Received: 22 September 2021 | Revised: 22 December 2021 | Accepted: 20 January 2022 | Accepted article online: 2 February 2022

DOI: 10.1002/pmic.202100103

Proteomics
Proteomics and Systems Biology

REVIEW

A practical guide to interpreting and generating bottom-up proteomics data visualizations

Julia Patricia Schessner | **Eugenia Voytik** | **Isabell Bludau**

Department of Proteomics and Signal Transduction, Max-Planck-Institute of Biochemistry, Planegg, Germany

Correspondence
Julia Patricia Schessner, Department of Proteomics and Signal Transduction, Max-Planck-Institut für Biochemie, Abt. Mann, E03, Am Klopferspitz 18, 82152 Planegg, DE, Germany.
Email: schessner@biochem.mpg.de

Julia Patricia Schessner and Eugenia Voytik contributed equally to this work.

Funding information
Swiss National Science Foundation Postdoc.Mobility fellowship, Grant/Award Number: P400PB_191046; Bayerisches Staatsministerium für Bildung und Kultus, Wissenschaft und Kunst, Grant/Award Number: Digimed Bayern; Max-Planck-Förderstiftung

Abstract

Mass-spectrometry based bottom-up proteomics is the main method to analyze proteomes comprehensively and the rapid evolution of instrumentation and data analysis has made the technology widely available. Data visualization is an integral part of the analysis process and it is crucial for the communication of results. This is a major challenge due to the immense complexity of MS data. In this review, we provide an overview of commonly used visualizations, starting with raw data of traditional and novel MS technologies, then basic peptide and protein level analyses, and finally visualization of highly complex datasets and networks. We specifically provide guidance on how to critically interpret and discuss the multitude of different proteomics data visualizations. Furthermore, we highlight Python-based libraries and other open science tools that can be applied for independent and transparent generation of customized visualizations. To further encourage programmatic data visualization, we provide the Python code used to generate all data figures in this review on GitHub (https://github.com/MannLabs/ProteomicsVisualization).

KEYWORDS
bottom-up proteomics, data visualization, open science, science communication

1 | INTRODUCTION

Mass spectrometry (MS)-based bottom-up proteomics allows comprehensive analysis of highly complex proteomes [1–6]. Thanks to recent technological advances that dramatically increased proteomic depth and throughput, MS technology is nowadays accessible to many non-expert labs either through core facilities or individual proteomics setups. Firstly, the field has witnessed a huge enhancement of instrumentation, exemplified by a new robust and high-throughput liquid chromatography (LC) system [7] and new types of mass spectrometers allowing peptide separation by ion mobility [8–13]. Secondly, these advances were accompanied by the development of high-throughput data acquisition techniques [14–19] and a burst of computational methods for proteomics data analysis [20–24]. Facilitated by increasingly powerful computational hardware and programming backends, computational proteomics has evolved into an independent, multidisciplinary field, but now presents a new barrier to scientists lacking expertise either in proteomics or bioinformatics.

Adequate data visualization is crucial to interpret data and communicate results of evermore complex experiments [25, 26]. A variety of data analysis tools have integrated visualization functions to address this need [27–30], but visualization is usually not among the highest priorities in the development of novel data analysis workflows and is

Abbreviations: BPI, base peak intensity; DDA, data dependent acquisition; DIA, data independent acquisition; DOI, digital object identifier; FDR, false discovery rate; LC, liquid chromatography; MS, mass spectrometry; PC, principal component; PCA, principal component analysis; PTM, post-translational modification; TIC, total ion chromatogram; XIC, extracted ion chromatogram

often an afterthought. Consequently, data assessment, interpretation and visualization often remain exclusive abilities of experts familiar with the data and capable of handling it programmatically. This drastically slows down method dissemination and knowledge transfer to a broader audience from different research fields. Due to this required expertise, communication with non-experts in proteomics is often suboptimal. While there are several reviews that either focus on standalone software tools [31, 32] or cover computational aspects of the visualization process by making an overview of available R libraries [33], they do not necessarily provide insight to non-experts in proteomics on why certain visualizations are important or how to interpret them.

In this review, we provide an overview of several common types of visualizations, focusing on their use and interpretation rather than the software. We also demonstrate how such visualization can be interactively created with Python, one of the most common programming languages in science that has a low threshold to learn and use. Following the main steps of proteomics data analysis, we first describe the visualization of raw data and peptide identification with a special focus on novel MS instrument types and data acquisition modes. Next, we cover the visualization of quantitative information on the level of proteins, peptides and post-translational modifications (PTMs). In light of the continuously increasing complexity of experimental designs, we also include strategies for visualizing multidimensional data and a primer on protein networks. For each visualization we describe its common use cases and relevance, what it shows, and what aspects of it are important for interpretation and reporting. In the final section, we describe how Python and community resources can be used to create and share customized data visualizations by utilizing both generic and specialized libraries. To make it easier for readers to adopt customized MS data visualization themselves, we provide fully documented Python code that was used to generate all data Figures presented in this review on GitHub: https://github.com/MannLabs/ProteomicsVisualization. With this review we want to enable researchers working on interdisciplinary projects to (1) critically assess proteomics data visualizations in publications, (2) discuss effectively with experts, and ultimately (3) turn their own data into visualizations that optimally communicate their results.

Statement of significance

We review data visualizations used to evaluate and communicate bottom-up proteomics data. Critical aspects are explicitly explained by presenting concrete use-cases of raw and processed proteomics data. As practical guidance, we highlight publicly available Python-based tools and provide our own codebase for data visualizations that are presented herein. This should help the interdisciplinary use of bottom-up proteomics by ensuring a common ground for data communication and by enabling independent data exploration and visualization.

2 | VISUALIZATION OF PROTEOMICS DATA

In brief, a standard MS-based bottom-up proteomics workflow can be described as follows (see fig. 1 in [6]). Proteins are enzymatically digested into short, MS-accessible peptides and separated using a LC setup that is directly coupled to a mass spectrometer (LC/MS setup). The MS then measures both intact peptide masses and the corresponding masses of peptide fragment ions that are generated on the fly, which is called tandem mass-spectrometry (LC-MS/MS setup). The resulting peptide and fragment ion spectra are then used to identify which peptides were present in the sample based on a reference proteome, commonly provided as species-specific protein FASTA file. With many strategies available, identified peptides are then quantified and their information is aggregated to the protein level by protein inference. Strategies for peptide and protein quantification vary from absolute quantification within samples to relative quantification across samples. A more detailed introduction to bottom-up proteomics is available elsewhere [34]. In table 1 we provide an overview of the analysis steps, visualizations and most important pitfalls/best practices covered in this review. Many of the recommendations we make apply beyond the proteomics field and many statistical aspects are beautifully explained in the "Points of significance" series in Nature Methods.

2.1 | Raw data visualization

At the heart of all proteomics projects is the raw data acquired by the MS [35] and unsatisfactory analysis results can often be traced back to low data quality. Evaluating the raw MS data quality is therefore a critical first step during data analysis, yet it is often neglected. Data quality is commonly assessed by visual exploration of the raw MS data, as it can reveal a variety of flaws of samples and instrumentation alike [31]. Alternately, various computational quality control methods are also available in the field and are extensively covered in literature [36]. In this section, we cover standard visualizations of raw MS data on precursor and fragment ion level and how to read them. For most of these visualizations either the MS vendors or the MS search software tools provide a graphical user interface. As one prominent option for visualizing data from public repositories we want to point out the PRIDE Inspector [37].

2.1.1 | Visualizations at the precursor level

Ion chromatograms. The first steps of data quality control should always include a performance assessment of the LC and the MS. This is commonly done by inspecting how many precursor ions reach the MS detector over time, visualized in the total ion chromatogram (TIC), showing the summed intensity of all detected precursor ions

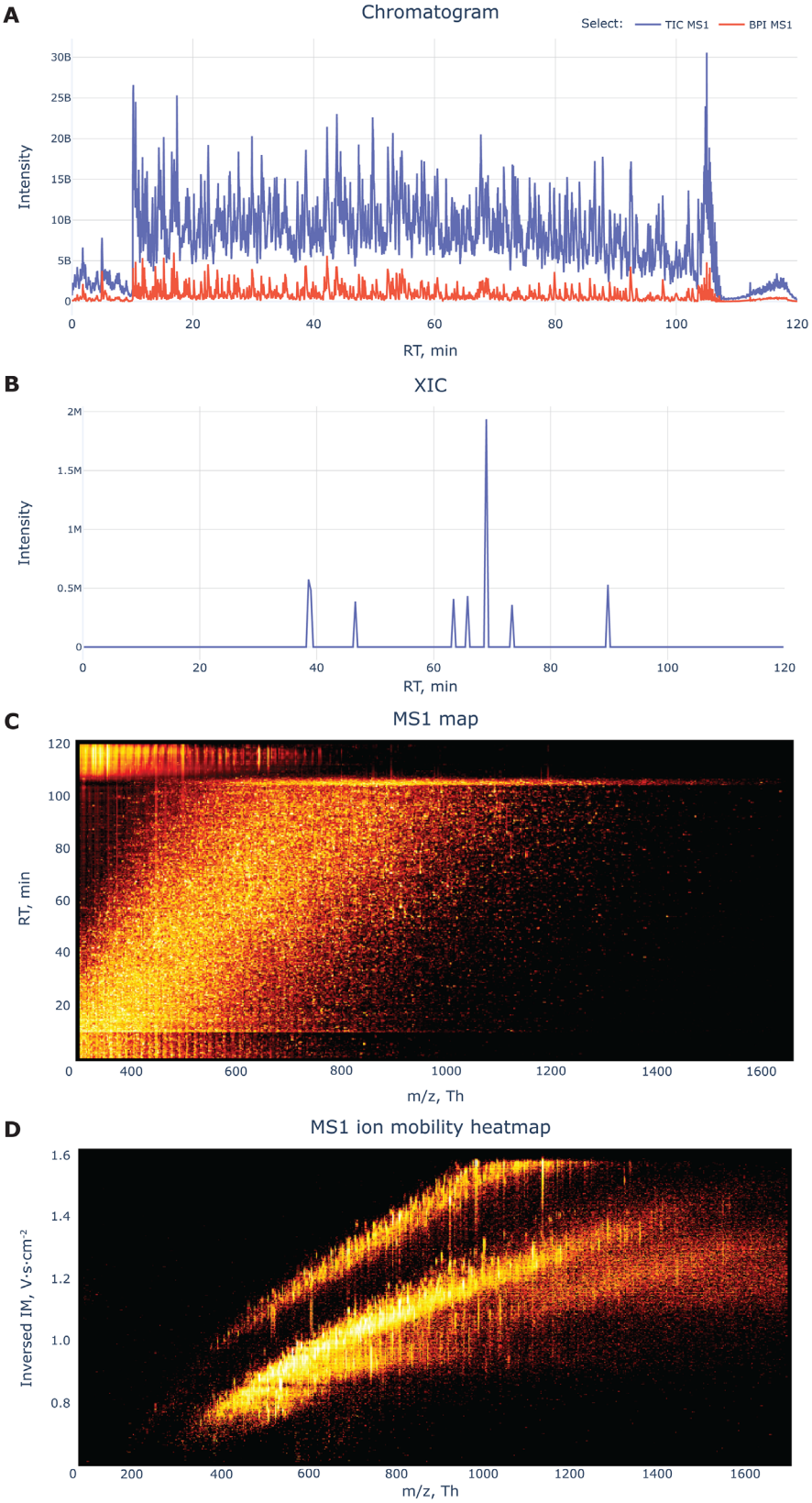

FIGURE 1 **Visualization of proteomics data at the precursor level.** (A-C) For these subFigures a dataset [62] from PXD012867 is used. (A) Total ion chromatogram (TIC) and base peak intensity (BPI) of MS1 data from 2 h nanoLC gradient measured on an Orbitrap based instrument. Low signal stretches in the first and last 10 min are due to loading time and LC flushing respectively. (B) Extracted ion chromatogram (XIC) for the analyte (m/z = 457.9978) with 5 ppm m/z tolerance. (C) Two-dimensional MS1 map showing the intensity of observed precursor masses across the whole retention time. (D) Two-dimensional MS1 ion mobility heatmap of precursor intensities acquired on an ion mobility separating time-of-flight instrument at a single time point, demonstrating a correlation of m/z and ion mobility (PXD017703, [107]).

