mspms: a R package and graphical interface for the processing and analysis of multiplex protease substrate profiling data.

Charlie Bayne1,2,3, Brianna Hurysz1,2,3, David J. Gonzalez2,3,4, Anthony O'Donoghue2,3

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

4Center for Microbiome Innovation, University of California at San Diego, La Jolla, California 92093, USA

Abstract:

Multiplex Substrate Profiling by Mass Spectrometry (MSP-MS) is a robust method for determining the substrate specificity of proteolytic enzymes. It is a valuable tool for researchers studying proteases, enabling the design of novel reporter substrates, inhibitors, diagnostics or protease-activated therapeutics. In addition, MSP-MS has been used to discover new enzyme activities in complex biological samples. However, analyzing the resulting mass spectrometry data is a complex process involving multiple steps, often presenting challenges for researchers without strong programming skills. To address this, we developed the mspms R package, complemented by a user-friendly graphical interface. This software facilitates efficient and reproducible MSP-MS data analysis, adhering to best practices. Here, we demonstrate its application by characterizing the sequence specificity of four well-studied cathepsin proteases. Our results validate the package's ability to accurately determine expected sequence specificities while offering enhanced capabilities compared to previous MSP-MS analysis tools.

Introduction:

Multiplex substrate profiling by mass spectrometry for proteases (MSP-MS) is a method for determining protease substrate specificity1. It involves incubating a rationally designed peptide library with a protease directly or a protease-containing biological sample under defined conditions (e.g. in the presence and absence of a protease inhibitor). The resulting cleaved peptides are then identified and quantified using mass spectrometry to reveal the enzyme's substrate preferences 2.

MSP-MS is a versatile technique accessible to a broad range of researchers due to its minimal requirements: a synthetic peptide library, a mass spectrometer, and a protease-containing sample. While conceptually straightforward, MSP-MS generates complex, high dimensional datasets that pose significant analytical challenges for biochemists to analyze. Rigorous data analysis is essential for accurate interpretation and involves multiple steps. These include data preprocessing, quality control, normalization, outlier determination, imputation of missing values, identification of cleavage motifs and subsequent statistical analysis and visualization.

A robust data analysis pipeline is essential for any experimental method generating complex data. Poorly documented analysis is a pervasive issue in biological research, contributing to the retraction of several high-profile papers in recent years3. This lack of transparency hinders data reproducibility and portability between labs, as it necessitates specialized programming knowledge. In academic settings, where code ownership often transitions with personnel, codebases can become opaque "black boxes," susceptible to errors as they evolve through decentralized use. While software engineering offers potential solutions4, their adoption in biology labs is often impeded by the required expertise and time investment.

To overcome these issues, we developed mspms: a well-documented, portable, and reproducible R package specifically designed for MSP-MS data analysis. This package adheres to best practices in software and data analysis, ensuring transparency and facilitating reproducible research. Recognizing that the potential user base may not be familiar with R programming, we also developed mspms-shiny, a graphical user interface with the core features of the mspms package. Mspms-shiny is available on the web at https://gonzalezlab.shinyapps.io/mspms\_shiny/, or can be downloaded for local use on any computer. Together, these user-friendly tools in combination with the powerful MSP-MS method provide a robust and widely applicable resource for the protease research community.

Here we demonstrate the effectiveness of mspms by analyzing publicly available MSP-MS data for several well-studied cathepsin proteases. Cathepsins are the most abundant lysosomal proteases that possess a range of diverse substrate specificities5. Among those profiled here, cathepsin A exhibits carboxypeptidase activity6, cathepsin B displays dipeptidyl carboxypeptidase activity7, cathepsin C has dipeptidyl aminopeptidase activity8, and cathepsin D possesses endoprotease activity9,10. As the specificity of these cathepsins have been well-documented, these enzymes serve as an ideal benchmark to evaluate the performance of mspms-facilitated analysis.

Materials and Methods

**Package Structure**

mspms contains 5 categories of functions: those that are responsible for preprocessing data, performing / visualizing QC checks, processing data, performing statistics, and visualizing experimental results.

**Data Preprocessing**

These functions aim to ingest data originating from different sources into a consistent format that can be used for subsequent analysis.

*Preprocessing*

Mspms is designed as a downstream analysis tool for peptide identification and quantification of computationally intensive software packages. While these software platforms produce largely equivalent information, their output formats often differ. To ensure compatibility with diverse software, mspms incorporates functions that standardize this data into a unified format for subsequent analysis. For instance, prepare\_peaks() and prepare\_pd() functions handle data exported from PEAKS11 and Proteome Discoverer12 respectively.

*All possible cleavages of peptide library*

It is necessary to identify all possible cleavage sites in a peptide library for use as a control dataset to compare to the experimentally determined cleavages sites. The control data is essential for the downstream cleavage 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 specified length.

**Quality Control Checks**

These functions allow the user to assess the quality of experimental data and diagnose potential problems.

*Determining the percentage of peptide library detected in experiment*

We provide the qc\_check() function to calculate the coverage of the peptide library that was detected in the MSP-MS experiment on a per sample basis. This function considers the number of full length (undigested) peptides that are detected relative to the number of full-length peptides that known to be present in the library mixture. In addition, the peptide coverage is also calculated when considering both full length peptides and cleaved products.

