



HAL
open science

Statistical analysis of quantitative peptidomics and peptide-level proteomics data with Prostar

Marianne Tardif, Enora Fremy, Anne-Marie Hesse, Thomas Burger, Yohann Couté, Samuel Wiczorek

► **To cite this version:**

Marianne Tardif, Enora Fremy, Anne-Marie Hesse, Thomas Burger, Yohann Couté, et al.. Statistical analysis of quantitative peptidomics and peptide-level proteomics data with Prostar. *Statistical Analysis of Proteomic Data*, 2426, Springer US, pp.163-196, 2023, *Methods in Molecular Biology*, 10.1007/978-1-0716-1967-4_9 . hal-03242797

HAL Id: hal-03242797

<https://hal.science/hal-03242797>

Submitted on 31 May 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Statistical analysis of quantitative peptidomics and peptide-level proteomics data with Prostar

Marianne Tardif^{1,†}, Enora Fremy¹, Anne-Marie Hesse², Thomas Burger¹,
Yohann Couté¹, Samuel Wiczorek^{1,*}

¹ Univ. Grenoble Alpes, INSERM, CEA, UMR BioSanté U1292, CNRS FR2048, 38000, Grenoble, France

² Univ. Grenoble Alpes, CNRS, INSERM, CEA, FR2048 38000, Grenoble, France

[†] marianne.tardif@cea.fr, ^{*} samuel.wiczorek@cea.fr

Abstract

Prostar is a software tool dedicated to the processing of quantitative data resulting from mass spectrometry-based label-free proteomics. Practically, once biological samples have been analyzed by bottom-up proteomics, the raw mass spectrometer outputs are processed by bioinformatics tools, so as to identify peptides and quantify them, notably by means of precursor ion chromatogram integration. From that point, the classical workflows aggregate these pieces of peptide-level information to infer protein level identities and amounts. Finally, protein abundances can be statistically analyzed to find out proteins that are significantly differentially abundant between compared conditions. Prostar original workflow has been developed based on this strategy. However, recent works have demonstrated that processing peptide-level information is often more accurate when searching for differentially abundant proteins, as the aggregation step tends to hide some of the data variabilities and biases. As a result, Prostar has been extended by workflows that manage peptide-level data, and this protocol details their use. The first one, deemed "peptidomics", implies that the differential analysis is conducted at peptide level, independently of the peptide-to-protein relationship. The second workflow proposes to aggregate the peptide abundances after their preprocessing (*i.e.*, after filtering, normalization and imputation), so as to minimize the amount of protein-level preprocessing prior to differential analysis.

Key words Statistical software, Data processing, Differential analysis, Label-free proteomics, Relative quantification

1 Introduction

Nowadays, discovery proteomics mainly relies on liquid chromatography coupled to tandem mass spectrometry (LC-MS/MS). It is often used in a bottom-up workflow, where proteins are digested by a protease into peptides, making MS-based analysis more powerful [1]. This pipeline routinely produces huge amount of raw data, which processing with bioinformatics tools turns into long lists of identified peptides. In addition, a quantitative information for these peptides is classically computed, most often through eXtracted Ion Chromatograms (XIC) [2]. This quantitative information can then be used to compare the relative abundance of peptides and proteins between samples. The relative nature of the quantification derives from the fact that a XIC amounts to the integral of an ion signal over time, which amplitude is not only correlated to peptide concentrations in the biological sample, but also to numerous physicochemical phenomena, which are highly context- and peptide-dependent. As a result, peptide quantifications cannot be used to compare the abundance of different peptides within a sample, but only to compare the abundances of a same peptide within distinct yet relatively similar samples. Although there exist techniques to correct these biases and tend toward a more absolute quantification measure [3], it remains ordinary to stick to relative quantification information to answer the majority of questions addressed by discovery proteomics. Notably, a recurrent question in discovery proteomics is how to thoroughly select a set of putative biomarkers out of the long list of identified analytes, based on their variation of abundance across the samples. This question being ubiquitous in proteomics, a host of biostatistics tools have been developed over the last years, among which a significant proportion embed various R packages into well-designed graphical user interfaces based on Shiny technology [4], such as notably, Prostar [5]. However, two views can be opposed: the first one is to aggregate the peptide-level identities and quantities at protein-level, and then to perform the statistical analysis at protein-level. The second one is to directly work at peptide-level, since peptides are the entities directly analysed by the mass spectrometer. Although the first approach is mainstream for historical reasons, it is now well-established that, at least in theory but not only, peptide-level processing is more reliable [6]. Therefore, we have extended Prostar workflows, which used to be protein-centric [7], to propose convenient strategies to analyze proteomics data at peptide level, either in a peptidomic fashion; or by postponing the peptide-to-protein

aggregation step. Although the user experience is largely unchanged throughout the various versions of Prostar, this protocol is best suited to version 1.24.X and later (*see Note 1*).

2 Material

2.1 Live demo mode

Before installing the software, any user can have a quick overview by testing its demo mode on the following URL: <http://live.prostar-proteomics.org>. This mode provides a direct access to the DAPARdata package, where some toy datasets are available, either in tabular or MSnset formats [8]. Concretely, the demo mode is accessible in the `Data manager` menu: on the corresponding tab, a dropdown-menu lists the datasets that are available through DAPARdata. After selecting a peptide-level dataset, click on `Load demo dataset`.

Any user can also test the website version on his own data, yet we do not recommend it since the server has limited computational capabilities that are shared between all the users connected at the same moment. Overloading is therefore possible, which would lead to data loss.

2.2 Hardware requirements

Prostar can either be installed on a desktop machine (local installation by the user) or a server. The present protocol focuses on the former install. For the latter one, we refer to the DAPAR and Prostar user manual [9]. Depending on the data size, a recent workstation is necessary (we advise a minimum of 8GB of RAM, although there are no strict constraints).

2.3 Software requirements

1. The operating system must either be Linux, Mac OS X or Windows.
2. A recent version of the R software (*see Note 2*) must be installed in a directory where the user has the read and write permissions.
3. Optionally, an IDE (Integrated Development Environment) such as R Studio [10] may be useful to conveniently deal with the various R package installs.

2.4 Software install

Prostar can be installed following two ways:

1. The “zero-install”, which is the easiest way, as it does not need a prior install of R. However, it is so far only available for Microsoft Windows machines. “Zero-install” is a portable version of Prostar without requiring any installation. It can directly be downloaded from: <http://prostar-proteomics.org/#zero-install>, as a zip file referred to as `Prostar_1.24.x.zip`. Unzip it into a directory with read/write permissions (*see Note 3*).
2. The stand-alone Bioconductor install, which is the standard method to install Bioconductor distributed software. R must be already installed. First, install Bioconductor package manager, and then Prostar, by copy-paste of the following commands:

```
install.packages("BiocManager")
BiocManager::install(version=BiocManager::version())
BiocManager::install("Prostar")
```

This will install Prostar together with all the required dependency packages.

2.5 Data type

The quantitative data should fit into a matrix-like representation where each line corresponds to a peptide and each column to a sample. Within the (i th, j th) cell of the matrix, one reads the abundance of peptide i in sample j .

2.6 Data size – number of peptides

Although strictly speaking, there is no lower or upper bound to the number of lines, it should be recalled that the statistical tools implemented in Prostar have been chosen and tuned to fit a discovery experiment dataset with large amount of peptides, so that the result may lack of reliability on too small datasets. Conversely, very large datasets are not inherently a problem, as R algorithms are well scalable, but one should keep in mind the hardware limitations of the desktop machine on which Prostar runs to avoid overloading.

2.7 Data size – number of samples

As for the number of samples (the columns of the dataset), it is necessary to have at least 2 conditions (or groups of samples) as it is not possible to perform relative comparison otherwise. Moreover, it is necessary to have at least 2 samples per condition (*see Note 4*), as otherwise, it is not possible to compute an intra-condition variance, which is a prerequisite to numerous processing.

2.8 Data format

The data table should be formatted in a tabulated file where the first line of the text file contains the column names. It is recommended to avoid special characters such as "]", "@", "\$", "%", etc. that are automatically removed. Similarly, spaces in column names are replaced by underscore ("_"). Dot (".") must be used as decimal separator for quantitative values. In addition to the columns containing quantitative values (*see Subheading 2.7*), the file may contain additional columns for metadata. Prostar supports any tabular file but is directly compatible with MaxQuant [11] and Proline [12] files (*see Chapter 4* for a protocol on Proline use). Alternatively, if the data have already been processed by Prostar and saved as an MSnset file [8], it is possible to directly reload them (*see Note 5*).

3 Methods

3.1 Starting Prostar

1. Prostar is launched differently depending on how it was installed:
 - (a) **Zero-install**: the unzipped folder contains an executable file (`Prostar.exe`) which directly launches Prostar in a webpage on the default internet navigator. At first launch, the latest version of Prostar is automatically downloaded from Bioconductor and silently installed. Once done, the user is invited to close the current page. Then, a new webpage with Prostar is automatically opened.
 - (b) **Bioconductor install**: run the following commands in a R console (Prostar will open in a new webpage):

```
library(Prostar)
Prostar()
```

2. Once Prostar is launched, it displays its welcome page (*see Figure 1*). Since no dataset is loaded, the main menu at the top of the page contains only the following items:
 - (a) `Prostar`: It contains some information about versions of Prostar and release notes.
 - (b) `Data manager`: It contains the different tools to load or convert a new dataset and export a dataset analysed by Prostar.
 - (c) `Help`: It gathers FAQ and a bug report form.
3. Once a peptide-level dataset has been loaded in Prostar (*see Subheading 3.2*), the content of the menus changes:
 - (a) the three Import tools (`Open MSnSet`, `Convert`, and `Demo mode`) from the menu `Data Manager` are hidden.
 - (b) Two items appear in the main menu: `Data processing (peptide)` and `Data mining`.
4. If one wants to change the current dataset, reloading Prostar first is necessary to avoid cache memory issues. Restarting Prostar can be done in two ways:
 - (a) Close the current webpage then restart Prostar as described previously (*see Subheading 3.1*).
 - (b) Restart the R session. For this, go to `Data manager` >> `Reload Prostar`. Then, click on the `Reload Prostar`. This action will restart Prostar with a fresh R session in which import options are enabled in the `Dataset manager` menu.

3.2 Data Loading

If the dataset under consideration is not a peptide dataset (each line of the quantitative table does not represent a peptide, but for instance a protein), do not apply the present protocol. If it is a protein dataset, please refer to [7]:

1. To upload data from tabular file (*see Notes 6 and 7*) (*i.e.*, stored in a file with one of the following extensions: `.txt`, `.csv`, `.tsv`, `.xls`, or `.xlsx`), go to the upper menu and click on `Data manager` >> `Convert data`.
2. Go to the `Select File` tab.
3. Choose the software used to produce the quantitative dataset (*see Note 8*).

