

# MSstats Version 4.0: Statistical Analyses of Quantitative Mass Spectrometry-Based Proteomic Experiments with Chromatography-Based Quantification at Scale

Devon Kohler, Mateusz Staniak, Tsung-Heng Tsai, Ting Huang, Nicholas Shulman, Oliver M. Bernhardt, Brendan X. MacLean, Alexey I. Nesvizhskii, Lukas Reiter, Eduard Sabido, Meena Choi,\* and Olga Vitek\*



Cite This: *J. Proteome Res.* 2023, 22, 1466–1482



Read Online

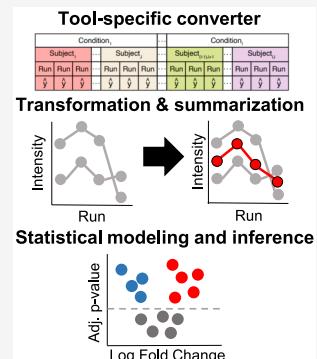
ACCESS |

Metrics & More

Article Recommendations

Supporting Information

**ABSTRACT:** The *MSstats* R-Bioconductor family of packages is widely used for statistical analyses of quantitative bottom-up mass spectrometry-based proteomic experiments to detect differentially abundant proteins. It is applicable to a variety of experimental designs and data acquisition strategies and is compatible with many data processing tools used to identify and quantify spectral features. In the face of ever-increasing complexities of experiments and data processing strategies, the core package of the family, with the same name *MSstats*, has undergone a series of substantial updates. Its new version *MSstats* v4.0 improves the usability, versatility, and accuracy of statistical methodology, and the usage of computational resources. New converters integrate the output of upstream processing tools directly with *MSstats*, requiring less manual work by the user. The package's statistical models have been updated to a more robust workflow. Finally, *MSstats*' code has been substantially refactored to improve memory use and computation speed. Here we detail these updates, highlighting methodological differences between the new and old versions. An empirical comparison of *MSstats* v4.0 to its previous implementations, as well as to the packages *MSqRob* and *DEqMS*, on controlled mixtures and biological experiments demonstrated a stronger performance and better usability of *MSstats* v4.0 as compared to existing methods.



**KEYWORDS:** Bioinformatics, Quantitative proteomics, Mass spectrometry, Statistical modeling, Statistical inference

## INTRODUCTION

Technological advances in bottom up liquid chromatography coupled with high-resolution tandem mass spectrometry (LC-MS/MS) have facilitated the quantification of changes in protein abundance across many conditions.<sup>1</sup> The experiments have increasingly complex experimental designs differing in number and type of conditions (group comparisons or repeated measures), and number and type of replicates (biological replicates, technical replicates, fractionation).<sup>2,3</sup> Additionally, numerous versions of chromatography-based acquisition methods, such as Data-Dependent Acquisition (DDA), Selected Reaction Monitoring (SRM), and Data-Independent Acquisition (DIA), now offer effective and complementary strategies for relative protein quantification with high accuracy and high throughput.<sup>4</sup> The experiments generate mass spectra, with features corresponding to peptide ions for DDA, peptide transitions for SRM, and peptide ions and fragment ions for DIA.

Regardless of the experimental aims and design, software tools are required to identify and quantify proteins in the resulting MS spectra. A variety of computational spectral processing tools, such as MaxQuant,<sup>5</sup> Skyline,<sup>6</sup> and Spectronaut,<sup>7</sup> extract, identify, and quantify features from the acquired spectra.<sup>8</sup> These tools are developed by different

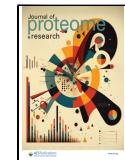
groups, optimized for different instruments and workflows, and make different decisions when feature identification or quantification are challenging. For example, the tools differ in strategies for quantifying feature intensities in individual LC-MS runs, e.g., peak area or peak height at apex. They also differ in reporting the intensity of the monoisotopic peak, the sum of multiple isotopic peaks, or relying on more advanced procedures. Moreover, the tools make different choices for reporting outlying or missing values, which occur in the presence of truncated or overlapped peaks, or when the abundance of an analyte is below the limit of detection.

The complexities of the experimental designs, data acquisitions and data processing create major challenges for the downstream statistical analyses, and may lead to inconsistent or erroneous results.<sup>9</sup> To accommodate these new challenges, methods for statistical analyses and their implementations must grow in parallel. Statistical methods

**Special Issue:** Methods for Omics Research 2023

**Received:** December 21, 2022

**Published:** April 5, 2023



must be both versatile enough to reflect various scientific questions and experimental designs, and robust enough to not overfit to a particular experiment, experimental design, or data processing tool. Implementations must utilize memory-efficient and time-efficient data structures and algorithms that scale to large data sets.<sup>10</sup>

