MassDash: A Web-based Dashboard for Targeted Mass Spectrometry Visualization

Justin C. Sing[‡1,2], Joshua Charkow[‡1,2], Mohammed AlHigaylan[1,2], Ira Horecka[1,2], Leon Xu[1,2], Hannes L. Röst*[1,2,3]

* corresponding author hannes.rost@mail.utoronto.ca
‡ Equal Contribution

1. Donnelly Centre for Cellular and Biomolecular Research, University of Toronto, Toronto, Ontario M5S 3E1, Canada
2. Department of Molecular Genetics, University of Toronto, Toronto, Ontario M5G 1A8, Canada
3. Department of Computer Science, University of Toronto, Toronto, Ontario M5G 1A8, Canada

Keywords

visualization, mass-spectrometry, data-independent-acquisition, validation, optimization, prototyping

Abstract

With the increased usage, diversity of methods and instruments being applied to analyze Data-Independent Acquisition (DIA) data, visualization is becoming increasingly important to validate automated software results. Here we present MassDash, a cross-platform, DIA mass spectrometry visualization software for comparing features and results across popular tools. MassDash provides a web-based interface and Python package for interactive feature visualizations and summary report plots across multiple automated DIA feature detection tools including OpenSwath, DIA-NN, and dreamDIA. Furthermore, MassDash processes peptides on the fly, enabling interactive visualization of peptides across dozens of runs simultaneously on a personal computer. MassDash supports various multidimensional visualizations across retention time, ion mobility, m/z, and intensity providing additional insights into the data. The modular framework is easily extendable enabling rapid algorithm development of novel peak picker techniques, such as deep learning based approaches and refinement of existing tools. MashDash is open-source under a BSD 3-Clause license and freely available at https://github.com/Roestlab/massdash.

Introduction

Data-Independent Acquisition (DIA) is a robust mass spectrometry technique that provides a thorough analysis of complex biological samples. DIA systematically samples fragment ions leading to high coverage and quantification accuracy,[1,2] making DIA a valuable technique for a wide variety of applications.[3–5] The proteomics field is widely adopting DIA for its ability to provide in-depth and accurate insights about complex biological systems with applications ranging from cancer research to systems biology[3,6–9]. Its adoption is met with continuous advancements in mass spectrometry technology, experiment design and sophisticated algorithms[2,10–14]. For example, the original peptide-centric DIA analysis strategy[2] continues to be refined with novel methods of library generation,[12] deep learning based peak pickers,[15] and statistical scoring[13,16,17]. Furthermore, library-free approaches, using deep learning predicted peptide libraries[13] or spectrum-centric approaches[18,19] have also increased in popularity. Additionally, these approaches can be applied to novel instrumentation and sampling strategies[20–27] and chemical multiplexing[28,29]. Considering the diverse strategies and applications in which DIA is applied, there is likely not one best analysis workflow for every application or instrument. Given the complexity of DIA data, it is unlikely that there is a universal set of parameters within a tool that can be applied ubiquitously. Moreover, the abundance of available tools may overwhelm individuals, potentially steering them toward suboptimal workflows. Thus, it is more important than ever for researchers to visualize the raw data as quality control to ensure that software is functioning as expected for the given context.

Although several visualization tools for DIA exist, each has its own limitations. TOPPView[30] and Skyline[31] are well-established open-source visualization and data exploration tools with vibrant communities. However, to prioritize performance, these tools are developed in low-level languages that make it challenging for novice users to prototype new features for novel analyses. TAPIR[32] and AlphaViz[33] are two visualization tools written in Python, a user-friendly high-level language, however, they come with limitations including the requirement to store data locally in the case of TAPIR or limited support to an individual commercial vendor form in the case of AlphaViz. Furthermore, both TAPIR and AlphaViz only support OpenSwath and DIA-NN outputs respectively. To that end, we sought to provide a user-friendly, Python-based DIA visualization tool that supports quick visualization from several automated feature detection tools and enables on-the-fly testing and prototyping of new algorithms.

We present MassDash, a novel platform tailored for researchers and analysts in the dynamic realm of DIA mass spectrometry. With a lightweight visualization library, it excels in prototyping algorithms (i.e., peak picking, scoring and multi-run alignment), scaling with multiple runs, and efficiently reading results from common DIA feature detection tools.

Seamlessly integrated with Streamlit, the user experience is enhanced, offering a powerful GUI for algorithm testing and parameter optimization. Accessible both locally and remotely through a web browser, this platform offers versatile data accessibility. Locally, it optimizes performance by leveraging high-speed network connections and minimizing I/O latency for swift data processing. Remotely, it excels in managing large datasets, ensuring enhanced accessibility and scalability. Furthermore, MassDash integrates with Jupyter Notebooks to allow for reproducible scripting of visualizations and rapid algorithm development in Python. MassDash's versatility aligns with modern research needs, emphasizing flexibility and accessibility in data analysis workflows. MassDash provides advanced capabilities for chromatogram visualization, algorithm testing, and parameter optimization, thus empowering researchers to customize analyses and streamline workflows.

Methods

Modular Framework Design

MassDash was designed to be a modular framework for DIA mass spectrometry data visualization and to allow researchers to easily integrate new functionalities, algorithms, and data processing methods. MassDash's frontend GUI is designed with Python's Streamlit library for its user-friendly simplicity and ability to deploy interactive web applications with ease. Streamlit enables rapid development and customization, making it an excellent choice for delivering an accessible and adaptable platform for researchers. The web application interface also allows for remote visualization such that raw mass spectrometry data does not need to be transferred to the user’s personal computer.

On The Fly Data Processing

MassDash loads the minimum amount of data required to maximize memory efficiency by only loading data as required when requested by the user. In practice, this means that data is only loaded for the precursor requested from the user. This allows for quick initialization of files and enables visualization of the same precursor across many runs even though the runs would be too large to fit into memory concurrently.

Python Library Integration

MassDash offers a robust Python library for seamless integration into scripts and Jupyter Notebooks, ensuring maximum reproducibility when reproducing plots and analyses. This Python library extends MassDash's functionality, allowing researchers to incorporate it into their Python scripts and Jupyter Notebooks. The integration of MassDash's Python library not only enhances the platform's adaptability but also provides users with the flexibility to incorporate it into their existing workflows or novel algorithms. This feature facilitates a more comprehensive and tailored approach to data analysis, enabling researchers to leverage MassDash's capabilities in

diverse computational environments and workflows. The Python library is accompanied by comprehensive documentation to help users adapt the library to their needs.