*Determining which peptides from library were not detected.*

The find\_nd\_peptides() function identifies the peptides in the library that were not detected in each sample.

*Visualizing peptide library coverage*

The percentage of the peptide library that is detected in each sample can be visualized using the plot\_qc\_check() function. A histogram containing the count of the number of samples per percentage missing is plotted with facets containing each group of replicates.

*Visualizing the library peptides that were not detected*

The results of the find\_nd\_peptides() function can be summarized and plotted using plot\_nd\_peptides(). This function plots the percentage of samples in the experiment where the indicated library peptide was not detected. Two facets are shown, one in which only full-length peptides are considered, and another where both full length and cleavage products are considered.

*Visualizing the distribution of peptides as a function of retention time*

The plot\_rt\_qc() function is used to plot a histogram containing the when aggregated by each group containing replicate samples.

**Data Processing**

*Data Normalization*

The normalyze() function performs median normalization as implemented in the NormalyzerDE R package13. 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*

The remove\_outliers() function detects outliers using Dixon’s Q test as implemented in the outliers R package 14. Values determined to be outliers (p < 0.05) are converted to NA (not applicable).

*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 15. Then values are imputed using these parameters utilizing the truncated normal distribution as implemented in the truncnorm R package 16. Outlier values are not imputed.

*Cleavage Motif Determination*

The experimentally detected peptides are assigned to the corresponding full length library peptide using the join\_with\_library() function. Cleavage motifs are then generated for each detected peptide using the add\_cleavage() function. This function reports amino acids on each side of the cleavage site, with the option to set a user defined distance from the scissile bond. If there is not an amino acid at a certain 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 or C terminus.

*Consolidating Data*

The prepare\_for\_stats() function converts data from a wide to long format, and appends all of the metadata contained in the design matrix. The polish() function then removes peptides that have been cleaved on both the N and C terminus, and then combines all cleavage motifs into one column of the data frame to facilitate downstream analysis.

**Standard normalization workflow**

To streamline the MSP-MS data normalization process, we provide the mspms() function. This comprehensive function integrates multiple essential steps into a single workflow. By inputting preprocessed data, the design matrix, the number of cleavage sites, and the peptide library, users can efficiently perform data normalization, outlier identification and removal, missing value imputation, cleavage motif determination, and data consolidation.

**Statistics / Preparation for Visualization**

Given the diversity of experimental designs and research objectives that utilize MSP-MS, a wide range of statistical tests can be applied. While we encourage researchers to implement their preferred statistical methods, mspms provides a foundation by offering a selection of commonly used statistical tests. These tools can serve as a starting point for more complex analyses.

*Anova*

The mspms\_anova() function can be used to perform an anova for each peptide when grouped by experimental condition to test for an effect of time as implemented in the Rstatix package. P values are subsequently corrected by determining the False Discovery Rate (FDR correction).

*Log2 Fold Change*

The log2 fold change for each condition relative to time 0 is calculated by the mspms\_log2fc() function.

*T tests*

Pairwise t-tests are performed as implemented in the Rstatix package 17 and are subsequently FDR corrected. A number of functions are provided for different use cases.

* mspms\_t\_tests(): calculates t-tests on data relative to time 0.
* log2fc\_t\_test(): performs t-tests and log2 fold change on data relative to time 0
* log2fct\_time(): calculates the log2 fold change and t-tests for each peptide keeping the condition constant and comparing user specified times.
* log2fct\_condition(): calculates the log2 fold change and t-tests for each peptide keeping the time constant and comparing user specified conditions.

*Cleavage site position within peptide*

We provide the count\_cleavages\_per\_pos() function to summarize the number of peptides in a data frame by the position within the peptide sequence at which they were cleaved.

*Icelogo*

We provide a series of functions to process data in order to create an iceLogo18.

* calc\_AA\_count\_of\_motif(): given a vector of inputs, this function returns a matrix of the counts of each amino acid at each position.
* calc\_AA\_prop\_of\_motif(): Given a matrix containing the count of amino acids at each position, this function calculates the proportions.
* calc\_AA\_motif\_zscore(): Calculates the zscores given the count and proportion of the amino acids at each position of a set of cleavages in reference to a background set.
* calc\_sig\_zscores(): given a matrix of zscores, determines which are significant given a user defined p value.
* calc\_AA\_percent\_difference(): calculates the percent difference of a matrix containing the proportions of amino acids at each position relative to a background matrix.
* calc\_AA\_fold\_change(): calculates the fold change of a matrix containing the proportions of amino acids at each position relative to a background matrix.
* prepare\_sig\_p\_diff(): filters matrix of amino acid counts by position to only contain those that are significant given a desired alpha threshold.
* prepare\_pd(): prepares a matrix with significant amino acid percent differences by position for plotting.
* prepare\_fc(): prepares a matrix with significant amino acid fold changes by position for plotting.
* prepare\_icelogo\_data(): combines the functions described above to generate an icelogo matrix of significantly enriched amino acids at each position of a cleavage motif.
* extract\_re(): extracts regular expressions specifying amino acids above a defined threshold cutoff of each position of a cleavage motif from an icelogo matrix.