TABLE 1 Overview of all visualizations presented in this review, including associated data, analysis steps, and pitfalls/recommendations

Data	Analysis step	Visualizations	Figure	Pitfalls/recommendations
MS1 raw data	Inspection of MS1 ion chromatograms to identify instrumentation and loading issues	Total ion chromatogram (TIC), Base peak intensity (BPI)	1A	Compare to a high-quality reference chromatogram matched by instrument, gradient, and sample complexity.
	Analysis of individual elution profiles	Extracted ion chromatogram (XIC)	1B	Mass range is critical: wide enough for mass errors, tight enough for specific selection.
	Inspection of precursor maps to identify instrumentation issues	Two-dimensional precursor maps	1C, 1D	Compare to a high-quality reference map. Different dimensions can be displayed.
MS2 raw data	Inspection of DDA peptide fragmentation	(mirrored) MS2 spectra and sequence fragmentation	2A, 2B	Number of fragments is crucial.
	Inspection of DIA peptide fragment groups	Two-dimensional or Three-dimensional elution profiles	2C, 2D	Elution peak shape should be highly correlated across fragments.
Peptide/PTM data	Map identified peptides to protein sequence	Non-overlapping traces	3A	Missed cleavages and repeated fragmentation are apparent.
	Map differential sequence coverage and external sequence features/PTMs	Overlapping traces per condition + external traces	3B	Missed cleavages are hidden in favor of differential coverage.
	Map PTM positions and quantities to sequences	Lollipop plot	3C	Different quantitative measures can be shown on y-axis.
Protein intensities	Dynamic range and normalization	Intensity histogram(s)	4A	Replicates should have similar shape.
	Proteome coverage	Protein rank plot	4B	Lower tail reveals depth limitation.
	Proteome correlation and reproducibility	Pairwise correlation plots and sample correlation heatmaps	4C, 4D	Use for small and high numbers of samples respectively.
Two-condition comparisons	Differential expression analysis by two-tailed tests	Volcano plots with square cutoffs/non-linear volcano lines	4E, 4F	Multiple hypothesis correction is mandatory. FDR and power (square cutoff) or s0 (non-linear cutoff) need to be reported.
	Enrichment analysis (e.g., by Fisher's exact test)	Variable visualization depending on experiment complexity	4G	The *p*-value is the most important parameter to display if fewer visual channels are available.
Multidimensional experiments	Dimensionality reduction to display complex datasets	Two-dimensional projection of proteins	5A, 5D, 5E	Algorithm determines topology (PCA/UMAP/tSNE).
	Reproducibility by PCA	PCA loadings plot	5B	Replicates should cluster.
	Variability contribution in PCA	Bar chart with all PCs	5C	Main discriminators of samples can be identified.
	Cluster analysis to group proteins and/or samples	Heatmap with marginal dendrograms	5F	Distance measure and clustering algorithm are key parameters, cutoffs are largely arbitrary.
	Display all features for a subset/summary of the data	Profile plot/parallel coordinates/radar plot	5G-I	Selection depends on data types and visibility of the key result.
Protein networks	Display protein distances	Weighted edge network	6A	Avoid hairballs, by parsimonious selection of nodes and edges, use a deterministic layouting algorithm.
	Display hierarchical groups	Hierarchical network	6B	Depends on underlying grouping.
	Display biological processes	Semantic network	6C	Indicate source for relationships.

against the retention time (blue line in Figure 1A). Problems that can be revealed inspecting the TIC are poor peak separation (very broad peaks), unstable spray or MS failure (intensity drops) and mistakes in sample preparation (low intensity, few peaks, unexpected overall shape) [38, 39]. Another major issue is saturation of the whole LC-MS system, for example, by overloading or contamination. This can be revealed by the base peak intensity (BPI) plot, which shows the intensity of the most abundant ions detected over time (red line in Figure 1A). If the system is saturated one can see plateaus in the BPI trace. It is generally advisable to have a reference TIC and BPI plot for the sample type and instrument setup used to be able to detect anomalies.

It can further be important to follow up on individual detected ions or groups of ions, to evaluate, for example, the spread of contaminants, the peak shape of quality control ions or the quality of identified peptide features. To this end, extracted ion chromatograms (XICs) are commonly used (Figure 1B). The desired mass and charge range is extracted from the raw data and its intensity is plotted against the retention time. In doing so it is important to set adequate boundaries to the mass range (m/z tolerance), accounting for mass errors and coeluting ions.

Precursor maps. To get an overview of the whole range of precursor masses detected along the retention time, a two-dimensional MS1 map can be used [40, 41]. It shows the intensity (color) of observed precursor masses (x-axis) across the chromatographic retention time (y-axis) as a heatmap (Figure 1C). Same as for the TIC, it is advisable to have a reference image for this to be able to see anomalies, as they could again hint at technical issues with the instrument.

Recent developments in MS instrumentation introduced ion mobility as an additional separation dimension [9, 11, 13], which should be evaluated in a similar way as the m/z dimension. Akin to the two-dimensional MS1 map, precursor signal intensities can be visualized in the ion mobility dimension against the m/z dimension (Figure 1D). This heatmap would be even more informative if it showed the intensity across all three dimensions (retention time, ion mobility and m/z). While this is in principle possible, the resulting visualizations are hard to interpret intuitively and improving them is one of the remaining challenges in proteomics data visualization [42].

2.1.2 | Visualizations at the fragment level

The first principal step of aggregating raw MS spectra into proteomic data is the identification of analyzed peptide sequences. The two required elements for sequence identification are the measured peptide fragment (MS2) spectra and the sequence search space, both of which depend on the acquisition mode and to a lesser extent the quantification strategy used [43]. We cover label-free data-dependent acquisition (DDA) and data-independent acquisition (DIA) here.

DDA. In the classical DDA approach the MS instrument isolates and fragments individual selected peptide ions from the precursor scan (MS1), most commonly the top-N most intense ones. The spectra are then searched against a sequence database that contains masses, sometimes also intensities, of peptide fragments from in silico protein digestion and fragmentation [44–46].

It can be important to manually evaluate the MS2 spectra and the identifications based on them, particularly when follow-up experiments hinge on a single or few proteins or even peptides. To do so, one can look at the individual MS2 spectra, highlighting the N-terminal and C-terminal fragment ions of the single selected precursor (Figure 2A). Underneath the spectrum itself, the sequence of the identified peptide and the position of identified N/C terminal fragment ions are indicated. Depending on the exact fragmentation method used, the peptide bond breaks at different positions, yielding different pairs of ions, most commonly b/y ions. Issues that can become apparent here are co-fragmentation of several peptides (many more fragments visible) or other isotopes of the same peptide (isotopic clusters for fragments), or poor fragmentation (very few ions and intense precursor peak). To check the quality of the peptide-spectrum-match against the library, mirrored spectra are commonly used (Figure 2B). Here the theoretical fragment masses are shown on a mirrored y-axis, which makes it immediately apparent which fragments are missing or should correctly be identified in the measured spectrum.

DIA. In DIA mode, instead of isolating a single precursor mass, mass ranges containing multiple precursors are isolated and fragmented for every MS1 scan, covering more precursors, but yielding more complex MS2 spectra. For a general introduction to DIA we suggest this review [49].

Due to the increased complexity, the simple MS2 spectrum visualizations lose most of their relevance and a spectral library containing only masses and intensities is no longer sufficient for identification. DIA libraries therefore additionally contain the retention time and if applicable the ion mobility of the precursor ions to narrow the search space at each time point [20, 48–50]. On top of the fragment masses, the exact coelution of fragments and their precursor is now crucial for scoring candidate identifications. To assess the quality of DIA identification, it is therefore most common to look at the elution profiles of all fragments associated with a specific precursor. Ideally, they should form a single sharp peak together with the precursor (Figure 2C). Indicators of peak misassignment would be peak shifts or blending additional peaks of individual fragments. Here, measuring ion mobility can lead to higher confidence, as fragments should correlate along this dimension as well. Both dimensions together can be visualized in heatmaps for the precursor and all its fragments in retention time and ion mobility space, colored by intensity (Figure 2D).

Additional complexity. Independent of the acquisition mode MS spectra can be complicated by peptide modifications, but the same visual techniques apply. Modifications can be either biologically generated PTMs (e.g., phosphorylation) [51, 52], artifacts introduced during sample preparation (e.g., oxidation) [53] or sample labelling techniques (e.g., TMT [54] or EASITag [55]). Depending on the exact type, modifications lead to additional peaks for neutral losses or reporter ion series in MS2 spectra, or even require an additional level of fragmentation (MS3) to acquire additional fragments. To interpret these complex spectra more specialized background knowledge that goes beyond the scope of this review is required.