A variety of statistical methods for detecting differentially abundant proteins have been developed to address these challenges. We loosely classify the existing methods into two groups: two-step and feature-based. Two-step methods first summarize the intensities of all the features of a protein in a run, and then subject the summaries to statistical modeling. In contrast, feature-based methods take as input quantified features, and specify a full statistical model at the feature level directly.<sup>11,12</sup> For example, *MSqRob*<sup>13,14</sup> includes functionalities for both a feature-based workflow, as well as summarization functionality for a two-step workflow. The package utilizes a linear mixed effects model to detect differential proteins and includes functionality to apply empirical Bayes variance estimation.<sup>15</sup> Meanwhile, *DEqMS*<sup>16</sup> takes as input protein-level summaries, reported by a data processing tool such as MaxQuant, and leverages the methods in the Limma R package<sup>17–19</sup> to moderate protein variance's through grouping proteins by their number of features in order to identify differentially abundant proteins. Other packages of note include *proDA*,<sup>20</sup> a two-step method which focuses on missing data and implements a probabilistic dropout based model to combine missing and observed values, *pmartr*,<sup>21</sup> another two-step method which focuses on the quality control, preprocessing of MS data, and Analysis of Variance-(ANOVA)<sup>22</sup> for statistical analysis, and *DAPAR*,<sup>23</sup> which implements a two-step method in a graphical user interface. Other packages, such as *QFeatures*<sup>24</sup> and *msImpute*,<sup>25</sup> focus on the first step of the two-step workflow, implementing functionalities for missing value imputation, and data preprocessing.

Although software tools implementing many methods exist, their functionalities often have limitations. First, the methods do not easily integrate with upstream spectral processing tools. The outputs of these tools typically need to be manually converted into the correct format before it can be used for the statistical analyses. Only a subset of implementations include functionality for data processing such as feature filtering, normalization, and missing value imputation, leaving the user to rely on other resources such as the *QFeatures*, or implement the steps on their own. The latter solution is particularly undesirable, as custom data processing leads to irreproducible results. Beyond data processing, while all the workflows provide functionalities for group comparison, their ability to distinguish biological and technical replicates is not always clear. This is undesirable for complex experimental designs, such as time-course and paired designs, where each replicate is measured in multiple conditions. Additionally, some of the methods require users to manually specify the model that fits their data. These methodological aspects challenge users with no statistical background, and can result in incorrectly specified models. Finally, not every implementation is optimized in terms of runtime and computational memory requirement for large data sets.

Here, we introduce *MSstats* version 4.0 (v4.0), a statistical methodology and core package in the family of R/Bioconductor packages designed for statistical analysis of experiments with chromatography-based quantification. Com-

pared to the previous versions of *MSstats*, *MSstats* v4.0 includes a series of methodological and technical improvements. Specifically, compared to *MSstats* v2.0, *MSstats* v3.0 includes substantial methodological improvements. These include updates to upstream data processing, such as missing value imputation, and switching from a feature-based to two-step modeling method. *MSstats* v4.0 implements the same statistical models as *MSstats* v3.0, but with a substantially refactored code that improves both the processing speed and memory use. While *MSstats* has undergone many methodological and technical improvements in recent years, there has not been a comprehensive review of the package and methods since *MSstats* v2.0.

Additionally, since its original publication, *MSstats* has expanded into a broad family of R/Bioconductor packages and methods. *MSstats* v4.0 is now designed for label-free acquisition and detection of relative changes in protein abundance, while other packages in the family address different types of experimental designs and biological objectives. Specifically, *MSstatsTMT*<sup>26</sup> is designed for experiments acquired via tandem mass tag (TMT) labeling. *MSstatsPTM*<sup>27</sup> focuses on experiments studying post-translational modifications (PTMs). Finally, *MSstatsShiny*<sup>28</sup> is an R-shiny based graphical user interface (GUI), which makes the *MSstats* methods accessible to users without programming background. While *MSstats* v4.0 itself is not directly designed for these specific use-cases, it serves as a backend for many. *MSstatsTMT* directly applies the summarization methods in *MSstats* v4.0 to each TMT mixture. *MSstatsPTM* utilizes the models in *MSstats* v4.0, while adding a statistical correction to remove confounding with the unmodified protein. *MSstats* v4.0 is at the core of all packages in the *MSstats* package family and the methods are leveraged without any input required by the user.

*MSstats* v4.0 includes direct converters for popular data processing tools. Converters for label-free DDA experiments include MaxQuant,<sup>5</sup> Fragpipe,<sup>29</sup> Skyline,<sup>6</sup> OpenMS,<sup>30</sup> Proteome Discoverer (Thermo Scientific), and Progenesis QI (Nonlinear Dynamics/Waters). SRM spectra can be quantified by Skyline, and MultiQuant (Sciex). DIA spectra can be processed by Skyline, Spectronaut<sup>7</sup> (Biognosys), DIA-Umpire,<sup>31</sup> OpenSWATH,<sup>32</sup> and DIA-NN.<sup>33</sup> After conversion, *MSstats* v4.0 includes multiple options for upstream data processing, including feature filtering and missing value imputation, providing users flexibility in preparing their data for modeling. *MSstats* v4.0 implements a two-step modeling method, first summarizing the feature intensities, and then fitting a linear mixed effects model to the summarized data. *MSstats* automatically adjusts the linear mixed effects model to fit the specific experimental design, greatly easing implementation complexity. Finally, *MSstats* v4.0 leverages several computational strategies to improve memory use and computation speed, including moving high-resource functions to the C++ programming language, and implementing a new workflow for larger-than-memory input files.