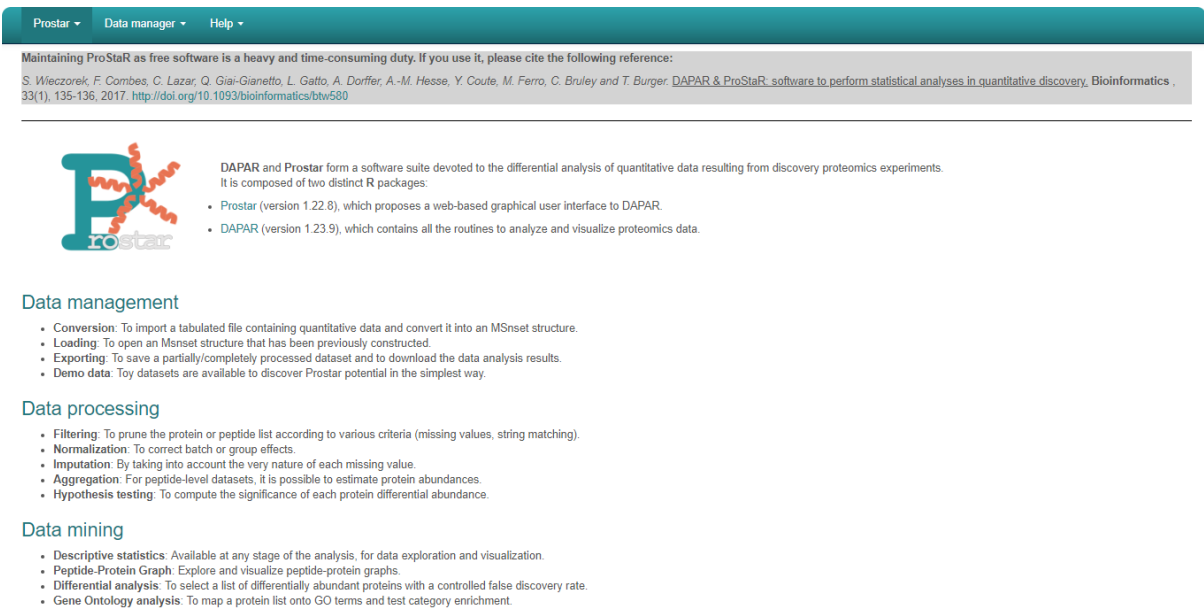


Figure 1: Prostar welcome page.

4. Click on and select the tabular file of interest.
5. Once the upload is complete, indicate it is a peptide dataset (*i.e.*, each line of the data table should correspond to a single peptide).
6. Indicate whether the data are already log-transformed or not. If not, they will be automatically log-transformed (*see Note 9*).
7. If the quantification software uses "0" in places of missing values, tick the last option (as in Prostar, 0 is considered a value, not a missing value).
8. Move on to the tab.
9. If the dataset already contains an ID column (a column in which each cell has a unique content, that can serve as an ID for the peptides), select the appropriate column name in the dropdown-menu . In any case, it is possible to use the option, which creates an artificial index.
10. (OPTIONAL) In (dropdown-menu), choose the column in the dataset that contains the IDs of proteins to which each peptide belongs to (*see Note 10*). As multiple computations of the peptide-level proteomic pipeline are based on this information, it is important to select the correct column. However, if a peptidomic analysis is anticipated (*i.e.*, without peptide-to-protein aggregation), it is not important. To check the content of the column: when a value is selected, a small preview of the content is displayed.
11. Move on to the tab.
12. Select the columns that contain the peptide abundances (one column for each sample of each condition). To select several column names in a row, click-on on the first one, and click-off on the last one. Alternatively, to select several names which are not continuously displayed, use the key to maintain the selection.
13. If, for each sample, a column of the dataset provides information on the identification method, *e.g.*, by direct MS/MS evidence, or by mapping (*see Note 11*), check the corresponding tick box. Then, for each sample, select the corresponding column. If none of these pieces of information is given, or, on the contrary, if all of them are specified with a different column name, a green logo appears, indicating it is possible to proceed (*see Note 12*) (*see Figure 2*). Otherwise (*i.e.*, the identification method is referenced for two different samples), then a red mark appears, indicating some corrections are mandatory.
14. Move on to the tab. This tab guides the user through the definition of the experimental design.
15. Fill the empty columns with as different names as biological conditions to compare (minimum 2 conditions and 2 samples per condition, *see Subheading 2.6*) and click on . If necessary, correct until the conditions are valid. When achieved, a green logo appears and the sample are reordered according to the conditions.
16. Choose the number of levels in the experimental design (either 1, 2 or 3), and fill the additional column(s) of the table (*see Note 13*).

Figure 2: Convert tool: `Exp. and feat. data` tab.

17. Once the design is valid, a green check logo appears (*see* Figure 3). Then, move on to the `Convert` tab.
18. Provide a name to the dataset to be created and click on the `Convert` button. This step may take some time as Prostar computes all relationships between peptides and their proteins (to anticipate on the connected component exploration, *see* Subheading 3.5; as well as on the peptide to protein aggregation, *see* Subheading 3.11).
19. As a result, a new MSnset structure is created and automatically loaded (this is why Prostar returns to the welcome page). This can be checked with the name of the file appearing in the upper right hand side of the screen, as a title to a new dropdown-menu. So far, it only contains `Original - peptide`, but other versions of the dataset will be added along the course of the processing.

3.3 Data Export

As importing a new dataset from a tabular file is a tedious procedure, we advise to save the dataset as an MSnset binary file right after the conversion. This makes it possible to restart the statistical analysis from scratch if a problem occurs without having to convert again the data. To do so:

1. Click on `Data manager` `Export`.
2. Choose MSnset as file format and provide a name to the object (*see* Note 14).
3. Click on `Download`.
4. Once downloaded, store the file in the appropriate directory.
5. It is possible to reload any dataset stored as an MSnset structure (*see* Note 5).

3.4 Descriptive Statistics

By clicking on `Data mining` `Descriptive statistics`, it is possible to access several tabs generating various plots (*see* Note 15) that provides a comprehensive and quick overview of the dataset (*see* Note 16):

1. On the first tab (`overview`), a brief summary of the quantitative data size is provided. It roughly amounts to the data summary that is displayed along with each dataset during the loading step of the demo mode.
2. On the second tab (`Quantification nature`), the barplots depicts the distribution of so-called cell metadata (*see* Note 17). The user selects the label to focus on; once done, three barplots are displayed. The left hand side barplot represents the number of peptides with the corresponding label in each sample. This number is written in the tooltip when the user hovers the mouse pointer over the bar along with the percentage of whole peptides this value corresponds to. The different colors correspond to the different conditions (or groups, or labels). The second barplot (in the middle) displays the distribution of the label of interest. The last barplot represents the same information as the previous one, yet, condition-wise.

Prostar ▾ Data manager ▾ Help ▾

1 - Select file 2 - Data id 3 - Exp. and feat. data 4 - Samples metadata 5 - Convert

If you do not know how to fill the experimental design, you can click on the "?" next to each design in the list that appear once the conditions are checked or got to the [FAQ](#) page.

1 - Fill the "Condition" column to identify the conditions to compare.

2 - Choose the type of experimental design and complete it accordingly ^[?]

Flat design (automatic)
 2 levels design (complete Bio.Rep column)
 3 levels design (complete Bio.Rep and Tech.Rep columns)

Correct conditions
 Correct design

Order by conditions ?

No ▾

Design

Sample.name	Condition	Bio.Rep
abundance_50fmol.1	50fmol	1
abundance_50fmol.2	50fmol	2
abundance_50fmol.3	50fmol	3
abundance_50fmol.4	50fmol	4
abundance_5fmol.1	5fmol	5
abundance_5fmol.2	5fmol	6
abundance_5fmol.3	5fmol	7
abundance_5fmol.4	5fmol	8

Figure 3: Convert tool: `Samples metadata` tab.

- The third tab (`Data explorer`) is dedicated to the visualization of data tables (see Figure 5): it makes it possible to view the content of the MSnSet structure. It is made of three tables, which can be displayed one at a time thanks to the radio button on the left menu. The first one (`Quantitative data`) contains quantitative values. The missing values are represented by empty cells; a legend of the colors indicates the type of quantification metadata. The second one (`Peptides metadata`) contains all the column of the dataset that are not the quantitative data. The third tab (`Experimental design`) summarizes the design, as defined at the import step (see Subheading 3.2, **Step 16**).
- In the fourth tab (`Corr. matrix`), it is possible to visualize to what extent the replicate samples correlates or not. The contrast of the correlation matrix can be tuned thanks to the color scale on the left hand side menu.
- A heatmap as well as the associated dendrogram is depicted on the fifth tab. The colors represent the intensities: Red for high intensities, green for low intensities and white for missing values. The dendrogram shows a hierarchical classification of the samples, so as to check that they are related according to the experimental design. It is possible to tune the clustering algorithm (see **Note 18**) that produces the dendrogram by adjusting the `distance` and `linkage` parameters, as described in the `hclust()` function of the R package `stats` [13].
- Tab 6 (`PCA`) shows different plots related to the Principal Component Analysis (see Figure 6). The plots are displayed only if the dataset does not contain any missing values.
- Tab 7 represents in a various ways the same information, that is the distribution of intensity values by replicates and conditions: respectively smoothed histograms (a.k.a. kernel density plots), boxplots, and violin-plots are used.
- Finally, the last tab displays a density plot of the variance (within each condition) conditionally to the log-intensities (see **Note 19**).

3.5 Peptide-protein Graph

Clicking on `Data mining` `Peptide-protein Graph` gives access to a graph visualization tool displaying the relationships between the peptides and the protein(s) they belong to (see **Note 20**). Peptides and proteins are depicted as nodes and an edge connects a (protein, peptide) couple when the peptide belongs to the protein. The set of all possible graphs in the dataset is computed when a new dataset is loaded in Prostar (see Subheading 3.2, **Step 18**). This visualization tool is divided into three tabs, so as to distinguish three types of graphs:

- `One-One Connected Components` (One-One CC): The table on the left hand displays all the One-One CCs, *i.e.*, graphs containing a single protein linked to its unique (and specific) peptide (*i.e.*, it is not shared with other

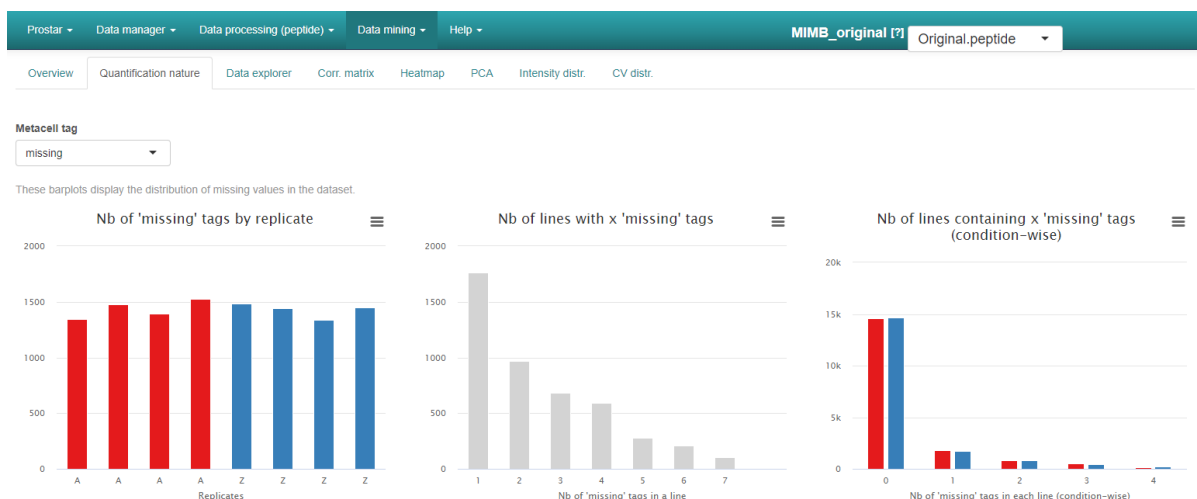


Figure 4: Descriptive statistics: `metadata` tab.

proteins) (*see Note 21*). By selecting an item in this table, it is possible to view the quantitative values of the peptide in consideration in the table on the right hand (*see Note 22*). Note that due to their large number and to limit memory load, One-One CCs are not displayed.

