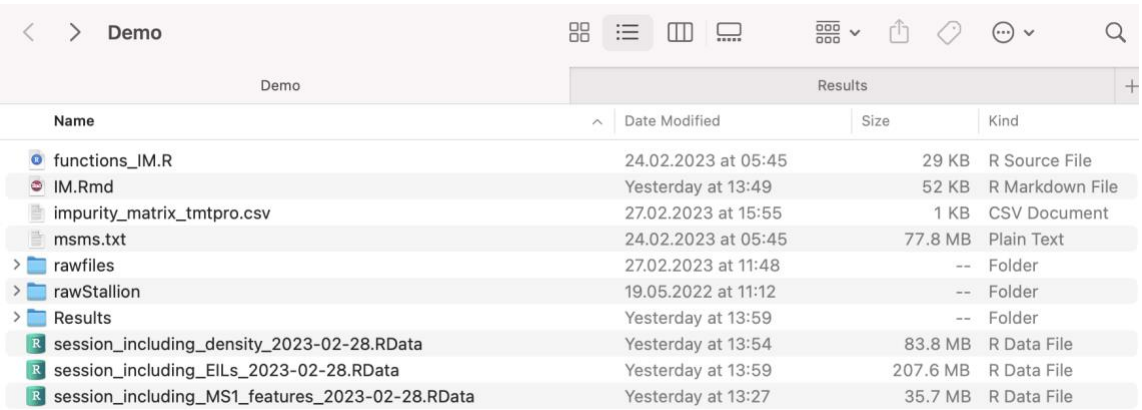
Site-To-Protein Normalization in Multiplex Proteomics

Overview

This document serves as a guide to the demo for the site-to-protein normalization workflow on GitHub. The demo comes with its own dataset based on which the entire workflow can be experienced from start to finish. I recommend running the demo first before applying the workflow to your own data – this allows you to get familiar with the required data input, the script's many parameters as well as the intermediate and final data output. We will first look at the required setup for running the workflow, and then how to run it. Finally, there is a description of the generated output. Currently, this workflow only supports MaxQuant output.

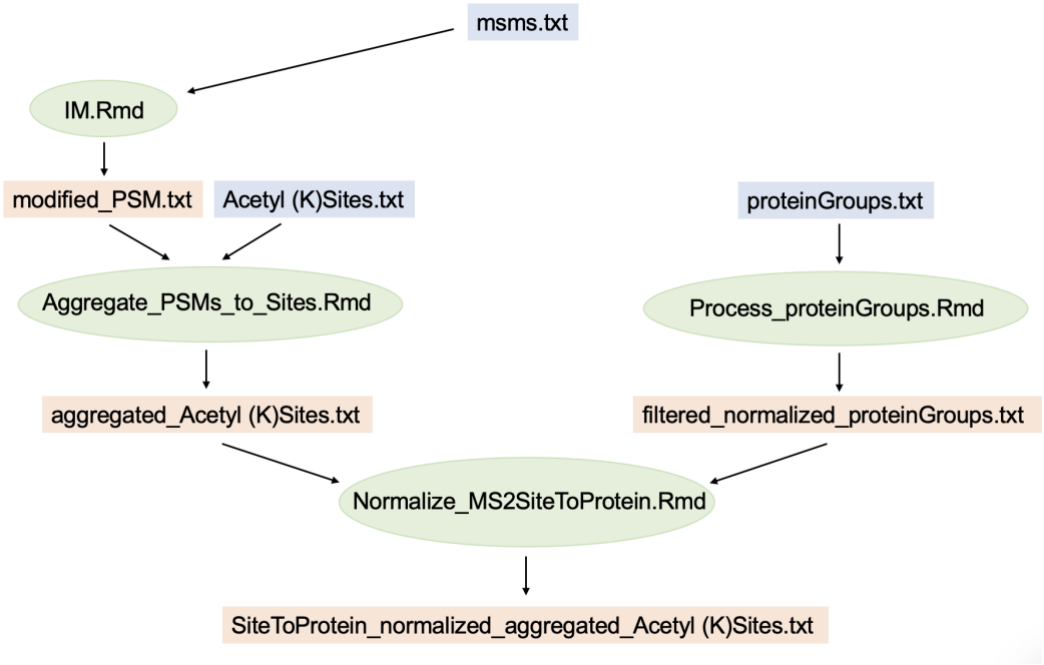
Required Setup

In general, the workflow of site-to-protein normalization builds on the workflow of interference modeling in multiplex proteomics which provides the necessary input data. This demo therefore starts where the demo of interference modeling left off (see GitHub repository named "InterferenceModeling_in_MultiplexProteomics"). That said, the required output of the first demo (named "modified_PSM.txt") is also available for download in the Demo folder of this GitHub repository, if you chose to skip the interference modeling part. For practical reasons, we assume that we just finished running the demo for interference modeling, which left us with the following data:



In the "Results" folder, we find the main output of the interference modeling workflow named "modified_PSM.txt". If you did not run the demo for interference modeling, download it now (contained in the Demo folder of this repository), and store it in a folder named "Results".