**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.

*Heatmap/* *Hierarchical Clustering*

We provide plot\_heatmap() to visualize a heatmap of the normalized intensity, where the rows and columns have been subjected to hierarchical clustering and the color of the cells represent the values the data as implemented in the heatmaply R package19. Euclidian distance and the complete agglomeration method are used to perform the hierarchical clustering.

*Principal Component Analysis*

We provide plot\_pca() as a convenience function to visualize the first two principal components of the data colored by time, and label shapes showing the experimental condition using the ggplot2 package20. Only complete cases are considered, the peptides determined to have outliers are omitted.

*Visualization of Sequence Specificity by iceLogo plots*

We provide plot\_icelogo() to create an iceLogo plot that illustrates the frequency of amino acids in each position surrounding the experimentally determined cleavage site when compared to all possible cleavage sites. A vector of user supplied sequence motifs are compared to the background of all possible combinations in the peptide library used for the experiment. The ggseqlogo R package 21 is then used to plot the underlying data.

The plot\_all\_icelogos() function generates icelogos representing enriched sequences relative to time 0 for each condition. This function compares cleavage sequences corresponding to peptides that are significant (p.adj < 0.05 and log2 fold change > 3) to a background of all possible sequences present in the initial peptide library.

*Time Course*

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

*Cleavage Position*

We provide the plot\_cleavage\_per\_pos() as a function to easily visualize the number of cleavages per position of the peptide library using the ggplot2 package.

**Standardized Report**

We provide the generate\_report() function to generate a .html file containing a standardized set of plots rendered from a .RMD template file. This function takes processed data, the design matrix, the number of cleavage sites, and the peptide library used in the experiment as arguments. This output is intended to be a standardized method to produce a generic mspms analysis that can quickly and easily be applied to any experiment.

**Data Used for Study**

Raw data from a previously reported MSP-MS experiment was acquired from the MassIVE Repository (accession number MSV00008595).22 In brief, these data were generated from a study that utilized a 228-member peptide library that was incubated with either X nM cathepsin A, X nM cathepsin B, X nM cathepsin C and X nM cathepsin. The concentration of each peptide in the library was 0.5 μM. After incubation at 37°C for defined time points, the reaction was quenched by addition of GuHCl. These samples were desalted with C18 spin columns, and ~0.4 μg of each sample was subjected to LC-MS/MS analysis using an Ultimate 3000 HPLC and Q-Exactive mass spectrometer. LC and MS parameters have been previously reported22.

**Peaks Processing**

Data from all mass spectrometry runs was processed using PEAKS software, ultimately producing a protein-peptides-id.csv and protein-peptides-lfq.csv file. For each sample experiment specific parameters were specified as follows: Q-Exactive instrument, HCD fragmentation, no enzyme. Scans were merged with a retention time window of 0.8 min, and precursor m/z error tolerance of 10 ppm. Precursor mass was corrected. Scans were filtered to include retention time between 0 and 95 min with a precursor mass tolerance of 10 ppm. For identification, a precursor mass tolerance of 20 ppm using monoisotopic mass and a fragmentation ion of 0.01 Da was specified. No PTMs were included in the search. The database used in the search was the 228-member peptide library described previously 2. FDR was estimated using decoy-fusion strategy. Label free quantification was performed with a mass error tolerance of 9 ppm, retention time shift tolerance of 3 minutes. Replicate samples were added to new groups.

The protein-peptides-lfq.csv file was prepared by navigating to the quantification options setting the normalization factor to “No normalization”, changing peptide filters to include all peptides (quality >= 0, Avg.Area >= 0, Peptide ID Count >=0, charge +1 - +10, and at least 1 confident sample). Protein filters were changed so no filtering occurs (Significance >=0). Supporting peptides were then exported and file name was changed to protein-peptides-lfq.csv.

The protein-peptides-id.csv file was prepared by adjusting the FDR setting to 1%. Supporting peptides were then exported and file name was changed to protein-peptides-id.csv.

Results:

The mspms R package was designed to be flexible and robust while providing functionality that prioritizes convenience and reproducibility of analysis. This is implemented as a package of modular functions that are unified by one of five goals, preprocessing the data, performing quality control checks, processing data, statistics/data preparation, and visualization (Figure 1A).

To evaluate the quality of the underlying mass spectrometry data from the MSP-MS experiments with cathepsins A, B, C and D, we applied the quality control functions of the mspms package to the data. Upon visualizing these results, we saw that > 90% of the full-length peptide library was detected for all samples at time zero, T0 (Supplementary Figure 1A), and >95% of the peptide library was detected when considering cleavage products in addition to the full-length peptides (Supplementary Figure 1B). Further investigation into the identity and consistency of the missing peptides showed that 5 peptides in the peptide library were not detected at any point in the study (Supplementary Figure 1C).