Here we empirically compared *MSstats* v4.0/v3.0 to *MSstats* v2.0 to *MSqRob*, and *DEqMS* on three controlled mixtures and two biological experiments. We demonstrated that the two-step statistical methodology is more versatile, being applicable to time-course and paired designs, as well as more accurate compared to the models in *MSstats* v2.0. *MSstats* v4.0 outperformed existing methods, while being easier to implement by users with limited statistical background. Finally, we contrasted the computational resource usage between *MSstats*

**Table 1.** Experimental Data Sets in this Manuscript<sup>a</sup>

Data Set	Type	Experimental Design			(MassIVE.quant or R Package)
		No. of Conditions	No. of Bio. Replicates	No. of Tech. Replicates	
1: Controlled Mixture-DDA–MaxQuant <sup>9</sup>	Group	4	1	3	RMSV00000249.2
2: Controlled Mixture–DIA–Spectronaut <sup>34</sup>	Group	2	1	3	RMSV00000250.2
3: Controlled Mixture–DDA–Skyline <sup>35</sup>	Group	5	1	3	RMSV00000261.1
4: Mouse–DDA–MaxQuant <sup>36</sup>	Paired	2	6	2	RMSV00000292.1
5: <i>S. cerevisiae</i> –DIA–Skyline <sup>37</sup>	Time course	6	3	1	RMSV00000251.1

<sup>a</sup>“Data Set” is the code name of the data set. “Type” is the type of experimental design (group comparison, paired, or time course). “No. of Conditions” states the number of conditions in the data set. “No. of Bio. Replicates” states the number of biological replicates per condition. “No. of Tech. Replicates” states the number of technical replicates per biological replicate. More details about experimental design and data processing steps for each data sets are available in next section, as listed in “Details in” column. The details of the experimental design, the raw data, the reports, the analysis scripts, the intermediate data processing output files including quantification, testing results are available in MassIVE.quant.

v2.0, v3.0, and v4.0 to highlight recent technological improvements. The package is open source and is available on Bioconductor and Github.

## EXPERIMENTAL PROCEDURES

### Data Overview and Availability

**Table 1** summarizes the data sets included in this manuscript. Three controlled mixtures were obtained by spiking proteins in known concentrations into a complex background or by diluting the same parent mixture. They covered a broad range of known fold changes between conditions and represent two DDA and one DIA data acquisition strategy. Two biological investigations demonstrated the applicability of the proposed approach across different experimental designs and acquisition strategies. The true composition of the biological experiments were unknown and the statistical methods could only be evaluated in terms of the differences between the outputs of *MSqRob*, *DEqMS*, *MSstats* v2.0, and *MSstats* v4.0. Finally, two simulated out of memory data sets were generated to illustrate the functionalities of *MSstatsBig*.

Details of data processing, software versions, R scripts with *MSstats* analyses, and result of statistical analyses for all data sets are available in MassIVE.quant (with direct links available in **Table 1**).

### Data Set 1: Controlled Mixture–DDA–MaxQuant<sup>9</sup>

The experiment was conducted as part of the 2015 study of the Proteome Informatics Research Group (iPRG) of the Association of the Biomedical Resource Facilities (ABRF). Six proteins were spiked into samples with *S. cerevisiae* proteome as the background. Four proteins were spiked with four different concentrations, forming a Latin Square design. Two more proteins were spiked in another four different concentrations in the same four biological samples. The concentrations are summarized in **Table 2**. Data from four mixtures were acquired in DDA mode with triplicate MS runs, resulting in 12 MS runs in total. Raw data were processed with MaxQuant.

### Data Set 2: Controlled Mixture–DIA–Spectronaut<sup>34</sup>

This study prepared two hybrid proteome samples, A and B, consisting of tryptic digests of human, *S. cerevisiae*, and *E. coli* proteomes. The proteomes were mixed in defined proportions, to yield expected peptide and protein ratios (A/B) of 1:1 for human, 2:1 for *S. cerevisiae* and 1:4 for *E. coli* proteins. Data from two mixtures were acquired in DIA mode with triplicate MS runs, resulting in 6 MS runs, and processed with Spectronaut. This manuscript used the raw files acquired on

**Table 2.** DDA:Choi207 Protein IDs and Concentrations of Spike-in Proteins

Protein	Concentration(fmol)			
	Sample1	Sample2	Sample3	Sample4
P44015	65	55	15	2
P44752	55	15	2	65
P44374	15	2	65	55
P44983	2	65	55	15
P44683	11	0.6	10	500
P55249	10	500	11	0.6
Background	+200ng yeast digest			

TripleTOF 6600, Iteration 2, with SWATH window number 64.