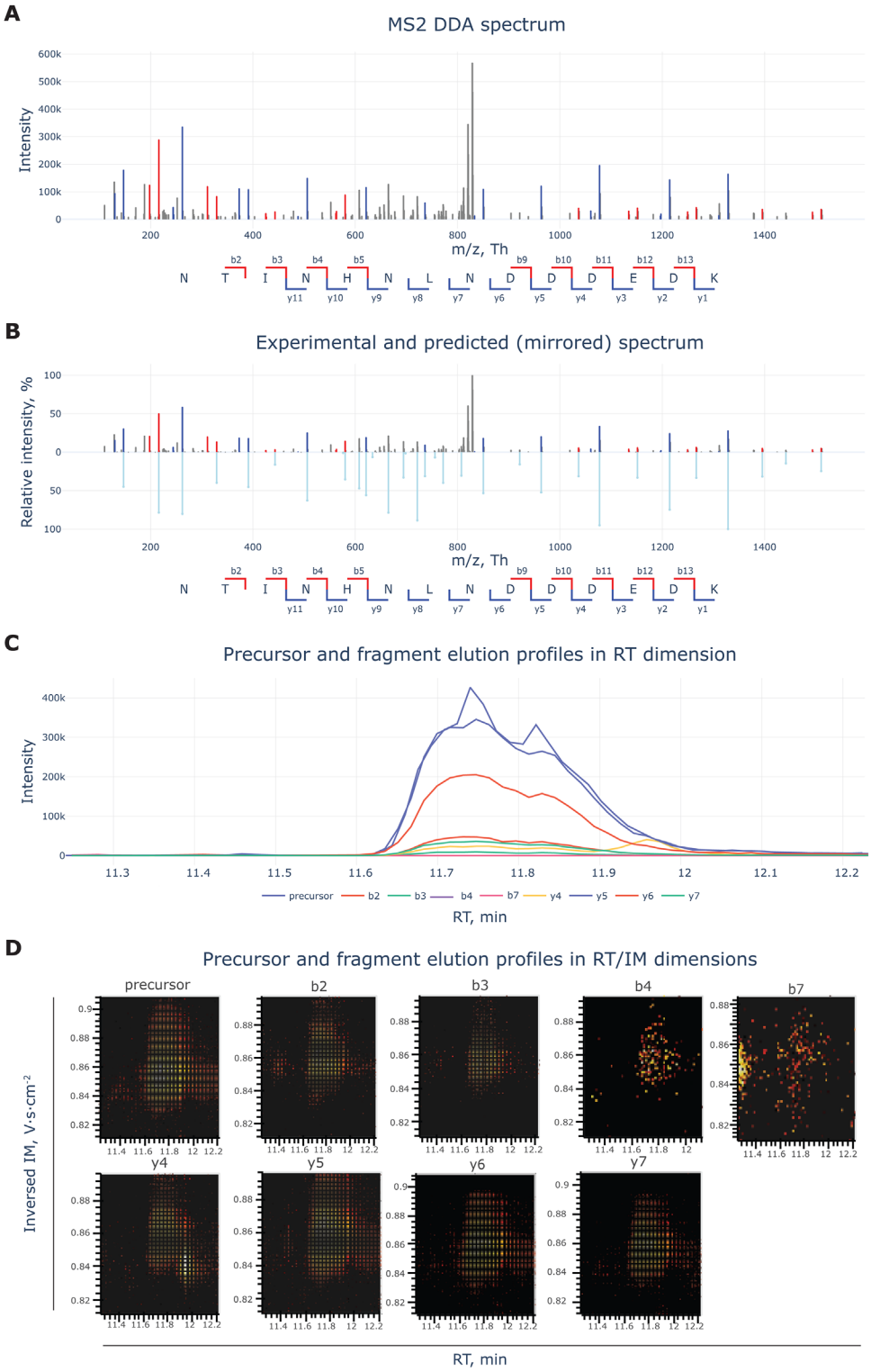

FIGURE 2 **Visualizations of proteomics data at the fragment ion level.** (A) Peptide MS2 spectrum generated by data dependent acquisition (PXD012867, [62]). The peptide sequence is annotated with the identified b- and y-ions. (B) Mirrored MS2 spectrum showing the experimental (top) and predicted (bottom) spectra for the same peptide as in A, confirming the correct identification (PXD012867, [62]). (C-D) Coelution of a peptide precursor and its fragment ions acquired on an ion-mobility separating time-of-flight instrument (PXD017703, [107]). (C) Extracted ion chromatograms in the elution time window of precursor and fragments nicely overlap. (D) Heatmaps of ion intensities in ion-mobility and retention time dimensions provide additional information on coelution in the ion-mobility dimension.

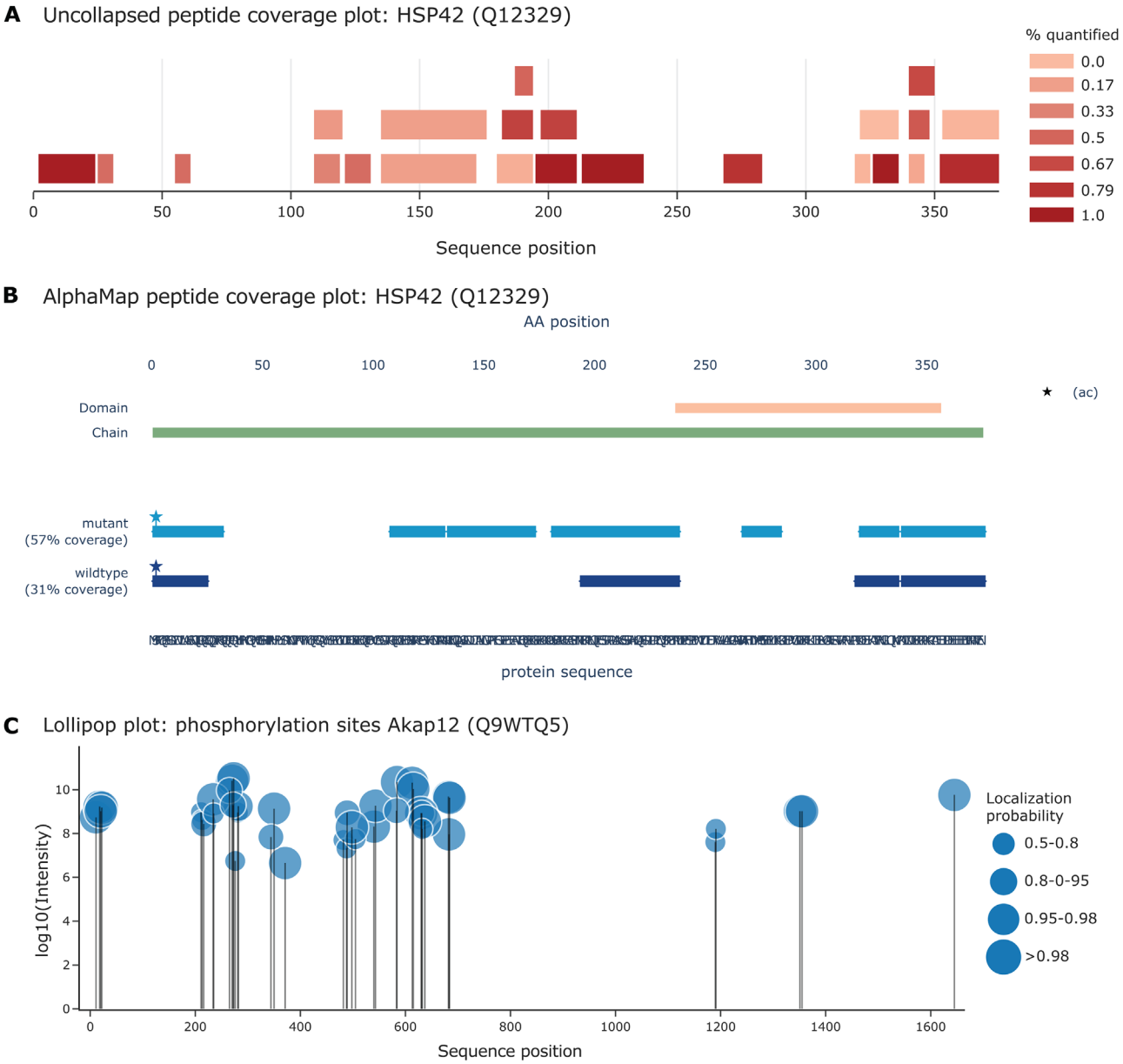

FIGURE 3 **Peptide visualization.** (A) Figure displaying peptide coverage along the protein sequence, overlap between peptides and identification frequency (color scale) (PXD012867, [62]). (B) Figure displaying differential peptide coverage across sets of samples with overlapping peptides collapsed into a single trace, PTMs (here only n-terminal acetylation) and external features (PXD012867, [62]). Generated using [58]. (C) Lollipop plot displaying phosphosites, their intensity and localization probability (bubble size) (PXD010697, [77]).

2.2 | Peptide and PTM visualization

When moving from raw data to aggregated peptide and protein quantifications, it is important to point out again that all bottom-up proteomics data is based on the identification of peptides rather than intact proteins. Therefore, assessing the coverage of protein sequences with identified peptides provides essential information. Sequence coverage can for example be assessed using a Figure in the style of the PeptideAtlas [56] (Figure 3A). Here, all unmodified peptides are displayed in a non-overlapping way along the protein sequence and are colored by their identification frequency across samples. This representation is well suited for assessing the reproducibility of peptide identification and to evaluate peptide overlaps caused by missed peptide cleavages. To evaluate differential sequence coverage between samples, overlapping peptides should be collapsed to a single line per sample to avoid clutter (Figure 3B).

If PTMs are measured, their position, intensity and localization probability can be visualized per modification site. If only the position needs to be visualized in the context of identified peptides, they can simply be added to these peptide views (start mark in Figure 3B). If a PTM's intensity and/or site probability are of interest a lollipop plot can be used (Figure 3C). These can for example be found on Phospho-

SitePlus [57]. Here, the size of the markers reflects the site probability and their vertical position reflects the intensity. For any of these visualizations it can be very informative to include additional annotation traces, for example, showing tryptic cleavage sites, and protein domains. This is for example possible using AlphaMap [58], which was also used here to create Figure 3B. With these visualizations in hand, various aspects of observed peptide and PTM signals associated with a protein of interest can be visualized and easily compared with data available in external databases. In doing so it is important to keep in mind that not all peptides are unique for just a single protein [59, 60].

2.3 | Protein quantity visualization and basic analysis

Aggregating peptide quantifications into protein quantifications is anything but a trivial task and highly depends on the inference strategy and quantification method used [61]. Agnostic to the quantification method, the assignment of peptides to proteins is not always uniquely possible, and therefore proteomics studies often talk about protein groups [59, 60]. These usually consist of any number of proteins that could be contained in the sample based on a set of shared non-unique, or "razor", peptides identified. Most protein groups consist of genetically closely related proteins, like isoforms or paralogs. From here on out we will focus on the analysis of protein groups independent of the inference and quantification method used, but want to point out that each quantification method comes with individual parameters and visualizations used for quality control. All following visualizations can in principle also be applied on the peptide level, but are mostly used on the protein level. We will start with the evaluation of single condition samples and simple two-condition comparisons by the example of a knock-out versus wildtype experiment [62] and then move on to more complex experimental designs and protein networks in the following sections.

Range and reproducibility. Once protein groups are quantified the first thing to look at is the distribution of their intensities. This is frequently done using log-intensity histograms (Figure 4A) or boxplots. These can indicate if certain samples have different intensity distributions, which might necessitate normalization, or a significantly reduced depth. They can further be used to assess the distribution of certain protein categories relevant to the downstream analysis, like imputed values or reverse database hits as in Figure 4A.

The dynamic range of a dataset is another important parameter as the measurement of low abundant proteins is a major limitation in untargeted bottom-up proteomics. To display it, a protein rank plot can be used (Figure 4B). Depending on the quantification method and the downstream processing, the y-axis can represent either raw intensity units or estimates of absolute protein quantities (e.g., iBAQ [63], proteomic ruler [64]). In full proteome studies, the highest abundant proteins typically include cytoskeletal and ribosomal proteins and, depending on the proteomic depth, the lower tail includes, for example, signaling proteins and transcription factors.

Next, it is important to assess the reproducibility of replicate samples and the general similarity of samples to compare. For a limited number of samples, multi-scatter plots displaying all pairwise log-intensity distributions and their correlations can be used (Figure 4C). For larger numbers of samples, where a visualization of all sample pairs is no longer feasible, reproducibility can be assessed by a heatmap of correlation values (Figure 4D), or alternatively by principal component analysis (see next chapter).

Volcano plots. The minimal comparative experiment spans two conditions with n biological replicates each. The standard analysis workflow for this is to perform multiple hypothesis corrected two-sample (Student's T-) tests [65, 66]. The multiple hypothesis correction is essential in any proteomics experiment, as *p*-values can be seemingly significant (i.e., very small) just by chance when making thousands of comparisons from the same dataset at once. Plotting the negative log10 of the (corrected) *p*-value against the difference in log-space for each protein leads to the classical volcano plot (Figure 4E-F).

The thresholds for calling a protein differentially abundant can be determined by one of two methods: (1) square cutoffs for *p*-value and fold-change (Figure 4E), or (2) non-linear volcano lines (Figure 4F). (1) For square cutoffs, the horizontal threshold is selected based on a desired multiple hypothesis testing corrected *p*-value (or FDR). The vertical fold-change cutoff is set with regard to the experimental power, which is the probability of detecting an effect of a certain size, given it actually exists. When using square cutoffs, the power should always be indicated as in Figure 4E, regardless of whether a fixed power is used to calculate the fold-change cutoff or the other way around [67]. (2) For nonlinear volcano lines, an s0 parameter is set instead of a specific fold-change cutoff [68]. The s0 parameter is added as a constant to all standard deviations used in the *t*-tests and can roughly be interpreted as the assumed systematic error of the measurements, thereby setting a lower bound on the fold-change as a function of the measured standard deviation.

In both methods the boundaries on the fold-change ensure that the biological variability exceeds the numerical variability introduced by measurement noise or imperfect normalization. Both methods are valid if applied correctly, but yield slightly different hitlists and are both highly dependent on the arbitrarily selected parameters. It should also be kept in mind that either method still has a false discovery rate and protein groups can be on either side of the boundaries by mistake. The boundaries rather serve the purpose of generating a statistically sound list for further downstream analysis. Importantly, multiple hypothesis correction always has to be performed and documented. Usually this is done either by Benjamini-Hochberg correction or by performing a permutation test. For square cutoffs the y-axis usually shows the corrected *p*-value (not done here to ease comparison).