2. `One-Multi Connected Components` (One-Multi CC): This interface is the same as for One-One CCs (as described above) but displays graphs containing a single protein linked to several (specific) peptides (*i.e.*, not shared with other proteins). Note that due to their large number and to limit memory load, One-Multi CC are not displayed either.
3. `Multi-Multi Connected Components` (In Multi-Multi CCs, a peptide may be specific to a protein or shared between several proteins): The dropdown-menu `Search for CC` let the user choose if he wants to see the list of Multi-Multi CCs as a table ("Tabular view") or a plot ("Graphical view"); each of these is displayed on the left hand side. The table is the same as for One-One CCs and One-Multi CCs. In the graphical view, each point represent a CC where the coordinates are the number of peptides as x-axis and the number of proteins as y-axis. The selection of a particular CC is possible by clicking on the corresponding point in the plot. Once a CC has been selected, the graph is plotted on the right hand side (*see Figure 7*): The medium-size nodes correspond to proteins while the small nodes are for peptides (specific peptides are blue while shared peptides are green). The three tables below show information about the nodes that have been selected by clicking on them in the graph (*see Figure 8*). The first table named "Proteins" lists the Ids of the proteins in the CC, while the two other tables shows the quantitative values of respectively "Specific proteins" and "Shared proteins". By default, all nodes are selected. Clicking on a particular node of the graph selects it as well as all of its neighbors (the other nodes -peptides or proteins- it connects to) and updates the tables below. The text-field `Peptide Info` allows to choose among the peptide metadata, the one to add in the tables "Specific proteins" and "Shared proteins" beside the quantitative values.

3.6 Filtering

The first processing tool of the peptidomics (or of the peptide-level proteomics) pipeline gives the opportunity to filter out the data according to some information stored in the quantitative data or peptides metadata. Each filtering tool is based on the same behaviour: the user builds a query (by choosing its parameters) that selects some lines in the dataset, to keep them or on the contrary, to remove them. Multiple filters can be run successively; in this case, the i th filtering is run on the dataset resulting from the $(i - 1)$ th filtering:

1. Click on `Data processing >> Filter data`.
2. The first tab (`Quanti. metadata filtering`) allows to filter peptides according to the nature of the quantitative values (stored in the so-called cell metadata, *see Subheading 3.4, Step 2*):
 - (a) Choose the data nature in the left hand dropdown-menu (`Nature of data to filter`). The possible values correspond to the quantification nature metadata such as described previously (*see Note 17*). It may be useful to keep in mind the hierarchy of the associated labels to understand the scope of filtering. For example, if "missing POV" is selected, then the filter will apply only on "missing POV" but if "missing" is selected, then all missing data are concerned ("missing POV" as well as "missing MEC").

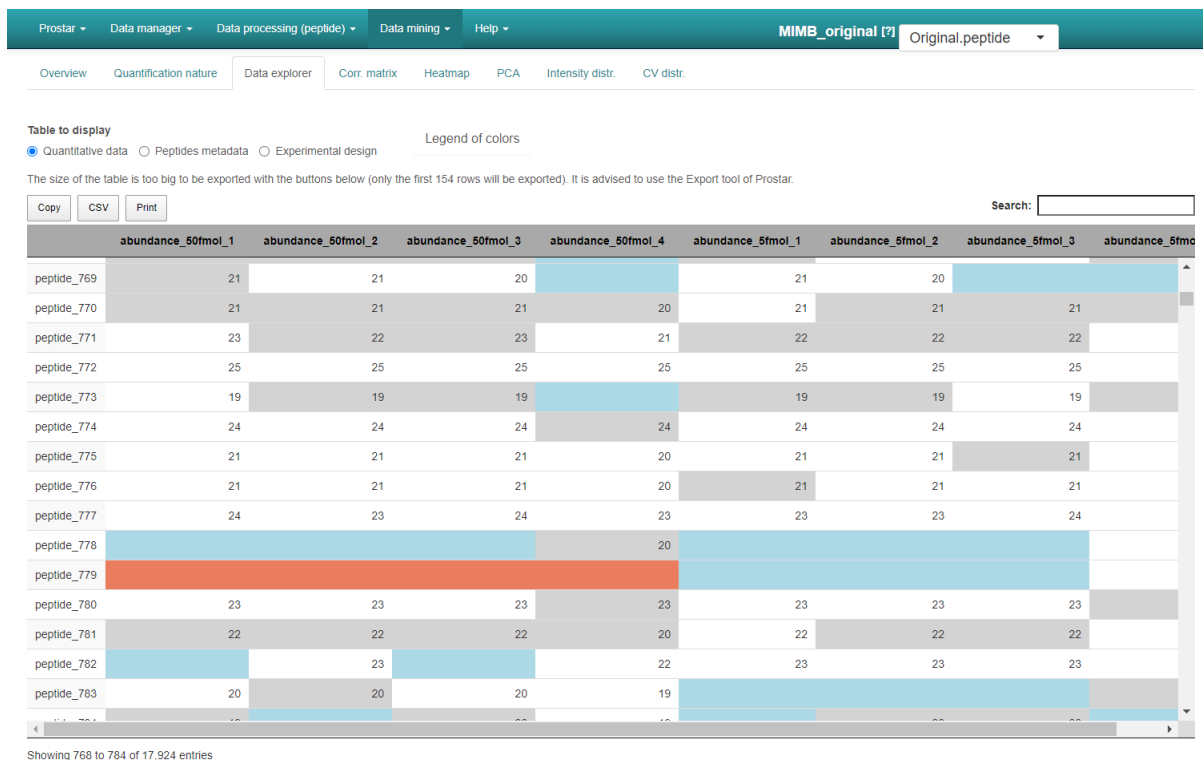


Figure 5: Descriptive statistics: `Data explorer` tab.

- (b) The next dropdown-menu (`Type of filter operation`) allows to choose whether the lines identified by the query are to be kept or deleted. If a given set of n lines is removed from a dataset containing N lines, then the resulting dataset will contain $N-n$ lines. On the contrary, if n lines are kept from an initial dataset containing N lines, then the resulting dataset will contain those n lines and the other $N-n$ lines are removed.
- (c) The `Scope` menu allows to consider either the entire experimental design, or to apply the filtering query on each group of samples (condition) separately. The available scopes are the following: "None": No filtering, the dataset is left unchanged, so directly go to next step `String-based filtering` (see Subheading 3.6, Step 3). "Whole matrix": The lines (across all conditions) which cell metadata contain a certain number of labels fitting with the user-defined formula are flagged. "Whole line": lines containing only the labels referred to in the queries are concerned. This is a special case of the "Whole matrix" scope leading to shortcut queries (e.g., to straightforwardly remove lines with only missing values). "Every condition": The concerned lines are those for which each condition contains a certain number of cells with the label of interest. "At least one condition": The concerned lines are those for which at least one condition contains a certain number of cells with the label of interest.
- (d) Whenever a scope is selected (except for the "Whole line" option), a series of three dropdown-menus are displayed to refine the query by setting conditions on the number of labels: `Nature of the threshold` allows to choose whether to select lines on the basis of an absolute counting, or on a percentage (see Note 23). Then, `Operator` defines the algebraic operator used to select lines associated with a threshold value (`Threshold`). If the threshold value is a percentage, the value must be a decimal number between 0 and 1. If it is a count value, the dropdown-menu proposes a list of several integers depending on the chosen scope: If the scope is equal to "Whole matrix", then the range of the threshold is from 0 to the total number of samples. If the scope is "For each condition" or "At Least One Condition", the possible values are the integers between 0 and the minimum amount of samples in all the conditions (i.e., the number of samples in the "smallest" condition).
- (e) Clicking on `Preview filtering` button displays a popup window showing the quantitative table of a random subset of the dataset (see Figure 10). This can be useful to better understand the behavior of the filters.
- (f) Visualize the effect of the filtering options without changing the current dataset by clicking on `Perform filtering`; the barplots are the same as in the tab `Descriptive Statistics` `Quantification nature` (see Subheading 3.4). If the filtering does not produce the expected effect, test another one. To do so, simply choose another method in the list and click again on `Perform filtering`. The plots are automatically updated. This action does not modify the dataset but offers a preview of the filtered data. Iterate this

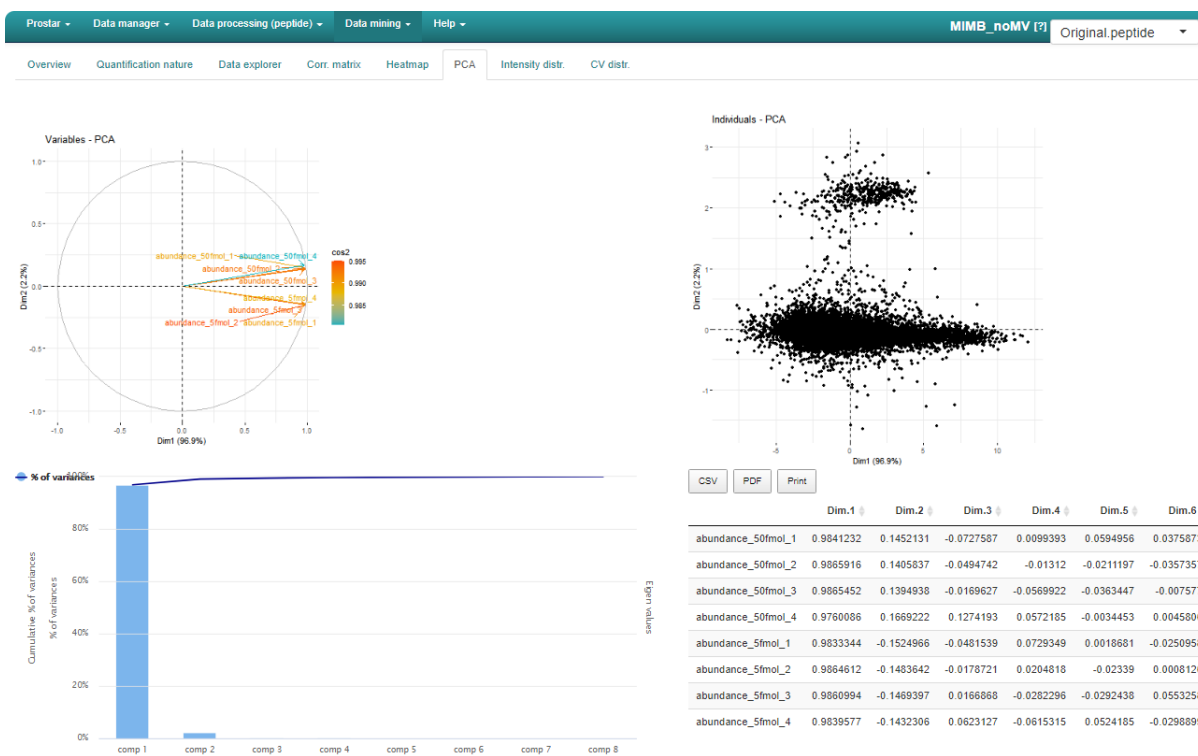


Figure 6: Descriptive statistics: **PCA** tab.

step as long as necessary. Each time a preview is run, a short summary is added in the table above the plots.