### Data Set 3: Controlled Mixture–DDA–Skyline<sup>35</sup>

Thirty proteins were prepared at 1.5 pmol/μL in three different subsets of 10 proteins each (**Table 3**). Proteins from these subsets were spiked into a 15 μg *E. coli* background in different amounts of five mixtures. The final amount of each protein in each mixture was either 100, 200, or 400 fmol/μg of *E. coli* background, indicated as 1, 2, and 4 in **Table 4**. Data from the mixtures were acquired in DDA mode with triplicate MS runs, resulting in 15 MS runs in total. Raw data were processed with Skyline.

### Data Set 4: Mouse–DDA–MaxQuant<sup>36</sup>

This experiment investigated Ataxin-2 (ATXN2) deficiency through Atxn2-knockout (Atxn2-KO) mice. Liver samples were taken from six wild type (WT) and six Atxn2-KO mice in a balanced paired design. Data were acquired in DDA mode with 1 technical replicate and processed with MaxQuant. Details on the specific instrument and instrument settings are available in the original publication.<sup>36</sup> We evaluated the ability of the statistical methods to correctly estimate the fold change and identify differentially abundant proteins between conditions.

### Data Set 5: *S. cerevisiae*–DIA–Skyline<sup>37</sup>

Cells of *S. cerevisiae* were cultured in biological triplicates, and sampled at six time points (0 min (T0), 15 min(T1), 30 min (T2), 60 min (T3), 90 min (T4), 120 min (T5)) after osmotic stress. Data were acquired in SWATH/DIA mode, and processed using Skyline. The analysis used additional information from 8 technical replicate MS runs on the same samples, and limited the analyses to peptides that were detected in at least 4 runs and CV less than 20% in these 8 runs. In this manuscript we compared five time points relative to T0 (T1 vs T0, T2 vs T0, T3 vs T0, T4 vs T0, T5 vs T0).

**Table 3.** SwissProt Accession Numbers of Proteins Spiked into the *E. coli* Background in Data Set 3: Controlled Mixture–DDA–Skyline

Subset 1		Subset 2		Subset 3	
Accession Number	Molecular Weight	Accession Number	Molecular Weight	Accession Number	Molecular Weight
P02701	16	P00915	29	Q3SX14	81
P00711	16	P02787	77	P00563	43
Q29443	78	P02663	24	P02769	69
Q29550	62	P01008	53	Q58D62	48
P0CG53	8	P00921	29	P00698	15
P68082	17	P05307	57	P00004	12
P00432	60	P61769	14	P00711	14
P02754	20	P02662	24	P00442	33
P24627	78	P01012	45	P01133	134
P80025	81	P02666	25	P02753	23

**Table 4.** Data Set 3: Controlled Mixture–DDA–Skyline Relative Protein Concentrations

Protein	Mixture				
	Mix 1	Mix 2	Mix 3	Mix 4	Mix 5
Protein subset1	1	2	4	2	1
Protein subset2	4	1	2	1	2
Protein subset3	2	4	1	2	1

The true changes in protein abundance were unknown. We evaluated the ability of the statistical methods to identify differentially abundant proteins between conditions.

#### Simulated Out-of-Memory Data

To illustrate the performance of *MSstatsBig*, we generated two out-of-memory data sets based on Data set 2: Controlled Mixture–DIA–Spectronaut. The first data set was created by replicating the Controlled Mixture 15 times, and the second was created by replicating it 30 times. The 15 copy data set was 17.3 GB in size, while the 30 copy was 34.6 GB.

### ■ BACKGROUND ON EXISTING STATISTICAL METHODS FOR DIFFERENTIAL ANALYSIS OF MS-BASED PROTEOMICS

#### Statistical Methods with Bioconductor Implementation

In this section we provide additional details on *MSqRob* (implemented in the *msqrob2* package on Bioconductor) and *DEqMS*. These two packages are closest to the scope of *MSstats* in that they (1) are designed for quantitative proteomic experiments with chromatography-based quantification (i.e., DDA, DIA or SRM data acquisitions), (2) have open-source implementations, (3) are compatible with multiple data processing tools, and (4) are easily installed and used via Bioconductor. Figure 1 overviews the functionalities of these packages, and additional details are given below.

**MSqRob.** The *MSqRob*<sup>13,14</sup> analysis workflow is documented in the package vignette. The workflow takes as input manually converted feature level data in a custom format. Once in the required format, the data is input into the *read-QFeatures()* function from the *QFeatures*<sup>24</sup> packages. The vignette suggests multiple preprocessing steps using both functions from the *QFeatures* package and manual implementations. The package includes methods for dealing with missing values, both converting censored values (such as 0) to NA, as well as missing value imputation. The package vignette itself does not mention missing value imputation, however the *impute()* function from *QFeatures* can be used to perform one

of 13 imputation methods. Finally, *MSqRob* utilizes the *normalize()* function from *QFeatures* for normalization, which gives users a wide variety of options to choose from.