Enrichment analysis. One common analysis to do downstream of a volcano analysis is to look at overrepresentation of biologically relevant groups of proteins (e.g., biological pathways of cellular compartments) in the hitlist compared to the overall proteome (methods reviewed in [69]). This is usually done by a Fisher's exact test [70] or gene set enrichment analysis (GSEA, [71]) based on systematic annota-

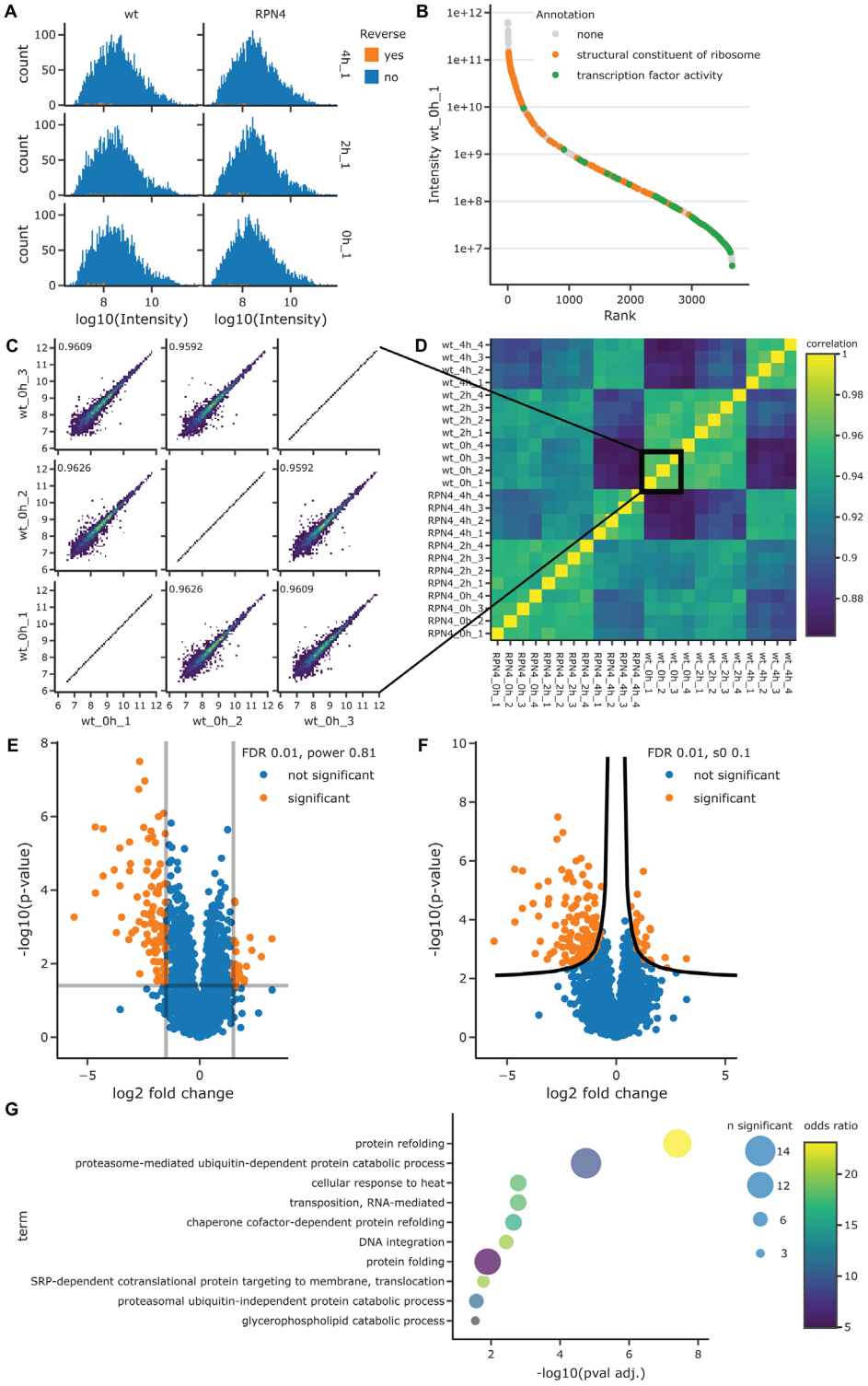

FIGURE 4 **Dataset properties and two-condition comparisons.** The data displayed in this Figure is taken from [62], where the principal comparison was drawn between wildtype and ΔRPN4 budding yeast cells (PXD012867). (A) Intensity histograms showing the distribution and number of protein groups are used to assess sample comparability. Hits from the reverse decoy database are annotated. (B) Protein rank plot from highest to lowest abundant proteins, illustrating the dynamic range. (C) Pairwise correlation plots demonstrate the biological and technical reproducibility. (D) Sample correlation matrix that is suitable to higher sample numbers than the pairwise correlation plot. It additionally illustrates sample grouping. (E, F) Volcano plots showing results of comparisons between two conditions, here between wildtype and ΔRPN4 samples. Multiple hypothesis testing was done by permutation and the FDR was set to 0.01. (E) Square significance cutoffs with minimal log2 fold change set to 1.5, which has a statistical power of 0.81. (F) Nonlinear volcano lines based on s0 = 0.1 adjusted *p*-value. (G) Enrichment analysis by Fisher's exact test for significant proteins from F. FDR = 5% after Benjamini-Hochberg correction. For all significant terms the corrected *p*-value, group size and the enrichment factor are displayed.

tions available, for example, through gene ontology [72, 73]. Often this is done using online tools that use the whole theoretical proteome as background. However, bottom-up proteomics is not able to quantify all proteins and unidentified proteins should not be included in the background for an enrichment analysis [74]. Thus, only tools that can consider the specific background should be used (e.g., String [75] or Panther [76]). The three main values resulting from an enrichment analysis per candidate group are enrichment factor, group size and multiple hypothesis testing corrected *p*-value, which can be visualized together (Figure 4G). From this one could now draw biologically relevant conclusions, linking the prior difference between the compared samples to enriched sets of protein groups. If differential enrichment in several samples is displayed, the x-axis can be used to display the different samples and the size can be switched from group size to *p*-value. Perseus is a common tool to generate many of the aforementioned visualizations and to run most underlying analyses, including the enrichment analysis [29]. However, given the output of the statistical analysis almost any comprehensive visualization tool can create these Figures.

2.4 | Multi-conditional and multidimensional experimental designs

With increasing throughput, thanks to improvements in MS instrumentation, more complex experimental designs became practical. Common multi-conditional designs include time course experiments [77] and profiling experiments across subcellular compartments [78] or protein complex fractionation [79]. Two- and multi-conditional designs can further be combined into multidimensional experiments with each other (e.g., measuring subcellular profiles over time [80] or in different genetic backgrounds [78]) and with additional variables (e.g., demographic parameters in clinical sample cohorts [81]). In this section we use a comparative spatial proteomics dataset [78] for demonstration purposes.

Dimensionality reduction. While the full scope of a two-condition experiment can easily be displayed in two-dimensional, higher dimensional experiments require dimensionality reduction for visualization. Just selecting two dimensions can be useful if a direct comparison is needed, but this will always disregard biological variability added by other dimensions. This is problematic because it can mask correlated or orthogonal effects.

One universal tool to incorporate these effects into dimensionality reduction is PCA [82]: The data is usually scaled and log-transformed and then linearly transformed onto a new coordinate system, such that the first component describes the largest fraction of the overall data variability and successive components decreasingly less. This effectively aggregates a large fraction of the data variability into fewer dimensions. This serves three purposes: First, any number of dimensions can be reduced to the main PCs to visualize all proteins and their annotation groups in two-dimensional (Figure 5A). Second, the contribution of each original dimension to the PCs (loading plot) serves as quality control for sample grouping (Figure 5B), where tight clustering of replicates should be apparent. Third, the variability contributed by each PC can inform on the independence of the acquired dimensions (Figure 5C). If many PCs have a similar contribution to the overall variability, this indicates independent underlying variables. In contrast, a single high variability PC often indicates that several of the underlying variables are at least partially dependent.

Other dimensionality reduction algorithms are tSNE [83] (Figure 5D) and UMAP [84] (Figure 5E). The major difference between PCA and tSNE/UMAP is that the latter performs non-linear transformations, whereby distances between individual proteins become incomparable. Their advantage is that they usually achieve visually more obvious separation of protein clusters in return and can provide performance benefits for two-dimensional clustering algorithms. In principle, these techniques can also be applied to a [sample x protein] rather than a [protein x sample] matrix to look at the data from a different perspective.

Heatmaps. A common visualization across different "omics" technologies are heatmaps with marginal dendrograms (Figure 5F). They can be used to understand the relations between samples and proteins alike. During the early stages of the analysis process heatmaps are often used similar to the PCA loadings plot to evaluate sample similarity. However, in contrast to the PCA plot they are based directly on the distance between the untransformed protein quantifications in each sample. Based on these distances a dendrogram is built, where branches of similar samples are grouped together. Additionally, it shows which groups of proteins follow a similar abundance pattern across samples, by building a vertical dendrogram across proteins in a similar fashion. The latter is particularly useful when it comes to a later stage of the analysis when proteins with specific behaviors of interest need to be grouped in order to form hypotheses about the underlying biology. Critical factors in creating and interpreting these heatmaps are the distance metric and clustering methods applied [85] to either axis and the normalization method that unifies the color scale across proteins. The distance is usually either euclidean distance or Pearson correlation. For normalization across samples z-scoring if often used.

Visualizing individual dimensions. The methods described above are most useful to display proteomic data across all measured dimensions. To show single dimensions (e.g., time course) or to combine proteomic data with other data types, different visualizations are better suited. The simplest way to display individual experimental dimensions is a line plot (Figure 5G), which works with continuous and categorical dimensions alike. Since showing the full proteomic scope would lead to clutter, we recommend either showing a relevant subset of proteins with thin lines, indicating density by opacity, or alternatively showing summary statistics. For some applications a radar plot might be preferred over a linear axis to ease interpretation (Figure 5I). Suitable applications include time course experiments along circadian cycles or biological slices of a bigger whole, for example, different organ tissues.

Mixing data types. If other data types (e.g., clinical parameters, additional "omics" data, quality parameters) are integrated with proteomic data, it is likely that none of the visualizations above can be applied. In that case one can turn to dimension plots having either parallel coordinates or categories. These have multiple parallel axes that can each represent a different data type with individual ranges.

