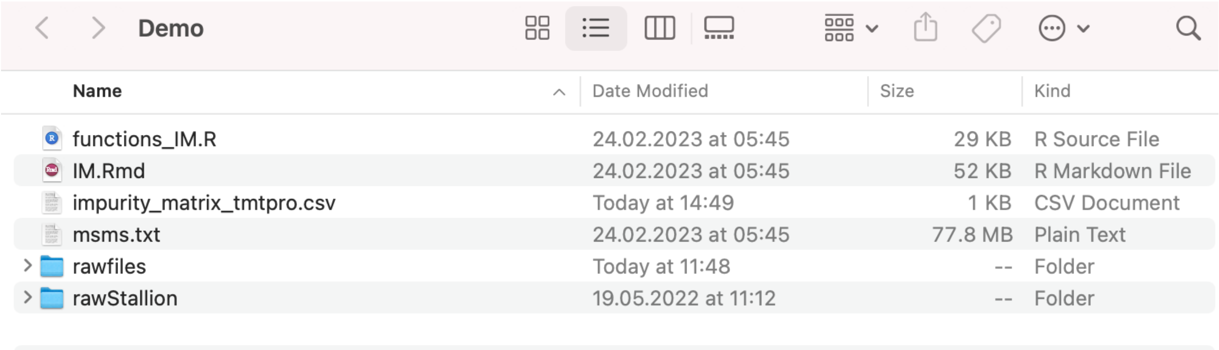
Interference Modeling
in Multiplex Proteomics

Overview

This document serves as a guide to the demo for the interference modeling workflow on GitHub. The demo comes with its own dataset, on which basis the workflow can be experienced from start to finish. I recommend running the demo first before applying the workflow to your own data – this will get you 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.

Required Setup

Each 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:



Name	Date Modified	Size	Kind
functions_IM.R	24.02.2023 at 05:45	29 KB	R Source File
IM.Rmd	24.02.2023 at 05:45	52 KB	R Markdown File
impurity_matrix_tmtpro.csv	Today at 14:49	1 KB	CSV Document
msms.txt	24.02.2023 at 05:45	77.8 MB	Plain Text
rawfiles	Today at 11:48	--	Folder
rawStallion	19.05.2022 at 11:12	--	Folder

Figure 1 - Overview of required setup of files and directories.

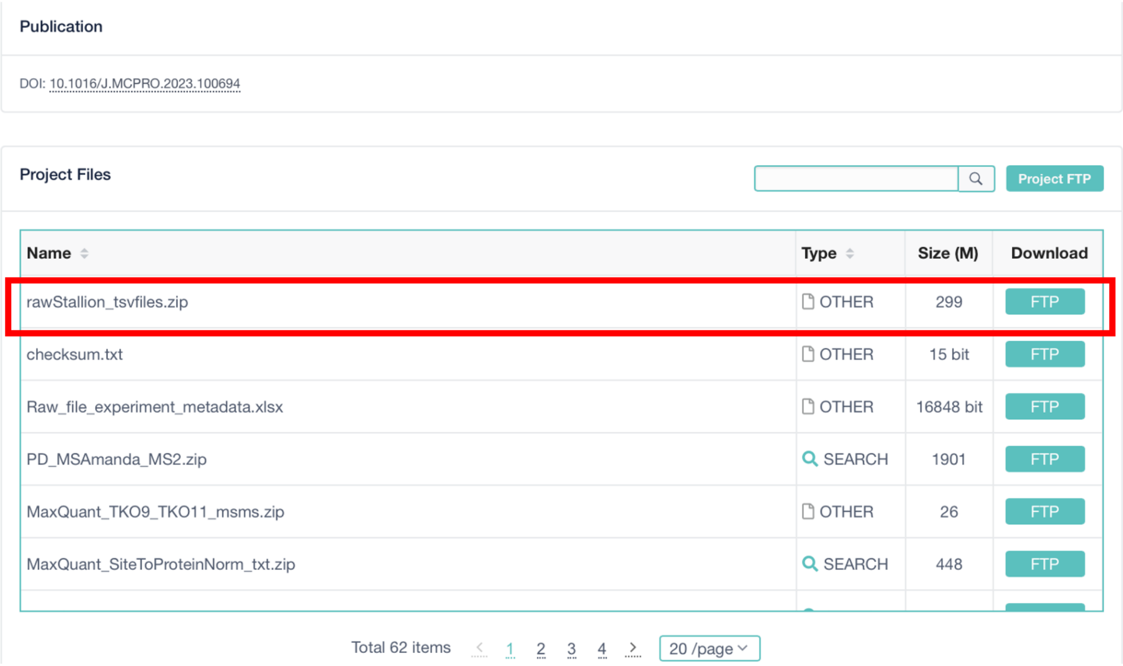
Let’s go over the individual files:

- **msms.txt** is a MaxQuant PSM table, i.e. a database search result that lists row-wise PSMs. Note that a FragPipe PSM table is also supported by the workflow. For this demo, you can download msms.txt on GitHub in the Demo folder. It is already prefiltered to make the file size smaller, since in this demo we are only interested in PSMs coming from measurements of acetyl-peptide enriched samples. 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 Thermo raw files (*.raw) corresponding to the PSMs contained in the PSM table (msms.txt). The raw files are required in order to extract scan-specific information like noise values etc. via the rawStallion tool (see below) and save them as tsv files, which can subsequently be read into R. For this demo, we need the following six raw files:

  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

  These raw files resulted from measurements of acetyl (K)-peptide enriched samples and they are available for download on PRIDE (identifier PXD040449, https://www.ebi.ac.uk/pride/archive/projects/PXD040449). Note that you can make this demo shorter by skipping directly to the tsv files and placing them in the folder “rawfiles” instead. The tsv files are also available for download on PRIDE (identifier PXD040449) and stored as “rawStallion_tsvfiles.zip”:



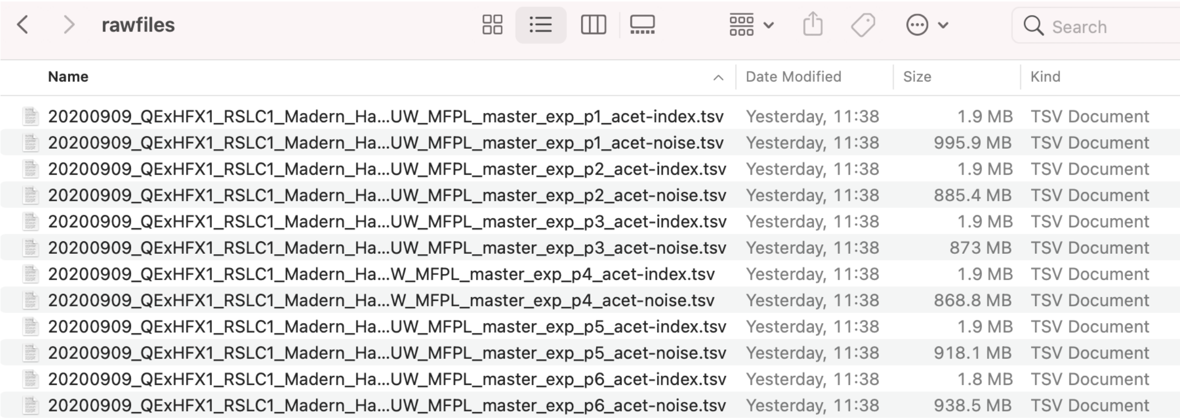
Publication

DOI: 10.1016/J.MCPRO.2023.100694

Project Files

Name	Type	Size (M)	Download
rawStallion_tsvfiles.zip	OTHER	299	FTP
checksum.txt	OTHER	15 bit	FTP
Raw_file_experiment_metadata.xlsx	OTHER	16848 bit	FTP
PD_MSAmanda_MS2.zip	SEARCH	1901	FTP
MaxQuant_TKO9_TKO11_msms.zip	OTHER	26	FTP
MaxQuant_SiteToProteinNorm_txt.zip	SEARCH	448	FTP

Figure 2 - tsv files for Demo available on PRIDE.