3. Move on to the second tab (**String-based filtering**), where it is possible to filter out peptides according to information stored as alpha-numerical strings in the peptide metadata (see **Note 24**):
 - (a) Among the columns constituting the peptide metadata listed in the dropdown-menu, select the one containing the information (see **Note 25**) of interest, e.g., "Contaminant" or "Reverse", (see **Note 26**). Then, specify in each case the prefix chain of characters that identifies the peptides to filter (see **Note 27**).
 - (b) Click on **Perform** to remove the corresponding peptides. A new line appears in the table listing all the filters that have been applied (see Figure 9).
 - (c) If other filters are necessary, iterate the above sub-steps.
4. Move on to the third tab (**Numerical filtering**), where it is possible to filter out peptides according to information stored as numerical values in the peptide metadata:
 - (a) As for the filter on the first tab, fill the dropdown-menus to build a query that select the lines concerned by the filter. The first one lists the name of the columns in the metadata and tries to guess the one with numerical values.
 - (b) Click on **Perform** to remove the corresponding peptides. A new line appears in the table listing all the filters that have been applied.
 - (c) If other filters are necessary, iterate the above sub-steps.
5. Once all the filters have been applied, move on to the last tab (**Visualize and Validate**) to check the set of filtered out peptides. This visualization tools works similarly as the **Data explorer** (see Subheading 3.4, **Step 3**).
6. Finally, click on **Save filtered dataset**. The new dataset is accessible and referred as "Filtered.peptide" in the version dropdown-menu (see Subheading 3.7).

3.7 Navigating through the dataset versions

Once the filters have been applied and the results saved, a new dataset is created, in the same way as after the data conversion (see Subheading 3.2, **Step 18**). It is referred to as "Filtered - peptide", and its name appears right below "Original - peptide" in the upper right drop-down menu, beside the dataset name (as illustrated

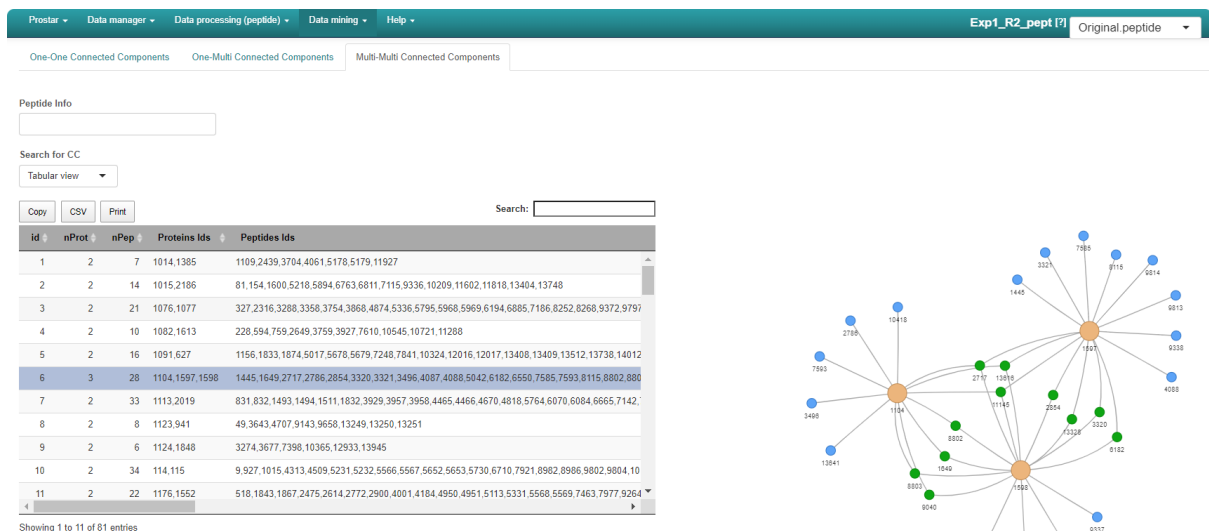


Figure 7: Peptide-Protein Graph: Zoom on the upper part of the **Multi-Multi Connected Components** tab.

in upper right corner of Figure 5). Unless modified, the newest created dataset is always the current dataset, *i.e.*, the dataset on which further processing will be applied. As soon as the current dataset is modified, all the plots and tables in Prostar are automatically updated. Thus, as soon as a new dataset is created, we suggest to go back to the descriptive statistics menu (*see* Subheading 3.4) to check the influence of the latest processing on the data. It is possible to have a dynamic view of the processing steps by navigating back and forth in the dataset versions, so as to see the graphic evolutions (*see* Note 28).

3.8 Normalization

The next processing step proposed by Prostar is data normalization (*see* Figure 11). Its objective is to reduce the biases introduced at any preliminary stage (such as for instance batch effects):

1. Choose the normalization method available in the dropdown-menu (**Normalization methods**). For each possible normalization, a short description is provided and the interface is automatically updated to display the method parameters that must be tuned (*see* Note 29).
 - (a) **None**: No normalization is applied.
 - (b) **Global quantile alignment**: The Quantile of the intensity distributions of all the samples are equated.
 - (c) **Sum by columns**: The total (un-logged) intensity values of all the samples are equated. The rationale behind is to normalize according to the total amount of biological material within each sample.
 - (d) **Quantile Centering**: A given quantile of the intensity distribution is used as reference, such as for instance the median, or a lower quantile depicting the lower limit of detection (*see* Note 30).
 - (e) **Mean Centering**: sample intensity distributions are aligned on their mean intensity values (and optionally, the variance distributions are equated to one).
 - (f) **LOESS**: The intensity values are normalized by means of a local regression model [14] of the difference of intensities as function of the mean intensity value (*see* [15] for implementation details).
 - (g) **vsn**: Variance Stabilizing Normalization, which wraps the method described in [16].
2. Possibly, indicate the normalization type (available in the dropdown-menu **Normalization type**). Notably, for most of the methods (in fact, all of them, but "Global quantile alignment"), it is necessary to indicate whether the method should apply to the entire dataset at once ("overall" option), or whether each condition should be normalized independently of the others ("within conditions" option).
3. For other parameters, which are specific to each method, the reader is referred to Prostar user manual, available through the **Help** menu (*see* Note 31).
4. Click on **Perform normalization**.
5. Observe the influence of the normalization method on the plots of the lower side panel. The middle graph can be switched from violin plot to box plot.
6. If the result of the normalization does not correspond to the expectations, first return to the initial situation by clicking on **reset** and change the normalization method (or change its tuning).

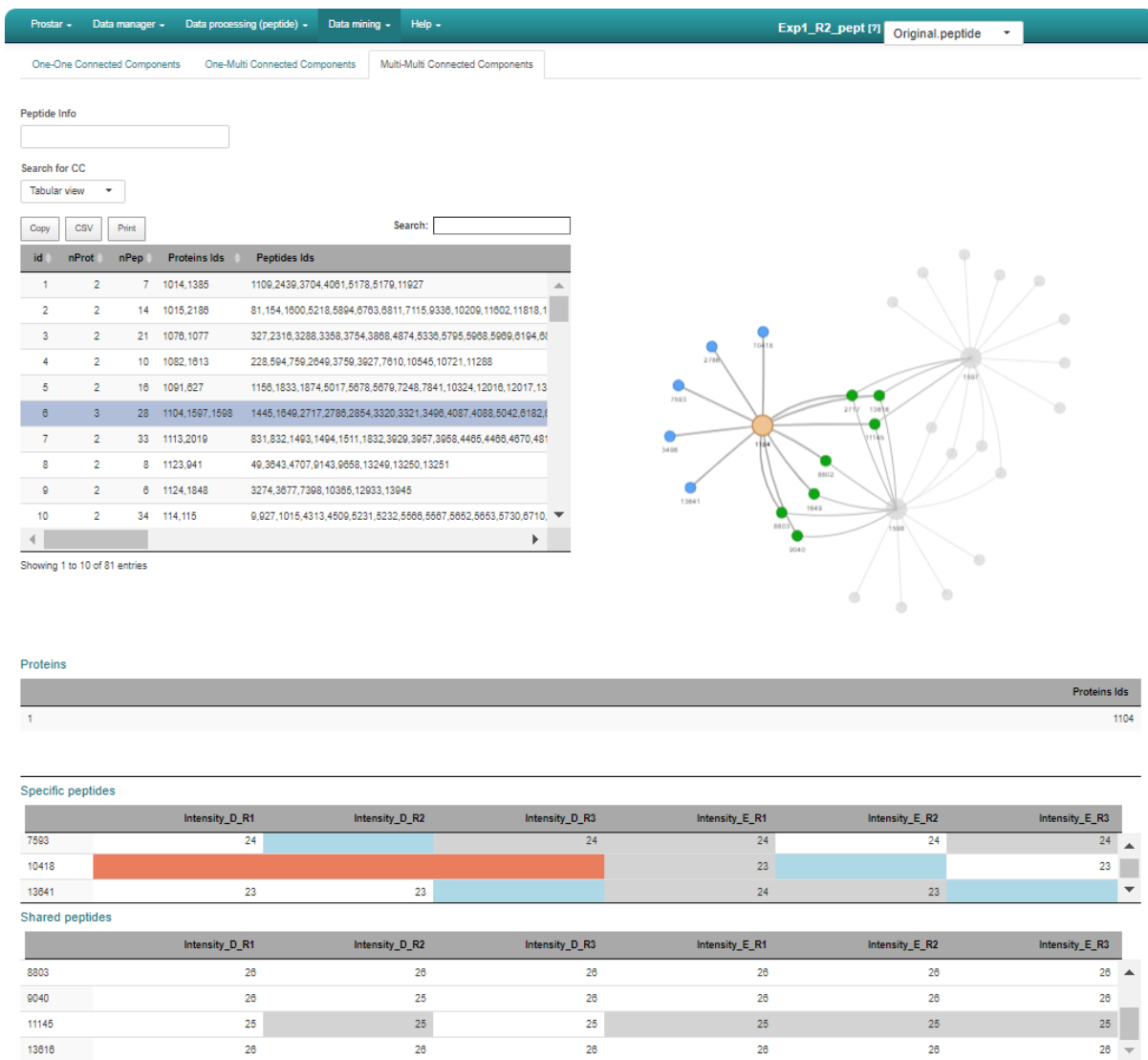


Figure 8: Peptide-Protein Graph: Multi-Multi Connected Components tab.

7. Once the normalization is effective, move on the `Validate` tab and click on `Save normalization`. Check that a new version appears in the dataset version dropdown-menu, referred to as "Normalized.peptide".

3.9 Missing values imputation

Prostar distinguishes two different types of missing values: POV (standing for Partially Observed Value) and MEC (standing for Missing in the Entire Condition). All the missing values for a given peptide (or protein in case of a protein dataset) in a given condition are considered POV if there is at least one observed value for this peptide in this condition. Alternatively, if all the intensity values are missing for this peptide in this condition, the missing values are considered MEC (see **Note 32**). At peptide level, several algorithms are proposed for the imputation of POV while the imputation of MEC is left as an option:

1. On the `Peptide imputation` tab (see Figure 12), select the algorithm to impute missing values in the `Algorithm` dropdown-menu.
2. If missing values are not going to be imputed at all, select "None". However, this option is not recommended since it may prevent subsequent hypothesis testing (see Subheading 3.13) as well as strongly impact the peptide-to-protein aggregation procedure (see Subheading 3.11).
3. If "imp4p" is selected, tune the number of iterations (default is 10). We strongly recommend the imp4p algorithm as it performs a diagnosis of the nature of each POV missing value, while it is not the case with the less refined "Basic Methods". Indeed, missing Value can be diagnosed as MNAR (Missing Not At Random) or MCAR (Missing Completely At Random) [17] (see **Note 33**). The more iterations there are