MassDash is powered by the pyOpenMS library which contains Python bindings for the OpenMS C++ library.[34] This provides tools for loading .mzML files and an implementation of the feature extraction algorithms as used by OpenSwath. Furthermore, MassDash also integrates the Bokeh[35] and Plotly[36] libraries for interactive visualization and the Torch[37] library for integrating new deep learning peak picking techniques.

Benchmarking Code Execution Time Performance

The major computation steps: feature identification data access, mass spectrometry (MS) data access initialization, MS data extraction, plot generation and plot rendering were tracked and recorded for 200 random peptide precursor selection interactions of MassDash's GUI. These major components were measured as time execution blocks, from the beginning of code execution until the end, using python's built-in module timeit.

Benchmarking Datasets

Example precursors showcased in this manuscript are all accessible via publicly available datasets, see data availability. A gold-standard *S. pyogenes* dataset containing raw mzML DIA data, sqMass files and OpenSwath peak boundaries were acquired from Gupta. et. al. 2023[38]. DIA-NN (version 1.8.0) was run on the 16 runs to identify features using the default settings and the same library as the originally published manuscript. Six diaPASEF files (Bruker .d) from a published two-proteome dataset from Meier et. al. 2020 were downloaded and converted to .mzML using diaPysef (version 1.0.10). OpenSwath results showcased are from the original publication[20]. DIA-NN analyses were conducted for the HY_200ng_15ng replicates and HY_200ng_45ng replicates. DIA-NN analysis was conducted on version 1.8.0 with the two proteome spectral library in the original publication[20] published in Meier et al. 2020 via the command line interface with the following settings: --reanalyse, --smart-profiling, --matrices, --report-lib-info --matrix-qvalue 0.05.

MassDash test files were also created from various publicly available datasets. Benchmarking sqMass files of a synthetic phosphopeptide dataset were acquired from the PRIDE archive (PXD004573) and were subsampled for a few precursor examples using SQLite and pandas[39] in Python. Ion mobility test files were prepared using 25pc_50ng raw data and its corresponding published and newly processed results. The OpenSwath (.osw) file was created by converting and subsetting the published .tsv results file (pyprophet_export_25pc_50ng.tsv). The DIA-NN report file was created by processing using the settings outlined above. Test .mzML files were created by converting the Bruker .d file using diaPysef (version 1.0.10) and filtered using pyOpenMS[34].

Results and Discussion

MassDash's Framework and Purpose

The increasing utilization and technical advancements in high-throughput mass spectrometry (MS), particularly in DIA, have driven the development of novel algorithms for data processing and deconvolution. However, the plethora of tools and workflows presents a problem for researchers attempting to identify the optimal approach for their data. We therefore present MassDash, a software platform for visualization and algorithm development capable of cross-workflow visualization and validation of DIA results with a capable Python back-end library for rapid iteration of algorithm development to support novel workflows (Fig. 1). This comprehensive and flexible platform addresses critical needs in the field, offering visualization and validation capabilities for existing tools and serving as a robust framework for generating algorithms tailored to the challenges posed by high-throughput mass spectrometry data.

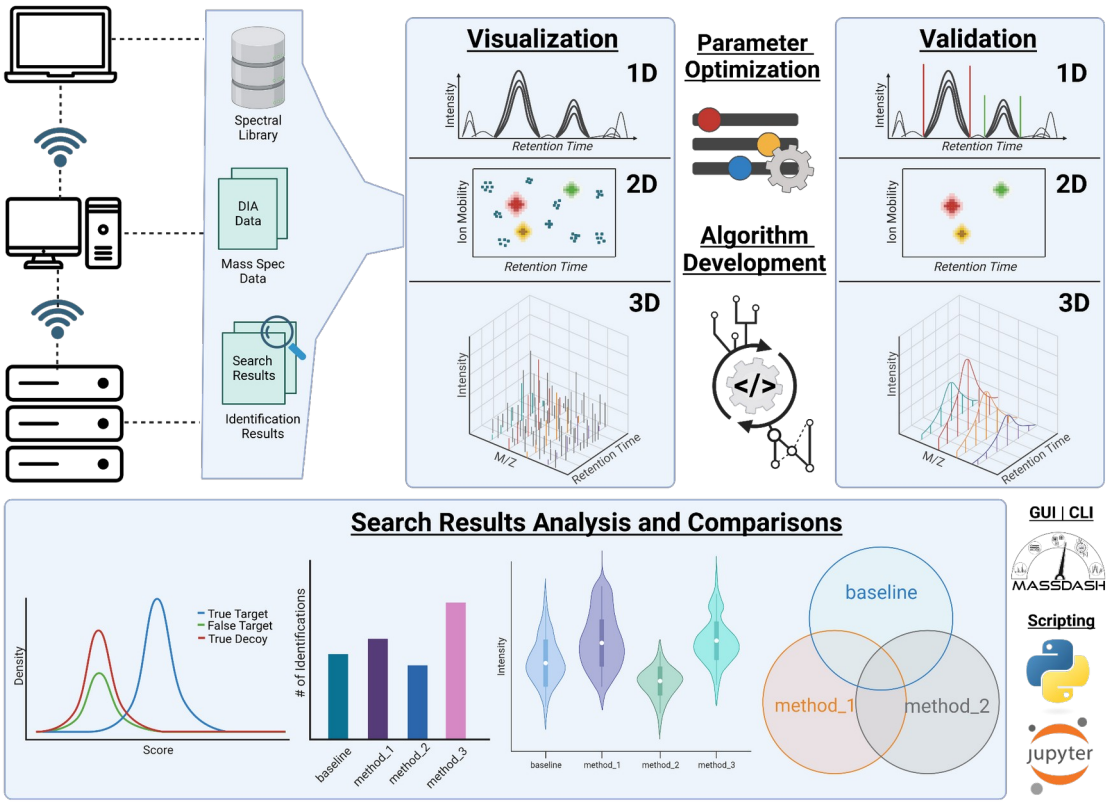

Figure 1. MassDash: A Comprehensive Toolkit for Streamlined DIA-MS Visualization, Analysis, Optimization, and Rapid Prototyping. This schematic provides an overview of the MassDash toolkit and its versatile workflows. Accessible through a graphical user interface

(GUI) deployed as a web application either locally or remotely, MassDash accepts mass spectrometry DIA/diaPASEF data, whether raw or in the form of post-extracted ion chromatograms. The tool offers diverse visualization options, including 1D, 2D, or 3D plots tailored to the data type. Raw data parameter optimization empowers users to finely tune and explore the dataset before initiating comprehensive targeted data extraction. Beyond its user-friendly interface, MassDash serves as a Python library, facilitating rapid algorithm development and testing. Users can delve into and compare search results derived from different methods, enhancing the tool's utility for robust data exploration and analysis. Created with BioRender.com (2024).

