mspms: a R package and graphical interface for the processing and analysis of multiplex substrate profiling by mass spectrometry for proteases (MSP-MS) data.

Charlie Bayne1,2,3, David Gonzalez2,3,6, Anthony O'Donoghue

Institutions:

1Biomedical Sciences Graduate Program, UC San Diego, La Jolla, California 92093, USA.

2Department of Pharmacology, University of California San Diego, La Jolla, California 92093, USA

3Skaggs School of Pharmacy and Pharmaceutical Sciences, University of California San Diego, La Jolla, California, USA

4Department of Pediatrics, UC San Diego, La Jolla, California 92093, USA

5Sharp Rees Stealy Medical, San Diego, CA

6Center for Microbiome Innovation, University of California at San Diego, La Jolla, California 92093, USA

7School of Pharmacy, University of Wisconsin, Madison, Wisconsin 53705, USA

**Abstract:**

Multiplex Substrate Profiling by Mass Spectrometry (MSP-MS) is a powerful method used to determine the substrate specificity of proteases. This method is of interest for many groups interested in the study of proteases and their role as regulators of many biological pathways whether applied to the study of disease states, the development of diagnostic and prognostic tests, generation of tool compounds, or rational design of protease targeting therapeutics. Analysis of the MS based data produced by MSP-MS is a multistep process involving detection and quantification of peptides, normalization, outlier detection, imputation, cleavage sequence identification, statistics, and data visualizations. This process can be challenging, especially for biologists/ mass spectrometrists with limited programming experience. To overcome these issues, we provide the mspms R package alongside a companion graphic user interface hosted at https://gonzalezlab.shinyapps.io/mspms\_shiny/ to facilitate the analysis of MSP-MS data utilizing good software/data analysis practices.

Introduction:

Multiplex substrate profiling by mass spectrometry for proteases (MSP-MS) is a method for determining protease substrate specificity. This method works by utilizing a rationally designed library of peptide sequences, incubating them in a protease/protease mixture, treating with any desired experimental conditions, and then detecting and quantifying the cleaved peptides through mass spectrometry1.

MSP-MS is applicable to a wide group of researchers as it only requires a synthetically created peptide library, mass spectrometer, and samples containing protease(s) to perform. While relatively simple conceptually, MSP-MS produces high dimensional data that can be very challenging for biologists or mass spectrometrists to analyze. Adequate interpretation of the MSP-MS data requires several steps. The data must be median normalized, outliers detected and removed, missing values imputed, cleavage motifs relative to the peptide library recognized, and then downstream statistics and visualizations must be performed.

A central component of any experimental method working on complex data, is the data analysis pipeline. Poorly documented analysis code poses a number of challenges. It has been a prevalent problem in the biological research world, leading to the retraction of several high-profile papers in recent years. There is also a logistical problem, as it limits the portability/ reproducibility of analysis lab to lab since it requires specific knowledge from the programmer to successfully utilize. This is commonly a problem in academic labs, where a researcher may develop code widely used for analysis within the lab before leaving for a new opportunity. Often, this results in an opaque codebase that may be treated as a black box and can ultimately result in erroneous data analysis, especially as the original code gets distorted over time through decentralized use by various researchers. Solutions for this exist in the domain of software engineering, but often times they are not successfully implemented in biology labs, due to the background knowledge required and the amount of time it would take to develop.

To overcome these issues, we developed mspms, which is a well-documented, portable, and reproducible R package for the analysis of MSP-MS data. This package brings good software practices to MSP-MS data analysis. We also recognize that there may be a wide community of researchers interested in the MSP-MS method but find the R programming environment to be inaccessible. For such cases, we provide mspms-shiny, a graphical user interface to the core features of the mspms package. Mspms-shiny is available on the web at https://gonzalezlab.shinyapps.io/mspms\_shiny/, or by downloading and running locally on the researcher’s computer of choice. These user-friendly tools in combination with the powerful MSP-MS method offers a robust resource that is widely applicable for the protease research community.

Here we describe the core features of the package and demonstrate their use through the analysis of a simple use case of MSP-MS data.

Methods:

**Code Availability**

Mspms is implemented in R and released under a MIT license.

This package is available on Bioconducter?

The development version of mspms is available on github at <https://github.com/baynec2/mspms>.

The shiny interface to mspms can be found hosted at <https://gonzalezlab.shinyapps.io/mspms_shiny/>

The code utilized in this manuscript can be found at

**Package structure**

mspms contains 4 broad categories of functions: those that are responsible for making the package widely useful, normalizing/processing data, performing statistics, and visualizing the data.