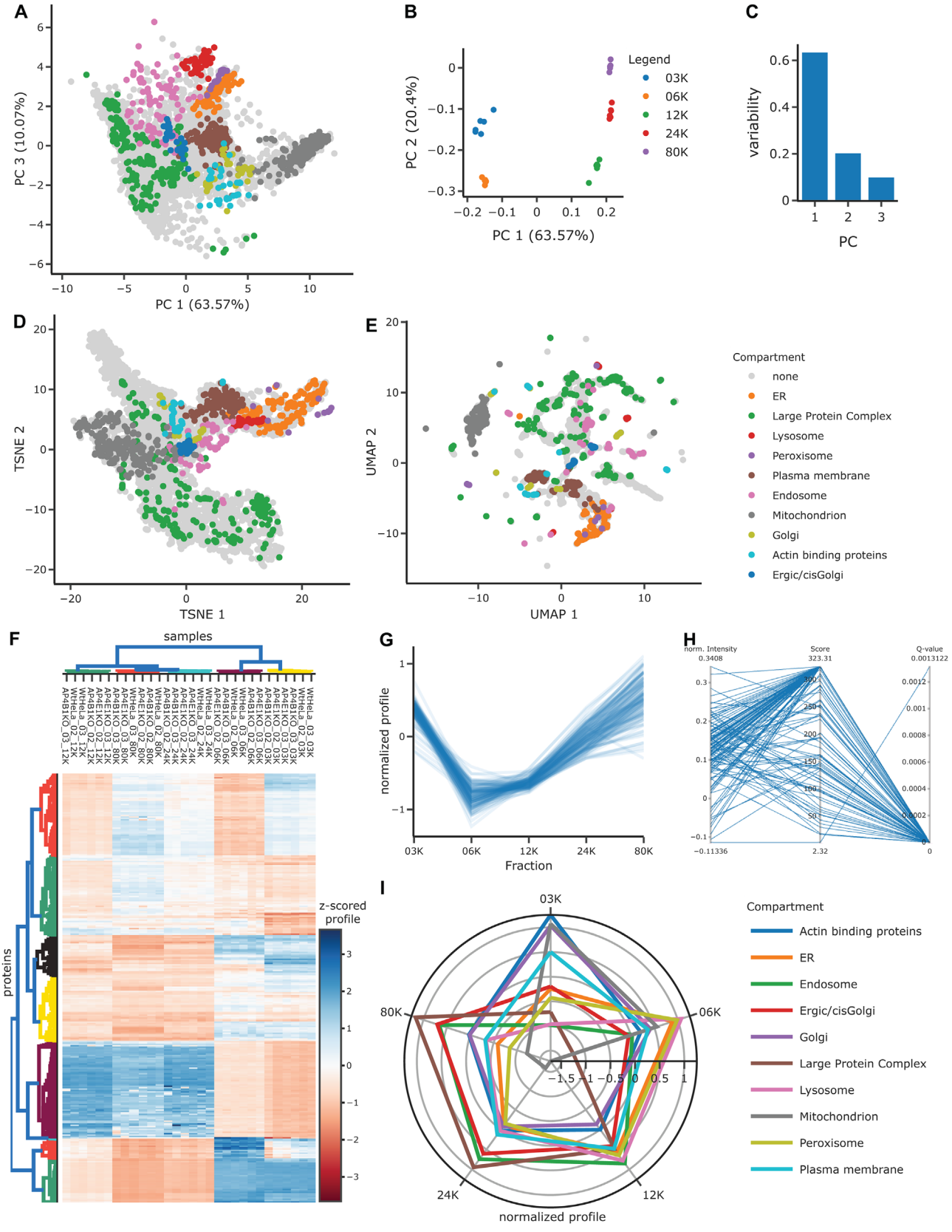

FIGURE 5 **Visualization of multidimensional experimental designs**. The data used for this Figure is a comparative spatial proteomics dataset from [78] (PXD010103). For organelle annotation marker proteins from [109] were used. (A-C) Dimensionality reduction by principal component analysis (PCA). (A) Projections onto PCs 1 and 3 show separation of protein groups into organelles. (B) Loading of PCs 1 and 2 with individual dimensions, that is, samples. Separation along PC1 is between < = 6K and > = 12K fractions, while PC2 separates fractions within each of these groups. As this represents the Eigenvectors of the PCA it is often represented with arrows instead of points. (C) Data variability explained by

Here, every line represents a dataset (e.g., protein or sample) and connects the data points across the parallel dimensions (Figure 5H). If all dimensions are categorical, the group sizes and membership combinations are displayed instead.

2.5 | Network representations of proteomic data

Many extensive proteomics studies, such as interactomics [86, 87], proteome profiling based [88104] or extensive clinical studies [89], focus on networks between proteins or could be mined for them. Any experiment that yields enough data to identify or quantify the physical or phenotypic relation between several pairs of proteins is sufficient to build a network, albeit of variable size. Since all networks are built from nodes and edges, many networks look similar at a first glance although they usually convey vastly different information. In proteomics, most often the nodes represent proteins and edges usually represent one of three types of information: physical interaction or proximity (interactomics), phenotypic similarity (profiling) or shared annotations (e.g., Gene Ontology). The most relevant distinctions made in graph theory are between weighted and unweighted networks and between directed and undirected networks. Additionally, the type of both nodes and edges can be homogenous throughout the network or not. For a general review of networks in biological systems see [47].

Different combinations of these characteristics give rise to three different types of networks often encountered in proteomics studies: (1) The most direct representation of measured relations between nodes are networks with homogenous node types and weighted edges (Figure 6A). Since a two-dimensional layout is often insufficient to convey edge properties accurately simply by length, additional visual channels like number of edges, color and thickness can be used. Groups of nodes are usually highlighted by color (e.g., query proteins vs interactors). (2) Akin to dendrograms, hierarchical networks (Figure 6B) convey information about the organization of proteins into groups. These networks are inherently directed, are often unweighted and generally have heterogeneous node types (e.g., protein complexes and proteins). (3) Incorporating extracted or annotated information about biological processes like protein regulation gives rise to semantic networks.

When reading or creating a network it is important to realize which type of network is used/required, what the main information behind nodes and edges is and how they are encoded in the visualization (see Fung et al. 2012 for more considerations). Depending on the degree of complexity and customization required, different tools can be used to create networks: Literature based interaction networks can be generated using STRING [76] and biological pathway graphs are provided by Reactome [90]. For networks based on quantifications provided by a researcher, many tools are available, including Cytoscape [91] - a very extensive and expandable standalone software - and Perseus, although it only contains limited network functionalities [92]. For scientists with programming experience several options exist, including the Cytoscape API [93], Python libraries like NetworkX [94] and graphviz (https://graphviz.readthedocs.io), the R library network [95], or igraph (https://igraph.org), which is available in both languages. A more specialized tool for clinical proteomics that aims to capture comprehensive prior knowledge is the clinical knowledge graph (CKG) [96].

3 | CUSTOM PROGRAMMATIC DATA VISUALIZATION

In the previous sections we have described several commonly used visualizations in the proteomics field, along with available software tools to create them. However, depending on the experimental design and specific focus of a study, it might still be challenging to find a fitting visualization in one of these tools. A scientist might want to create something entirely novel, or just customize the Figure beyond the capabilities of the tool that you are using. Besides these practical limitations, the data visualization process can also contribute to low transparency and reproducibility in scientific papers by use of closed source software and lack of documentation [97]. These challenges can be mastered by programming the visualizations oneself and sharing the code appropriately. Thus, in this section we describe how Python in combination with established open code/science tools can be used to generate customized proteomics visualizations transparently.

3.1 | Proteomics data visualization in Python

For this review we chose Python as a programming language, because it is widely known for its readability and versatility, as well as a shallow learning curve for new developers and a very active, supportive and collaborative community. The latter is particularly useful considering that "open code" and community engagement can benefit researchers by saving time and funding resources [98]. As a primer for proteomics visualization in R, we recommend [33]. Similar to R, Python already has a large variety of well-documented and well-maintained libraries for scientific computing [99]. Although Python has only been in widespread use in the computational proteomics field for roughly

individual PCs. Only the first three PCs are shown here, as they jointly cover > 90% of the data variability. (D) Projection onto non-linear tSNE dimensions. This has a similar density as the PCA, but different arrangement of organelles. (E) Projection onto non-linear UMAP dimensions. Although this shows the same dataset as A and D, clusters are a lot more visible because the local density is increased. (F) Heatmap with marginal dendrograms (complete linkage) of all organelle marker proteins. Samples are clustered by Pearson correlation, proteins by euclidean distance. (G) Line plot showing profiles along the subcellular dimension of all ER marker proteins. (H) Parallel coordinates plots can be used to relate proteomic data to other data dimensions that use different scales. Here, showing the identification score and q-value together with the normalized protein intensity in one sample (same proteins as in G). (I) Radar plot displaying average profiles per organellar marker group.

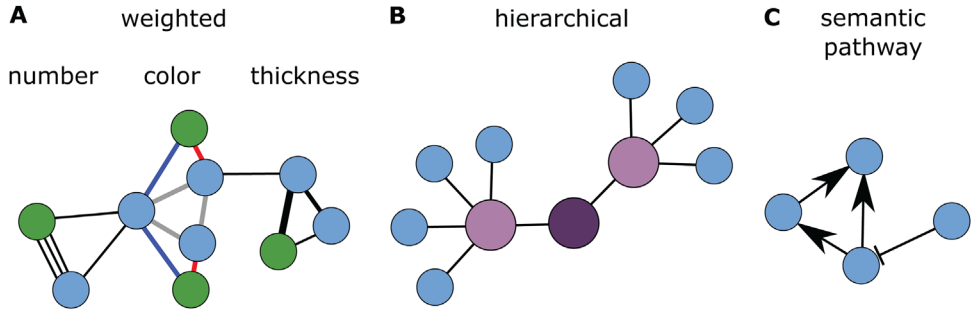

FIGURE 6 **Common network types encountered in proteomics studies.** These are only schematics, in reality these networks are a lot more extensive and often turn into "hairballs" that are hard to read. (A) Weighted protein network with variable visual channels used to encode edge weights. Node color often shows query versus neighboring proteins. (B) Hierarchical network showing group membership of proteins. (C) Semantic pathway network describing biological processes.

TABLE 2 Selection of open-source software libraries for proteomics data analysis and visualization in Python

Library	Description
pymzML [110,111]	An mzML data parser for fast access and handling of the data with integrated data visualization.
Pyteomics [112,113]	A framework for proteomics data analysis, supporting different data formats.
pyOpenMS [114]	A library for the analysis of proteomics and metabolomics data.
multiplierz [115]	A scriptable framework for access to manufacturers' formats via mzAPI.
PaDuA [116]	A Python package optimized for the processing and analysis of quantified (phospho)proteomics data.
AlphaTims [117]	A Python package for efficient accession and visualization of Bruker TimsTOF raw data.
AlphaMap [58]	A Python package for the visual annotation of proteomics data on the peptide level with sequence specific knowledge.
spectrum_utils [118]	A Python package for processing and visualization of MS/MS spectra.

a decade, a number of libraries for MS data accession and specialized analysis tasks are already established (Table 2).

Similar to this data analysis stack, many data visualization libraries exist that are differently well suited for different purposes. Static plots in Python can be generated using Matplotlib [100] or Seaborn [101]. Both libraries are highly versatile, but Seaborn adds additional functionality on top of Matplotlib, for example, it offers more choices for plot styles and colors. Interactive plots are particularly useful for exploratory data analysis by providing data on demand and basic tools like zooming, selecting, rotating, and so on. These can be built in libraries such as Bokeh (docs.bokeh.org) and Plotly (https://plotly.com). Plotly is very popular in the scientific field due to the high number of unique visualizations, including three-dimensional and scientific use cases. Thus, we also used it throughout the code used to generate the Figures in this review.

One overall challenge of data visualization is how to efficiently handle big data. Big data is particularly challenging, because the simultaneous display of thousands of data points usually leads to occlusion of information (as can be seen in Figure 5A) and oftentimes misinterpretation. Common workarounds are down sampling, reduced opacity (as in Figure 5E), replacement by summary statistics (as in Figure 5F) and more. While these methods can often improve data display, the full data scope should always be evaluated and in many cases, it cannot be replaced. An easy way to visualize it without occlusion is offered by the Datashader library (https://datashader.org). It rasterizes the data space similar to a histogram, but in two-dimensional and encodes the number of points per two-dimensional bin by color (Figure 1C, Figure 3C). This facilitates quick visualization of patterns or structures in big data sets.

Due to the amount of data contained in most proteomic studies, there is usually more biological insights to be gained than can be described in a single publication. While uploading datasets to repositories is generally mandatory nowadays, data can be made even more accessible by providing a dedicated online resource or even an analysis service with embedded interactive visualizations. Python provides several libraries that integrate data analysis and visualization capabilities with modern web frameworks to create browser based graphical user interfaces, examples being Dash (https://dash.plotly.com), Streamlit (https://docs.streamlit.io) and Panel (https://panel.holoviz.org).

Using a combination of the scientific Python stack, the generalized visualization libraries and web engines, several visualization tools and resource pages for the proteomics field have already been created [46,58,96,102-104].

3.2 | Open science tools

To enable full accessibility, transparency and reusability of custom visualizations we briefly introduce several existing open science and open source principles and tools.