After data preprocessing, *MSqRob* includes summarization functionality (via *QFeatures*). The package does not require summarization, and provides the option for differential analysis using the individual protein features or summarized intensities. Multiple summarization strategies are available, including Tukey's median polish,<sup>38</sup> robust regression summarization, or simply taking the mean, median, or sum of the protein features. Finally, the package performs differential analysis using a linear mixed effects model (*msqrob()* function). The users must specify the correct model for their data, and this requires potentially nontrivial statistical expertise. Multiple options exist for estimating the parameters of the models, including ridge penalty, M-estimation, and Empirical Bayes. Once the model is fit, contrast-based analysis is used for group comparison. Finally, the package itself does not include plotting functionalities, although the vignette leverages plotting functions from the *Limma* package for manually created plots.

**DEqMS.** The *DEqMS*<sup>16</sup> analysis workflow is described in the package vignette. The package provides functionality for both label-free and tandem mass tag (TMT) labeled workflows. The package does not include any direct converters for spectral processing tools and requires manual conversion by the user. For data preprocessing, the package includes a normalization function (*equalMedianNormalization()*) which implements median normalization. Beyond normalization, there are no other data preprocessing options implemented in the package, although the vignette does suggest a few manual prepossessing steps. For summarization, the package includes the summarization function *medianSummary()*. The summarization function requires the user to select a reference column, which is used to calculate the relative peptide ratios. In the function example, this reference column is a reference channel in a TMT experiment. There was no example of using this function for label-free experiments.

For modeling, *DEqMS* implements a version of *Limma* which leverages feature counts for variance moderation. The model takes as input run-level summaries for each protein, grouped into an individual column per MS run. The model is implemented as a 4 step workflow. The first 3 steps follow the classic *Limma* modeling workflow. This includes fitting a linear model, contrast based group comparison, and variance moderation. After the *Limma* model is fit, *DEqMS* groups proteins with the same number of features together, and individually moderates the variance of the feature-groupings

	Method	MSqRob	DEqMS	MSstats v2.0	MSstats v4.0
Input	Quantification with Acquisition	DDA, DIA	DDA, DIA	DDA, DIA, SRM (with heavy reference peptides)	DDA, DIA, SRM (with heavy reference peptides)
	Metadata	Information about Run, Condition, Biological Replicate (Technical Replicate if needed)	Information about Run, Condition, Biological Replicate	Information about Run, Condition, Biological Replicate	Information about Run, Condition, Biological Replicate, (Technical Replicate, Fraction if needed)
Pre-processing	Tool-specific converter	N/A - Limitation : Users should prepare data in the required format.	N/A - Limitation : Users should prepare data in the required format.	N/A - Limitation : Users should prepare data in the required format.	To link outputs of data processing tools to MSstats, eight(8) converters preprocess outputs of 8 upstream data processing tools and make the required format of MSstats : 1. [DIAUmpiretoMSstatsFormat] – DIA-Umpire 2. [MaxQtomSstatsFormat] – MaxQuant 3. [OpenMSstoMSstatsFormat] – OpenMS 4. [OpenSWATHtoMSstatsFormat] – OpenSWATH 5. [PDtosMSstatsFormat] – Proteome Discoverer 6. [ProgenesisstoMSstatsFormat] – Progenesis Q 7. [SkylinetomSstatsFormat] – Skyline 8. [SpectronautomSstatsFormat] – Spectronaut 9. [DIANNtoMSstatsFormat] – DIA NN
	Data check-up	N/A	N/A	N/A	All converters apply not only the same preprocessing filtering, but also the tool-specific data manipulation. Here are all filtering strategies: - Filter <ul style="list-style-type: none"><li>- Decoy or iRT proteins</li><li>- Shared peptides</li><li>- Peaks with id confidence score below cutoff</li><li>- Contaminant peptides or peptides including M or oxidation M sequence</li></ul> - Aggregate multiple peaks per feature and sum: maximum peak or sum of peaks - Filter <ul style="list-style-type: none"><li>- Features observed in one run</li><li>- Proteins with one feature</li></ul> - Integrate metadata to corresponding MS runs
Transformation and normalization	Not included in package itself. Recommends QFeatures package functions <ul style="list-style-type: none"><li>- [normalize]</li><li>- [logTransform]</li><li>- [filterFeatures]</li></ul>	[equalMedianNormalization]		[dataProcess] - Log2 or 10 transformation for peak intensities - Four normalization options <ul style="list-style-type: none"><li>- No normalization</li><li>- Equalize median</li><li>- Quantile</li><li>- Using standard proteins or peptides</li></ul>	
Fractionated samples	N/A	N/A	N/A	N/A	Aggregate multiple fractions per sample in [dataProcess]
Statistical model	Missing value imputation	Qfeatures [impute] function. Model is designed to handle missing values	N/A	N/A	Assumes values are missing due to low abundance. Imputes low intensity values using an AFT model - [dataProcess]
	Summarization	Not included in package itself. Recommends [aggregateFeatures] from QFeatures. - Summarization not required	[medianSummary] - Median summarization for each run - Requires reference column	N/A	Robust summarization per runs across features, by TMP - [dataProcess] Additionally, mean-based summarization is available as an option.
	Statistical modeling and inference	[msqrob] - Linear mixed-effects model with Empirical Bayes moderation - User specified model	Follows Limma workflow and includes the [spectraCounteBayes] function to moderate variance based on feature counts	Linear mixed-effects model – [groupComparison] Users select the fixed or random effects	Linear mixed-effects model fit using run-level summarized data – [groupComparison] - Automatically decide and fit the best linear mixed-effects model, based on the recognized experimental design
	Subject quantification	N/A	N/A	Subject-level measurements using final statistical model – [quantification]	Subject-level measurements using run-level summarization – [quantification]
	Power calculation	N/A	N/A	Calculate statistical power based on the median of estimated variances from the fitted model for hypothesis testing – [designSampleSize]	
	Sample Size Calculation	N/A	N/A	Calculate sample size based on the estimated variance from the fitted model for hypothesis testing – [designSampleSize]	
Plotting functionality	N/A	Variance plots grouped by feature counts <ul style="list-style-type: none"><li>- [VarianceBoxplot]</li><li>- [VarianceScatterplot]</li></ul>	[dataProcessPlots] - Profile, QC, and Condition Plots [groupComparisonPlots] - Heatmap, Volcano, and Comparison Plots	[dataProcessPlots] - Profile, QC, and Condition Plots [groupComparisonPlots] - Heatmap, Volcano, and Comparison Plots	