MassDash is a modular framework, allowing easy integration of new functionalities and algorithms. Leveraging the Streamlit library, MassDash provides a user-friendly graphical user interface (GUI) accessible via a web browser. Written in Python, it seamlessly integrates with other Python packages, enhancing its capacity for rapid algorithm development—particularly valuable for testing and implementing deep learning algorithms. Moreover, MassDash functions not only as a GUI but also as a versatile Python package that can be seamlessly incorporated into scripting environments, including Jupyter Notebooks (Sup. Fig. 5). Furthermore, MassDash utilizes Sqlite and pyOpenMS[34] for efficient and low-memory data access and retrieval across multiple mass spectrometry runs. Benchmark tests on major computing operations reveal that data extraction is the most time-consuming process, averaging around 0.71 seconds for a single post-extracted ion chromatogram file (XIC, sqMass) and 7.97 seconds for 16 runs of post XIC data (Fig. 2, Sup. Fig. 1, Sup. Table 1). When analyzing and visualizing raw DIA data, the extraction time is approximately 5.26 seconds for a single file and 9.42 seconds for two files. This demonstrates that if you have pre-extracted sqMass files, MassDash can efficiently and quickly extract and render XICs. In contrast, extraction from raw data requires more computation, resulting in a longer wait time; however, it still completes within seconds.

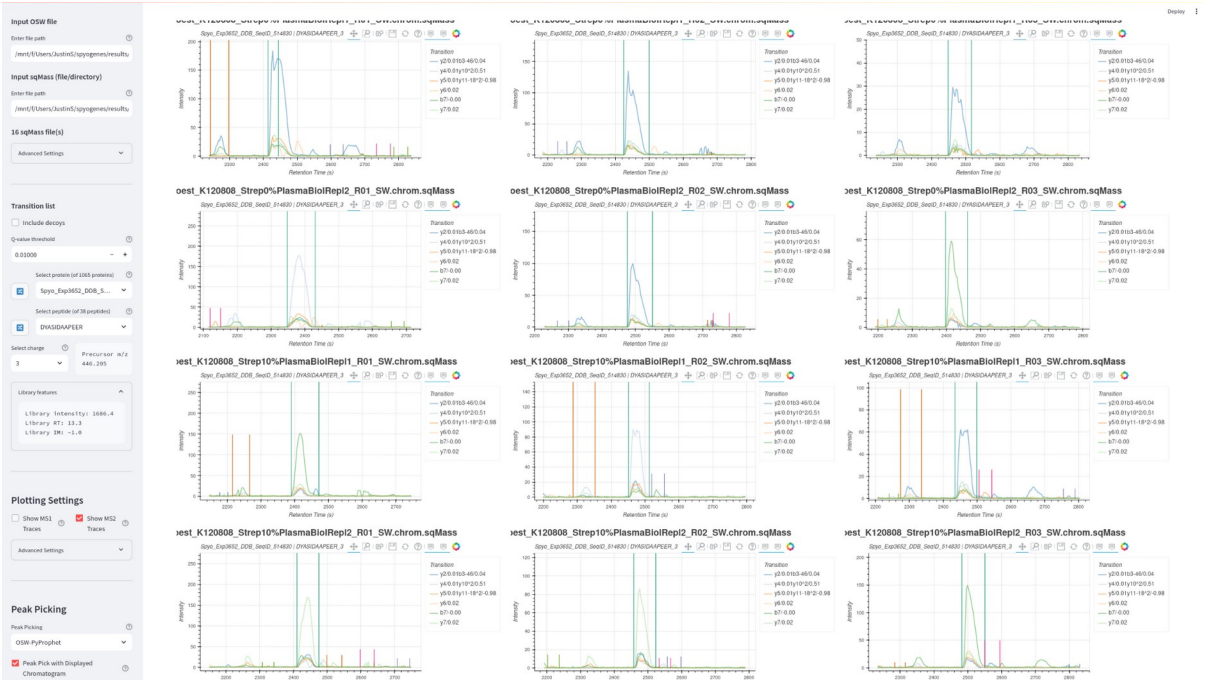

Figure 2. Visualization and Validation of Multiple Experiments with MassDash. Extracted Ion Chromatograms (XICs) are displayed for the protein Spyo_Exp3652_DDB_SeqID_514830 (charge +2) across 16 runs. The left sidebar allows users to choose the protein/peptide for plotting XICs and select different available peak-picking methods for assessment. While most runs consistently pick the most intense peak, it is noticeable that the peak in the first run is split in half by the peak-picking software (OpenSwath). This highlights the importance of validating software tools involved in data processing to ensure result quality. The data used in this example is sourced from the gold standard *S. pyogenes* data, available on PeptideAtlas (PASS01508).

The Importance of Validation and Visualization

Many automated feature detection tools typically incorporate methods to control the number of false discoveries, thereby offering a degree of confidence in the resulting identifications.[40] However, current statistical approaches in the field use a family-wise error rate (FWER) which does not provide a probability for an individual peak to be correctly picked. This works well for global analyses, but is problematic if only a few proteins are selected for follow-up analyses (as in many wet-lab biological experiments or biomarker studies) since the false positive rate among a selected subgroup of data points could be much higher than among the total number of peptides[41]. Therefore, manual validation of results is crucial to ensure that quantifications obtained from data processing tools accurately extract signals for the correct peak. This involves confirming that peaks are not incorrectly cut off, avoiding the return of questionable peaks, and ensuring consistent selection of the same peak across multiple runs (Fig. 2). This is evident in Figure 3, which depicts instances where automated software tools align with manual inspection (Fig. 3a), situations where tools converge on the same feature, but manual

validation reveals it as questionable due to a strong precursor signal but has weak fragment ion signals (Fig. 3b), and scenarios where tools identify entirely different features (Fig. 3c). MassDash facilitates the validation of feature detection tools by overlaying their detected features onto the extracted data. This capability extends beyond a single dimension, enabling the visualization of two to three-dimensional plots to assess the performance of data extraction parameters across different dimensions, including mass-to-charge, retention time, and ion mobility (Sup. Fig. 3, Sup. Fig. 4).

