ProteoModlR: An R suite for quantitative proteomics pathway modeling

Mojdeh Shakiba, Paolo Cifani and Alex Kentsis

December 7, 2015

Contents

1 Overview

ProteoModlR is an open-sourced R suite for quantitative mass spectrometry analysis of the relative concentration of proteins and the stoichiometry of post-translational chemical modifications.

1.1 Applicability

Due to its modular design and flexible analysis pipeline, ProteoModlR allows for seamless integration with existing proteomics software, such as MaxQuant and Skyline, as well as with statistical and pathway analysis tools. It facilitates analysis and visualization of quantitative proteomics data enabling researchers to assess differential activation of functional cellular processes.

1.2 Functionalities

ProteoModlR performs three user-customizable functions: *quality control*, *normalization* and *analysis*. In the first step, *quality control*, the data is checked for the presence of all specified columns in the correct order. Data is filtered to remove peptides with invalid intensity measurement (i.e. missing values or zero intensity across all conditions and all runs). Next, the data is classified based on the possibility to perform exact or approximate calculations of the relative protein abundance and protein stoichiometry according to the modification specified by the user (e.g. phosphorylation, acetylation, etc.). *Abundances* based on MS intensity measurements correspond to the amount of a given peptide. *Exact* abundance is calculated using the intensities from peptides not bearing the modification of interest. If all available peptides from a protein present the selected modification, only *approximate* abundance is computed using modified peptides to infer changes in abundance. *Stoichiometry* is defined for any given peptide and modification as the fraction of the total peptide amount bearing the selected modification. If there are intensity measurements for both the modified and unmodified forms of a given peptide, *exact* stoichiometry is computed. On the other hand, if the unmodified peptide is not present in the dataset, *approximate* stoichiometry is calculated instead based on the signal intensity of the modified peptide relative to other unmodified peptides from the same protein (if available) or as the change in the abundance of the modified peptide itself. The filtered and classified data are exported as a comma-separated value (CSV) file into the working directory (*filtered_timestamp.csv* file), along with a bar graph summarizing the number of peptides and proteins remaining after each step of the quality control module (*Filtering_tracker.pdf*).

The second step is *normalization.* Here, the user can choose from three normalization options: normalization by isotopologue, normalization to internal reference peptide(s), and/or normalization to total ion current. If an isotopically labeled standard is used, *Normalization by isotopologue* equalizes the intensity of the chosen isotope across the samples, keeping the ratio of the heavy-labeled to the light-labeled forms of each peptide unchanged. *normalization to internal reference peptide(s)* divides the intensity of each peptide by the geometric mean of the intensities of the specified reference peptides [1]. In order to guide better selection of reference peptides that remain unchanged across experimental conditions, a figure of the abundance of each reference peptide acorss the various conditions is generated and saved in the working directory (*reference_peptideSequence.pdf*). If *normalization to total ion current* is selected, the intensity of each peptide is divided by the total ion current for that experiment. If no normalization is selected, the data are returned unchanged. Otherwise, the normalized values replace the input intensities in the data and the data are exported as a CSV file into the working directory (*Normalized_timestamp.csv*).

The third step is *analysis*, where peptide site occupancies and/or abundances are calculated. Site occupancy refers to the fraction of the modified peptide ($\frac{modified}{unmodified+modified}$). Exact site occupancy calculations can be made in the case where the unmodified form of the peptide has been measured, while approximate calculation can be done by taking the ratio of the modified peptide to another peptide in the same protein, or by simply looking at the change in the abundance of the modified peptide itself. Abundance of a given peptide is obtained using the measured intensities: exact abundance is indicated by measuring the intensity of peptide sequences with no modifications, while approximate abundance is obtained using modified peptides.

Depending on the user input on whether to restrict the analysis to exact calculations or to allow for approximate output, the software selects the appropriate peptides based on the classification performed in quality control. In addition, if the user selects a reference state, all calculations are reported as log_2 fold-change relative to the reference state. The analyzed data is exported as a CSV file into the working directory (*Exact/Approx_Abund_timestamp.csv*).

1.3 Integration with other computational tools

ProteoModlR accepts as input any CSV formatted file with the appropriate columns. This type of file can be produced by a variety of programs, such as Skyline and MaxQuant [1-3]. The output of ProteoModlR is also in a CSV format and is thus compatible with pathway analysis tools, such as Cytoscape (http://www.cytoscape.org),and with statistical analysis software for downstream analyses [4-5].

1.4 Availability

ProteoModlR and its documentation are available for download at
http://github.com/kentsisresearchgroup/ProteoModlR.

1.5 Software Requirements

ProteoModlR is an R suite and requires the following R packages, all of which are available through CRAN (https://cran.r-project.org): 'plyr','ggplot2','reshape2'.

2 Input File Format

The input file must contain 12 columns in the following order: `Protein`, `Peptide`, `Pathway`, `Modification`, `Gene.Name`, `Position`, `Protein.Name`, `Condition`, `PatientID`, `Label`, `Run`, `Intensity`, even if the content of the columns are left empty. Skyline users are encouraged to customize the output to fit this format. MaxQuant output (*evidence.txt* table) can be easily re-formatted using external CSV editors, so that a single intensity value per row is provided.

1. `Protein`: This column stores the protein identifier, preferably as UniProt ID [*www.uniprot.org*]. If multiple identifiers are listed, the program takes the first 6 characters corresponding to the base of the first UniProt ID.

2. `Peptide`: This column stores the peptide sequence in single letter code.