Firstly, it is important to fully document what any code is doing and to provide necessary context, akin to wet-lab protocols and documentation. A modern software development tool supporting this is Jupyter

(https://jupyter.org), which is compatible with Python, R and Julia. It integrates code, execution output (e.g., visualizations) and static documentation in a single interactive, but freezable file format. The documentation is written in the very simple markdown syntax, which allows standard text formatting and inclusion of complex elements like images and formulas. In recent MS-based proteomics publications, one already sees links to the study specific code provided in Jupyter [98,105]. Given a suitable Python environment and access to the data anybody can thereby reproduce results transparently. In case local hardware is limiting code execution, community resources can be used. Specifically, Google provides a free but powerful Jupyter notebook environment called Google Colab [106].

Secondly, it is important to share code publicly and since code usually continues evolving after publication it is crucial to transparently keep track of code versions, dependencies and contributions. The community standard tool for version control is Git, complemented by the public hosting service GitHub [107], which is free to use for scientific projects. Beyond sharing versioned code, it is also a social coding platform that enables community contributions like peer-review and ensures transparent attribution of code contributions to authors. For code that requires interactive execution, or creates interactive elements, GitHub provides integration online hosting solutions like Binder (https://mybinder.org). To create persistent and citable digital object identifiers (DOIs) for code repositories, Zenodo (https://zenodo.org) can be used directly from GitHub.

To give new developers an easy entry point and an example of what these tools can do, we applied them to the Python code we wrote to create the data visualizations in this review. The repository is hosted on the GitHub (https://github.com/MannLabs/ProteomicsVisualization), which includes a link to the hosted interactive version in Binder and installation instructions for a computational proteomics Python environment and a short guide on how to contribute custom visualizations for others to reuse.

4 | CONCLUSION

In this review we have summarized data visualizations specific to the proteomics field, from raw data to complex experimental designs. As this field is rapidly progressing and highly translational, we decided to not only cite existing tools for visualization, but to further provide guidance towards creating common data visualizations programmatically and interpreting them critically and correctly. As the options for experimental design are constantly evolving we could not cover all flavors of proteomics data visualization herein. It will be exciting to see how interactive web technologies and virtual reality will improve the way we visually explore proteomics data in the years to come, especially with regard to current limitations on three-dimensional visualization. Lastly, we want to encourage our readers to try out different visualization types and visual channels interactively for the data they have at hand and to view data visualization as a creative, yet crucial step of science and science communication.

ACKNOWLEDGMENTS

This study was supported by The Max-Planck Society for Advancement of Science. Eugenia Voytik acknowledges funding by the Bavarian State Ministry of Health and Care through the research project DigiMed Bayern (www.digimed-bayern.de). IB acknowledges funding support from her Postdoc.Mobility fellowship granted by the Swiss National Science Foundation (P400PB_191046). The authors would like to acknowledge all our colleagues at the Department for Proteomics and Signal Transduction, particularly the head of the department, Matthias Mann, and the head of the research group on systems biology of membrane trafficking, Georg Borner, who constantly support us in our work provided valuable feedback. Special thanks go to Alexandra Davies, a cell biologist and MS-expert, who helped us tailor this review to our target audience. Further valuable feedback was given to us by Sander Willems, Peter Treit and Vincent Albrecht.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Julia Patricia Schessner and Eugenia Voytik devised and wrote the manuscript. All authors contributed code and edited the manuscript.

DATA AVAILABILITY STATEMENT

Proteomics data from the following ProteomeExchange repositories were reused to generate Figures in this study: PXD012867, PXD017703, PXD010697, PXD010103.

ORCID

Julia Patricia Schessner https://orcid.org/0000-0003-3361-9830
Eugenia Voytik https://orcid.org/0000-0003-4776-0771
Isabell Bludau https://orcid.org/0000-0002-2601-238X

REFERENCES

1. Kelstrup, C. D., Jersie-Christensen, R. R., Batth, T. S., Arrey, T. N., Kuehn, A., Kellmann, M., & Olsen, J. V. (2014). Rapid and deep proteomes by faster sequencing on a benchtop quadrupole ultra-high-field Orbitrap mass spectrometer. . *Journal of Proteome Research*, *13*, 6187–6195.
2. Linscheid, N., Santos, A., Poulsen, P. C., Mills, R. W., Calloe, K., Leurs, U., Ye, J. Z., Stolte, C., Thomsen, M. B., Bentzen, B. H., Lundegaard, P. R., Olesen, M. S., Jensen, L. J., Olsen, J. V., & Lundby, A. (2021). Quantitative proteome comparison of human hearts with those of model organisms. *PLOS Biology*, *19*(4), e3001144.
3. Michalski, A., Cox, J., & Mann, M. (2011). More than 100,000 detectable peptide species elute in single shotgun proteomics runs but the majority is inaccessible to data-dependent LC-MS/MS. *Journal of Proteome Research*, *10*, 1785–1793.
4. Nagaraj, N., Wisniewski, J. R., Geiger, T., Cox, J., Kircher, M., Kelso, J., Pääbo, S., & Mann, M. (2011). Deep proteome and transcriptome mapping of a human cancer cell line. *Molecular Systems Biology*, *7*(1), 548.
5. Müller, J. B., Geyer, P. E., Colaço, A. R., Treit, P. V., Strauss, M. T., Oroshi, M., Doll, S., Winter, S. V., Bader, J. M., Köhler, N., Theis, F., Santos, A., & Mann, M. (2020). The proteome landscape of the kingdoms of life. *Nature*, *582*, 592–596.

6. Aebersold, R., & Mann, M. (2016). Mass-spectrometric exploration of proteome structure and function. *Nature*, *537*, 347–355.
7. Bache, N., Geyer, P. E., Bekker-Jensen, D. B., Hoerning, O., Falkenby, L., Treit, P. V, Doll, S., Paron, I., Müller, J. B., Meier, F., Olsen, J. V, Vorm, O., & Mann, M. (2018). A novel LC system embeds analytes in pre-formed gradients for rapid, ultra-robust proteomics. *Molecular and Cellular Proteomics*, *17*, 2284–2296.
8. Beck, S., Michalski, A., Raether, O., Lubeck, M., Kaspar, S., Goedecke, N., Baessmann, C., Hornburg, D., Meier, F., Paron, I., Kulak, N. A., Cox, J., & Mann, M. (2015). The impact II, a very high-resolution quadrupole time-of-flight instrument (QTOF) for deep shotgun proteomics. *Molecular and Cellular Proteomics*, *14*, 2014–2029.
9. Buryakov, I. A., Krylov, E. V., Nazarov, E. G., & Rasulev, U. K. (1993). A new method of separation of multi-atomic ions by mobility at atmospheric pressure using a high-frequency amplitude-asymmetric strong electric field. *International Journal of Mass Spectrometry and Ion Processes*, *128*(3), 143.–148.
10. Hebert, A. S., Prasad, S., Belford, M. W., Bailey, D. J., McAlister, G. C., Abbatiello, S. E., Huguet, R., Wouters, E. R., Dunyach, J.-J., Brademan, D. R., Westphall, M. S., & Coon, J. J. (2018). Comprehensive single-shot proteomics with FAIMS on a hybrid orbitrap mass spectrometer. *Analytical Chemistry*, *90*, 9529–9537.
11. Silveira, J. A., Michelmann, K., Ridgeway, M. E., & Park, M. A. (2016). Fundamentals of trapped ion mobility spectrometry part II: Fluid dynamics. *Journal of the American Society for Mass Spectrometry*, *27*, 585–595.
12. Rodriguez-Suarez, E., Hughes, C., Gethings, L., Giles, K., Wildgoose, J., Stapels, M. E., Fadgen, K. J., Geromanos, S. P. C., Vissers, J., Elortza, F. I., & Langridge, J. (2013). An ion mobility assisted data independent LC-MS strategy for the analysis of complex biological samples. *Current Analytical Chemistry*, *9*(2), 199–211.
13. Helm, D., Vissers, J. P. C., Hughes, C. J., Hahne, H., Ruprecht, B., Pachl, F., Grzyb, A., Richardson, K., Wildgoose, J., Maier, S. K., Marx, H., Wilhelm, M., Becher, I., Lemeer, S., Bantscheff, M., Langridge, J. I., & Kuster, B. (2014). Ion mobility tandem mass spectrometry enhances performance of bottom-up proteomics. *Molecular & Cellular Proteomics*, *13*(12), 3709–3715.
14. 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), O111.016717.
15. Meier, F., Geyer, P. E., Winter, S. V., Cox, J., & Mann, M. (2018). BoxCar acquisition method enables single-shot proteomics at a depth of 10,000 proteins in 100 minutes. *Nature Methods*, *15*, 440–448.
16. Meier, F., Beck, S., Grassl, N., Lubeck, M., Park, M. A., Raether, O., & Mann, M. (2015). Parallel accumulation-serial fragmentation (PASEF): Multiplying sequencing speed and sensitivity by synchronized scans in a trapped ion mobility device. *Journal of Proteome Research*, *14*, 5378–5387.
17. Geromanos, S. J., Vissers, J. P. C., Silva, J. C., Dorschel, C. A., Li, G.-Z., Gorenstein, M. V., Bateman, R. H., & Langridge, J. I. (2009). The detection, correlation, and comparison of peptide precursor and product ions from data independent LC-MS with data dependant LC-MS/MS. *Proteomics*, *9*, 1683–1695.
18. Messner, C. B., Demichev, V., Wendisch, D., Michalick, L., White, M., Freiwald, A., Textoris-Taube, K., Vernardis, S. I., Egger, A. S., Kreidl, M., Ludwig, D., Kilian, C., Agostini, F., Zelezniak, A., Thibeault, C., Pfeiffer, M., Hippenstiel, S., Hocke, A., von Kalle, C., … Ralser, M. (2020). Ultra-high-throughput clinical proteomics reveals classifiers of COVID-19 infection. *Cell Systems*, *11*, 11–24.e4.
19. Messner, C. B., Demichev, V., Bloomfield, N., Yu, J. S. L., White, M., Kreidl, M., Egger, A. S., Freiwald, A., Ivosev, G., Wasim, F., Zelezniak, A., Jürgens, L., Suttorp, N., Sander, L. E., Kurth, F., Lilley, K. S., Mülleder, M., Tate, S., & Ralser, M. (2021). Ultra-fast proteomics with Scanning SWATH. *Nature Biotechnology*, *39*(7), 846–854.
20. 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*, 41–44.
21. Gessulat, S., Schmidt, T., Zolg, D. P., Samaras, P., Schnatbaum, K., Zerweck, J., Knaute, T., Rechenberger, J., Delanghe, B., Huhmer, A., Reimer, U., Ehrlich, H.-C., Aiche, S., Kuster, B., & Wilhelm, M. (2019). Prosit: Proteome-wide prediction of peptide tandem mass spectra by deep learning. *Nature Methods*, *16*, 509–518.
22. 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*, 513–520.
23. Tiwary, S., Levy, R., Gutenbrunner, P., Salinas Soto, F., Palaniappan, K. K., Deming, L., Berndl, M., Brant, A., Cimermancic, P., & Cox, J. (2019). High-quality MS/MS spectrum prediction for data-dependent and data-independent acquisition data analysis. *Nature Methods*, *16*, 519–525.
24. Zhou, X.-X., Zeng, W.-F., Chi, H., Luo, C., Liu, C., Zhan, J., He, S.-M., & Zhang, Z. (2017). pDeep: Predicting MS/MS spectra of peptides with deep learning. *Analytical Chemistry 89*, 12690–12697.
25. Gehlenborg, N., O'Donoghue, S. I., Baliga, N. S., Goesmann, A., Hibbs, M. A., Kitano, H., Kohlbacher, O., Neuweger, H., Schneider, R., Tenenbaum, D., & Gavin, A. C. (2010). Visualization of omics data for systems biology. *Nature Methods* 2010 7:3, *7*, S56–S68.
26. Pavlopoulos, G. A., Secrier, M., Moschopoulos, C. N., Soldatos, T. G., Kossida, S., Aerts, J., Schneider, R., & Bagos, P. G. (2011). Using graph theory to analyze biological networks. *BioData Mining*, *4*, 1–27.
27. Adams, K. J., Pratt, B., Bose, N., Dubois, L. G., John-Williams, L. St., Perrott, K. M., Ky, K., Kapahi, P., Sharma, V., MacCoss, M. J., Moseley, M. A., Colton, C. A., MacLean, B. X., Schilling, B., Thompson, J. W., & Consortium, A. D. M. (2020). Skyline for small molecules: a unifying software package for quantitative metabolomics. *Journal of Proteome Research*, *19*, 1447–1458.
28. 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*, *14*(5), 1400–1410.
29. Tyanova, S., Temu, T., Sinitcyn, P., Carlson, A., Hein, M. Y., Geiger, T., Mann, M., & Cox, J. (2016). The Perseus computational platform for comprehensive analysis of (prote)omics data. *Nature Methods*, *13*, 731–740.
30. Tyanova, S., Temu, T., Carlson, A., Sinitcyn, P., Mann, M., & Cox, J. (2015). Visualization of LC-MS/MS proteomics data in MaxQuant. *Proteomics*, *15*, 1453–1456.
31. Oveland, E., Muth, T., Rapp, E., Martens, L., Berven, F. S., & Barsnes, H. (2015). Viewing the proteome: How to visualize proteomics data?. *PROTEOMICS*, *15*, 1341–1355.
32. Perez-Riverol, Y., Wang, R., Hermjakob, H., Müller, M., Vesada, V., & Vizcaíno, J. A. (2014). Open source libraries and frameworks for mass spectrometry based proteomics: A developer's perspective. *Biochimica et Biophysica Acta - Proteins and Proteomics*, *1844*, 63–76.
33. Gatto, L., Breckels, L. M., Naake, T., & Gibb, S. (2015). Visualization of proteomics data using R and bioconductor. *Proteomics*, *15*, 1375–1389.
34. Sinha, A., & Mann, M. (2020). A beginner's guide to mass spectrometry–based proteomics. *The Biochemist*, *42*(5), 64–69.
35. Deutsch, E. W. (2012). File formats commonly used in mass spectrometry proteomics. *Molecular & Cellular Proteomics*, *11*, 1612–1621.