Before we continue towards unbiased site-to-protein normalization, we should first consider where we are and what we actually want to do. The table "modified_PSM.txt" contains PSM-wise information on quantified acetyl(K) peptides and comes with some extra columns, including a column named EIL (Estimated Interference Level) as well as normalized reporter ion intensities. The plan is to first carry over this PSM-wise information to acetyl site level (as given by MaxQuant's "Acetyl (K)Sites.txt") via aggregation, and then perform unbiased site-to-protein normalization by accounting for reporter ion interference on both site and protein level. In this demo, we will go through the workflow of normalizing MS2-quantified site abundances to MS3-quantified protein abundances (more specifically, FAIMS-MS3 quantification of proteins with real-time-search on), which requires us to follow this specific workflow, as shown in the README of the GitHub repository:



In blue: MaxQuant output; in green: R-scripts; in orange: Intermediate or final data output.

Arrows in the above diagram indicate input and output directionality of data. The demo of interference modeling already concluded the first step, seen top-left in the above diagram. Evidently, we still have some steps ahead of us and we require some additional input data (in blue) and R-scripts (in green) to run the entire pipeline. Let's look at the files we need:

- **Acetyl (K)Sites.txt** is a MaxQuant site table. It needs to come from the same database search that generated "msms.txt". You can find this file on GitHub in the Demo folder. Alternatively, the data is available on PRIDE (identifier PXD040449) among the search results contained in "MaxQuant_SiteToProteinNorm_txt.zip".

- **proteinGroups.txt** is a MaxQuant protein table. It needs to come from the same database search that generated "msms.txt". This table contains additional intensity columns corresponding to MS3-based quantification of unmodified peptides (i.e. "proteome"). To ensure that MaxQuant produces these extra columns in the protein table output, the respective raw files were set as their own experiment during raw file configuration of the

MaxQuant database search. You can find this file on GitHub in the Demo folder. Alternatively, the data is available on PRIDE (identifier PXD040449) among the search results contained in "MaxQuant_SiteToProteinNorm_txt.zip".

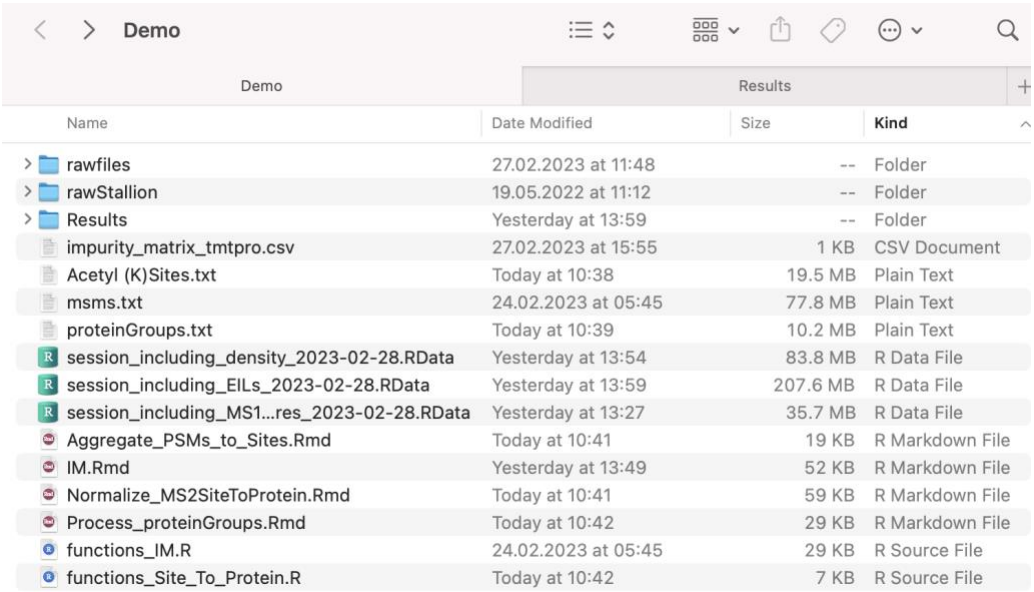
- **Aggregate_PSMs_to_Sites.Rmd** is an R Markdown script to perform the aggregation of PSM level information contained in "modified_PSM.txt" to site level information contained in "Acetyl (K)Sites.txt". This file is located in the main folder of the repository on GitHub. Its output is a file called "aggregated_Acetyl (K)Sites.txt", which will automatically be saved in the "Results" folder.