We next applied the mspms package to investigate global data patterns using principal coordinate and unsupervised hierarchical clustering analyses. Principal component analysis revealed tight clustering of replicates withing each experimental group (defined by condition and timepoint), as evidenced by the 95% confidence intervals of the eclipses surrounding the points (Figure 2A). Replicates from identical conditions and timepoints exhibited near-perfect clustering while distinct peptide groups demonstrated differential abundance across condition and timepoints (Figure 2B). ~~Notably, T0 samples also clustered based on cathepsin type, with clear distinctions between Cathepsin A & C, and Cathepsin B & D groups (Figure 2A +B).~~

We then analyzed the significant differences for each cathepsin relative to time 0 by performing t-tests, calculating the log2 fold change, and plotting the resulting data as volcano plots, observing many significantly increased peptides (as defined as peptides having a log2 fold change >3 and p.adj <= 0.05 at any time point relative to time 0, after incubation with each cathepsin (Figure 3A). The number of significantly different peptides were observed to increase for each cathepsin as a function of time (Supplementary Figure 2).

To evaluate the cleavage location preference of each enzyme, we examined where each scissile bond was located within the 14-mer peptides (Figure 3B). We observed a clear enrichment of cleaved substrates at the C terminus of the peptide library for cathepsin A and B, while substrates cleaved by cathepsin C sites were enriched at the N terminus. Cathepsin A and B could be distinguished from each other as cathepsin B generally removed two amino acids from the C-terminus and so the majority of cut sites were at position 12 (between amino acid 12 and 13) with sequential processing of these peptides occurring at position 10. In contrast, cathepsin A cleaves one amino acid from the C-terminus and therefore cleavage at position 13 was more abundant than at position 12. Subsequent trimming of peptides occurred at positions 11, 10, 9 and 8.

Cathepsin D is an endoprotease and prefers to hydrolyze peptides at sites that are more distal from N- and C-termini. Therefore the location of cleavage sites within the peptide library were distributed within the central location of the peptides with position 8 being the most frequent (Figure 3B).

To visualize amino acid substrate preferences at each position, we used mspms to conduct an iceLogo analysis, generating substrate specificity plots for eight positions surrounding the cleavage site (P4′ to P4). Only cleavage sequences from peptides determined to have a log2 fold change ≥ 3 and FDR adjusted p value of ≤ 0.05 relative to time 0 were considered. These sequences were then compared to the background of all possible cleavage sequences present in the peptide library (n = 2,964) where residues with a p value ≤ 0.05 are reported. We observed that Cathepsin A displayed a strong preference for cleavage of peptides containing norleucine (n), leucine and phenylalanine in the P1 and P1′ position, while aspartic acid was the least preferred amino acid in these positions. Isoleucine, proline, phenylalanine and valine were preferred in the P2 position, no enrichment of any individual amino acid in the P3 position, while phenylalanine was enriched in P4. As cleavage occurs at the C-terminus of peptides, there was no amino acids present in the P2′ to P4′ position, which is illustrated with X (Figure 3C).

Cathepsin B showed a strong preference for tryptophan at P2′, phenylalanine and norleucine at P1′, and lysine, norleucine, and arginine at P1. As this enzyme prefers to remove two amino acids from the C-terminus, there was a high frequency of X in the P3′ and P4′ positions. Cathepsin C showed a strong preference of X at P4 and P3, and an aversion for X at P3′- P4′. Histidine was preferred at P2 and P1, while phenylalanine was preferred at P1’ and P2’ (Figure 3C). Lastly, Cathepsin D frequently cleaved peptides with phenylalanine and norleucine at P1 and P1’ and arginine in P2, P2′ and P3′. Since this is an endopeptidase, there was an aversion for X at P4-P2 and P2′-P4′ (Figure 3C).

Discussion:

Prior to the development of the mspms package, MSP-MS data analysis relied on a collection of in house R scripts. While functional, this approach suffered from several critical limitations. First, these scripts consisted of a monolithic code base that was decentralized and not well documented. Second, the analysis process was fragmented, involving manual data manipulation between intermediate spreadsheets making it challenging to track data transformations and ensure reliability. Finally, adapting the scripts to accommodate experimental variations required significant R programming expertise, limiting their flexibility and accessibility.

In creating the mspms R package, we have addressed these limitations. (1) This tool is accessed through a centralized location, and functionality is broken down into discrete, modular functions. This logical approach makes package maintenance/testing and the development of new features easy to perform. (2) Analysis performed by the mspms package is entirely self-contained, facilitating reproducible research while allowing us to create mspms-shiny, a GUI application that allows for the user to access the core features of the package without requiring any R programming knowledge. To enable this, key features of iceLogo, a highly cited Java based application (ref) that is no longer actively supported, were implemented in R. Our implementation of iceLogo is even more adaptable to mspms experiments, as it can handle nonstandard amino acid letters, such as X which represents a post terminal site, and lowercase n representing the non-natural amino acid, norleucine. (3) Mspms was designed for broad applicability, supporting diverse experimental designs. For example, mspms allows users to define a peptide library, opening up the possibility to easily extend MSP-MS analysis beyond the previously described 228-peptide library. Additionally, mspms can analyze the cleavage sequence motifs of any number of amino acids surrounding the cleavage site, while previous analysis was limited to four amino acids on each side of the cleavage site (Supplementary Figure 4). Lastly, mspms supports analysis of data exported by multiple upstream proteomics software. PEAKS and Proteome Discoverer are currently supported, and there is the option to easily add more based on user demand in the future. As there is substantial variability in the proteomic software of choice lab to lab, this is a crucial feature enabling broad applicability of the tool.