36. Bittremieux, W., Valkenborg, D., Martens, L., & Laukens, K. (2017). Computational quality control tools for mass spectrometry proteomics. *PROTEOMICS*, *17*(3-4), 1600159.
37. Perez-Riverol, Y., Xu, Q.-W., Wang, R., Uszkoreit, J., Griss, J., Sanchez, A., Reisinger, F., Csordas, A., Ternent, T., Del-Toro, N., Dianes, J. A., Eisenacher, M., Hermjakob, H., & Vizcaíno, J. A. (2016). PRIDE inspector toolsuite: Moving toward a universal visualization tool for proteomics data standard formats and quality assessment of proteomexchange datasets. *Molecular & Cellular Proteomics : MCP*, *15*, 305–317.
38. Noga, M., Sucharski, F., Suder, P., & Silberring, J. (2007). A practical guide to nano-LC troubleshooting. *Journal of Separation Science*, *30*, 2179–2189.
39. Rudnick, P. A., Clauser, K. R., Kilpatrick, L. E., Tchekhovskoi, D. v., Neta, P., Blonder, N., Billheimer, D. D., Blackman, R. K., Bunk, D. M., Cardasis, H. L., Ham, A. J. L., Jaffe, J. D., Kinsinger, C. R., Mesri, M., Neubert, T. A., Schilling, B., Tabb, D. L., Tegeler, T. J., Vega-Montoto, L., … Stein, S. E. (2010). Performance metrics for liquid chromatography-tandem mass spectrometry systems in proteomics analyses. *Molecular & Cellular Proteomics*, *9*, 225–241.
40. Cox, J., & Mann, M. (2008). MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nature Biotechnology*, *26*, 1367–1372.
41. Avtonomov, D. M., Raskind, A., & Nesvizhskii, A. I. (2016). BatMass: A Java software platform for LC-MS data visualization in proteomics and metabolomics. *Journal of Proteome Research*, *15*, 2500–2509.
42. Meier, F., Park, M. A., & Mann, M. (2021). Trapped ion mobility spectrometry and parallel accumulation–serial fragmentation in proteomics. *Molecular & Cellular Proteomics*, *20*, 5378–5387.
43. Ting, Y. S., Egertson, J. D., Payne, S. H., Kim, S., MacLean, B., Käll, L., Aebersold, R., Smith, R. D., Noble, W. S., & MacCoss, M. J. (2015). Peptide-centric proteome analysis: An alternative strategy for the analysis of tandem mass spectrometry data. *Molecular & Cellular Proteomics*, *14*(9), 2301–2307.
44. Perkins, D. N., Pappin, D. J. C., Creasy, D. M., & Cottrell, J. S. (1999). Probability-based protein identification by searching sequence databases using mass spectrometry data. 3551–3567.
45. Cox, J., Neuhauser, N., Michalski, A., Scheltema, R. A., Olsen, J. v., & Mann, M. (2011). Andromeda: A peptide search engine integrated into the Maxquant environment. *Journal of Proteome Research*, *10*, 1794–1805.
46. Strauss, M. T., Bludau, I., Zeng, W.-F., Voytik, E., Ammar, C., Schessner, J., Ilango, R., Gill, M., Meier, F., Willems, S., , (2021). AlphaPept, a modern and open framework for MS-based proteomics. *bioRxiv*, 2021.07.23.453379.https://www.biorxiv.org/content/10.1101/2021.07.23.453379.
47. Koutrouli, M., Karatzas, E., Paez-Espino, D., & Pavlopoulos, G. A. (2020). A guide to conquer the biological network era using graph theory. *Frontiers in Bioengineering and Biotechnology*, *8*, https://doi.org/10.3389/fbioe.2020.00034.
48. Hu, A., Noble, W. S., Wolf-Yadlin, A., Hu, A., Noble, W. S. & Wolf-Yadlin, A. (2016). Technical advances in proteomics: new developments in data-independent acquisition F1000Research, *5*, 419.
49. Ludwig, C., Gillet, L., Rosenberger, G., Amon, S., Collins, B. C., & Aebersold, R. (2018). Data-independent acquisition-based SWATH - MS for quantitative proteomics: A tutorial. *Molecular Systems Biology*, *14*(8), e8126.
50. 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* 2015 12:3, *12*, 258–264.
51. Havilio, M., & Wool, A. (2007). Large-scale unrestricted identification of post-translation modifications using tandem mass spectrometry. *Analytical Chemistry*, *79*, 1362–1368.
52. Boersema, P. J., Mohammed, S., & Heck, A. J. R. (2009). Phosphopeptide fragmentation and analysis by mass spectrometry. *Journal of Mass Spectrometry*, *44*, 861–878.
53. Wiśniewski, J. R., Zettl, K., Pilch, M., Rysiewicz, B., & Sadok, I. (2020). "Shotgun" proteomic analyses without alkylation of cysteine. *Analytica Chimica Acta*, *1100*, 131–137.
54. Thompson, A., Schäfer, J., Kuhn, K., Kienle, S., Schwarz, J., Schmidt, G., Neumann, T., & Hamon, C. (2003). Tandem mass tags: A novel quantification strategy for comparative analysis of complex protein mixtures by MS/MS. *Analytical Chemistry*, *75*, 1895–1904.
55. Virreira Winter, S., Meier, F., Wichmann, C., Cox, J., Mann, M., & Meissner, F. (2018). EASI-tag enables accurate multiplexed and interference-free MS2-based proteome quantification. *Nature Methods*, *15*, 527–530.
56. Desiere, F. (2006). The PeptideAtlas project. *Nucleic Acids Research*, *34*(90001), D655–D658.
57. Hornbeck, P. V., Zhang, B., Murray, B., Kornhauser, J. M., Latham, V., & Skrzypek, E. (2015). PhosphoSitePlus, 2014: Mutations, PTMs and recalibrations. *Nucleic Acids Research*, *43*(D1), D512–D520.
58. Voytik, E., Bludau, I., Willems, S., Hansen, F. M., Brunner, A.-D., Strauss, M. T., & Mann, M. (2021). AlphaMap: An open-source Python package for the visual annotation of proteomics data with sequence-specific knowledge. *Bioinformatics*, *38*, 849–852.
59. Claassen, M. (2012). Inference and validation of protein identifications. *Molecular & Cellular Proteomics*, *11*(11), 1097–1104.
60. Nesvizhskii, A. I., Keller, A., Kolker, E., & Aebersold, R. (2003). A statistical model for identifying proteins by tandem mass spectrometry. *Analytical Chemistry*, *75*, 4646–4658.
61. Matzke, M. M., Brown, J. N., Gritsenko, M. A., Metz, T. O., Pounds, J. G., Rodland, K. D., Shukla, A. K., Smith, R. D., Waters, K. M., McDermott, J. E., & Webb-Robertson, B.-J. (2013). A comparative analysis of computational approaches to relative protein quantification using peptide peak intensities in label-free LC-MS proteomics experiments. *Proteomics*, *13*, 493–503.
62. Schmidt, R. M., Schessner, J. P., Borner, G. H. H., & Schuck, S. (2019). The proteasome biogenesis regulator Rpn4 cooperates with the unfolded protein response to promote ER stress resistance. *eLife*, *8*, https://doi.org/10.7554/elife.43244.
63. Schwanhäusser, B., Busse, D., Li, N., Dittmar, G., Schuchhardt, J., Wolf, J., Chen, Wei, & Selbach, M. (2011). Global quantification of mammalian gene expression control. *Nature*, *473*(7347), 337–342.
64. Wiśniewski, J. R., Hein, M. Y., Cox, J., & Mann, M. (2014). A "Proteomic Ruler" for protein copy number and concentration estimation without spike-in standards. *Molecular & Cellular Proteomics*, *13*(12), 3497–3506.
65. Krzywinski, M., & Altman, N. (2014). Comparing samples—part I. *Nature Methods*, *11*(3), 215–216.
66. Krzywinski, M., & Altman, N. (2014). Comparing samples—part II. *Nature Methods*, *11*(4), 355–356.
67. Krzywinski, M., & Altman, N. (2013). Power and sample size. *Nature Methods*, *10*(12), 1139–1140.
68. Tusher, V. G., Tibshirani, R., & Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences*, *98*, 5116–5121.
69. Maleki, F., Ovens, K., Hogan, D. J., & Kusalik, A. J. (2020). Gene set analysis: Challenges, opportunities, and future research. *Frontiers in Genetics*, *11*, https://doi.org/10.3389/fgene.2020.00654.
70. Fisher, R. A. (1922). On the interpretation of χ 2 from contingency tables, and the calculation of P. *Journal of the Royal Statistical Society*, *85*(1), 87–94.
71. Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., & Mesirov, J. P. (2005). Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression pro-