- **Process_proteinGroups.Rmd** is an R Markdown script to perform filtering and between-sample normalization (etc.) of protein reporter ion intensities contained in "proteinGroups.txt". This file is located in the main folder of the repository on GitHub. Its output is a file named "filtered_normalized_proteinGroups.txt", which will be saved in the "Results" folder. In this demo, "Process_proteinGroups.Rmd" further requires the otherwise optional input of an isotopic impurity matrix to correct for isotopic impurities of TMT labels. Fortunately, we can use the same impurity matrix already used in the interference modeling workflow named "**impurity_matrix_tmtpro.csv**". If it is not yet in your working directory, you can download this file on GitHub in the Demo folder.

- **Normalize_MS2SiteToProtein.Rmd** is an R Markdown script to perform the final step of site-to-protein normalization. This file is located in the main folder of the repository on GitHub. It produces an output called "SiteToProtein_normalized_aggregated_Acetyl (K)sites.txt", which will be stored in the "Results" folder.

- **functions_Site_To_Protein_norm.R** contains functions automatically sourced by the script "Normalize_MS2SiteToProtein.Rmd". This file is located in the main folder of the repository on GitHub.

Now download the required files from GitHub and put them into your folder where you run the demo. It should end up looking like this:



Demo

Demo | Results

Name	Date Modified	Size	Kind
rawfiles	27.02.2023 at 11:48	--	Folder
rawStallion	19.05.2022 at 11:12	--	Folder
Results	Yesterday at 13:59	--	Folder
impurity_matrix_tmtpro.csv	27.02.2023 at 15:55	1 KB	CSV Document
Acetyl (K)Sites.txt	Today at 10:38	19.5 MB	Plain Text
msms.txt	24.02.2023 at 05:45	77.8 MB	Plain Text
proteinGroups.txt	Today at 10:39	10.2 MB	Plain Text
session_including_density_2023-02-28.RData	Yesterday at 13:54	83.8 MB	R Data File
session_including_EILs_2023-02-28.RData	Yesterday at 13:59	207.6 MB	R Data File
session_including_MS1...res_2023-02-28.RData	Yesterday at 13:27	35.7 MB	R Data File
Aggregate_PSMs_to_Sites.Rmd	Today at 10:41	19 KB	R Markdown File
IM.Rmd	Yesterday at 13:49	52 KB	R Markdown File
Normalize_MS2SiteToProtein.Rmd	Today at 10:41	59 KB	R Markdown File
Process_proteinGroups.Rmd	Today at 10:42	29 KB	R Markdown File
functions_IM.R	24.02.2023 at 05:45	29 KB	R Source File
functions_Site_To_Protein.R	Today at 10:42	7 KB	R Source File

Running the Program

We can now follow the pipeline as indicated in the directed graph above. We run the two scripts "Aggregate_PSMs_to_Sites.Rmd" and "Process_proteinGroups.Rmd" to prepare the input for the final script "Normalize_MS2SiteToProtein.Rmd". Just as the script "IM.Rmd" in the interference modeling workflow, these three scripts have their first two code blocks dedicated to 1) loading required packages and 2) specifying relevant parameters. Before running each script, make sure that the required packages are installed.

The parameters in each scrip's parameter section are already configured to the specifics of the demo. Nonetheless, make sure to understand each parameter by reading the respective comments above the lines of code. If specified incorrectly, the program will produce errors down the line. Everything outside of the parameter code blocks does not need to be changed by the user. If a parameter is described as "Optional", the parameter is not required for successfully running the program, as some steps in the workflow can skipped. Set optional parameters to their default value to skip corresponding code blocks. The default values of optional parameters are described in the comments of the code.

Like in the interference modeling workflow, each code block performs a specific task in the respective script and often produces intermediate output (visual and/or textual) of interest. The comments in the code should provide the necessary understanding of what is happening. Code blocks described as "Optional" can be skipped, since they are not required for successfully running the program. In particular: Note that both scripts "Aggregate_PSMs_to_Sites.Rmd" and "Process_proteinGroups.Rmd" contain some parameters and/or optional code blocks that allow for more stringent filtering of features based on defined thresholds of score, intensity and PSM-wise PPF (Precursor Purity Fraction) values. Currently, the parameters in each script are set to filter as little as possible. In a real experiment, I recommend to adjust those filter in order to get rid of features with subpar data quality.

Output

The final script "Normalize_MS2SiteToProtein.Rmd" produces an output table named "SiteToProtein_normalized_aggregated_Acetyl (K)Sites.txt" that is stored in the "Results" folder. Note that this table will only list sites that could be normalized to corresponding protein level (i.e. unmodified peptides of the same proteins), which naturally requires independent quantification on both site and protein level! Hence, some sites might have been dropped.

