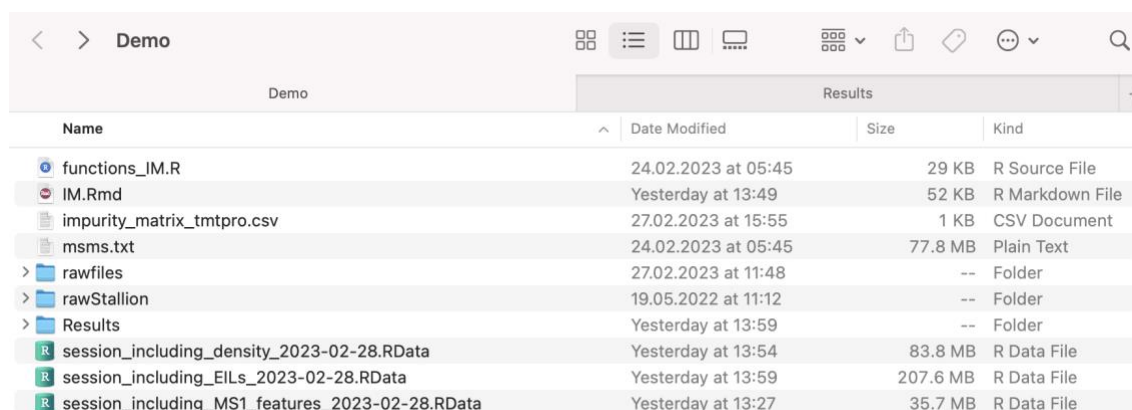# Site-To-Protein Normalization in Multiplex Proteomics

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. We recommend running the demo first before applying the workflow to your own data – this will get any new user familiar with the required data input, the script's many parameters as well as the intermediate and final data output. In the course of this user guide, we will look at the required setup for running the workflow and then how to run it on the basis of the available demo dataset. Finally, there is a description of the generated output. Please note that as of now, this workflow supports MaxQuant output only.

Required Setup

This workflow of site-to-protein normalization builds on the workflow of interference modeling in multiplex proteomics to provide the necessary input data. This demo therefore starts where the demo of interference modeling left off (please see GitHub repository: https://github.com/maxperutzlabs-ms/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 choose to skip the demo for 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:

Figure 1 - *Overview of files and directories after running the interference modeling demo. This setup serves as the starting point for this demo.*

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, please download it now (it is stored 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). This metric is crucial in unbiased site-to-protein normalization, as it provides an estimate of the degree of interference/ratio compression for each PSM. As described in our paper (https://doi.org/10.1016/j.mcpro.2023.100694), if we can account for the varying degrees of ratio compression on both PTM site and underlying protein level, we have a means for unbiased site-to-protein normalization in multiplex proteomics.

Our plan now is to first carry over the PSM-wise information about EIL and reporter ion intensities to the acetyl site level (as given by MaxQuant's "Acetyl (K)Sites.txt") via aggregation. Then we 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 quantified proteins with real-time-search on). This configuration requires us to follow the following workflow as visualized in the README of the GitHub repository:
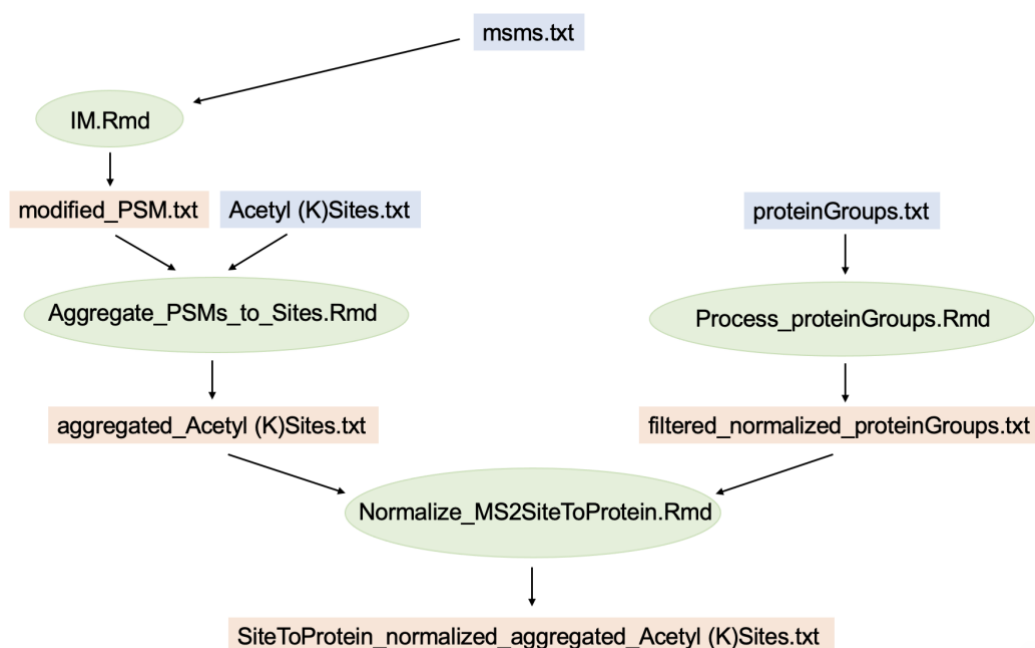