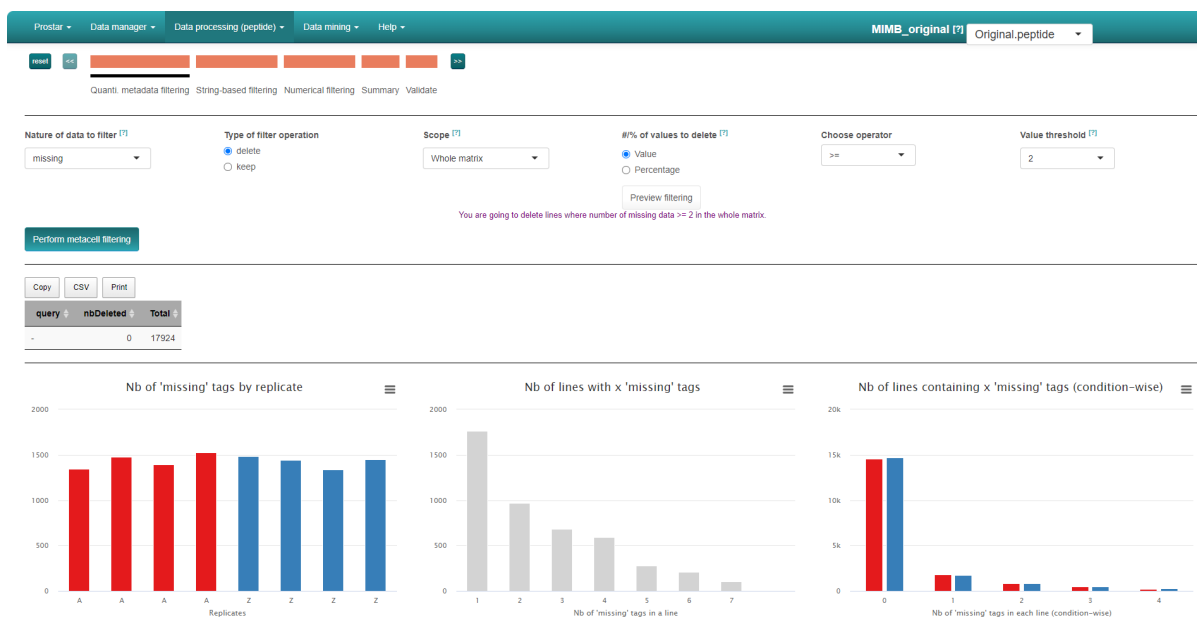


Figure 9: Snapshot of the filtering module.

the more precise the imputation is. You can choose to impute the MEC values at this step (MECs are always considered MNAR missing values) by checking `Impute MEC also` (see Note 34).

4. If MEC imputation is decided, tune the required parameters: The first parameter is the upper bound of the MNAR imputed values, expressed as a quantile of the distribution of observed values in each sample (default is centile 2.5). The second parameter is the probability distribution defined on the range of MNAR values, either "uniform" or "beta": The latter is more conservative, as it provides more weight on values that are close to the detection limit, which reduces the risk of creating artificial differentially abundant peptides (and thus proteins).
5. If "Basic Methods" is selected, choose the appropriate method in the `Methods` dropdown-menu and tune its parameters. The three basic methods proposed are as follows and the user is invited to consult the manual for a detailed description. Be aware that these methods only deal with POV missing values (MECs are not imputed for now) and do not make any diagnosis of the nature of these missing values.
 - (a) "Det. quantile": each missing value within a given sample is replaced by a deterministic value (usually a low value) (see Note 35). When using "Det. quantile", the list of imputation values for each sample appears above the graphics on the right panel.
 - (b) "KNN": K-nearest neighbors estimates each missing value by the mean of the observed values of other peptides with a similar intensity pattern (called neighbors).
 - (c) "MLE": Maximum Likelihood Estimation estimates the expected abundance value of each peptide in each condition and use this as replacement of missing values.
6. Click on `Perform Imputation`. If there are yet missing values in the imputed dataset (i.e., the MECs were not imputed), a message alerts the user that the aggregation of peptides into proteins may fail (see Note 36). If the user wants to avoid possible issues with further peptide-to-protein aggregation, it is necessary to click on `Reset` to reinitialize the imputation tool and impute its dataset with "imp4p" and the option "Include MEC also".
7. Observe the influence of the chosen imputation method on the plots of the lower hand side panel. If the result does not correspond to the expectations, change the imputation method (or its tuning) by clicking on `reset` and return to the first step of of imputation.
8. Once the imputation is effective, move on the `Save` tab and click on `Save imputation`. Check that a new version appears in the dataset version dropdown-menu, referred to as "Imputed.peptide" (see Note 37).

3.10 Hypothesis Testing for peptidomics data

For peptide datasets that do not contain any missing values, or for those where these missing values have been imputed, it is possible to test whether each peptide is significantly differentially abundant between the compared conditions and next proceed to differential analysis:

1. Click on `Data processing (peptide)` `>>` `Hypothesis testing` (see Figure 13).

delete lines where number of missing data ≥ 2 in each condition.

Example dataset

- original dataset
 simulate filtered dataset

	A_1	A_2	A_3	B_1	B_2	B_3
peptide_1	96	3	7	52	2	97
peptide_2		52	50	0	100	79
peptide_3			43	64	62	38
peptide_4				36	49	95
peptide_5					65	38
peptide_6						80
peptide_7						
peptide_8		12	2		11	60
peptide_9		20	42			85
peptide_10			92			65

Figure 10: Simulate filtering on a toy example: The query objective is to “delete lines where there are more than 2 missing values in each condition”.

2. Follow the very same four steps as for hypothesis testing with protein datasets (*see* Subheading 3.13).
3. After saving, check that a new version appears in the dataset version dropdown-menu, referred to as "HypothesisTest.peptide".
4. Then, this new peptide dataset, containing the p-values and fold change (FC) cutoff for the desired contrasts, can be explored in the [Differential analysis](#) tabs available in the [Data mining](#) menu (*see* Subheading 3.14).
5. Follow the very same six steps as for differential analysis with protein datasets (*see* Subheading 3.14).
6. From the [FDR](#) tab, you may download data and save any plot/table of interest (*see* Figure 14).

3.11 Aggregation

From a peptide dataset, it is possible to aggregate the peptide intensities to construct a new protein-level dataset. For each parent protein, aggregation can only apply to a series of non-missing values (yielding a numerical values) or to a series of missing values (yielding a missing value), but not to a combination of both (*see* **Note 38**). If missing value imputation was not performed before aggregation, a warning is displayed and the list of excluded peptides (and thus proteins) is provided to the user (*see* **Note 39**). To simplify things in practice, an option is to systematically impute all the missing value upstream aggregation, however, it may lead to the imputation of peptides that the practitioner would not trust in the first place. To perform aggregation:

1. Click on the corresponding option in the [Data mining](#) > [Aggregation](#) menu. The lower panel provides two barplots depicting the protein distribution according to their number of peptides: either all of them (*see* Figure 15, right-hand plot), or only those which are specific to a single protein (left-hand plot).
2. A key step is to decide whether and how to include shared peptides in the aggregation by clicking one of the three radio buttons under "Include shared peptides". Option "No" excludes shared peptides (*see* **Note 40**). Option "Yes (as protein specific)" processes shared peptides as if they were specific to each of their parent proteins (*see* **Note 41**). The option "Yes (redistribution)" computes a proportional redistribution of the intensity of shared peptides among proteins (*see* **Note 42**).
3. Choose whether to consider all peptides or only the first N most abundant ones for each protein by clicking the appropriate button under "Consider".
4. Choose the aggregation operator, by clicking either on [sum](#) or [mean](#) under "Operator". Note that for the redistribution option of the shared peptides, only the mean operator is allowed (*see* **Note 43**).



Figure 11: Normalization tab: The intensity density plot, the violon plot (alt. box plot) and the distortion plot make it possible to visualize the influence of each normalization method.

5. Click on **Perform aggregation** and wait for the message "aggregation done".
6. Move on to the next tab, **Add metadata**, and select the columns of the metadata table that must be aggregated to form protein-level metadata. This will create columns with the same headers as in the original peptide dataset. The result of metadata aggregation will be a comma-separated string concatenation of the unique values available at peptide level (duplicated values are not repeated). It is recommended to select at least the peptide identifier field, so that it remains possible to link protein and peptide information afterwards.
7. Move on to the **Save** tab and save the aggregation step. A protein dataset is created and loaded in memory for further processing. The interface automatically switches to the homepage. The new aggregated dataset is accessible and referred as "Aggregated.protein" in the version dropdown-menu.
8. At this stage, it may be wise to export the dataset as a MSnSet or excel file (*see* Subheading 3.3), in order to be able to carry out several different processes without having to start the aggregation step again (*see* **Notes 44** and **45**).
9. As hereafter described (*see* Subheading 3.12), the aggregated protein dataset can be explored (**Descriptive Statistics** menu), analyzed (**Data mining** menu) and processed (**Data processing** menu) very much the same way as a peptide dataset. However, the new dataset being a protein one, the aggregation process does not appear anymore in the **Data processing** menu.

3.12 Aggregated protein dataset preprocessing

1. Additional filtering can be applied at the protein level if necessary. The menu displays the same interface and options as for the peptide level (*see* Subheading 3.6). In particular, if only specific peptides were accounted for during the aggregation (*see* **Note 46**), proteins with no specific peptide will thus have missing values in all conditions. It is thus necessary to filter these proteins as "empty lines" (*see* Subheading 3.6, **Step 2**) (*see* **Note 40**). If a protein-level filter is applied, saving is necessary (leading to a new dataset referred to as "Filtered.protein" in the version dropdown-menu).
2. Protein level normalization can be applied if necessary. The menu displays the same interface and options as for the peptide level (*see* Subheading 3.8). If a protein-level normalization is applied, saving is necessary (leading to a new dataset referred to as "Normalized.protein" in the version dropdown-menu).
3. Depending on the options chosen during peptide imputation step (*see* Subheading 3.9), the aggregated protein dataset may still contain missing values, as previously detailed (*see* Subheading 3.11). This occurs when either no imputation was performed at peptide level; or when "imp4p" was applied without processing the MECs (*see* Subheading 3.9, **Step 3**); or when one of the "Basic Methods" was preferred. In such cases, imputation is a necessary step to proceed to hypothesis testing and differential analysis of the protein level dataset. The interface and options for protein imputation slightly differ from those at peptide-level and the user should refer to [7] for a step-by-step description (*see* **Note 37**). If a protein-level imputation is applied, saving is necessary (leading to a new dataset referred to as "Imputed.protein" in the version dropdown-menu).