**Functions to make mspms widely useful**

These functions intend to extend mspms analysis to the largest number of circumstances possible. Many of these functions are in continuous development based on issues raised by specific research groups on GitHub to address their specific needs.

*Preprocessing*

Mspms is designed to be downstream of computationally intensive solutions for detecting and quantifying peptides. These solutions generally produce outputs that contain the same information but are formatted differently. To make mspms compatible with all of these different software solutions we provide functions that parse them into a consistent format that is then capable of being operated on by downstream functions.

Prepare\_peaks() and prepare\_pd() are used to prepared exported data from peaks and proteome discoverer respectively for data analysis. Additional file converters from popular MS analysis software are in development (ie. Spectronaut, FragPipe, and MaxQuant).

*All possible cleavages of peptide library*

It is necessary to determine all of the possible cleavages of a defined length in a peptide library for downstream analyses such as recognition motif visualization. We provide the calculate\_all\_cleavages() function for this use case. This function takes a vector of given peptide sequences, and then splits them up into all possible cleavages of a user specified length.

**Data Processing/ Normalization**

*Data Normalization*

Median normalization is performed as implemented in the NormalyzerDE R package2. The intensity from a peptide in a sample is first divided by the median of all peptides in the sample. This value is then multiplied by the mean of median of sum of intensities of all peptides in all samples.

*Outlier Removal*

Outliers are detected using a dixon test as implemented in the outliers R package 3. Values determined to be outliers (p < 0.05) are converted to NA.

*Data Imputation*

Values for missing data (peptides detected as 0) are imputed. First, a univariate distribution is fit on the lowest range of values as determined by noise \* length. using the MASS R package 4. Then values are imputed using these parameters utilizing the truncated normal distribution as implemented in the truncnorm R package 5. Outlier values are not imputed.

*Cleavage Motif*

Cleavage motifs are generated for each detected peptide. The experimentally detected peptide sequence is mapped to the corresponding peptide sequence in the peptide library it was derived from. Then, the number of amino acids a user specified distance from the cleavage motif are reported. If there is not an AA at that position of the peptide library because it is past the terminal end, that position of the cleavage motif is represented as a X. This is performed for detected peptides that were cleaved at the N terminus, the C terminus, and both the N and C terminus.

**Statistics**

There are many statistics that could be performed on MSP-MS data depending on the experimental design. Most are best left to the researcher to implement on their own. Here we provide convenient implementations of a small subset of statistical tests that we have found to be generally useful for all MSP-MS experiments.

*T tests*

Pairwise t-tests are performed for each peptide grouped by experimental condition relative to time 0 as implemented in the Rstatix package 6. P values are subsequently FDR corrected.

*Anova*

An anova is performed for each peptide grouped by experimental condition to test for an effect of time as implemented in the Rstatix package. P values are subsequently FDR corrected.

**Data Visualization**

Similar to our approach with statistics, we provide functions for plotting a subset of graphs that we have found to be generally useful for MSP-MS experiments.

*Visualization of Specificity through Sequence Motifs*

We provide a plot\_cleavage\_motifs() function to visualize sequence specificity using sequence motif logos. A vector of user supplied sequence motifs are compared to the background of all possible combinations in the peptide library used for the experiment. This is implemented in the same manner as IceLogo. The ggseqlogo R package 7 is then used to plot the underlying data.

*Heatmap/Hierchical Clustering*

We provide plot\_heatmap() to visualize a heatmap of the normalized intensity, where the rows and columns have been subjected to hierchical clustering and the color of the cells represent the values the data as implemented in the heatmaply R package8. Euclidian distance and the complete agglomeration method are used to perform the hierchical clustering.

*PCA*

We provide plot\_pca() as a connivence function to visualize the first two principle components of the data colored by time, and label shapes showing the experimental condition using the ggplot2 package9. Only complete cases are considered, the peptides determined to have outliers are omitted.

*Time Course*

We provide plot\_time\_course() as a convivence function to visualize the intensity of peptides over time by condition using the ggplot2 package.

Results:

To illustrate the utility of mspms, we ……….

Discussion:

mspms is a simple-to-use tool, that is specifically designed for biologists or mass spectrometrists who wish to analyze data produced through the MSP-MS method.

Acknowledgements

Charlie Bayne was supported in part by the UCSD Graduate Training Program in Cellular and Molecular Pharmacology through an institutional training grant from the National Institute of General Medical Sciences, T32 GM007752.