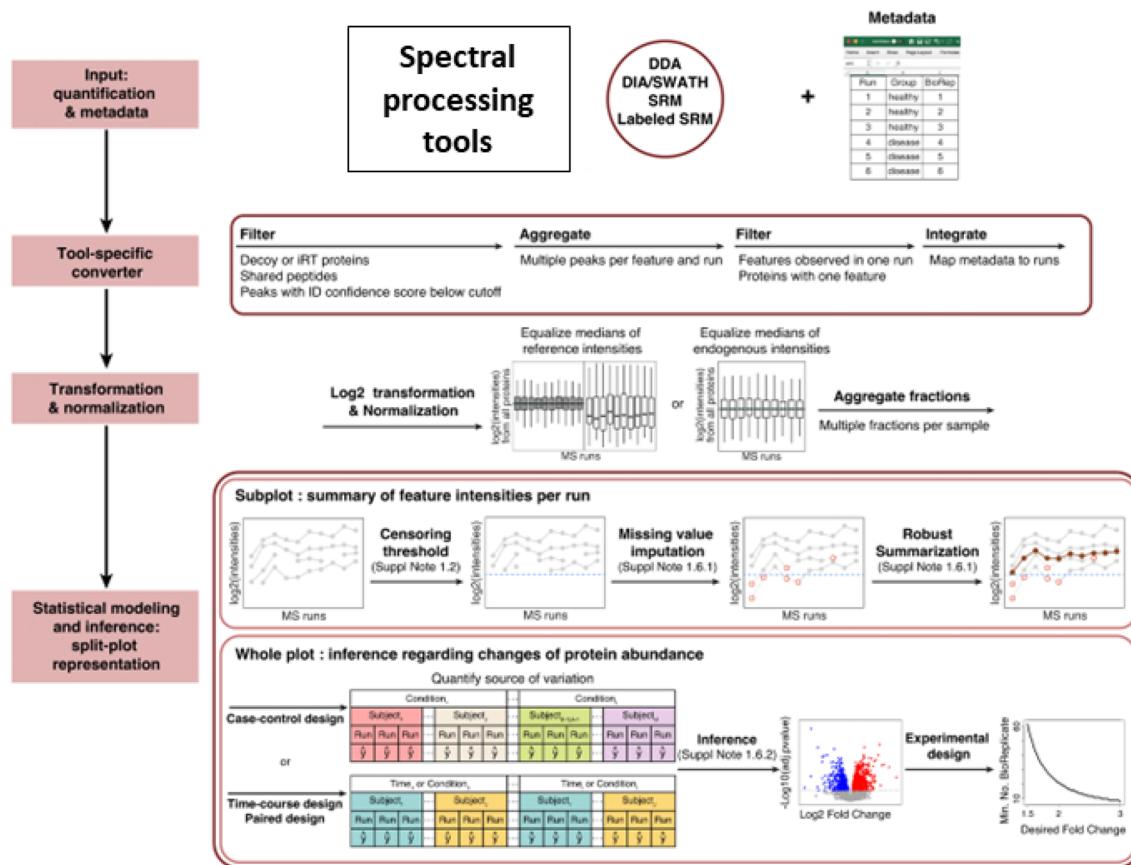
**Figure 1.** Functionality and workflow comparison between MSstats v4.0, MSstats v2.0, MSqRob, and DEqMS. function [function] indicates the corresponding function name in MSstats, MSqRob, or DEqMS. N/A indicates functionality that was not available.

together. In this way, proteins with low features do not have their variances artificially reduced by proteins with high features, and vice versa. DEqMS includes two plotting functions (*VarianceBoxplot()* and *VarianceScatterplot()*) to visualize the results of the feature grouping variance

moderation. Other plots of the results must be manually created.

