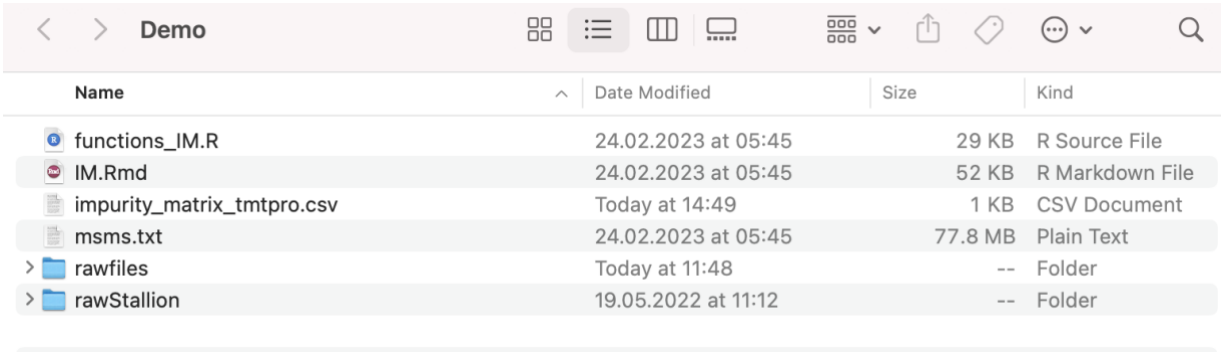
Demo for:
Interference Modeling in Multiplex Proteomics

Overview

This document serves as a guide to the demo for the interference modeling 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 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.

Required Setup

Every time you want to use this workflow, I recommend creating a new folder (here: "Demo") in which to put all the necessary software (i.e. scripts and tools) as well as the required data input to run the script. For this demo, the required setup looks like this:



Let's go over the individual files:

- **msms.txt** is a MaxQuant PSM table. You can find this file on GitHub in the Demo folder. While the corresponding MaxQuant database search encompasses many different raw files, in this demo, we are only interested in PSMs coming from measurements of acetyl-peptide enriched samples. The msms.txt on GitHub has already been filtered accordingly in order to minimize the file's overall size. The unfiltered version of this file (which ultimately generates the exact same output since the script would otherwise perform this filtering) is available on

PRIDE (identifier PXD040449) among the search results contained in "MaxQuant_SiteToProteinNorm_txt.zip".

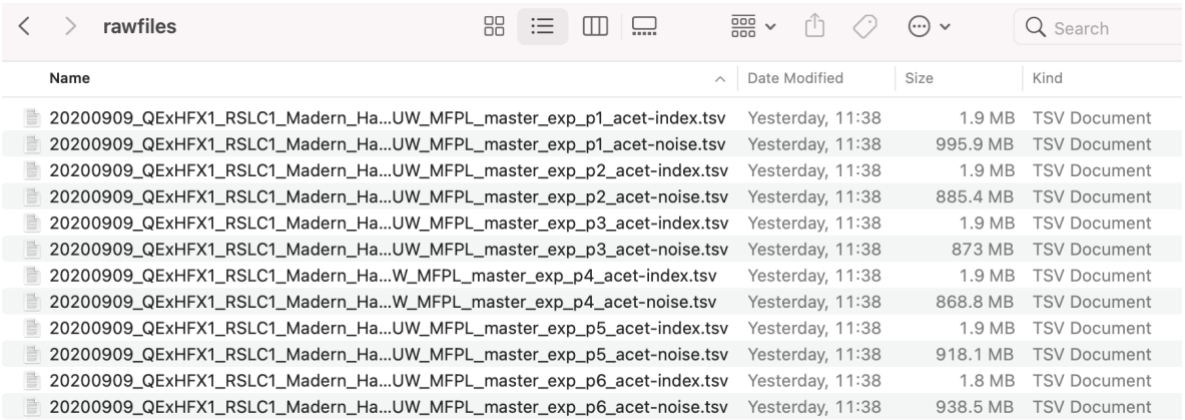
- **rawfiles** is a subfolder that should be created. It needs to contain the corresponding Thermo raw files (*raw) of PSMs in the PSM table after filtering for relevant raw files. For this demo, we need the following six raw files that correspond to acetyl-peptide enriched measurements:

  20200909_QExHFX1_RSLC1_Madern_Hartl_UW_MFPL_master_exp_p1_acet.raw
  20200909_QExHFX1_RSLC1_Madern_Hartl_UW_MFPL_master_exp_p2_acet.raw
  20200909_QExHFX1_RSLC1_Madern_Hartl_UW_MFPL_master_exp_p3_acet.raw
  20200909_QExHFX1_RSLC1_Madern_Hartl_UW_MFPL_master_exp_p4_acet.raw
  20200909_QExHFX1_RSLC1_Madern_Hartl_UW_MFPL_master_exp_p5_acet.raw
  20200909_QExHFX1_RSLC1_Madern_Hartl_UW_MFPL_master_exp_p6_acet.raw

  The raw files are available for download on PRIDE (identifier PXD040449).

  Note that we advise that all the raw files used in the workflow are of similar nature, i.e. they come from the same experiment and even the same sub-experiment (here: acetylome measurements) within the experiment. If there are different kinds of raw files in your PSM table (e.g. measurements of unmodified peptides, acetyl-peptides, and phospho-peptides), it is therefore best to run them all separately through this workflow. The reason is that they are best normalized and thus interference-corrected independent from each other, since distinct types of peptides (i.e. unmodified, acetyl, phospho, etc.) can differ in their relative sample/ intensities when coming from real biological experiments.

- **rawStallion** is a Windows command-line application that reads relevant information (e.g. noise values, intensity values, etc.) from Thermo raw files and writes them to two tsv files per raw file. You can download rawStallion here: https://github.com/fstanek/rawStallion. If you don't have access to a Windows operating system, you can find the corresponding tsv files for this demo on PRIDE (identifier PXD040449) as "rawStallion_tsvfiles.zip". Download the data, unzip it and put the tsv files into the rawfiles folder:



  Using the tsv files instead of Thermo raw files lets you run the demo while skipping the section requiring rawStallion.

- **impurity_matrix_tmtpro.csv** is a csv file that contains an isotopic impurity matrix specific to the labeling reagents used in the experiment. In this matrix, rows reflect relative contribution of individual reagents to reporter ion channels ordered along the columns. You can find impurity_matrix_tmtpro.csv on GitHub in the Demo folder.

- **IM.Rmd** is the R Markdown script to perform the entire workflow. This file is located in the main folder of the repository on GitHub.

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

Running the Program

Open the script IM.Rmd in R studio and make sure your working directory is set to the folder (here "Demo") that contains the necessary software and data described above. We can then proceed to go through the script.

The first code block loads multiple required packages:

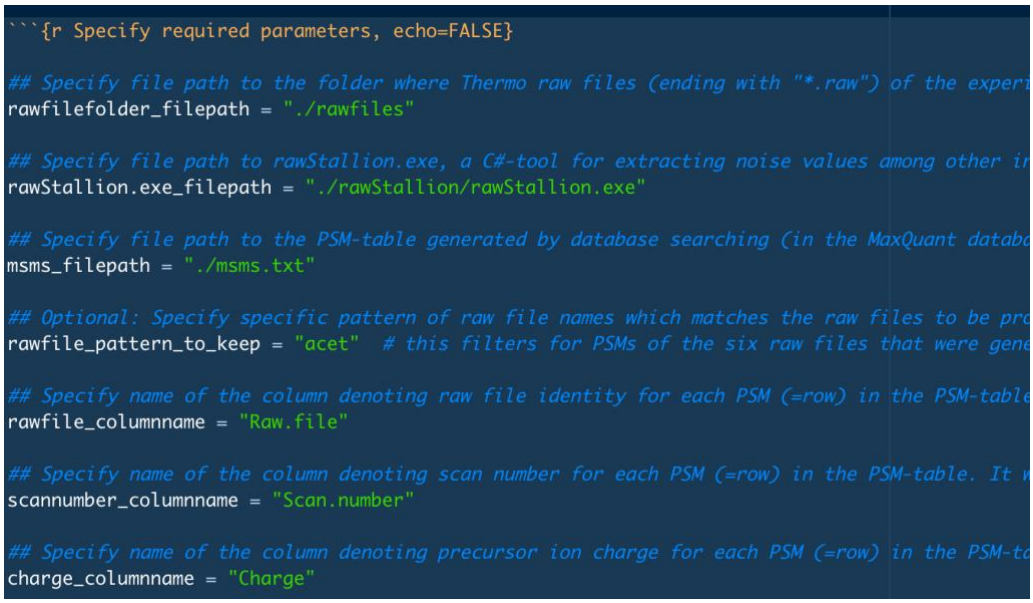
```
```{r Load required packages and functions, echo=FALSE, message=FALSE, warning=FALSE}