In this work, we demonstrate the core features of mspms as applied to publicly available MSP-MS data profiling several well-studied cathepsin proteases spanning all three families of cathepsins as cathepsin A is a serine protease, cathepsins B and C are cysteine proteases and cathepsin D is an aspartic acid protease. We use the mspms package to profile the substrate specificity of each of these enzymes and our results align with previously published data (refs).

A critical initial step in MSP-MS data analysis is a rigorous quality control assessment. Given the method's reliance on a specific peptide library, calculating the percentage of library peptides detected in each sample provides a straightforward yet powerful indicator of data integrity. Anomalously low detection rates signal potential instrument or experimental problems.

While theoretically all peptides in a defined library should be detectable in each MSP-MS sample, practical limitations, such as mass spectrometer performance and LC conditions, often hinder complete detection. Identifying consistently missing peptides is crucial for troubleshooting and method optimization.

Potential causes for missing peptides include library preparation errors or suboptimal LC-MS parameters. To address the latter, mspms provides a histogram of peptide retention times, aiding in gradient optimization for better peptide distribution and improved detection.

Application of our quality control metrics to the current dataset revealed high detection rates for the majority of library peptides. A small subset of peptides were not detected in any sample. Further inspection into the retention time histogram suggests that the LC-MS method used with these samples could be further optimized to promote better detection by spreading out the peptides more evenly over the gradient (Supplementary Figure 2). While complete library coverage is ideal, the data generated here for the cathepsin assays is of sufficient quality to make meaningful conclusions about the underlying biology.

Subsequent to data quality assessment, a high-level analysis of the data is essential to verify experimental outcomes. To facilitate this exploration, the mspms package offers, principle ordination analysis23 or unsupervised hierarchical clustering analysis visualized via a heatmap by using the plot\_pca() or plot\_heatmap() functions, respectively. These methods allow the user to inspect the broad level features of the data and ensure that the experiment performed as expected. A proper MSP-MS experiment will have been designed with appropriate positive and negative controls, allowing the user to definitively know what was expected and how the observed results agree.

Initial high-level analysis of our MSP-MS data showed a tight clustering of replicates by experimental group (Figure 2A, B), suggesting good agreement between replicate samples. Inspection of the heatmap showed clear clusters of peptide cleavage products that were distinct to each condition (Figure 2B). Furthermore, several of these distinct clusters of peptides were seen to increase as a function of incubation time. Together, this shows that different cathepsins produce different cleavage products when evaluated using the MSP-MS technique, suggesting different substrate preferences.

Interestingly, inspection of the T0 samples showed a clear separation by cathepsin present in the sample. This was unexpected as the cathepsins at T0 were denatured in 4M urea2 prior to exposure to the peptide library and are not anticipated to be catalytically active (citation for this? Any chance they could actually be active in 4M urea?). Thus the observation that the T0 samples clustered separately by condition suggested the presence of batch effects between experimental groups. These batch effects could be due to either the sample preparation factors, or mass spectrometry runs. In our analysis, these batch effects are appropriately controlled for by restricting our analysis to only consider differences for each condition relative to time 0, as has typically done when analyzing MSP-MS data. However, it is important to note that removal of these batch effects would make additional types of analysis appropriate to perform (i.e. comparing cathepsin B cleavage products after 30 minutes of incubation to cathepsin A cleavage products at 30 minutes). Eliminating these batch effects would require additional experiments to identify appropriate randomization and blocking schemes, that could in turn easily be evaluated using the mspms framework.

Employing the mspms R package, we sought to validate the expected substrate specificity of each cathepsin profiled through the use of cleavage position and iceLogo plots. Our investigation of cathepsin A revealed carboxypeptidase activity, as previously described in the literature 6. Analysis of cathepsin B revealed clear evidence of expected di-carboxypeptidase activity24,7, as indicated by the high frequency of cleavages at position 12 which corresponds to a loss of a dipeptide from the C terminal end. Cleavage at position 10 suggests subsequent removal of a second dipeptide from the same substrate.

We then analyzed cathepsin C, also known as dipeptidyl peptidase I (DPP-I), which is a lysosomal dipeptidyl-peptidase belonging to the cysteine cathepsin protease family. This enzyme has relatively broad substrate specificity and removes dipeptides from the N-terminus of the cleavage site. It has been reported that this enzyme is unable to cleave substrates when the N-terminal amino group is blocked or when a proline residue is found in either the P1 or P1′ position. In addition, N-terminal lysine or arginine blocks cleavage8. Our mspms based analysis showed clear dipeptidyl aminopeptidase activity as the most frequently cleaved position is between the 2nd and 3rd amino acids (position 2) which corresponds to the removal of a dipeptide from the N-terminus. Peptides were also cleaved at position 4 that increased over time which suggests removal of a second dipeptide from the N terminal end. The dipeptidyl aminopeptidase activity is reflected in the icelogo by the strong preference for X at P4 and P3.