#### Previously Published Version of MSstats

Figure 1 details the implementation of MSstats v2.0. MSstats v2.0 supported DDA, DIA, and SRM acquisitions, however it



**Figure 2.** MSstats v4.0 workflow and place in the bottom up LC-MS/MS proteomics analysis pipeline. MSstats takes as input the output of spectral processing tools used for identification and quantification and an annotation file matching MS runs to experimental metadata, such as conditions and biological replicates. Tool-specific converters perform data filtering and aggregation. Next the data is transformed onto the  $\log_2$  scale and normalized. Finally, statistical modeling and inference is performed, with optional plotting to view the results.

did not support more complex designs, including those with technical replicates, fractionation, time course, and paired designs. Similarly to other existing methods, MSstats v2.0 did not include any tool-specific converters. The users had to manually convert their data set into the required format, which was a challenge for users without an in-depth knowledge of the MSstats formatting and advanced computational skills. In terms of data preprocessing, MSstats v2.0 included limited options and functionalities. The `dataProcess()` function gave users the option to either  $\log_2$  or  $\log_{10}$  transform peak intensities, and four options for data normalization (equalizing medians across runs, quantile normalization, peptide or protein standard normalization, or not performing normalization). There were no options for missing value imputation, feature filtering, or feature summarization.

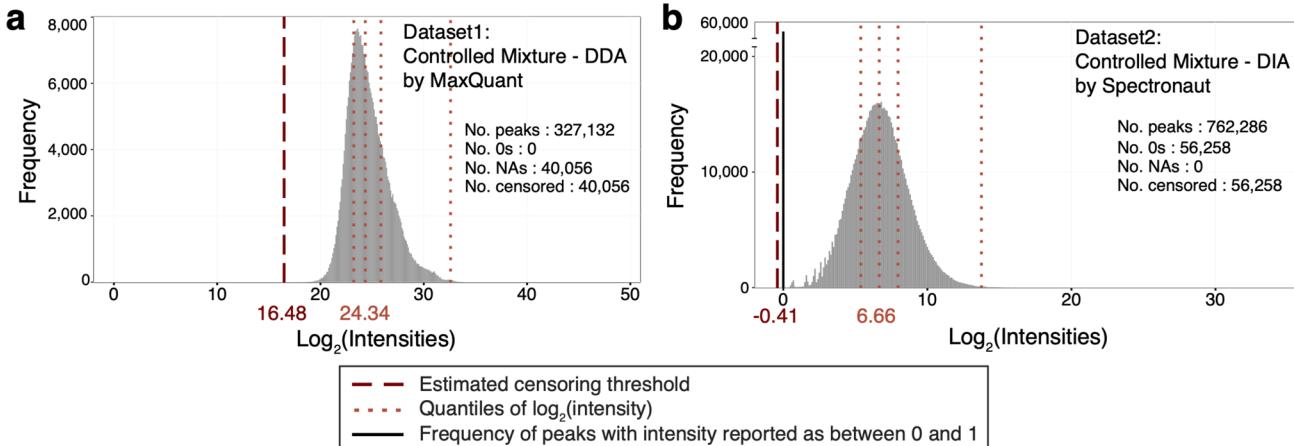
MSstats v2.0's `groupComparison()` function implemented a linear mixed-effects model for differential analysis. The model was feature-based, taking as input quantified features, and specifying a full statistical model at the feature level. This resulted in high computational complexity and issues of numeric stability of the model fit in some cases. Additionally, while MSstats v2.0 implemented a mixed-effects model, it required the user to manually specify the fixed and random effects. This was a challenging step for users with limited statistical background. Finally, after modeling was performed, MSstats v2.0 implemented functionality for future experiment planning, including power and sample size calculations in the `designSampleSize()` function.

## RESULTS

### Statistical Methods and Implementations in MSstats v4.0

**Adaptive Converters Statistically Interpret and Format the Outputs of Data Processing Tools.** MSstats v4.0 takes as input the  $\log_2$ -intensities of peaks obtained by data processing tools, step 1 of Figure 2, (as opposed to ratios of peak intensities between the groups), to facilitate statistical modeling.<sup>39</sup> Since spectral processing tools differ in both analytical strategies and output formats, MSstats v4.0 employs tool-specific converters (Step 2 of Figure 2), which ensure that the output of the tools is interpreted and formatted correctly.

Converters in MSstats v4.0 follow a consistent workflow implemented in the Bioconductor package `MSstatsConvert`.<sup>40</sup> First, raw data from the tools are reshaped into long format, with columns denoting protein ID, peptide sequence, precursor charge, fragment ion and product charge, information about labeling, run ID, and intensity value. Each run is then annotated with labels for condition and biological replicate using the information provided in the accompanying “annotation” file. The “annotation” file maps run names to the experimental design, and must be created manually by the user. Details on how to create this file and example “annotation” files for different experimental designs can be found in Supporting Information Section 1. This minimum set of columns constitutes the MSstats format. Whenever applicable, the user can add information about the ID of technical run and fraction,