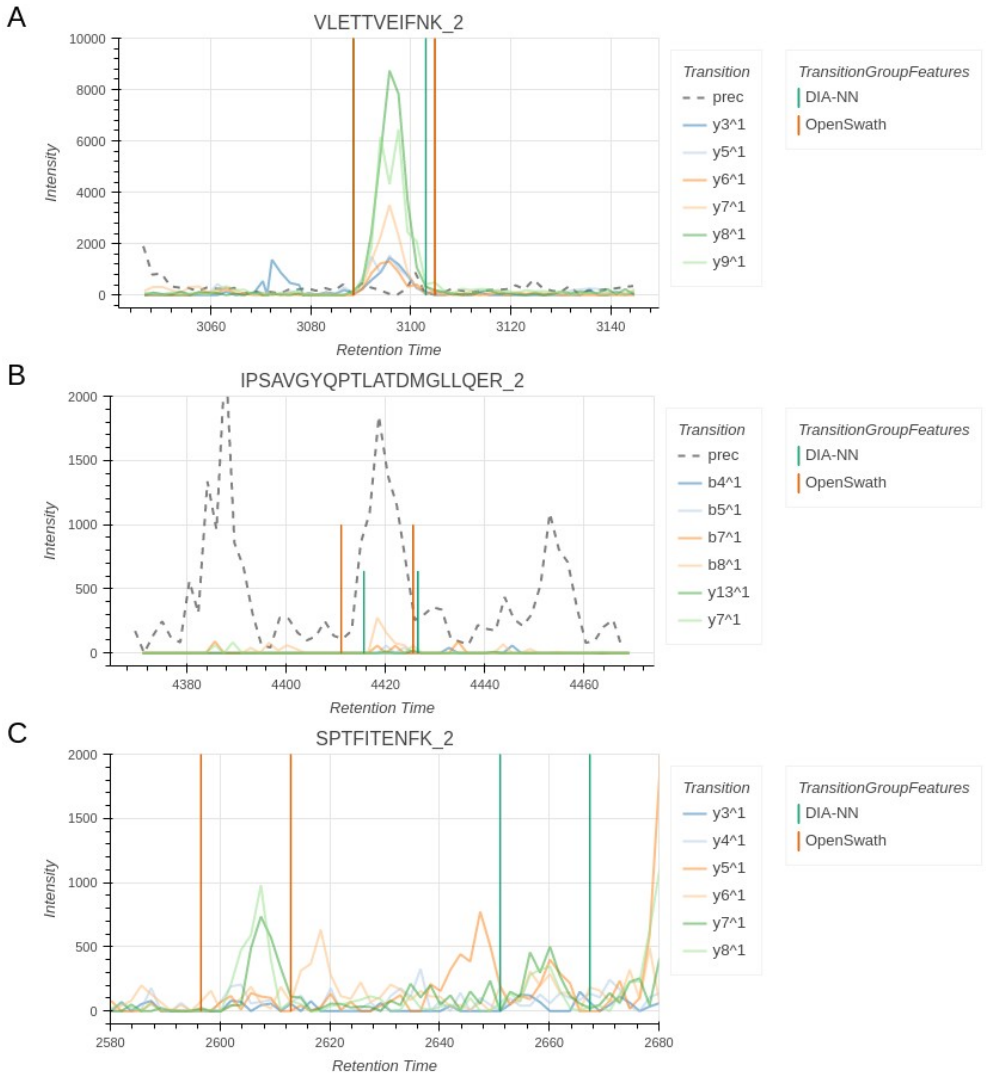

Figure 3: Automated feature detection tools are not always straightforward. Chromatograms of select precursors in a diaPASEF experiment with precursor boundaries shown for DIA-NN (green) and OpenSwath (orange). A) Example of a precursor where automated software tools are in agreement with manual inspection. B) Example of a precursor where both automated feature detection tools significantly identify a questionable precursor. Although this precursor has a strong precursor signal, the fragment ion signals are weak and provide questionable evidence of

the precursor's presence. C) Example where automated feature detection tools identify different features for the same precursor. Both features chosen by the software tools look reasonable for this noisy precursor. Manual visualization is helpful to determine which feature, if any, to accept as the true feature.

It is crucial not only to visually validate the detected features but also to compare different feature detection tools because this comparative analysis enables assessment of an algorithm's performance, data specific parameter optimization, tool selection, evaluate reproducibility, and address algorithmic biases (Fig. 3, Fig. 5, Sup. Fig. 2). Given the multiplexed nature of DIA data and the continuous development of novel tools, it becomes essential to compare and evaluate different methods. In our benchmarking dataset, *S. pyogenes* (PASS01508), we illustrate the comparison between three feature detection tools—OpenSwath, DIA-NN, and DreamDIA—at a 1% FDR at the peptide level (Sup. Fig 2). Across 16 runs, both deep-learning algorithms (DIA-NN, DreamDIA) identify roughly the same number of 11,000 peptides, while OpenSwath identifies around 9,800 peptides. Notably, OpenSwath and DreamDIA exhibit higher variance in the number of identifications across the 16 runs, whereas DIA-NN shows a much lower variance. Regarding the coefficient of variation (CV), their performance is fairly similar, although DIA-NN and DreamDIA have a greater number of outliers with a CV above 250. Additionally, the comparison reveals peptides uniquely identified by each software tool, emphasizing the importance of visual validation to explore intriguing case examples and understand the nuances of identification methods.

MassDash as a tool for Parameter Optimization

With the diverse application of DIA to many different contexts and instruments, the default settings for a given software tool may not always be appropriate for the given experiment. MassDash's plug-and-play functionality provides a conducive environment for the swift integration of different algorithms, streamlining the process of exploring and optimizing analytical approaches. This flexibility enables researchers to effortlessly implement and test various algorithms for diverse tasks, including feature extraction, peak picking, and other DIA-MS analysis methods of interest.