From the iceLogo, we were not able to detect evidence supporting the aversion of cathepsin C to proline at P1 or P1′, or lysine and arginine at P1. However, the flexible nature of the mspms R package permitted further exploration of the data which allowed us to filter the data using a regular expression allowing us to eliminate the cleavage products that would not be expected to be cleaved by cathepsin C when considering the aforementioned substrate specificities ("XX[^L|K|X][^P][^P]..."). We saw the remaining peptides (that would be expected to be cleaved based on the literature) were significantly increased over time (Supplementary Figure 4). To evaluate the opposite case, we also filtered our dataset to only include peptides that would not be expected to be cleaved by cathepsin C, including peptides with a lysine or arginine at the P2 position, or a proline at P1 or P1’, finding no significant differences in mean intensity when comparing 0 to 90 minutes of incubation. We additionally compared all peptides present in the experiment, as well as a reversed rule ("...[^P][^P][^L|K|X]XX"), finding no significant differences in mean peptide intensity over time for both cases. Together, this suggests that the previously described rules for cathepsin C specificity are discernable using the mspms package, but additional features defining optimal substrate specificity may also exist.

Finally, we profiled cathepsin D, a lysosomal aspartyl protease that has endoprotease activity. It is known to prefer hydrophobic residues on both sides of the cleaved bond (P1 and P1′)10,9. Mspms based analysis showed that cathepsin D has endopeptidase activity, both by the position plot, and iceLogo. The iceLogo further displays a clear preference for phenylalanine (F) and norleucine (n), which are both hydrophobic residues.

Together, these results show that the mspms R package is able to analyze MSP-MS data recapitulating expected substrate specificities. While we analyzed four well-studied cathepsins in the current work, this type of analysis can be extended to any set of samples profiled by the MSP-MS method, whether that be complex biological mixtures that contain numerous proteases, or purified proteases of interest. Due to the involvement of proteases in a wide variety of biological processes and the importance of determining their substrate specificities, we expect mspms to be a widely used tool facilitating the analysis of MSP-MS data across many research groups.

Conclusions

Here we present the mspms R package and show that it is a robust, flexible, and simple-to-use tool allowing researchers to analyze data produced by the MSP-MS method to probe protease specificity in their sample(s) of choice.

Availability of Source Code and Requirements:

* Project name: mspms
* Project home page: <https://github.com/baynec2/mspms>.
* Operating system(s): Platform independent
* Programming language: R
* Other requirements: R 4.1.0, dplyr, magrittr, MASS, NormalyzerDE, outliers, purrr, stats, tidyr, truncnorm, utils, ggplot2, ggseqlogo, heatmaply, readr, readxl, rstatix, stringr, tibble, SummarizedExperiment, rlang, forcats, ggpubr, rmarkdown, DT.
* License: MIT
* Bioconductor repository: Submitted, TBD.
* Shiny app instance: <https://gonzalezlab.shinyapps.io/mspms_shiny/>
* Shiny app repository: <https://github.com/baynec2/mspms-shiny>
* Reproducible vignette: <https://github.com/baynec2/mspms>/vignettes/mspms\_analysis.RMD
* Manuscript repo: <https://github.com/baynec2/mspms-manuscript>.

Availability of Supporting Data and Materials:

All data used to build this manuscript can be found in the github repository for the manuscript.

Figures

**A black and white diagram

Description automatically generated**

**Figure 1. Overview of the mspms R package and cathepsin MSP-MS profiling.** Schematic of the functions contained within the mspms R package.