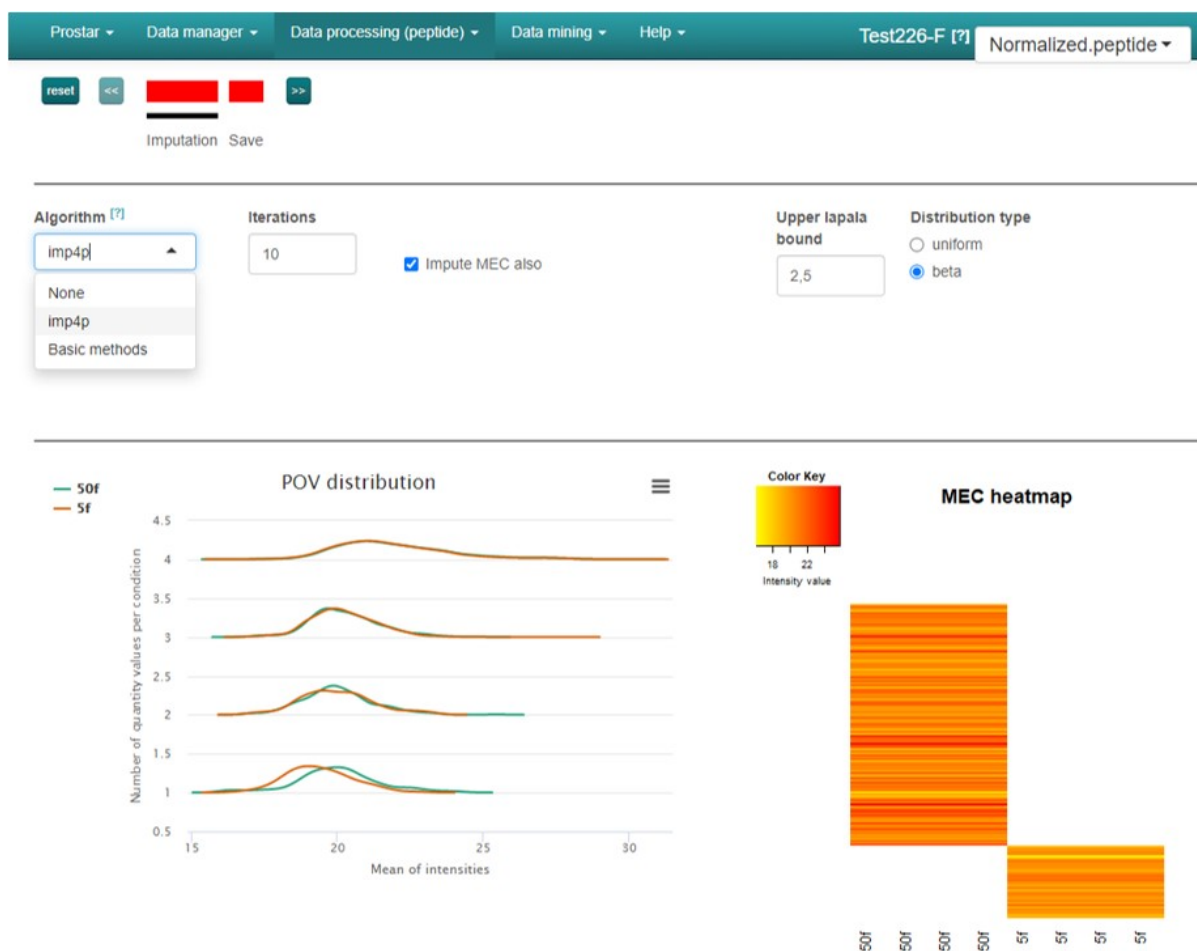


Figure 12: Peptide imputation tab: Result of imputation of both the “Partially Observed Values” and the “Missing in the Entire Condition” missing values.

3.13 Hypothesis Testing for aggregated protein dataset

Hypothesis testing can be conducted regardless of the version of the protein dataset (either aggregated, filtered, normalized or imputed) as long as it does not contain missing values. To do so, click on [Data processing \(protein\)](#) [Hypothesis testing](#). The steps are similar to those that can be applied to a peptide dataset as part of a peptidomics strategy (see Subheading 3.10):

1. Choose the test contrasts: In case of two conditions to compare, there is only one possible contrast. However, in case of $N > 2$ conditions, several pairwise contrasts are possible. Notably, it is possible to perform N tests of the "1vsAll" type, or $N(N-1)/2$ tests of the "1vs1" type.
2. Choose the type of statistical test, between limma [15] or t-test (either Welch or Student). This makes appear a density plot of the logarithmized fold-change (logFC) (as many density curves on the plot as contrasts).
3. Thanks to the logFC density plot, tune the logFC threshold (see Note 47). Refresh the plot to view the impact of the chosen threshold by clicking on [Perform log FC plot](#). The corresponding logFC value is indicated for convenience.
4. Run the tests by clicking on [Perform log FC plot](#), move forward to the [Save](#) tab and click [Save significance test](#) to preserve the results (i.e., all the computed p-values). Then, a new dataset is created, containing the p-values and logFC cutoff for the desired contrasts. It can be explored as any other dataset ([Data mining](#) [Differential analysis](#) menu).
5. Check that a new version appears in the dataset version dropdown-menu, referred to as "HypothesisTest.protein".

3.14 Differential Analysis

The differential analysis of a "Hypothesis.Test" version of a protein level dataset can be conducted in the same way as for a peptide dataset as part of a peptidomics strategy (see Subheading 3.10). To do so, click on

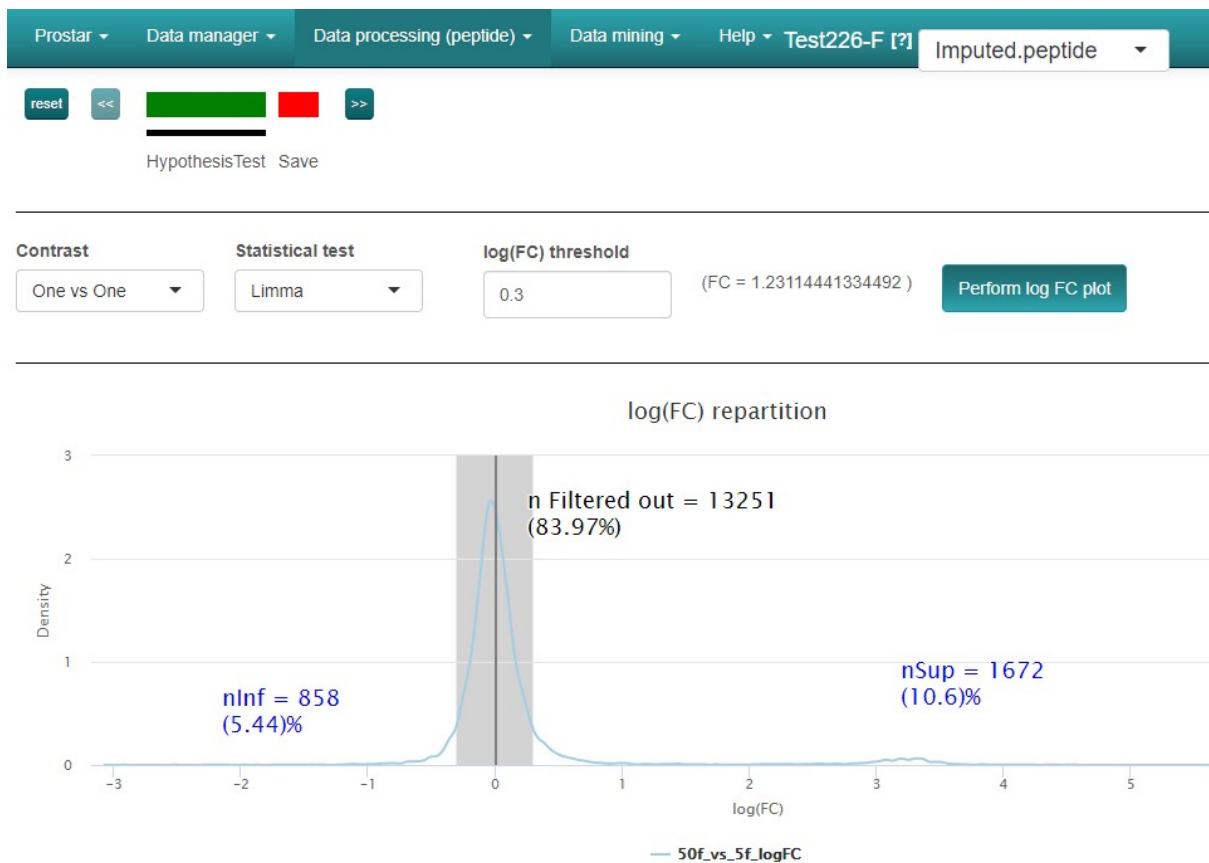


Figure 13: Tuning of the null hypothesis significance testing (to prepare the differential analysis at peptide level).

Data mining >> Differential Analysis :

1. Select a pairwise comparison of interest. The corresponding volcano plot is displayed. At this stage, it is possible to specify the field of the tooltip to appear on the Volcano (for example the protein identifier for immediate recognition of the proteins of interest).
2. Possibly, swap the logFC axis with the corresponding tick box, depending on layout preferences.
3. If an imputation step for missing values has been performed at the peptide (or protein) level, some proteins may have an excellent p-value, while a too great proportion of their intensity values (within the two conditions of interest in this comparison) are in fact imputed/recovered values, so that they are not trustworthy. To avoid such proteins become false discoveries, it is possible to discard them (by forcing their p-value to 1). To do so, fill in the last parameters of the right hand side menu, which are similar to the filtering options described earlier (see Subheading 3.6, Step 2).
4. Click on `Perform p-value push` and move on to the `pvalue calibration` tab.
5. Tune the calibration method, as indicated in [18] as well as in Prostar user manual.
6. Move on to the next tab and adjust the FDR threshold (see Figure 16). Clicking on one protein in the volcano plot makes it appears in the lower table.
7. Save any plot/table of interest and move on to the next tab (`Summary`) to have a comprehensive overview of the differential analysis parameters.
8. Possibly, go back to step 1 to process another pairwise comparison.

It is possible to go on with the current protein list, and to explore the underlying functional profiles using `GO analysis` proposed in the `Data mining` menu [7].

4 Notes

1. Prostar versions which are posterior to 1.24 may slightly differ from what is described in this protocol. However, the general spirit of the graphical user interface remains unchanged allowing any user accustomed to an earlier version to easily adapt to a newer version.

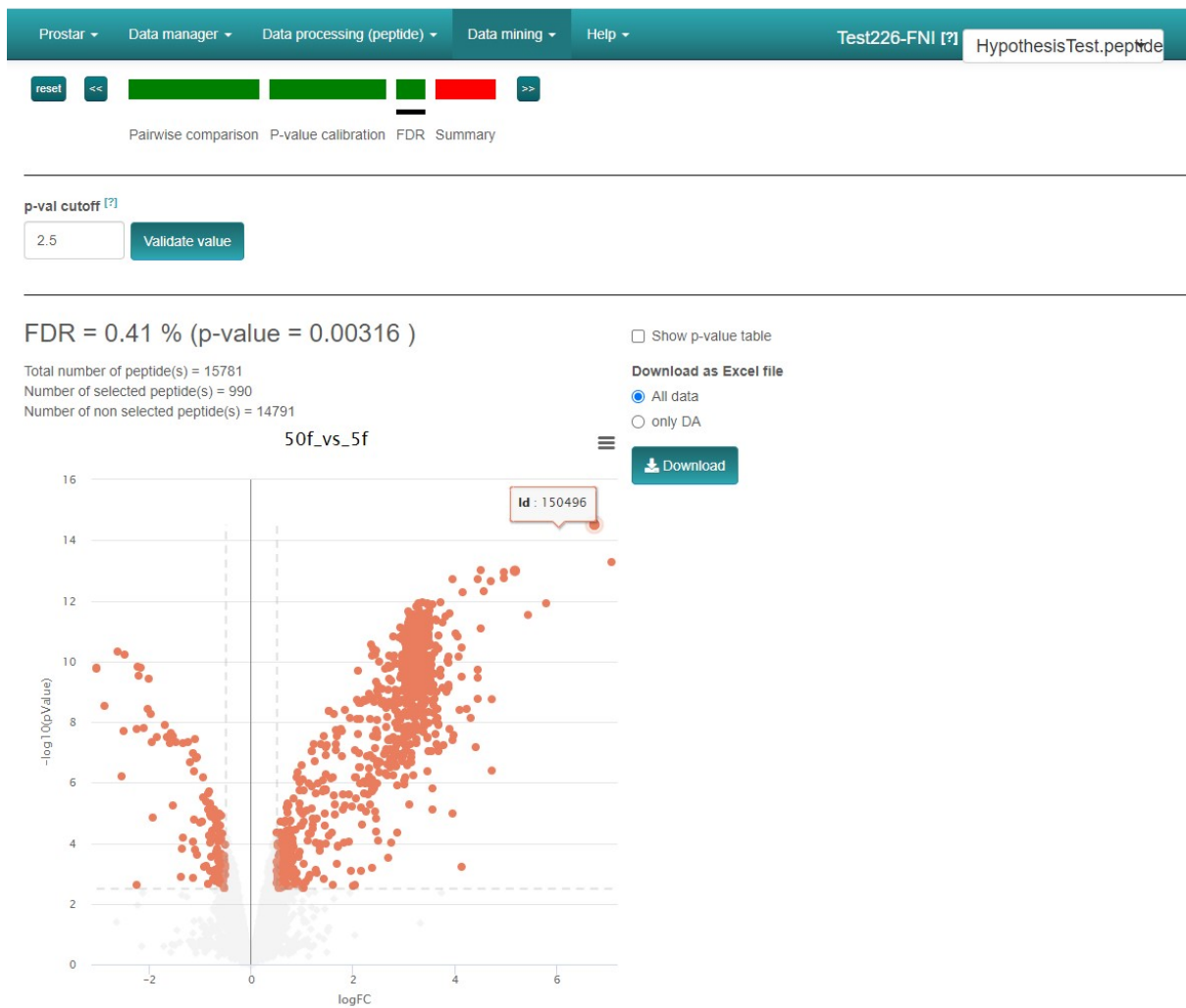


Figure 14: Volcano plot to visualize the differentially abundant peptides according to a user-specified FDR.