For example, the optimal extraction parameters across retention time, m/z, and ion mobility may be different for different experiments or even for different precursors in the same experiment. Although enlarging extraction windows leads to higher sensitivity, this comes at the cost of a decrease in selectivity. Since MassDash contains built-in peak pickers and can perform on-the-fly extraction, different extraction parameters can be tested and visualized in an interactive environment. For example, in OpenSwath, peak picking is performed by extracting and summing up signals across a specific area of ion mobility followed by peak picking on the resulting

chromatogram. A commonly used ion mobility extraction window for diaPASEF data is 0.06 1/k0 since this provides buffer room to account for differences between the library and feature ion mobilities. However, in this diaPASEF experiment, for precursor *NAFLESELDEKENLLESVQR_3* the extraction window of 0.06 appears too large since there is interference in the precursor signal leading to an overestimation in the peak boundaries (Fig. 4A). Visualizing the 2D heatmap, it is clear that there is an interfering precursor above (*NPSFSEEPWLEIK_2)* and that refining the ion mobility extraction window should improve the chromatogram signal-to-noise (Fig. 4B, 5C). After refining the ion mobility extraction window the transition chromatograms are cleaner allowing for a more accurate quantification estimation of peak boundaries (Fig. 4D).

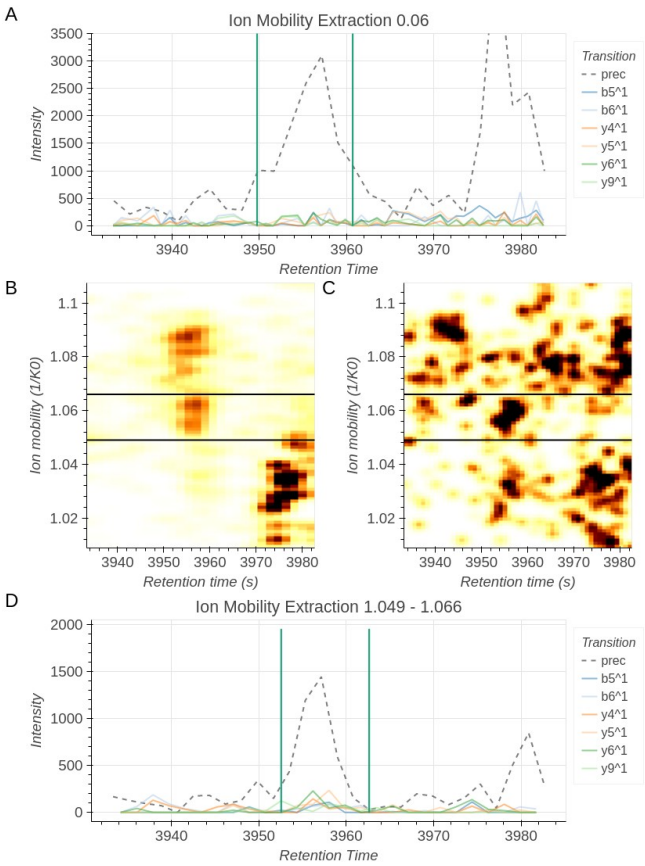

Figure 4: Multidimensional plotting can be used for extraction width optimization. A) Chromatogram of precursor *NAFLESELDEKENLLESVQR_3* with an ion mobility extraction width of 0.06, centered around the precursor (1.028 - 1.088). This is the extraction window commonly used for OpenSwath to account for variations in ion mobility. The fragment ion signals are very noisy and the precursor signal contains an interfering peak on the left side leading to an overestimation of the peak group width. B) 2D chromatogram (retention time vs ion mobility) with Gaussian blur and histogram equalization of the MS1 signal of precursor

NAFLESELDEKENLLESVQR_3. Through visualization, the extraction window (black box) was manually drawn to exclude the precursor above (*NPSFSEEPWLEIK_2*). C) 2D chromatogram with Gaussian blur and histogram equalization of the amalgamated MS2 transition signals. Ion mobility boundaries used for MS1 are in agreement with MS2 2D chromatogram. D) Chromatogram of precursor *NAFLESELDEKENLLESVQR_3* with an ion mobility extraction of 1.049 - 1.066. Boundaries were manually determined from figure B) and C). With the updated extraction window the precursor and fragment ion peaks are cleaner leading to easier detection of the peak group boundaries.

Additionally, MassDash's Python backend facilitates seamless integration with numerous Python packages enabling rapid method prototyping and testing. For example, MashDash supports the native OpenSwath peak picker meaning that peak picker parameter optimization can be done interactively and the new optimal parameters can be used in a subsequent OpenSwath analysis. One instance in which peak-picking parameters may need to be adjusted is in phosphoproteomic datasets. These datasets have a high prevalence of peptide isoforms which due to their similarity may partially coelute. This partial co-elution would require less smoothing to be performed by the peak picker to obtain accurate boundaries. For example, using the OpenSwath peak picker with high Savitzky–Golay smoothing, the peak picker may mistakenly group the two coeluting peptides into one peak (Fig. 5 orange line). Decreasing the smoothing parameter allows the peak picker to correctly distinguish between the two partially coeluting peaks (Fig. 5 green line).

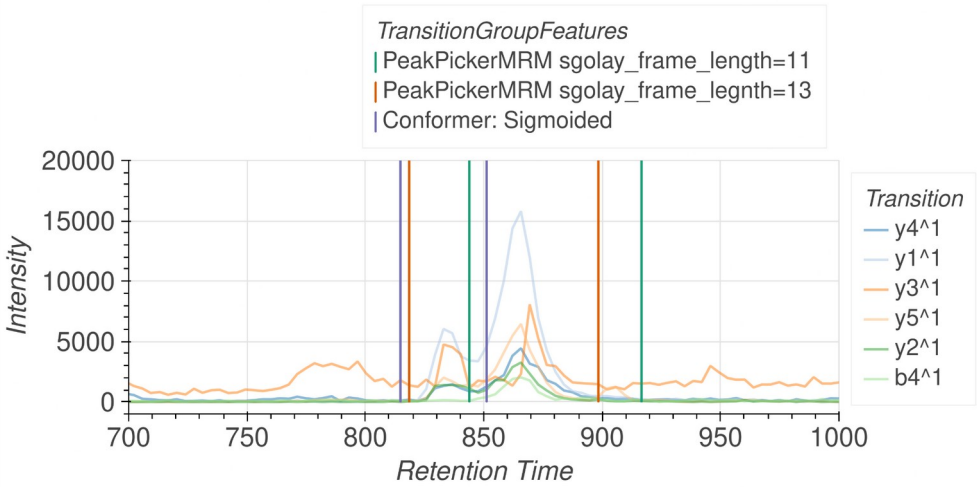

Figure 5: Algorithm Development With MassSeer. Chromatogram of phosphopeptide NKESPT(UniMod:21)KAIVR(UniMod:267) with a charge state of 3. In this chromatogram two peptides are closely eluting with one another Three different peak pickers are shown the MRMPeakPicker (OpenSwath default peak picker) with an sgolay_frame_length of 11, MRMPeakPicker with a sgolay_frame_length of 15 and the deep learning Conformer peak

picker. Each peak picker picks different peak boundaries indicating the need to optimize peak pickers for specific experimental contexts.