3. `Pathway`: This column may store the pathway identifier to which each protein corresponds. This classification is entirely for post-analysis purposes and will simply follow the remainder of the data to the output. It can be used afterwards for pathway analysis in other software.

4. `Modification`: This column stores the modification(s) for a given peptide. Use of UniMod nomenclature [*www.unimod.org*] is highly recommended. Note that the software tolerates minor variations in the modification identifiers. The use of dash in the modification ID (e.g. di-methyl instead of dimethyl) should be avoided. Peptides with no detected modification must be labeled "unmodified".

5. `Position`: This column may store the position within the peptide to which the modification in the corresponding column belongs.

6. `Gene.Name`: This column may store the gene name or gene ontology (GO) identifier for each protein.

7. `Protein.Name`: This column may store the protein name.

8. `Condition`: This column stores the experimental condition to which each peptide/protein corresponds. For instance, in a case-control study, this will be "healthy" or "diseased", while for a time-point study this will be "T1", "T2, etc.

9. `PatientID`: This column may store a unique patient/sample identifier.

10. `Label`: This column stores the isotopologue labeling for each peptide (e.g. "H" for heavy-labeled and "L" for light-labeled isotopologues). For label-free datasets, label all entries as "L".

11. Run: This column stores the replicate ID for a given sample. For example if an experiment was performed in 3 technical replicates, each measurement may be labeled 1 to 3 in this column. If a single replicate is analyzed, label all entries as ”1”.

12. Intensity: This column stores the original, untransformed measure of abundance (i.e. MS intensity) as calculated using other software such as Skyline.

3 Workflow

3.1 Quality Control and Normalization

Once the data is made available in the working environment, the following code can be used to call both the *normalization* and the *quality control* functions with the desired inputs:

```
Normalize(data, iso.norm = "", internal.norm = "", tot.current=F, mod="")
```

Normalize takes in the data as a data frame. While the data file provided does not need to be exported from any specific program, the data should contain the 12 columns covered in Section 2. In addition to the data, the user can specify 5 arguments that determine the type of normalization to be performed (if any):

1. iso.norm takes a character string corresponding to the reference isotopologue to which the other isotopologue will be normalized (e.g. iso.norm = "L"). If no isotopologue normalization is desired, enter ”NA”.

2. internal.norm takes in a vector of character strings corresponding to the peptide sequence of the reference peptides to which other peptides are normalized (e.g. internal.norm = "ATDVIVP"). If more than one reference peptide is indicated (e.g. internal.norm = c("ATDVIVP","AAATDVI")), a geometric mean is taken [6]. If no reference peptide normalization is desired, enter ”NA”.

3. tot.current takes in a boolean input indicating whether or not normalization to total ion current is to be made (e.g. tot.current = T); mod takes in a string corresponding to the protein modification of interest (e.g. mod="phospho").

Any of the normalization options can be left unused but a modification must be selected for any analysis to be performed by the program. The function Normalize will internally call the function for quality control and perform the filtering and classification needed for the next step. The data outputted from both the quality control step and the normalization step is saved in the working directory (*Normalized_time_stamp.csv* file).

3.2 Analysis

The function Analyze calculates the approximate and/or exact site occupancy and/or abundance of the peptides:

```
Analyze(data, stoich="", abund="", ref.state="")
```

The first argument in the function call corresponds to the output of the function `Normalize`. `stoich` takes in either `"Exact"` or `"Approximate"`, depending on the user's preference for calculating the site occupancy of peptides for which either exact or approximate calculations are possible. `abund` also takes in either `"Exact"` or `"Approximate"`, depending on the user's preference for exact or approximate abundance calculations. It is important to note that the decision as to whether exact or approximate calculations can be performed for a given peptide is made in the quality control stage, depending on the modification of a peptide and the existence of its unmodified form in the raw dataset. As such, if for instance exact abundance calculations are selected by the user, peptides for which only approximate abundance can be calculated are eliminated in the output of the function `Analyze`. The last argument in the `Analyze` function call is `ref.state`, which provides the user with the option to normalize to a reference condition (e.g. `ref.state="Disease"`).

4 References

1. Cox J, Mann M: "MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification". Nat Biotechnol 2008, 26:13671372.
2. Cox J, Neuhauser N, Michalski A, Scheltema RA, Olsen JV, Mann M: "Andromeda: a peptide search engine integrated into the MaxQuant environment". J Proteome Res 2011, 10:17941805.
3. MacLean B, Tomazela DM, Shulman N, Chambers M, Finney GL, Frewen B, Kern R, Tabb DL, Liebler DC, MacCoss MJ: "Sky-line: an open source document editor for creating and analyzing targeted proteomics experiments". Bioinformatics 2010, 26:966968.
4. Choi M, Chang C-Y, Clough T, Broudy D, Killeen T, Mac-Lean B, Vitek O: "MSstats: an R package for statistical analysis of quantitative mass spectrometry-based proteomic experiments". Bio-informatics 2014, 30:25242526.
5. Linding R, Jensen LJ, Pasculescu A, Olhovsky M, Colwill K, Bork P, Yaffe MB, Pawson T: "NetworKIN: a resource for explor-ing cellular phosphorylation networks". Nucleic Acids Res 2008, 36(Database issue):D6959.
6. Vandesompele J, De Preter K, Pattyn F, Poppe B, Van Roy N, De Paepe A, Speleman F: "Accurate normalization of real-time quantitative RT-PCR data by geometric averaging of multiple internal control genes", Genome Biol. 2002; 3(7).