**Demo for:**

**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 intended workflow can be experienced from start to finish. I recommend running the demo first before trying the workflow on 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.

## Required Setup

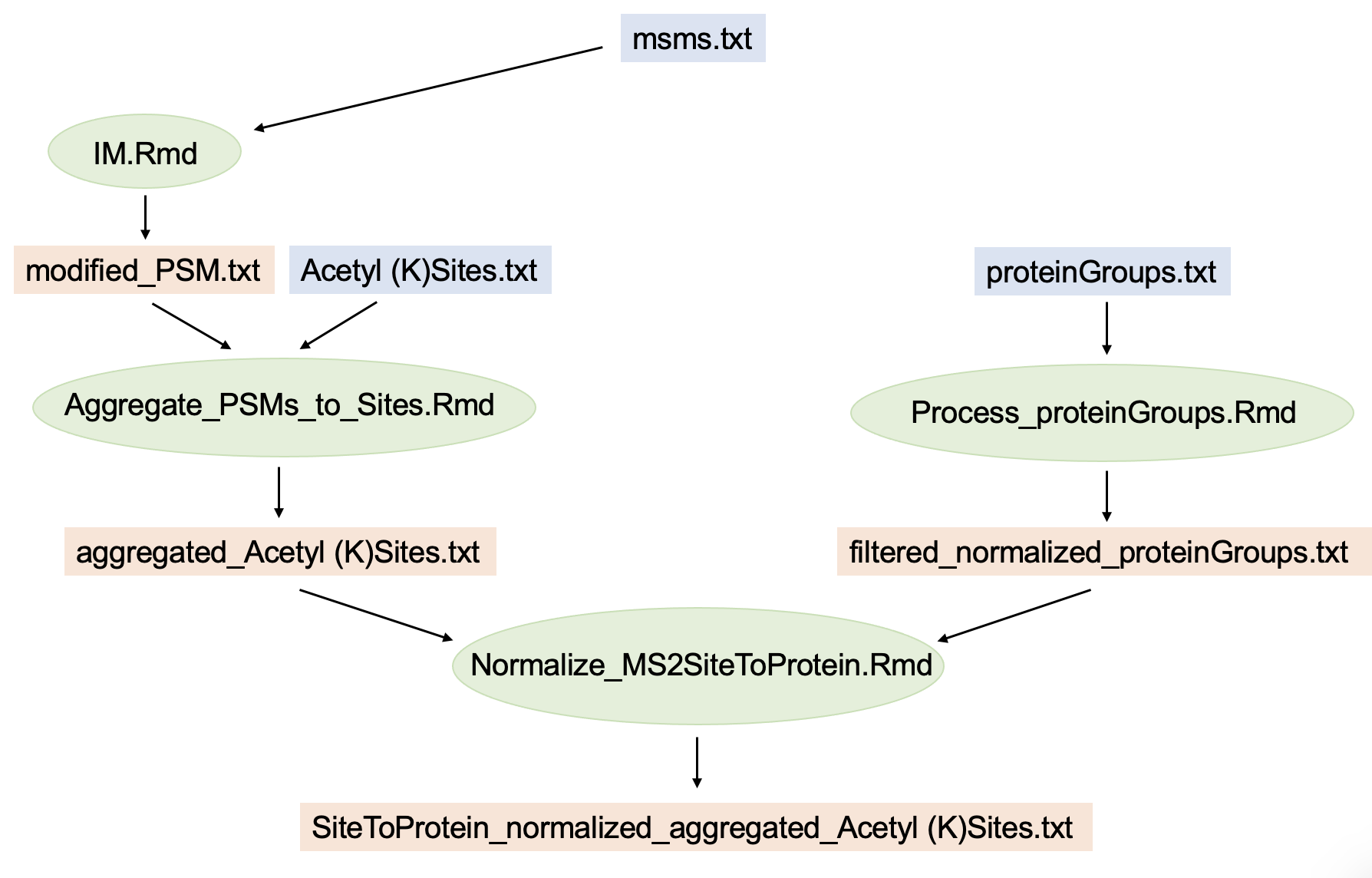
# In general, the workflow of site-to-protein normalization builds on the workflow of interference modeling in multiplex proteomics which provides the necessary input data. This demo therefore starts where the demo of interference modeling left off (see GitHub repository named “InterferenceModeling\_in\_MultiplexProteomics”). However, 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 of interference modeling, which left us with the following data in the folder we named “Demo”:

Graphical user interface, text, application, email

Description automatically generated

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 of interference modeling, download it now and put it into a folder named “Results”.

Before we add any additional files and 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 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 now transfer this PSM-wise information to acetyl site level (as given by MaxQuant’s “Acetyl (K)Sites.txt”) via aggregation, and then use this updated information on site level to perform site-to-protein normalization while accounting for reporter ion interference at MS2 level. For this demo, we will go through the workflow of normalizing MS2-quantified site abundances to MS3-quantified protein abundances, which will require us to follow this specific workflow 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. As can be seen, we still have some steps ahead of us and 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’s site table output. It needs to originate from the same data base search that produced msms.txt (used in the interference modeling workflow to obtain “modified\_PSM.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’s protein table output. It needs to originate from the same data base search that produced msms.txt (used in the interference modeling workflow to obtain “modified\_PSM.txt”). This table contains additional intensity columns coming from MS3-based quantification of unmodified peptides (i.e. “proteome”). To ensure these extra columns in MaxQuant’s output, the respective raw files were set as their own experiment in the data base 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. It outputs a file called “aggregated\_Acetyl (K)Sites.txt”, which will 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 MaxQuant’s “proteinGroups.txt”. This file is located in the main folder of the repository on GitHub. It outputs file called “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 in the workflow. This file is located in the main folder of the repository on GitHub. It outputs a file called “SiteToProtein\_normalized\_aggregated\_Acetyl (K)sites.txt”, which will be saved 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.

Download the required files from GitHub and put them into your folder where you run the demo, which should now look like this:

Graphical user interface, text, application, email

Description automatically generated

## 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 IM.Rmd in the interference modeling workflow, these scripts have their first two code sections dedicated to 1) loading required packages and 2) specifying 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 read and understand each parameter by reading the comments above each line of code. If specified incorrectly, the program will produce errors down the line. Everything outside of the parameter code section 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 these sections. The default values of optional parameters are described in the comments.

Like in the interference modeling workflow, each subsequent code section 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 sections described as “Optional” can be skipped, since they are not required for successfully running the program.

Note that both Aggregate\_PSMs\_to\_Sites.Rmd and Process\_proteinGroups.Rmd contain some parameters or optional code blocks that allow for more stringent filtering of features based on score, intensity and PSM-wise PPF (Precursor Purity Fraction) values. Currently, the parameters are set to filter as little as possible.

**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 only lists sites that could be normalized to corresponding protein level (i.e. unmodified peptides of the same proteins), which often results in filtering of sites despite good data quality. The table contains multiple new 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).

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.

Additionally, the scripts Aggregate\_PSMs\_to\_Sites.Rmd and Process\_proteinGroups.Rmd produce intermediate output tables that are stored in the Results table. These tables might be relevant on their on their own as they still provide the full list of sites and proteins prior to filtering for sites with quantified underlying proteins.