A close-up of a diagram

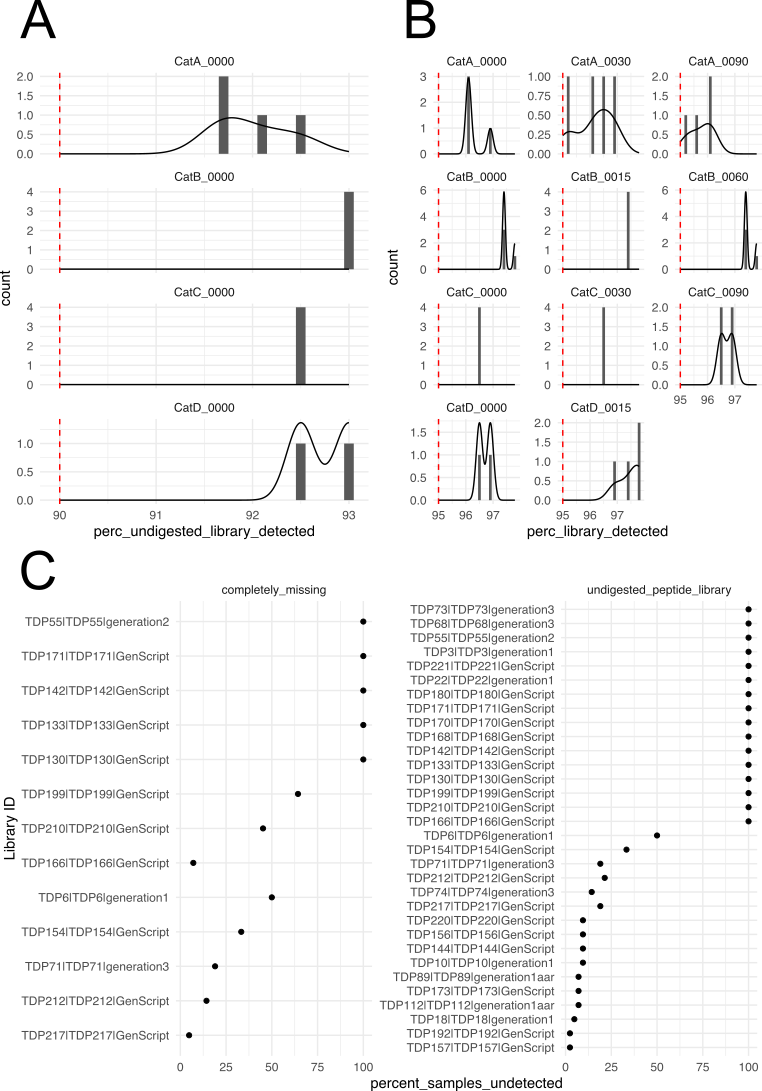
Description automatically generated

**Figure 2. Global visualization of MSP-MS data.** (A) Principal component analysis displaying PC1 and PC2. Samples are colored by time, while the shape and line type show the type of cathepsin with eclipses representing the 95% confidence interval. (C) Heatmap showing the results of the experiment as clustered using unsupervised hierarchical clustering. Rows of the heatmap represent the samples while columns represent the peptides. Color of the heatmap cells represent the normalized, column centered, and scaled values. White cells represent those that were found to be outliers and converted to NA. Colored bars to the right of the heatmap indicate the cathepsin and time of the samples in each row. Colored bars corresponding to each peptide in the columns display whether the corresponding peptide is a full-length peptide belonging to the 228-member peptide library (non -cleaved, dark blue) or a cleavage product (cleaved, blue).

A collage of graphs and diagrams

Description automatically generated**Figure 3. Differentially abundant peptide cleavages over time.** (A) Volcano plots displaying the log 2-fold change of the timepoint as indicated by color relative to and -log10 FDR corrected p values for each cathepsin. (B) Plot showing the number of significant cleavage events at each position of the peptide library (as defined as having a log2fc >= 3 and FDR adjusted p values <= 0.05) (C) Icelogo plots as implemented in the mspms package. Amino acid residues (with X representing positions past the terminus) four positions to the left and right of the cleavage site are displayed. Only residues with significantly higher proportions relative to the proportion of all possible cleavage sequences present in the initial peptide library (pval <= 0.05) are shown, with the height representing the percentage differences.

Supplementary Data:

****

**Supplementary Figure 1. Quality Control Evaluation.** (A) Histogram displaying the count of samples for each sample grouping at time 0 as a function of the percentage of full-length peptides detected. (B) Histogram displaying the count of samples for each sample grouping as a function of the percentage of both the full-length and cleavage products mapping to the 228-peptide library used in the experiment. (C)



**Supplementary Figure 2. Retention Time Evaluation.** Histogram displaying the count of peptides detected as a function of retention time, with the width of bins set to 1 minute. The percentage of solvent B over time is shown overlayed as a dashed red line.



**Supplementary Figure 3. Number of Significant Differences Relative to T0 as a Function of MSP-MS Incubation Time.** The number of significantly enriched peptides relative to time 0 as shown per duration of time incubated with the indicated cathepsin.

**Supplementary Figure 4. Assessment of Cathepsin C for Reported Substrate Preferences.** Mean normalized intensities for peptides compared between T0 and T90 for cathepsin C. (A) Cleavages expected to increase over time based on cathepsin C specificities reported in the literature, peptides with a proline adjacent to the cleavage site (P1 or P1’ position), or with N terminal residues of lysine or arginine have been excluded. Peptides expected to not be cleaved based on reported cathepsin C specificities, (B) peptide cleavage products with Lysine or Arginine at the N terminus, (C) peptide cleavage products with proline at position P1, (C) peptide cleavage products with proline at P1’, peptide cleavage products with the reversed rule from , (F) all peptides . Statistics show the results of T-tests. P values are denoted as follows: ns > 0.05 \* <= 0.05, \*\* <= 0.01, \*\*\* <= 0.001, \*\*\*\* <= 0.0001.

****

**Supplementary Figure 5 . Icelogo Analyses With Extended Cleavage Motifs.** Icelogo analysis of peptide cleavage motifs containing the 6 amino acids before and after the significantly enriched peptides with detected cleavage sites relative to all possible within the background universe of the 228-peptide library used for the experiment.

Abbreviations:

MSP-MS: Multiplex Substrate Profiling by Mass Spectrometry. T0: Time zero. PCA: Principal Component Analysis. PC1: Principal component 1. PC2: Principal component 2.