Figure 2 *- Overview of site-to-protein normalization workflow when normalizing MS2-quantified sites to MS3-quantified proteins. In blue: MaxQuant output; in green: R-scripts; in orange: Intermediate or final data output.*

Arrows in the above diagram indicate input and output data directionality. Data input (taken from the MaxQuant search results) is shown in blue, data output is shown in orange, and R-scripts are shown in green. As we can see, the demo of interference modeling already concluded the first step, seen top-left in the above diagram, thereby producing "modified_PSM.txt". This is where we start for this demo. Looking at the rest of the graph, we see that we still have some steps ahead of us which requires some additional input data and R-scripts to run the entire workflow. Let's first go over all the additional files shown in Figure 2 that we require:

- **Acetyl (K)Sites.txt** is a MaxQuant site table. It needs to be part of the same database search result that generated "msms.txt", because this the ID columns of the two tables cross-reference each other, thus allowing the aggregation step. 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 the 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". In short, the information of EIL values and between-sample-normalized reporter ion intensities is carried over. 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.

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

Figure 3 - *Overview of required setup of all directories and files to complete the demo.*

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. Please refer to the respective comments above the lines of code in order to more insight into the meaning of each parameter. If specified incorrectly, the program might 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. We hope that the comments in the code provide the necessary understanding of what each code section is doing. Code blocks described as "Optional" can be skipped, since they are not required for successfully running the program.

Please 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 permissive as possible. Generally, we recommend to adjust those filter parameters 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. Please 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").*

Please 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.

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 in their own way 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.

Additional Comments

For the sake of simplicity, this demo only covered the normalization of MS2-quantified site intensities to MS3-quantified protein intensities. On MS3-quantified protein level, we make the (simplifying) assumption that no ratio compression effects are present (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, in addition to MS2-quantified sites (see Figure 4 below). Other than this, the rest of the workflow stays the same. In the diagram below, the full workflow to perform normalization of MS2-quantified site abundances to MS2-quantified protein abundances is depicted:
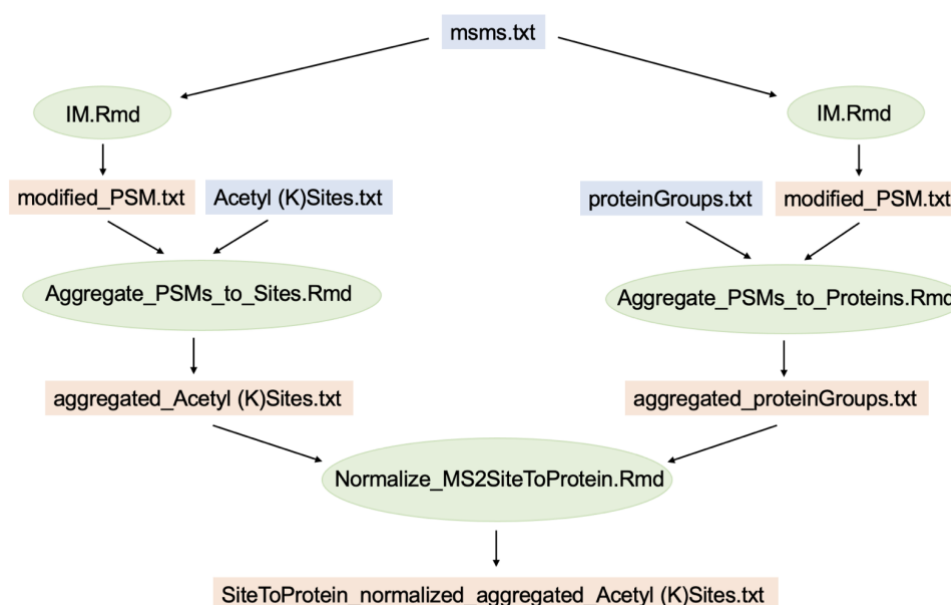

Figure 4 - *Overview of the site-to-protein normalization workflow when normalizing MS2-quantified sites to MS2-quantified proteins. In blue: MaxQuant output; in green: R-scripts; in orange: Intermediate or final data output.*

The demo user can try this workflow for themselves – half of the workflow (i.e. the left side in Figure 4) has already been performed as part of this demo. All the required input data is available on PRIDE (identifier PXD040449) among the search results named "MaxQuant_SiteToProteinNorm_txt.zip". The required raw files from measurements of unmodified peptides (i.e. "proteome") via MS2-based quantification are named:

<div align="center">

20201030_QExHFX1_RSLC1_Madern_Hartl_UW_MFPL__complexity_P2
20201030_QExHFX1_RSLC1_Madern_Hartl_UW_MFPL__complexity_P5

</div>

They can also be downloaded on PRIDE (identifier PXD040449). Please not that the workflow shown in Figure 4 will require a second instance of the script "IM.Rmd", as well as the corresponding PSM level output. Therefore, we recommend setting up and run the "protein-half" of this pipeline (i.e. the right half in Figure 4) in a different directory than where the other half is located.