protein.tsv
protein <- prolfqua::tidy_MSFragger_combined_protein(protein)
# remove proteins identified by a single peptide
protein <- protein |> dplyr::filter(unique.stripped.peptides > 1)

# annotate the data
merged <- dplyr::inner_join(annotation, protein)
atable <- prolfqua::AnalysisTableAnnotation$new()
atable$fileName = "raw.file"
# specify column containing protein identifiers
atable$hierarchy[["protein_Id"]] = "protein"

# column with protein abundances
atable$setWorkIntensity("total.intensity")

# the factors of the analysis
atable$factors[["dilution."]] = "sample"
atable$factors[["run"]] = "run_ID"

config <- prolfqua::AnalysisConfiguration$new(atable)

adata <- prolfqua::setup_analysis(merged, config)
lfqdata <- prolfqua::LFQData$new(adata, config)
# remove small intensities
lfqdata$remove_small_intensities()
```

Miscellaneous



```
> d <- prolfqua::prolfqua_data('data_ionstar')$filtered()
Column added : nr_peptide_Id_IN_protein_Id
> lfqd <- prolfqua::LFQData$new(d$data, d$config)
> lfqd$get
```

Figure 6: The screenshot displays the command-line completion (tab completion) of RStudio on the `prolfqua::LFQData` R6 object. In the example, it shows the getter methods of the object.

```r
file.path(system.file(package = "prolfqua"),
          "Figures/hexStickerProlfqua.png") |>
  knitr::include_graphics()
```

Figure 7: Sticker maintainer: Witold E. Wolski; License: Creative Commons Attribution CC-BY. Feel free to share and adapt, but don't forget to credit the author.

Session information

```
## R version 4.2.0 (2022-04-22)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Debian GNU/Linux 11 (bullseye)
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.13.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods
## [7] base
##
## other attached packages:
## [1] lmerTest_3.1-3 lme4_1.1-29    Matrix_1.4-1
##
## loaded via a namespace (and not attached):
##  [1] nlme_3.1-157        webshot_0.5.3       RColorBrewer_1.1-2
##  [4] progress_1.2.2      httr_1.4.2          prolfqua_0.3.8
##  [7] numDeriv_2016.8-1.1 tools_4.2.0         backports_1.4.1
## [10] R6_2.5.1            vipor_0.4.5         DBI_1.1.1
## [13] lazyeval_0.2.2      colorspace_2.0-0    tidyselect_1.1.0
## [16] gridExtra_2.3       prettyunits_1.1.1   compiler_4.2.0
## [19] cli_3.3.0           rvest_1.0.2         xml2_1.3.2
## [22] plotly_4.10.0       labeling_0.4.2      bookdown_0.26
## [25] scales_1.1.1        systemfonts_1.0.0   stringr_1.4.0
## [28] digest_0.6.29       minqa_1.2.4         rmarkdown_2.14
## [31] svglite_2.1.0       pkgconfig_2.0.3     htmltools_0.5.2
## [34] fastmap_1.1.0       htmlwidgets_1.5.3   rlang_1.0.2
## [37] readxl_1.4.0        rstudioapi_0.13     generics_0.1.0
## [40] farver_2.0.3        jsonlite_1.8.0      dplyr_1.0.4
## [43] car_3.0-13          magrittr_2.0.3      kableExtra_1.3.4
## [46] Rcpp_1.0.8.3        ggbeeswarm_0.6.0    munsell_0.5.0
## [49] abind_1.4-5         lifecycle_1.0.1     stringi_1.5.3
## [52] yaml_2.2.1          carData_3.0-5       MASS_7.3-57
## [55] grid_4.2.0          ggrepel_0.9.1       forcats_0.5.1
## [58] crayon_1.5.1        lattice_0.20-45     cowplot_1.1.1
## [61] splines_4.2.0       hms_1.0.0           knitr_1.31
## [64] pillar_1.4.7        ggpubr_0.4.0        boot_1.3-28
## [67] ggsignif_0.6.3      glue_1.6.2          evaluate_0.14
## [70] data.table_1.13.6   BiocManager_1.30.17 png_0.1-7
## [73] vctrs_0.4.1         nloptr_2.0.3        cellranger_1.1.0
## [76] gtable_0.3.0        purrr_0.3.4         tidyr_1.1.2
## [79] assertthat_0.2.1    cachem_1.0.6        ggplot2_3.3.6
## [82] conflicted_1.1.0    xfun_0.31           broom_0.8.0
## [85] rstatix_0.7.0       viridisLite_0.3.0   tibble_3.0.6
## [88] pheatmap_1.0.12     beeswarm_0.4.0      memoise_2.0.0
## [91] ellipsis_0.3.2      BiocStyle_2.24.0
```