Name	Date Modified	Size	Kind
20200909_QExHFX1_RSLC1_Madern_Ha...UW_MFPL_master_exp_p1_acet-index.tsv	Yesterday, 11:38	1.9 MB	TSV Document
20200909_QExHFX1_RSLC1_Madern_Ha...UW_MFPL_master_exp_p1_acet-noise.tsv	Yesterday, 11:38	995.9 MB	TSV Document
20200909_QExHFX1_RSLC1_Madern_Ha...UW_MFPL_master_exp_p2_acet-index.tsv	Yesterday, 11:38	1.9 MB	TSV Document
20200909_QExHFX1_RSLC1_Madern_Ha...UW_MFPL_master_exp_p2_acet-noise.tsv	Yesterday, 11:38	885.4 MB	TSV Document
20200909_QExHFX1_RSLC1_Madern_Ha...UW_MFPL_master_exp_p3_acet-index.tsv	Yesterday, 11:38	1.9 MB	TSV Document
20200909_QExHFX1_RSLC1_Madern_Ha...UW_MFPL_master_exp_p3_acet-noise.tsv	Yesterday, 11:38	873 MB	TSV Document
20200909_QExHFX1_RSLC1_Madern_Ha...W_MFPL_master_exp_p4_acet-index.tsv	Yesterday, 11:38	1.9 MB	TSV Document
20200909_QExHFX1_RSLC1_Madern_Ha...W_MFPL_master_exp_p4_acet-noise.tsv	Yesterday, 11:38	868.8 MB	TSV Document
20200909_QExHFX1_RSLC1_Madern_Ha...UW_MFPL_master_exp_p5_acet-index.tsv	Yesterday, 11:38	1.9 MB	TSV Document
20200909_QExHFX1_RSLC1_Madern_Ha...UW_MFPL_master_exp_p5_acet-noise.tsv	Yesterday, 11:38	918.1 MB	TSV Document
20200909_QExHFX1_RSLC1_Madern_Ha...UW_MFPL_master_exp_p6_acet-index.tsv	Yesterday, 11:38	1.8 MB	TSV Document
20200909_QExHFX1_RSLC1_Madern_Ha...UW_MFPL_master_exp_p6_acet-noise.tsv	Yesterday, 11:38	938.5 MB	TSV Document

Figure 3 - tsv files for Demo in the “rawfiles” folder.

- **rawStallion** is a Windows command-line application using Thermo RawFileReader software 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 will let you run the demo without having installed rawStallion.

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

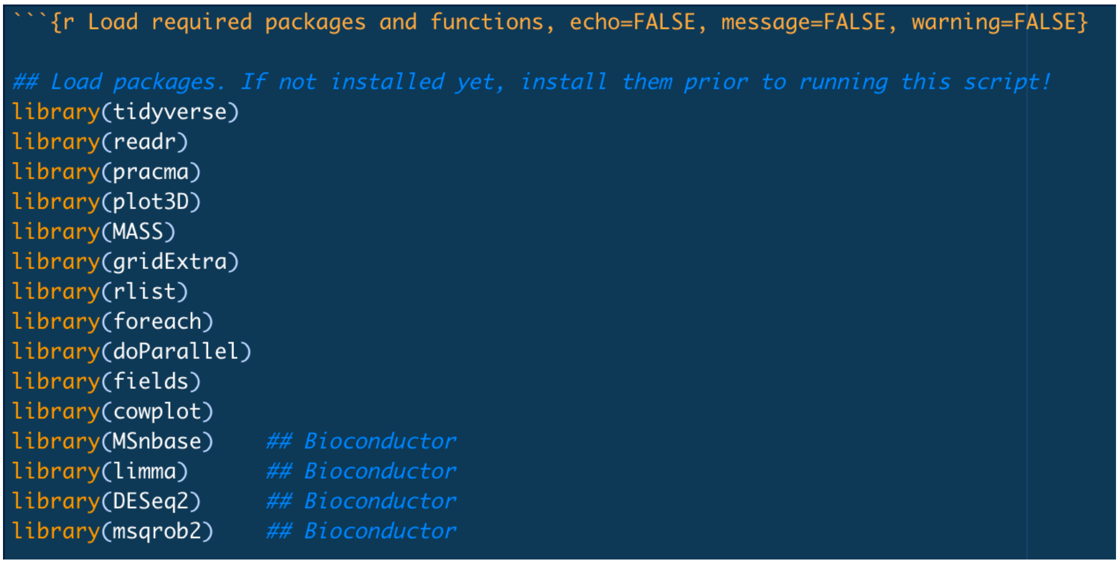
- **functions_IM.R** contains functions automatically sourced by the main script “IM.Rmd”. This file is located in the main folder of the repository on GitHub.

- **impurity_matrix_tmtpro.csv** is a csv file that contains an isotopic impurity matrix specific to the multiplexing label reagents used in the experiment. This file is required because the workflow extracts reporter ion intensities from MS2 spectra, and subsequently aims to impurity-correct them using this matrix. Rows in the matrix reflect relative contribution of individual reagents to reporter ion channels ordered along the columns. You can find “impurity_matrix_tmtpro.csv” required for this Demo on GitHub in the Demo folder. Note that for any other dataset, ideally the impurity matrix is always manually adjusted using the isotopic impurity information on the product sheet that comes with the labeling kit.