The output table contains multiple additional intensity columns, which are already normalized between samples. These are: Site intensities (no suffix), interference-adjusted site intensities (suffix *"__IFadjust"*), underlying protein intensities (suffix *"__underlyingProtein"*), interference-adjusted underlying protein intensities (suffix *"__underlyingProtein_IFadjust"*), site-to-protein normalized abundances that are likely biased by varying degrees of ratio compression in individual site and protein pairs (suffix *"__siteToProtein"*), and finally interference-adjusted site-to-protein normalized abundances that mitigate this aforementioned bias (suffix *"__siteToProtein_IFadjust"*).

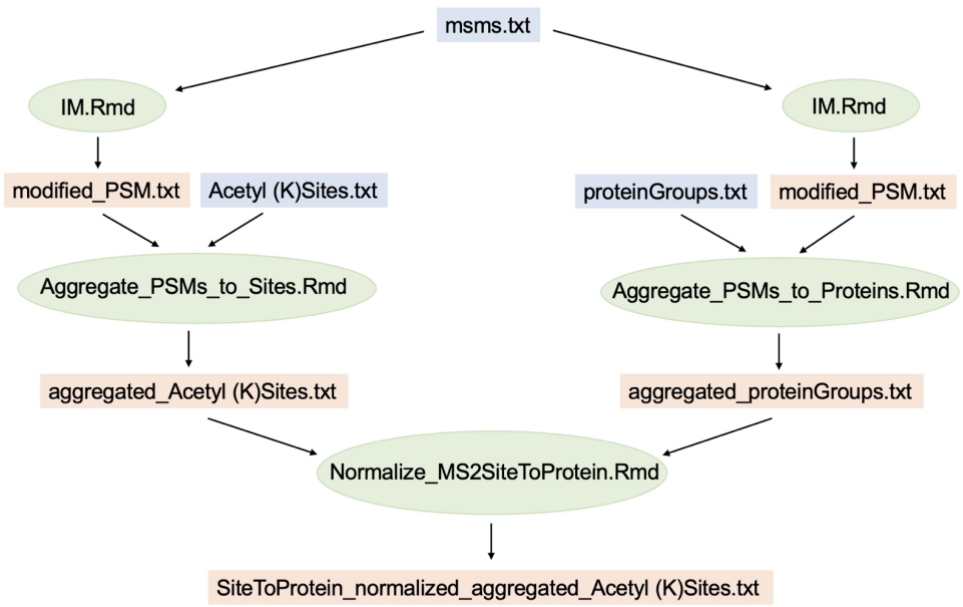
Note that "interference-adjusted" here does not imply "interference-corrected". Instead it means that the interference levels in each individual site and protein pair have been equalized to reach the same level such that subsequent ratio-building is unbiased (i.e. not biased by different levels of reporter ion interference).

If a batch vector was specified in the parameter section of "Normalize_MS2SiteToProtein.Rmd", the output table contains additional columns for all intensity types that are batch corrected via the comBat algorithm (additional suffix "*__batchCorr*"). Further, the table contains ANOVA p-values and other metrics of interest.

Further, the scripts "Aggregate_PSMs_to_Sites.Rmd" and "Process_proteinGroups.Rmd" produce intermediate output tables that are stored in the "Results" folder. These tables might be relevant on their own since they still contain the full list of sites and proteins (prior to filtering based on independent measurement on both site and protein level), including columns for normalized as well as interference-corrected (in case of the site table) intensities.

Comments

Note that for the sake of simplification, this demo only covered the normalization of MS2-quantified site intensities to MS3-quantified protein intensities for which no ratio compression effects are assumed (i.e. EIL = 0). However, the workflow also supports normalization to MS2-quantified protein intensities where EIL is assumed to be ≥ 0. This requires an extra round of interference modeling for PSMs of MS2-quantified unmodified peptides (i.e. "proteome) to ultimately estimate the degree of interference in MS2-quantified proteins. The remaining workflow stays the same and uses the same methodology. In the diagram below, the full workflow to perform normalization of MS2-quantified site abundances to MS2-quantified protein abundances is depicted:



You can try this workflow for yourself – one half of the workflow (i.e. the left side in the above diagram) has already been performed as part of this demo anyway. All the necessary input data is available on PRIDE (identifier PXD040449) among the search results "MaxQuant_SiteToProteinNorm_txt.zip". The raw files from measurements of unmodified peptides (i.e. "proteome") via MS2-based quantification are:

20201030_QExHFX1_RSLC1_Madern_Hartl_UW_MFPL__complexity_P2
20201030_QExHFX1_RSLC1_Madern_Hartl_UW_MFPL__complexity_P5

They can also be downloaded on PRIDE (identifier PXD040449).
Notably, this workflow will require a second instance of the script "IM.Rmd", as well as the corresponding PSM level output. Therefore, set up and run the "protein-half" of this pipeline (i.e. the right half in the above diagram) in a different folder to where the other half is located.