Competing Interests:

The authors declare that they have no competing interests.

Funding:

Charlie Bayne and Brianna Hurysz were 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.

Author contributions

C.B wrote the R package, shiny app, documentation, and manuscript; B.H contributed to software development. D.J.G. and A.J.O oversaw the project.

Acknowledgements

We would like to acknowledge Dr. Zhenze Jiang for his initial conception of the previous R scripts, as well as Dr. Lawrence Liu and Dr. Michael Yoon for maintaining and updating the scripts. We also thank Dr. Jiang and Dr. Yoon for running the cathepsins MSP-MS experiment analyzed in this manuscript. Finally, we would like to acknowledge Diego F. Trujillo for suggestions to improve mspms and for educating new users how to perform the data analysis.

References

1. O’Donoghue, A. J. *et al.* Global identification of peptidase specificity by multiplex substrate profiling. *Nat Methods* **9**, 1095–1100 (2012).

2. Rohweder, P. J., Jiang, Z., Hurysz, B. M., O’Donoghue, A. J. & Craik, C. S. Multiplex substrate profiling by mass spectrometry for proteases. in *Methods in Enzymology* vol. 682 375–411 (Elsevier, 2023).

3. Miller, G. A Scientist’s Nightmare: Software Problem Leads to Five Retractions. *Science* **314**, 1856–1857 (2006).

4. Gentleman, R. C. *et al.* Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology* (2004).

5. Yadati, T., Houben, T., Bitorina, A. & Shiri-Sverdlov, R. The Ins and Outs of Cathepsins: Physiological Function and Role in Disease Management. *Cells* **9**, 1679 (2020).

6. Hiraiwa, M. Cathepsin A/protective protein: an unusual lysosomal multifunctional protein. *Cellular and Molecular Life Sciences (CMLS)* **56**, 894–907 (1999).

7. Yoon, M. C., Hook, V. & O’Donoghue, A. J. Cathepsin B Dipeptidyl Carboxypeptidase and Endopeptidase Activities Demonstrated across a Broad pH Range. *Biochemistry* **61**, 1904–1914 (2022).

8. Stojkovska Docevska, M. & Novinec, M. Cathepsin C: structure, function, and pharmacological targeting. *Rare Dis Orphan Drugs J* **2**, (2023).

9. Ferguson, J. B., Andrews, J. R., Voynick, I. M. & Fruton, J. S. The Specificity of Cathepsin D. *Journal of Biological Chemistry* **248**, 6701–6708 (1973).

10. Sun, H. *et al.* Proteolytic Characteristics of Cathepsin D Related to the Recognition and Cleavage of Its Target Proteins. *PLoS ONE* **8**, e65733 (2013).

11. Ma, B. *et al.* PEAKS: powerful software for peptide *de novo* sequencing by tandem mass spectrometry. *Rapid Comm Mass Spectrometry* **17**, 2337–2342 (2003).

12. Orsburn, B. C. Proteome Discoverer—A Community Enhanced Data Processing Suite for Protein Informatics. *Proteomes* **9**, 15 (2021).

13. Willforss, J., Chawade, A. & Levander, F. NormalyzerDE: Online Tool for Improved Normalization of Omics Expression Data and High-Sensitivity Differential Expression Analysis. *J. Proteome Res.* **18**, 732–740 (2019).

14. Dixon, W. J. Processing Data for Outliers. *Biometrics* **9**, 74 (1953).

15. Ripley, B. *et al.* MASS: Support Functions and Datasets for Venables and Ripley’s MASS. (2024).

16. Mersmann, O., Trautmann, H., Steuer, D. & Bornkamp, B. truncnorm: Truncated Normal Distribution. (2023).

17. Kassambara, A. rstatix: Pipe-Friendly Framework for Basic Statistical Tests. (2023).

18. Colaert, N., Helsens, K., Martens, L., Vandekerckhove, J. & Gevaert, K. Improved visualization of protein consensus sequences by iceLogo. *Nat Methods* **6**, 786–787 (2009).

19. Galili, T., O’Callaghan, A., Sidi, J. & Sievert, C. heatmaply: an R package for creating interactive cluster heatmaps for online publishing. *Bioinformatics* **34**, 1600–1602 (2018).

20. Wickham, H. *et al.* ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics. (2024).

21. Wagih, O. ggseqlogo: a versatile R package for drawing sequence logos. *Bioinformatics* **33**, 3645–3647 (2017).

22. Jiang, Z. *et al.* Differential Neuropeptidomes of Dense Core Secretory Vesicles (DCSV) Produced at Intravesicular and Extracellular pH Conditions by Proteolytic Processing. *ACS Chem. Neurosci.* **12**, 2385–2398 (2021).

23. Jolliffe, I. T. & Cadima, J. Principal component analysis: a review and recent developments. *Phil. Trans. R. Soc. A.* **374**, 20150202 (2016).

24. Yoon, M. C. *et al.* Selective Neutral pH Inhibitor of Cathepsin B Designed Based on Cleavage Preferences at Cytosolic and Lysosomal pH Conditions. *ACS Chem. Biol.* **16**, 1628–1643 (2021).