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:



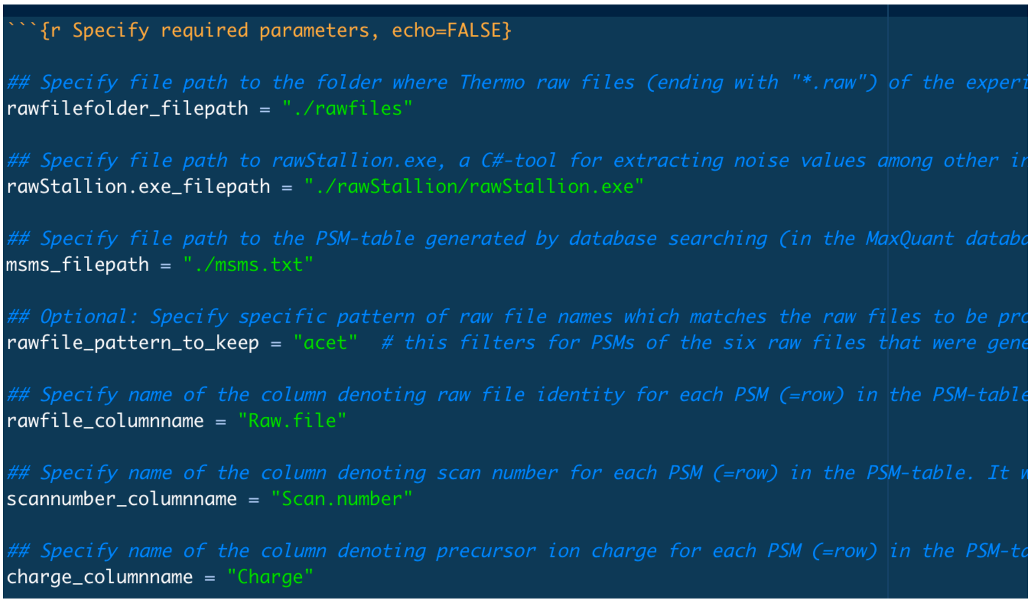
~~~
```{r Load required packages and functions, echo=FALSE, message=FALSE, warning=FALSE}

## 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
~~~

Figure 4 –Code block loading required R and Bioconductor packages.

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 websites (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 to configure the program to your specific data input. Anything outside of this code block does not require input from the user! In its current form as available on GitHub, the script's parameter settings are configured to make the demo run.

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"
~~~

Figure 5 –Code block listing customizable parameters of the workflow

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, code block after code block. Each 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 to what is happening.

Roughly, the script performs the following steps in order:

1) Reading in data and modifying it (e.g. filtering for raw files of interest, filtering out contaminants, etc.).
2) Using rawStallion to extract raw file-specific variables (e.g. noise-values, intensity values, etc.) and save them as tsv files.
3) Calculation of raw-file specific variables using the tsv files (e.g. reporter intensities, PPF, TIW, etc.).

4) Calculation of other variables for modeling (e.g. peptide density, number of labels per peptide, empirical peptide classes, etc.)
5) Modeling via robust multiple linear regression.
6) Using estimated model parameters to calculate EIL values (Estimated Interference Levels).
7) Performing between-sample normalization.
8) Performing interference-correction based on EIL values on normalized intensity data, which generates interference-corrected normalized intensity data.
9) Exporting all data as modified PSM table with extra variables (columns).

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 reloaded 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 named Results (“Demo/Results/modified_PSM.txt”). This table is a modified version of the input PSM table but contains multiple additional columns that are generated while running the script. 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” of this demo serves as input to the demo for site-to-protein normalization in multiplex proteomics (see another GitHub repository named “SiteToProteinNormalization_in_MultiplexProteomics”). Hence you can continue from here if you choose to follow the second demo.

Comments

Note that we advise that all the raw files used in the workflow at a time 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 – all from the same experiment), it is best to run them all separately through this workflow. The reason for that is that they are best normalized and thus interference-corrected independently from each other, since distinct types of peptides (i.e. unmodified, acetyl, phospho, etc.), even within the same experiment, can differ in their relative sample contribution and thus background interference level.