Novel algorithm development

MassDash's modularity and Python interface promote the rapid development of novel tools, including deep learning-based peak pickers. This empowers researchers to test and validate machine learning and deep learning algorithms effortlessly within a ready-to-use modular environment. Many AI frameworks, being in Python, further facilitate integration with MassDash, enhancing accessibility and streamlining the incorporation of cutting-edge methodologies. MassDash's adaptability extends to the utilization of saved trained models, providing a valuable tool for researchers to assess the performance of machine learning and deep learning algorithms on their specific datasets. This feature enhances the platform's utility in exploring cutting-edge approaches, advancing algorithm development, and contributing to the ongoing evolution of DIA-MS methodologies. To demonstrate this, we have built-in support for the Conformer deep learning peak picker[15]. Interestingly, when applying the Conformer peak picker to the phosphopeptide chromatogram above, the small peak is chosen as the peak of highest confidence in contrast to the OpenSwath peak picker (Figure 4 purple line). This might be because the Conformer peak picker also takes into account library retention time and relative ratios of fragment ions. MassDash's object-oriented codebase allows users to easily extend the application and implement and test their own peak picker.

Conclusions

In conclusion, we present MassDash, a novel modular software framework for mass spectrometry DIA visualization, which offers a streamlined, user-friendly interface for rapid quality control visualization, prototyping, and algorithm testing. Its Python library integrates seamlessly into scripting environments like Jupyter Notebooks, enhancing accessibility and adaptability. MassDash's capacity for on-the-fly data processing, efficient memory usage and, rapid algorithm development in Python distinguishes it from existing tools, providing researchers with a versatile platform for visualization and validation of DIA data. Overall, MassDash's practical features make it a valuable asset for researchers navigating the complexities of high-throughput mass spectrometry data analysis.

Data Availability

Initial implementation and testing was performed using the gold standard *S. pyogenes* dataset, available on PeptideAtlas (PASS01508), as well as a synthetic phosphorylation dataset available on PRIDE (PXD004573). Testing of raw targeted extraction for diaPASEF data was done using a two-proteome mixture dataset available on PRIDE (PXD017703). Ion mobility test files were created from the 50ng dilution set (PXD017703, 20190816_TIMS05_MA_FlMe_diaPASEF_25pc_50ng_A2_1_26.d)

Acknowledgments

We are thankful to Röst lab members for testing, validation and input throughout the development of the tool. We would also like to thank Mingxuan Gao for his input and help with incorporating DreamDIA into the MassDash. J.C.S. was supported by the European Research Area Network Personalized Medicine Cofund (PerProGlio). J.C. was supported by CIHR (#506078) and the Ontario Graduate Scholarship. M.A. was supported by the Chan Zuckerberg Initiative (#513040). I.H. was supported by Early Research Award (ERA) (#511828) and the University of Toronto Open Fellowship. L.X. was supported by NSERC (#506685).

Contributions

J.C.S., J.C., and H.L.R. designed the study. J.C.S. and J.C. designed and developed the software and M.A, L.X. and I.H. contributed to development. J.C.S., J.C., M.A, L.X., I.H. and H.L.R. contributed to writing and analysis. H.L.R. supervised the study.

Competing Interests

The authors declare no competing interests.

References

1. Gillet, L. C., Navarro, P., Tate, S., Röst, H., Selevsek, N., Reiter, L., Bonner, R., & Aebersold, R. (2012). Targeted Data Extraction of the MS/MS Spectra Generated by Data-independent Acquisition: A New Concept for Consistent and Accurate Proteome Analysis *. *Molecular & Cellular Proteomics*, *11*(6). https://doi.org/10.1074/mcp.O111.016717

2. Röst, H. L., Rosenberger, G., Navarro, P., Gillet, L., Miladinović, S. M., Schubert, O. T., Wolski, W., Collins, B. C., Malmström, J., Malmström, L., & Aebersold, R. (2014). OpenSWATH enables automated, targeted analysis of data-independent acquisition MS data. *Nature Biotechnology*, *32*(3), Article 3. https://doi.org/10.1038/nbt.2841

3. Bruderer, R., Bernhardt, O. M., Gandhi, T., Miladinović, S. M., Cheng, L.-Y., Messner, S., Ehrenberger, T., Zanotelli, V., Butscheid, Y., Escher, C., Vitek, O., Rinner, O., & Reiter, L. (2015). Extending the Limits of Quantitative Proteome Profiling with Data-Independent Acquisition and Application to Acetaminophen-Treated Three-Dimensional Liver Microtissues. *Molecular & Cellular Proteomics : MCP*, *14*(5), 1400–1410. https://doi.org/10.1074/mcp.M114.044305

4. Navarro, P., Kuharev, J., Gillet, L. C., Bernhardt, O. M., MacLean, B., Röst, H. L., Tate, S. A., Tsou, C.-C., Reiter, L., Distler, U., Rosenberger, G., Perez-Riverol, Y., Nesvizhskii, A. I., Aebersold, R., & Tenzer, S. (2016). A multicenter study benchmarks software tools for label-free proteome quantification. *Nature Biotechnology*, *34*(11), Article 11. https://doi.org/10.1038/nbt.3685

5. Collins, B. C., Hunter, C. L., Liu, Y., Schilling, B., Rosenberger, G., Bader, S. L., Chan, D. W., Gibson, B. W., Gingras, A.-C., Held, J. M., Hirayama-Kurogi, M., Hou, G., Krisp, C., Larsen, B., Lin, L., Liu, S., Molloy, M. P., Moritz, R. L., Ohtsuki, S., … Aebersold, R. (2017). Multi-laboratory assessment of reproducibility, qualitative and quantitative performance of SWATH-mass

spectrometry. *Nature Communications*, *8*(1), Article 1. https://doi.org/10.1038/s41467-017-00249-5