## Load packages. If not installed yet, install them prior to running this script!
library(tidyverse)
library(readr)
library(pracma)
library(plot3D)
library(MASS)
library(gridExtra)
library(rlist)
library(foreach)
library(doParallel)
library(fields)
library(cowplot)
library(MSnbase)    ## Bioconductor
library(limma)      ## Bioconductor
library(DESeq2)     ## Bioconductor
library(msqrob2)    ## Bioconductor
```

Make sure these packages are installed prior to running the script. Regular R packages can be installed within R-studio. To install Bioconductor packages, visit the respective Bioconductor website (e.g. https://bioconductor.org/packages/release/bioc/html/MSnbase.html) and follow the instructions in the "Installation" section.

The second code block is where the user is required to specify the input parameters needed to successfully run the script. These parameters aim configure the program to the specific data input. Anything outside of this code block does not need input from the user. In its current form, the script's parameters are configured to make the demo work.
```

Here is a screenshot of the top few parameters:



```
```{r Specify required parameters, echo=FALSE}

## Specify file path to the folder where Thermo raw files (ending with "*.raw") of the experi
rawfilefolder_filepath = "./rawfiles"

## Specify file path to rawStallion.exe, a C#-tool for extracting noise values among other in
rawStallion.exe_filepath = "./rawStallion/rawStallion.exe"

## Specify file path to the PSM-table generated by database searching (in the MaxQuant databa
msms_filepath = "./msms.txt"

## Optional: Specify specific pattern of raw file names which matches the raw files to be pro
rawfile_pattern_to_keep = "acet"  # this filters for PSMs of the six raw files that were gene

## Specify name of the column denoting raw file identity for each PSM (=row) in the PSM-table
rawfile_columnname = "Raw.file"

## Specify name of the column denoting scan number for each PSM (=row) in the PSM-table. It w
scannumber_columnname = "Scan.number"

## Specify name of the column denoting precursor ion charge for each PSM (=row) in the PSM-ta
charge_columnname = "Charge"
```

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. If a parameter is described as “Optional”, specifying this parameter is not required for successfully running the program, as some steps in the workflow can skipped. Set optional parameters to their default value (e.g. NULL or “”) to skip these sections. The default values of optional parameters are mentioned in the comments.

Once all parameters are specified, the entire script can be executed. Each subsequent code block performs a specific task and often produces intermediate output (visual and/or textual) of interest. This output should hopefully be insightful once you read the corresponding paper (“A causal model of ion interference enables assessment and correction of ratio compression in multiplex proteomics”). Additionally, the comments in the code should provide the necessary understanding of what is happening.

At several points during the workflow, the script will create a session image and save it in the working directory, e.g. “session_including_MS1_features_2023-02-28.RData”. These sessions can be loaded at a later time point via the R function load() to access or recreate part of the workflow without starting from scratch again.

Code blocks described as “Optional” can be skipped, since they are not required to successfully run the program. Note that some code blocks will take some time to run, especially if there are many raw files to be processed.

Output

The script produces an output table called “modified_PSM.txt” that is stored in a folder called Results (“Demo/Results/modified_PSM.txt”). This table contains multiple additional columns that are generated while running the script from beginning to end. Notable column additions are: Normalized reporter intensity columns (suffix “_norm”); normalized interference-corrected reporter intensity
```

columns (suffix "_norm__interference_corrected"); and the columns EIL (Estimated Interference Level) and PPF (Precursor Purity Fraction).
Note that the output table "modified_PSM.txt" serves as input to the demo for site-to-protein normalization in multiplex proteomics (see GitHub repository named "SiteToProteinNormalization_in_MultiplexProteomics"). This allows you to continue from here if you choose to follow the second demo.