files. *Proceedings of the National Academy of Sciences*, *102*, 15545–15550.
72. Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., & Sherlock, G. (2000). Gene ontology: Tool for the unification of biology. The gene ontology consortium. *Nature Genetics*, *25*, 25–29.
73. The gene ontology resource: Enriching a GOld mine. (2021). *Nucleic Acids Research*, *49*, D325–D334.
74. Khatri, P., & Draghici, S. (2005). Ontological analysis of gene expression data: current tools, limitations, and open problems. *Bioinformatics*, *21*(18), 3587–3595.
75. Snel, B. (2000). STRING: a web-server to retrieve and display the repeatedly occurring neighbourhood of a gene. *Nucleic Acids Research*, *28*(18), 3442–3444.
76. Mi, H., Ebert, D., Muruganujan, A., Mills, C., Albou, L. P., Mushayamaha, T., & Thomas, P. D. (2021). PANTHER version 16: A revised family classification, tree-based classification tool, enhancer regions and extensive API. *Nucleic Acids Research*, *49*, D394–D403.
77. Brüning, F., Noya, S. B., Bange, T., Koutsouli, S., Rudolph, J. D., Tyagarajan, S. K., Cox, J., Mann, M., Brown, S. A., & Robles, M. S. (2019). Sleep-wake cycles drive daily dynamics of synaptic phosphorylation. *Science*, *366*(6462), https://doi.org/10.1126/science.aav3617.
78. Davies, A. K., Itzhak, D. N., Edgar, J. R., Archuleta, T0 L., Hirst, J., Jackson, L. P., Robinson, M. S., & Borner, G. H. H. (2018). AP-4 vesicles contribute to spatial control of autophagy via RUSC-dependent peripheral delivery of ATG9A. *Nature Communications*, *9*(1), https://doi.org/10.1038/s41467-018-06172-7.
79. Bludau, I., Heusel, M., Frank, M., Rosenberger, G., Hafen, R., Banaei-Esfahani, A., van Drogen, A., Collins, B. C., Gstaiger, M., & Aebersold, R. (2020). Complex-centric proteome profiling by SEC-SWATH-MS for the parallel detection of hundreds of protein complexes. *Nature Protocols*, *15*(8), 2341–2386.
80. Jean Beltran, P. M., Mathias, R. A., & Cristea, I. M. (2016). A portrait of the human organelle proteome in space and time during cytomegalovirus infection. *Cell Systems*, *3*, 361–373.e6.
81. Pangratz-Fuehrer, S., Genzel-Boroviczény, O., Bodensohn, W., Eisenburger, R., Scharpenack, J., Geyer, P. E., Müller-Reif, J. B., van Hagen, N., Müller, A. M., Jensen, M. K., Klein, C., Mann, M., & Nussbaum, C. (2021). Cohort profile: the MUNICH preterm and term clinical study (MUNICH-PreTCI), a neonatal birth cohort with focus on prenatal and postnatal determinants of infant and childhood morbidity. *BMJ Open*, *11*(6), e050652.
82. Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, *24*(7), 498–520.
83. Van Der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html.
84. Mcinnes, L., Healy, J. & Melville, J. (2020). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. https://arxiv.org/abs/1802.03426
85. Do, J. H., & Choi, D.-K. (2008). Clustering approaches to identifying gene expression patterns from DNA microarray data. *Molecules and Cells*, *25*, 279–288.
86. Hein, M. Y., Hubner, N. C., Poser, I., Cox, J., Nagaraj, N., Toyoda, Y., Gak, I. A., Weisswange, I., Mansfeld, J., Buchholz, F., Hyman, A. A., & Mann, M. (2015). A human interactome in three quantitative dimensions organized by stoichiometries and abundances. *Cell*, *163*, 712–723.
87. Huttlin, E. L., Bruckner, R. J., Navarrete-Perea, J., Cannon, J. R., Baltier, K., Gebreab, F., Gygi, M. P., Thornock, A., Zarraga, G., Tam, S., Szpyt, J., Gassaway, B. M., Panov, A., Parzen, H., Fu, S., Golbazi, A., Maenpaa, E., Stricker, K., Guha Thakurta, S., … Gygi, S. P. (2021). Dual proteome-scale networks reveal cell-specific remodeling of the human interactome. *Cell*, *184*, 3022–3040.e28.
88. Kirkwood, K. J., Ahmad, Y., Larance, M., & Lamond, A. I. (2013). Characterization of native protein complexes and protein isoform variation using size-fractionation-based quantitative proteomics. *Molecular & Cellular Proteomics : MCP*, *12*, 3851–3873.
89. Shi, Y., Ding, Y., Li, G., Wang, L., Osman, R. A., Sun, J., Qian, L., Zheng, G., & Zhang, G. (2021). Discovery of novel biomarkers for diagnosing and predicting the progression of multiple sclerosis using TMT-based quantitative proteomics. *Frontiers in Immunology*, *12*, https://doi.org/10.3389/fimmu.2021.700031.
90. Joshi-Tope, G. (2004). Reactome: A knowledgebase of biological pathways. *Nucleic Acids Research*, *33*(Database issue), D428–D432.
91. Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., & Ideker, T. (2003). Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Research*, *13*(11), 2498–2504.
92. Rudolph, J. D., & Cox, J. (2019). A network module for the perseus software for computational proteomics facilitates proteome interaction graph analysis. *Journal of Proteome Research*, *18*, 2052–2064.
93. Otasek, D., Morris, J. H., Bouças, J., Pico, A. R., & Demchak, B0 (2019) Cytoscape automation: Empowering workflow-based network analysis. *Genome Biology*, *20*(1), https://doi.org/10.1186/s13059-019-1758-4.
94. Hagberg, A, Swart, P. & S Chult, D. (2008). Exploring network structure, dynamics, and function using networkx. https://www.osti.gov/biblio/960616
95. Butts, C. T. (2008). Network: A package for managing relational data inR. *Journal of Statistical Software*, *24*(2), 1–36.
96. Santos, A., Colaço, A. R., Nielsen, A. B., Niu, L., Strauss, M., Geyer, P. E., Coscia, F., Albrechtsen, N. J. W., Mundt, F., Jensen, L. J., & Mann, M. (2022). A knowledge graph to interpret clinical proteomics data. *Nature Biotechnology*, https://doi.org/10.1038/s41587-021-01145-6.
97. Baker, M. (2016). 1,500 scientists lift the lid on reproducibility. *Nature*, *533*, 452–454.
98. Bittremieux, W., Adams, C., Laukens, K., Dorrestein, P. C., & Bandeira, N. (2021). Open science resources for the mass spectrometry-based analysis of SARS-CoV-2. *J. Proteome Res.*, *20*, 1464–1475.
99. Perez, F., Granger, B. E., & Hunter, J. D. (2011). Python: An ecosystem for scientific computing. *Computing in Science & Engineering*, *13*(2), 13–21.
100. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95.
101. Waskom, M., Botvinnik, O., O'Kane, D., Hobson, P., Lukauskas, S., Gemperline, D. C., Augspurger, T., Halchenko, Y., Cole, J. B., Warmenhoven, J., Ruiter, J., Pye, C., Hoyer, S., Vanderplas, J., Villalba, S., Kunter, G., Quintero, E., Bachant, P., Martin, M. & Qalieh, A. …, (2017). mwaskom/seaborn: v0.8.1 (September 2017). Zenodo. https://doi.org/10.5281/zenodo.883859
102. Petras, D., Phelan, V. V., Acharya, D., Allen, A. E., Aron, A. T., Bandeira, N., Bowen, B. P., Belle-Oudry, D., Boecker, S., Cummings, D. A., Deutsch, J. M., Fahy, E., Garg, N., Gregor, R., Handelsman, J., Navarro-Hoyos, M., Jarmusch, A. K., Jarmusch, S. A., Louie, K., Maloney, K. N., Marty, M. T., Meijler, M. M., Mizrahi, I., Neve, R. L., Northen, T. R., Molina-Santiago, C., Panitchpakdi, M., Pullman, B., Puri, A. W., Schmid, R., …, (2021). GNPS Dashboard: Collaborative exploration of mass spectrometry data in the web browser. *Nature Methods*, https://doi.org/10.1038/s41592-021-01339-5.
103. Hansen, F. M., Tanzer, M. C., Brüning, F., Bludau, I., Stafford, C., Schulman, B. A., Robles, M. S., Karayel, O., & Mann, M. (2021). Data-independent acquisition method for ubiquitinome analysis reveals regulation of circadian biology. *Nature Communications*, *12*(1), https://doi.org/10.1038/s41467-020-20509-1.

104. Martin-Jaular, L., Nevo, N., Schessner, J. P., Tkach, M., Jouve, M., Dingli, F., Loew, D., Witwer, K. W., Ostrowski, M., Borner, G. H. H., & Théry, C. (2021). Unbiased proteomic profiling of host cell extracellular vesicle composition and dynamics upon HIV-1 infection. *The EMBO Journal*, *40*(8), e105492.
105. Meier, F., Köhler, N. D., Brunner, A.-D., Wanka, J.-M. H., Voytik, E., Strauss, M. T., Theis, F. J., & Mann, M. (2021). Deep learning the collisional cross sections of the peptide universe from a million experimental values. *Nature Communications*, *12*(1), https://doi.org/10.1038/s41467-021-21352-8.
106. Bisong, E. (2019). Google Colaboratory. In E. Bisong (Ed.), *Building machine learning and deep learning models on google cloud platform: A comprehensive guide for beginners* (pp. 59–64). Apress. https://doi.org/10.1007/978-1-4842-4470-8_7.
107. Perez-Riverol, Y., Gatto, L., Wang, R., Sachsenberg, T., Uszkoreit, J., Leprevost, F. da V., Fufezan, C., Ternent, T., Eglen, S. J., Katz, D. S., Pollard, T. J., Konovalov, A., Flight, R. M., Blin, K., & Vizcaíno, J. A. (2016). Ten simple rules for taking advantage of git and GitHub. *PLOS Computational Biology*, *12*(7), e1004947.
108. 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*, 1229–1236.
109. Itzhak, D. N., Tyanova, S., Cox, J., & Borner, G. H. H. (2016). Global, quantitative and dynamic mapping of protein subcellular localization. *eLife*, *5*, https://doi.org/10.7554/elife.16950.
110. Bald, T., Barth, J., Niehues, A., Specht, M., Hippler, M., & Fufezan, C. (2012). pymzML-Python module for high-throughput bioinformatics on mass spectrometry data. *Bioinformatics*, *28*(7), 1052–1053.
111. Kösters, M., Leufken, J., Schulze, S., Sugimoto, K., Klein, J., Zahedi, R. P., Hippler, M., Leidel, S. A., & Fufezan, C. (2018). pymzML v2.0: Introducing a highly compressed and seekable gzip format. *Bioinformatics*, *34*, 2513–2514.
112. Goloborodko, A. A., Levitsky, L. I., Ivanov, M. V., & Gorshkov, M. V. (2013). Pyteomics—a Python framework for exploratory data analysis and rapid software prototyping in proteomics. *Journal of the American Society for Mass Spectrometry*, *24*(2), 301–304.
113. Levitsky, L. I., Klein, J. A., Ivanov, M. V, & Gorshkov, M. V. (2019). Pyteomics 4.0: Five years of development of a Python proteomics framework. *Journal of Proteome Research*, *18*, 709–714.
114. Röst, H. L., Schmitt, U., Aebersold, R., & Malmström, L. (2014). pyOpenMS: a Python-based interface to the OpenMS mass-spectrometry algorithm library. *Proteomics*, *14*, 74–77.
115. Alexander, W. M., Ficarro, S. B., Adelmant, G., & Marto, J. A. (2017). multiplierz v2.0: A Python-based ecosystem for shared access and analysis of native mass spectrometry data. *Proteomics*, *17*(15-16), 1700091.
116. Ressa, A., Fitzpatrick, M., van den Toorn, H., Heck, A. J. R., & Altelaar, M. (2019). PaDuA: A Python library for high-throughput (phospho)proteomics data analysis. *J Proteome Research*, *18*, 576–584.
117. Willems, S., Voytik, E., Skowronek, P., Strauss, M. T., & Mann, M. (2021). AlphaTims: Indexing trapped ion mobility spectrometry–TOF data for fast and easy accession and visualization. *Molecular & Cellular Proteomics*, *20*, 100149.
118. Bittremieux, W. (2020). spectrum_utils: A python package for mass spectrometry data processing and visualization. *Analytical Chemistry*, *92*(1), 659–661.

How to cite this article: Schessner, J. P., Voytik, E., & Bludau, I. (2022). A practical guide to interpreting and generating bottom-up proteomics data visualizations. *Proteomics*, e2100103. https://doi.org/10.1002/pmic.202100103