6. Li, K. W., Gonzalez-Lozano, M. A., Koopmans, F., & Smit, A. B. (2020). Recent Developments in Data Independent Acquisition (DIA) Mass Spectrometry: Application of Quantitative Analysis of the Brain Proteome. *Frontiers in Molecular Neuroscience*, *13*. https://www.frontiersin.org/articles/10.3389/fnmol.2020.564446

7. Li, J., Smith, L. S., & Zhu, H.-J. (2021). Data-independent acquisition (DIA): An emerging proteomics technology for analysis of drug-metabolizing enzymes and transporters. *Drug Discovery Today: Technologies*, *39*, 49–56. https://doi.org/10.1016/j.ddtec.2021.06.006

8. Krasny, L., & Huang, P. H. (2021). Data-independent acquisition mass spectrometry (DIA-MS) for proteomic applications in oncology. *Molecular Omics*, *17*(1), 29–42. https://doi.org/10.1039/D0MO00072H

9. Zhu, F.-Y., Chen, X., Song, Y.-C., Lam, L. P. Y., Tobimatsu, Y., Gao, B., Chen, M.-X., & Cao, F.-L. (2023). SWATH-MS-based proteogenomic analysis reveals the involvement of alternative splicing in poplar upon lead stress. *Genome Research*, *33*(3), 371–385. https://doi.org/10.1101/gr.277473.122

10. Teleman, J., Röst, H. L., Rosenberger, G., Schmitt, U., Malmström, L., Malmström, J., & Levander, F. (2015). DIANA—algorithmic improvements for analysis of data-independent acquisition MS data. *Bioinformatics*, *31*(4), 555–562. https://doi.org/10.1093/bioinformatics/btu686

11. Ting, Y. S., Egertson, J. D., Bollinger, J. G., Searle, B. C., Payne, S. H., Noble, W. S., & MacCoss, M. J. (2017). PECAN: Library Free Peptide Detection for Data-Independent Acquisition Tandem Mass Spectrometry Data. *Nature Methods*, *14*(9), 903–908. https://doi.org/10.1038/nmeth.4390

12. Searle, B. C., Pino, L. K., Egertson, J. D., Ting, Y. S., Lawrence, R. T., MacLean, B. X., Villén, J., & MacCoss, M. J. (2018). Chromatogram libraries improve peptide detection and quantification by

data independent acquisition mass spectrometry. *Nature Communications*, *9*(1), Article 1. https://doi.org/10.1038/s41467-018-07454-w

13. Demichev, V., Messner, C. B., Vernardis, S. I., Lilley, K. S., & Ralser, M. (2020). DIA-NN: Neural networks and interference correction enable deep proteome coverage in high throughput. *Nature Methods*, *17*(1), Article 1. https://doi.org/10.1038/s41592-019-0638-x
14. Sun, B., Smialowski, P., Aftab, W., Schmidt, A., Forne, I., Straub, T., & Imhof, A. (2023). Improving SWATH-MS analysis by deep-learning. *PROTEOMICS*, *23*(9), 2200179. https://doi.org/10.1002/pmic.202200179
15. Xu, L. L., & Röst, H. L. (2020). *Peak Detection On Data Independent Acquisition Mass Spectrometry Data With Semisupervised Convolutional Transformers* (arXiv:2010.13841). arXiv. https://doi.org/10.48550/arXiv.2010.13841
16. Reiter, L., Rinner, O., Picotti, P., Hüttenhain, R., Beck, M., Brusniak, M.-Y., Hengartner, M. O., & Aebersold, R. (2011). mProphet: Automated data processing and statistical validation for large-scale SRM experiments. *Nature Methods*, *8*(5), Article 5. https://doi.org/10.1038/nmeth.1584
17. Gao, M., Yang, W., Li, C., Chang, Y., Liu, Y., He, Q., Zhong, C.-Q., Shuai, J., Yu, R., & Han, J. (2021). Deep representation features from DreamDIAXMBD improve the analysis of data-independent acquisition proteomics. *Communications Biology*, *4*(1), Article 1. https://doi.org/10.1038/s42003-021-02726-6
18. Tsou, C.-C., Avtonomov, D., Larsen, B., Tucholska, M., Choi, H., Gingras, A.-C., & Nesvizhskii, A. I. (2015). DIA-Umpire: Comprehensive computational framework for data-independent acquisition proteomics. *Nature Methods*, *12*(3), Article 3. https://doi.org/10.1038/nmeth.3255
19. Kong, A. T., Leprevost, F. V., Avtonomov, D. M., Mellacheruvu, D., & Nesvizhskii, A. I. (2017). MSFragger: Ultrafast and comprehensive peptide identification in mass spectrometry–based proteomics. *Nature Methods*, *14*(5), Article 5. https://doi.org/10.1038/nmeth.4256

20. Meier, F., Brunner, A.-D., Frank, M., Ha, A., Bludau, I., Voytik, E., Kaspar-Schoenefeld, S., Lubeck, M., Raether, O., Bache, N., Aebersold, R., Collins, B. C., Röst, H. L., & Mann, M. (2020). diaPASEF: Parallel accumulation–serial fragmentation combined with data-independent acquisition. *Nature Methods*, *17*(12), Article 12. https://doi.org/10.1038/s41592-020-00998-0

21. Bekker-Jensen, D. B., Martínez-Val, A., Steigerwald, S., Rüther, P., Fort, K. L., Arrey, T. N., Harder, A., Makarov, A., & Olsen, J. V. (2020). A Compact Quadrupole-Orbitrap Mass Spectrometer with FAIMS Interface Improves Proteome Coverage in Short LC Gradients*. *Molecular & Cellular Proteomics*, *19*(4), 716–729. https://doi.org/10.1074/mcp.TIR119.001906

22. Reilly, L., Lara, E., Ramos, D., Li, Z., Pantazis, C. B., Stadler, J., Santiana, M., Roberts, J., Faghri, F., Hao, Y., Nalls, M. A., Narayan, P., Liu, Y., Singleton, A. B., Cookson, M. R., Ward, M. E., & Qi, Y. A. (2023). A fully automated FAIMS-DIA mass spectrometry-based proteomic pipeline. *Cell Reports Methods*, *3*(10), 100593. https://doi.org/10.1016/j.crmeth.2023.100593