2. We advise to use the latest version of R and to make regular updates, as it will guarantee the compatibility with the latest Prostar developments.

3. To ensure full compatibility and debugging, the zip file is often available later than the corresponding Bioconductor release (up to one month later).

4. With only two replicates per condition, the computations are tractable. It does not mean that statistical validity is guaranteed. Classically, 3 replicates per condition are considered a minimum in case of a controlled experiment with a small variability, mainly issuing from technical or analytical repetitions. Analysis of complex proteomes between conditions with a large biological variability requires more replicates per condition (5 to 10).

5. To reload a dataset that has previously been stored as an MSnset file, go to [Data manager](#) [Open MSnset file](#) and simply browse the file system to find the desired file.

6. Before uploading a real dataset, any user can test Prostar thanks to the demo mode. This mode provides a direct access to the DAPARdata packages where some toy datasets are available, either in tabular or MSnset formats. Concretely, the demo mode is accessible in the [Data manager](#) menu: on the corresponding tab, a dropdown-menu lists the datasets that are available through DAPARdata. After selecting a peptide-level dataset, click on [Load demo dataset](#).

7. The DAPARdata package also contains tabular versions (in txt format) of the datasets available in the demo mode. Thus, it is also possible to test the import/export/save/load functions of Prostar with these toy datasets. Concretely, one simply has to import them from the folder where the R packages are installed, in the following sub-folder: `../R/R-4.x.x/library/DAPARdata/extdata`. Note that each dataset is also available in the MSnset format, but these datasets should not be considered to test conversion functions from/to tabular formats.

Prostar ▾ Data manager ▾ Data processing (peptide) ▾ Data mining ▾ Help ▾ Test226-F [?] Imputed.peptide ▾

reset << [Color bar] >>

Aggregation Add metadata Save

Include shared peptides [?]

No

Yes (as protein specific)

Yes (redistribution)

Consider

all peptides

N most abundant

Operator

Mean

Sum

Perform aggregation

Only specific peptides

All (specific & shared) peptides

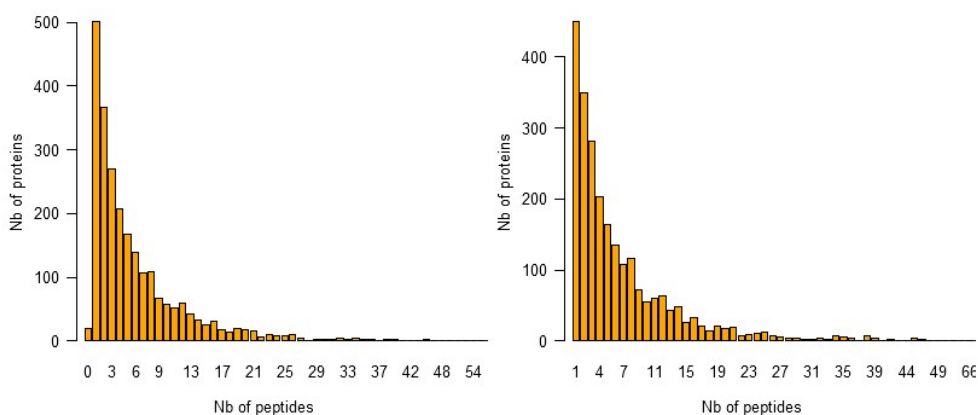


Figure 15: Options in the **Aggregation** tab.

8. It is important to specify the quantification software since this choice conditions the way with which Prostar recodes the origin of quantification. In the current version, the choice is limited to Proline and MaxQuant software. With MaxQuant, the file "peptides.txt" is appropriate. Notably, it contains a column referred to as "Identification_type", which will be instrumental (as it contains the "by Matching" and "by MS/MS" qualifiers, *see* Subheading 3.2, **Step 13**). With Proline, the export of the **Display Abundances** >> **Peptides** window is appropriate (*see* **Chapter 4**). The user should be careful to include the "Quant. PSMs count" columns (checked by default) which will determine the origin of each peptide quantification in each sample: The numerical value amounts to the number of MS/MS identifications so that a non-zero value indicates at least one MS/MS evidence; on the contrary, "0" means the identification has been transferred to the XIC from another run (In other words, it conveys the same information as the "Identification_type" column in MaxQuant output, and it will be used similarly, *see* Subheading 3.2, **Step 13**).

9. Prostar cannot process non-log-transformed data. Thus, do not try to cheat the software by indicating data on their original scale are log-transformed.

10. If a peptide belongs to several proteins, their respective IDs must be separated by a comma.

11. Here, "mapping" refers to the the abundance value being recovered via the **Match Between Run** option with MaxQuant, or the **Cross Assignment** option with Proline.

12. However, the content of the specified columns are not checked, so that it is the user's responsibility to select the correct ones.

13. In case of difficulty, either to choose the adapted design hierarchy or to fill the table design, it is possible to click on the interrogation mark beside the sentence "Choose the type of experimental design and complete it accordingly". Except for flat design, which are automatically defined, it displays an example of the corresponding design. It is possible to rely on this example to precisely fill the design table.

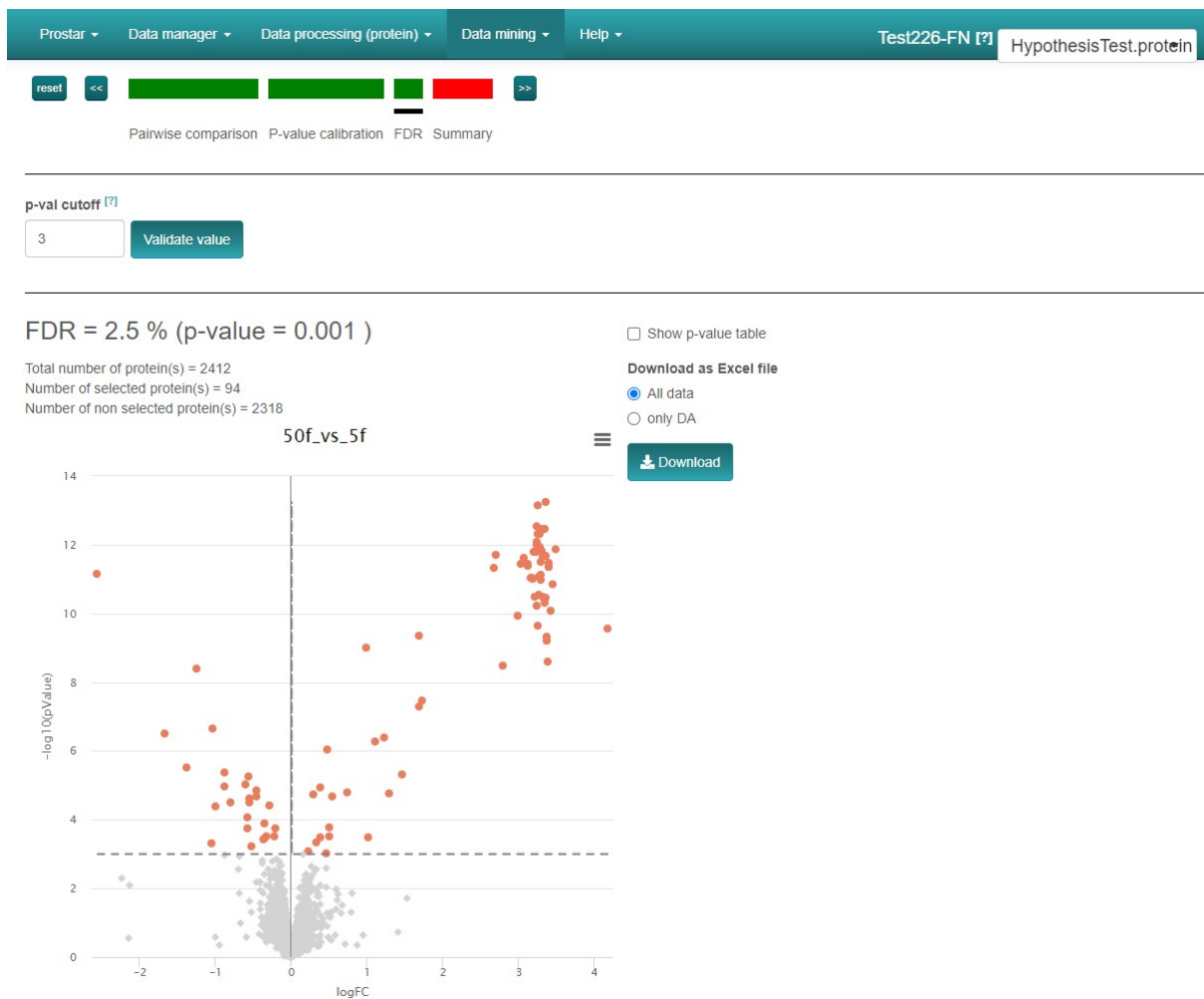


Figure 16: Volcano plot to visualize the differentially abundant proteins according to a user-specified FDR.

14. Alternatively, it is possible to export data as excel spreadsheets or as a zip containing text files. This has no interest in case of a preliminary export; however, it may be useful to share a dataset once the statistical analysis is completed.


15. The user can download the plots showed in Prostar by right-clicking on the plot. A contextual menu appears and let the user choose either `Save image as` or `Copy image`. In the latter case, the user has to paste the image in appropriate software.

16. It is essential to regularly go back to these tabs, so as to check that each processing step has produced the expected results.

17. Cell metadata are metadata associated to each cell of the quantitation table, and encoding the nature of the quantification value for each peptide and each sample. For a peptide, there are three main types of quantification, each contain several subtypes: (i) Quantitative data ("quanti"): The peptide under consideration is either directly "identified" (e.g., "By MS/MS" tag in MaxQuant) or "recovered" (indirectly found, e.g., "By matching" tag in MaxQuant). If the very status ("identified" or "recovered") is unknown, it is simply labelled "quanti". (ii) Missing value ("missing"): The peptide may be a Partially Observed Value ("missing POV") or Missing in Entire Condition ("missing MEC"). (iii) Imputed value ("imputed"): Imputed value are of two types, "imputed POV" or "imputed MEC".

18. Computing the heatmap and the dendrogram may be very computationally demanding, depending on the dataset.

19. As is, the full-scale plot is often difficult to read, as the high variances are concentrated on the lower intensity values. For this reason, the plot is pre-zoomed and focused on the lower intensity values. It is possible to reset the zoom by clicking on `Reset zoom` and to interactively zoom in on any part of the plot by clicking and dragging on the plot.