23. Moseley, M. A., Hughes, C. J., Juvvadi, P. R., Soderblom, E. J., Lennon, S., Perkins, S. R., Thompson, J. W., Steinbach, W. J., Geromanos, S. J., Wildgoose, J., Langridge, J. I., Richardson, K., & Vissers, J. P. C. (2018). Scanning Quadrupole Data-Independent Acquisition, Part A: Qualitative and Quantitative Characterization. *Journal of Proteome Research*, *17*(2), 770–779. https://doi.org/10.1021/acs.jproteome.7b00464

24. Juvvadi, P. R., Moseley, M. A., Hughes, C. J., Soderblom, E. J., Lennon, S., Perkins, S. R., Thompson, J. W., Geromanos, S. J., Wildgoose, J., Richardson, K., Langridge, J. I., Vissers, J. P. C., & Steinbach, W. J. (2018). Scanning Quadrupole Data-Independent Acquisition, Part B: Application to the Analysis of the Calcineurin-Interacting Proteins during Treatment of Aspergillus fumigatus with Azole and Echinocandin Antifungal Drugs. *Journal of Proteome Research*, *17*(2), 780–793. https://doi.org/10.1021/acs.jproteome.7b00499

25. Distler, U., Łącki, M. K., Startek, M. P., Teschner, D., Brehmer, S., Decker, J., Schild, T., Krieger, J.,

Krohs, F., Raether, O., Hildebrandt, A., & Tenzer, S. (2023). *midiaPASEF maximizes information content in data-independent acquisition proteomics* (p. 2023.01.30.526204). bioRxiv. https://doi.org/10.1101/2023.01.30.526204

26. Skowronek, P., Krohs, F., Lubeck, M., Wallmann, G., Itang, E. C. M., Koval, P., Wahle, M., Thielert, M., Meier, F., Willems, S., Raether, O., & Mann, M. (2023). Synchro-PASEF Allows Precursor-Specific Fragment Ion Extraction and Interference Removal in Data-Independent Acquisition. *Molecular & Cellular Proteomics*, *22*(2), 100489. https://doi.org/10.1016/j.mcpro.2022.100489
27. Heil, L. R., Damoc, E., Arrey, T. N., Pashkova, A., Denisov, E., Petzoldt, J., Peterson, A. C., Hsu, C., Searle, B. C., Shulman, N., Riffle, M., Connolly, B., MacLean, B. X., Remes, P. M., Senko, M. W., Stewart, H. I., Hock, C., Makarov, A. A., Hermanson, D., … MacCoss, M. J. (2023). Evaluating the Performance of the Astral Mass Analyzer for Quantitative Proteomics Using Data-Independent Acquisition. *Journal of Proteome Research*, *22*(10), 3290–3300. https://doi.org/10.1021/acs.jproteome.3c00357
28. Minogue, C. E., Hebert, A. S., Rensvold, J. W., Westphall, M. S., Pagliarini, D. J., & Coon, J. J. (2015). Multiplexed Quantification for Data-Independent Acquisition. *Analytical Chemistry*, *87*(5), 2570–2575. https://doi.org/10.1021/ac503593d
29. Derks, J., Leduc, A., Wallmann, G., Huffman, R. G., Willetts, M., Khan, S., Specht, H., Ralser, M., Demichev, V., & Slavov, N. (2023). Increasing the throughput of sensitive proteomics by plexDIA. *Nature Biotechnology*, *41*(1), Article 1. https://doi.org/10.1038/s41587-022-01389-w
30. Sturm, M., & Kohlbacher, O. (2009). TOPPView: An Open-Source Viewer for Mass Spectrometry Data. *Journal of Proteome Research*, *8*(7), 3760–3763. https://doi.org/10.1021/pr900171m
31. MacLean, B., Tomazela, D. M., Shulman, N., Chambers, M., Finney, G. L., Frewen, B., Kern, R., Tabb, D. L., Liebler, D. C., & MacCoss, M. J. (2010). Skyline: An open source document editor for creating and analyzing targeted proteomics experiments. *Bioinformatics*, *26*(7), 966–968.

https://doi.org/10.1093/bioinformatics/btq054

32. Röst, H. L., Rosenberger, G., Aebersold, R., & Malmström, L. (2015). Efficient visualization of high-throughput targeted proteomics experiments: TAPIR. *Bioinformatics*, *31*(14), 2415–2417. https://doi.org/10.1093/bioinformatics/btv152

33. Voytik, E., Skowronek, Patricia, Zeng, Wen-Feng, Tanzer, Maria C., Brunner, Andreas-David, Thielert, Marvin, Strauss, Maximilian T, Willems, Sander, & Mann, Matthias. (n.d.). *AlphaViz: Visualization and validation of critical proteomics data directly at the raw data level | bioRxiv*. Retrieved November 14, 2023, from https://www.biorxiv.org/content/10.1101/2022.07.12.499676v1

34. Röst, H. L., Schmitt, U., Aebersold, R., & Malmström, L. (2014). pyOpenMS: A Python-based interface to the OpenMS mass-spectrometry algorithm library. *Proteomics*, *14*(1), 74–77. https://doi.org/10.1002/pmic.201300246

35. Bokeh Development Team. (2018). *Bokeh: Python library for interactive visualization*. https://bokeh.pydata.org/en/latest/

36. Inc, P. T. (2015). *Collaborative data science*. Plotly Technologies Inc. https://plot.ly

37. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, *32*. https://papers.nips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html

38. Gupta, S., Sing, J. C., & Röst, H. L. (2023). Achieving quantitative reproducibility in label-free multisite DIA experiments through multirun alignment. *Communications Biology*, *6*(1), Article 1. https://doi.org/10.1038/s42003-023-05437-2

39. McKinney, W. & others. (2010). Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference*, *445*, 51–56.

40. *Target-Decoy Search Strategy for Mass Spectrometry-Based Proteomics—PMC*. (n.d.). Retrieved January 9, 2024, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2922680/

41. Rosenberger, G., Bludau, I., Schmitt, U., Heusel, M., Hunter, C. L., Liu, Y., MacCoss, M. J., MacLean, B. X., Nesvizhskii, A. I., Pedrioli, P. G. A., Reiter, L., Röst, H. L., Tate, S., Ting, Y. S., Collins, B. C., & Aebersold, R. (2017). Statistical control of peptide and protein error rates in large-scale targeted data-independent acquisition analyses. *Nature Methods*, *14*(9), Article 9. https://doi.org/10.1038/nmeth.4398