20. This functionality is available only if for a peptide-level proteomics analysis. In fact, for a peptidomic analysis, the parent protein column was not necessarily defined (*see* Subheading 3.2, **Step 10**), making peptide-protein relationship analysis useless.
21. The first column, named "Proteins Ids" corresponds to the value selected in the dropdown-menu called `protein IDs` (*see* Subheading 3.2, **Step 10**), while the column named "Peptides Ids" corresponds to the value selected in the dropdown-menu `ID definition` in the loading process of a peptide dataset (*see* Subheading 3.2, **Step 9**).
22. The color legend is the same as in the `Data Explorer` tab (*see* Subheading 3.4, **Step 3**).
23. The latter case may be more adapted in case of unbalanced designs, *i.e.*, datasets in which conditions have not the same number of samples.
24. Peptide metadata are additional columns of the dataset which contains metadata attached to each line, *i.e.*, each peptide, as opposed to cell metadata.
25. To work correctly, the selected column must contain information encoded as a string of characters. For each peptide, the beginning of the corresponding string is compared to a given prefix. If the prefix matches, the peptide is filtered out. Otherwise, it is retained. Note that the filter only operates a prefix search (at the beginning of the string), not a general tag match search (anywhere in the string). Similarly, filters based on regular expressions are not implemented.
26. In datasets outputted from MaxQuant, metadata indicates under a binary form which peptides are reversed sequences (resulting from a target-decoy approach) and which are potential contaminants. Both of them are indicated by a "+" in the corresponding column (the other peptides having an NA instead). It is thus possible to filter both reversed and contaminants out by indicating "+" as the prefix filter. However, if adequately encoded, filtering on other type of information is possible.
27. If one has no idea of the prefixes, it is possible to switch to the `Descriptive Statistics`  `Data Explorer` (*see* Subheading 3.4, **Step 3**), so as to visualize the corresponding metadata.
28. It is not possible to keep in parallel several datasets or multiple versions of a dataset at a particular level (for instance, a dataset filtered using various rules). Thus, if one goes on with the next processing steps on an older dataset, or if one goes back to a previous step and restart it, the new results will overwrite the previously saved ones at this same step, without updating other downstream processing, leading to possible inconsistencies.
29. This list corresponds to the normalization methods available in Prostar version 1.24; future versions will possibly propose slightly different methods.
30. This normalization method should not be confused with Global quantile alignment.
31. It should be noted that the choice of a normalization method and its tuning is highly data dependent, so that a single protocol cannot be proposed. The data analyst should gather expertise on the normalization methods, so as to be able to choose soundly. Thus, we advise to refer to Prostar user manual, as well as to the literature describing the normalization methods (to be found in the `Help` menu).
32. As a result, all the missing values are either POV or MEC. Moreover, for a given peptide across several conditions, the missing values can be both POV and MEC, even though within a same condition they are all of the same type.
33. The routine used for imputation with `imp4p` depends on the final MNAR or MCAR diagnosis.
34. Considering that imputation of MECs is always risky, the checkbox lets the alternative to impute MECs at the protein level (*i.e.*, after the aggregation process), despite the risk that the missing value distribution makes the aggregation impossible (*see* Subheading 3.11).
35. If "Det. quantile" is used, we advise to use a small quantile of the intensity distribution to define the imputation value, for instance, 1–5% depending on the stringency you want to apply on peptides quantified only in one condition of a pairwise comparison. In case of a dataset with too few peptides, the lower quantile may amount to instable values. In such a case, we advise to use a larger quantile value (for instance 10% or 20%) but to use a smaller multiplying factor so as to keep the imputation value reasonably small with respect to the detection limit.
36. During the aggregation process, it may be necessary to aggregate quantified peptides and missing valued peptides. As in such cases it is impossible to correctly estimate the protein abundance, the aggregation is cancelled.
37. When exporting the data in the Microsoft Excel format, imputed values are displayed in a colored cell so that their origin POV and MEC can easily be distinguished.
38. The reason why combining missing and non-missing values is not authorized is given in [17]: Broadly speaking, it would amount to implicitly impute the missing value(s) with a value that does not change the aggregated value.

39. The rules established in the current 1.24 version to manage the aggregation of missing values may be refined in future versions.
40. If shared peptides are not included in the aggregation step, one faces the risk of losing some proteins (those which do not have a single specific peptide). This is visible on the "Only specific peptides" barplot where a few proteins may appear to have 0 peptides.
41. Thus, the intensity of each shared peptide is fully accounted for all the proteins it belongs to. This straightforward solution is obviously not the most rigorous aggregation model.
42. To achieve this proportional redistribution, the intensities of shared peptides are iteratively split and shared proportionally to the parent protein abundances. The iteration goes on until the redistribution process does not change the protein abundances any more.
43. The reason why the sum operator is not allowed is the following: an iterative sum would endlessly inflate the protein abundance and convergence would never be reached.
44. By consulting the Excel export, you will notice several columns that provide information on the number and nature of the aggregated peptides, either at the whole experiment level, or at each single replicate level. Moreover, the missing values are colored according to their type (POV and MEC, if any) using the same color code as in Prostar.
45. Each protein abundance value is associated with a cell metadata directly inherited from those of the peptides. In general, it contains the same label as all the children peptides, namely "identified", "recovered", "imputed POV" or "imputed MEC". However, if all the children peptides do not have the same labels, more generic labels are used: a mix of "identified" and "recovered" peptides leads to a protein-level quantitation value, referred to as "quanti". In the case of peptides tagged with a mix of "imputed POV and "imputed MEC, the protein is simply referred to as "imputed". Finally, in the case of a mix of quantified peptides (*i.e.*, "identified" or "recovered"), and of imputed peptides (*i.e.*, "imputed POV" or "imputed MEC" or "imputed"), the protein is labelled as "combined".
46. Depending on the chosen option: "Include shared peptides" set to "No" (*see* Subheading 3.11, **Step 2**).
47. It is important to tune the logFC threshold to a small enough value, to avoid discarding too many proteins [19, 20]: it is essential to keep enough remaining proteins for the next coming FDR computation step (*see* Subheading 3.14), as (1) FDR estimation is more reliable with many proteins, (2) FDR, which relates to a percentage, does not make sense on too few proteins.

Acknowledgement

This work was supported by grants from Agence Nationale de la Recherche under: ProFI project (Proteomics French Infrastructure, ANR-10-INBS-08), GRAL project, a program from the Chemistry Biology Health (CBH) Graduate School of University Grenoble Alpes (ANR-17-EURE-0003), DATA@UGA and SYMER projects (ANR-15-IDEX-02) and MIAI @ Grenoble Alpes (ANR-19-P3IA-0003).

References

- [1] Zhang Y, Fonslow BR, Shan B, Baek MC, Yates III JR (2013) Protein analysis by shotgun/bottom-up proteomics. *Chemical reviews* 113(4):2343–2394, <https://doi.org/10.1021/cr3003533>
- [2] Ong SE, Foster LJ, Mann M (2003) Mass spectrometric-based approaches in quantitative proteomics. *Methods* 29(2):124–130, [https://doi.org/10.1016/s1046-2023\(02\)00303-1](https://doi.org/10.1016/s1046-2023(02)00303-1)
- [3] Schwanhäusser B, Busse D, Li N, Dittmar G, Schuchhardt J, Wolf J, Chen W, Selbach M (2011) Global quantification of mammalian gene expression control. *Nature* 473(7347):337–342, <https://doi.org/10.1038/nature10098>
- [4] Beeley C (2013) Web application development with R using Shiny. Packt Publishing Ltd, <https://github.com/PacktPublishing/Web-Application-Development-with-R-Using-Shiny-third-edition>
- [5] Wieczorek S, Combes F, Lazar C, Gai Gianetto Q, Gatto L, Dorffer A, Hesse AM, Coute Y, Ferro M, Bruley C, Burger T (2017) Dapar & prostar: software to perform statistical analyses in quantitative discovery proteomics. *Bioinformatics* 33(1):135–136, <https://doi.org/10.1093/bioinformatics/btw580>
- [6] Goeminne LJ, Argentini A, Martens L, Clement L (2015) Summarization vs peptide-based models in label-free quantitative proteomics: performance, pitfalls, and data analysis guidelines. *Journal of proteome research* 14(6):2457–2465, <https://doi.org/10.1021/pr501223t>

- [7] Wiczorek S, Combes F, Borges H, Burger T (2019) Protein-level statistical analysis of quantitative label-free proteomics data with prostar. In: *Proteomics for Biomarker Discovery*, Springer, pp 225–246, https://doi.org/10.1007/978-1-4939-9164-8_15
- [8] Gatto L, Lilley KS (2012) MSnbase-an R/Bioconductor package for isobaric tagged mass spectrometry data visualization, processing and quantitation. *Bioinformatics* 28(2):288–289, <https://doi.org/10.1093/bioinformatics/btr645>
- [9] Wiczorek S, Combes F, Burger T (2018) DAPAR and ProStaR user manual. In: *Bioconductor*. https://www.bioconductor.org/packages/release/bioc/vignettes/Prostar/inst/doc/Prostar_UserManual.pdf?attredirects=0
- [10] RStudio Team (2015) RStudio: Integrated Development for R. RStudio, Inc., Boston, MA. <http://www.rstudio.com/>
- [11] Cox J, Mann M (2008) Maxquant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nature Biotechnology* 26:1367–1372, <https://doi.org/10.1038/nbt.1511>
- [12] Bouyssié D, Hesse AM, Mouton-Barbosa E, Rompais M, Macron C, Carapito C, Gonzalez de Peredo A, Couté Y, Dupierris V, Burel A, et al. (2020) Proline: an efficient and user-friendly software suite for large-scale proteomics. *Bioinformatics* 36(10):3148–3155, <https://doi.org/10.1093/bioinformatics/btaa118>
- [13] R-Core-Team (2020) stats package. URL <https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/hclust>, r package version 3.6.2
- [14] Cleveland WS (1979) Robust locally weighted regression and smoothing scatterplots. *Journal of the American statistical association* 74(368):829–836, <https://doi.org/10.1080/01621459.1979.10481038>
- [15] Smyth GK (2005) Limma: linear models for microarray data. In: *Bioinformatics and computational biology solutions using R and Bioconductor*, Springer, pp 397–420, https://doi.org/10.1007/0-387-29362-0_23
- [16] Huber W, Von Heydebreck A, Sültmann H, Poustka A, Vingron M (2002) Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics* 18(suppl_1):S96–S104, https://doi.org/10.1093/bioinformatics/18.suppl_1.S96
- [17] Lazar C, Gatto L, Ferro M, Bruley C, Burger T (2016) Accounting for the multiple natures of missing values in label-free quantitative proteomics data sets to compare imputation strategies. *Journal of proteome research* 15(4):1116–1125, <https://doi.org/10.1021/acs.jproteome.5b00981>
- [18] Giai Gianetto Q, Combes F, Ramus C, Bruley C, Couté Y, Burger T (2016) Calibration plot for proteomics: A graphical tool to visually check the assumptions underlying fdr control in quantitative experiments. *Proteomics* 16(1):29–32, <https://doi.org/10.1002/pmic.201500189>
- [19] Giai Gianetto Q, Couté Y, Bruley C, Burger T (2016) Uses and misuses of the fudge factor in quantitative discovery proteomics. *Proteomics* 16(14):1955–1960, <https://doi.org/10.1002/pmic.201600132>
- [20] Wiczorek S, Gianetto QG, Burger T (2019) Five simple yet essential steps to correctly estimate the rate of false differentially abundant proteins in mass spectrometry analyses. *Journal of proteomics* 207:103441, <https://doi.org/10.1016/j.jprot.2019.103441>