**Figure 3.** Distribution of  $\log_2$  transformed and normalized intensities, after normalization. (a) Data set 1: Controlled Mixture–DDA–MaxQuant. (b) Data set 2: Controlled Mixture–DIA–Spectronaut. Median normalization equalized the medians of  $\log_2$ -intensities of all the features between the MS runs. “No. peaks” reports the number of peaks with  $\log_2$  intensity  $> 0$ . “No. 0s” is the number of  $\log_2$  intensity = 0, “No. NAs” is the number of intensities reported as “NA” by the data processing tool. “No. censored” is the number of censored intensities as defined by *MSstats* v4.0. Dotted orange lines indicate the 25, 50, 75, and 99.9th percentiles ( $\hat{q}_{25}$ ,  $\hat{q}_{50}$ ,  $\hat{q}_{75}$ , and  $\hat{q}_{99.9}$ ) of the  $\log_2$  intensities exceeding 0. Dashed dark red lines indicate the censoring threshold, estimated by *MSstats* v4.0.

along with optional columns for use with filtering. This step of the workflow is tool-specific.

The following steps are generic and can be applied to any data set in the described format. They consist of filtering (for example by Q-values or removing contaminants), handling isotopic peaks, removing features with few measurements across runs, removing shared peptides (i.e., peptides assigned to more than one protein by the spectral processing tool), aggregating features with duplicated measurements in a run, optional filtering of proteins identified by a single feature, and adding annotation. These operations are implemented by the *MSstatsPreprocess()* function in *MSstatsConvert* package. The next step of the workflow selects or aggregates fractionated runs and creates balanced design, with one row in the data table for every run for each feature, even if the intensity value is missing. These operations are performed by the *MSstatsBalancedDesign()* function from *MSstatsConvert* package.

The *MSstatsConvert* package enables creation of new converters consistent with the intended *MSstats* workflow. With this package, users can create converters for yet unsupported tools (including custom workflows) using the same logic as in *MSstats* converters with minimal coding, as the only tool-specific step is the reshaping of raw data.

Figure 3 visualizes feature intensities from two data sets in the [Experimental Procedures](#) section: Data Set 1: Controlled Mixture–DDA–MaxQuant and Data Set 2: Controlled Mixture–DIA–Spectronaut. Since spectral processing tools differ in their approaches to reporting missing values, the number of 0 and “NA” values varied across tools. Data set 1 quantified by MaxQuant reported no small intensities and no 0s, but a large amount of “NA” values. Meanwhile Data set 2 quantified by Spectronaut reported a large amount of 0s but no “NA” values. MaxQuant reports missing values as “NA”, while Spectronaut reports them as 0. The converters in *MSstats* v4.0 automatically interpret the missing values, while taking into account the details of the upstream spectral processing tool, and perform time-consuming (and potentially error-prone) data-preprocessing for the user. Beyond missing values, Data set 2 (DIA) had more peaks than Data set 1 (DDA). The distributions of  $\log_2$ -intensities of the peaks, for example the

medians of  $\log_2$ -intensities, were different for both data sets and tools. These differences are not necessarily problematic for peptide and fragment ions that fall in the linear regime of the dynamic range, as similar conclusions regarding changes in protein abundance can be reached from different intensity values. However, they do have important implications for low-abundant analytes.

To maximize the between-tools consistency of the analysis for low-abundant analytes *MSstats* v4.0 learns, separately for each experiment and tool, a threshold for “high-confidence”  $\log_2$ -intensities (Step 4 of Figure 2). The threshold is a tuning parameter, defined as the 0.1th percentile of the  $\log_2$ -intensities in the linear regime of the dynamic range, and estimated as follows. Define  $q_p$  the  $p$ th percentile of all the  $\log_2$ -intensities that exceed 0. In particular, the median is  $q_{50}$ , the 25th percentile is  $q_{25}$ , and the 75th percentile is  $q_{75}$  (dotted lines in Figure 3). *MSstats* v4.0 estimates  $\hat{q}_{0.1}$  as

$$\hat{q}_{0.1} = \hat{q}_{25} - [\hat{q}_{99.9} - \hat{q}_{75}] \quad (1)$$

(dashed lines in Figure 3). The estimation assumes a symmetric distribution of the  $\log_2$ -intensities in the linear regime of the dynamic range, and uses  $\hat{q}_{99.9} - \hat{q}_{75}$  to learn the deviation  $\hat{q}_{25} - \hat{q}_{0.1}$ .

*MSstats* v4.0 views all “NA” and all the values below  $\hat{q}_{0.1}$  as censored, i.e., unreliable or missing due to the low abundance of the underlying analyte, with two exceptions. The first exception is Skyline, which reports low-intensity values, and uses “NA” for intensities of truncated or overlapped peaks. For Skyline, *MSstats* v4.0 views “NA” as not associated with low-abundant analytes, and assumes that they are missing at random. The second exception is “NA” reported by any data processing tool for the intensities of reference peptides, e.g., in SRM experiments that use labeled references, or for any other standards. Since reference peptides are not expected to be low abundant, their intensities are also viewed as missing at random, and kept as “NA” by *MSstats* v4.0. This step is optional in the *MSstats* v4.0 workflow.

**Split-Plot Approach Provides Robust and Accurate Statistical Analysis for Diverse Experimental Designs.** In this section, we discuss statistical modeling and analysis in the

a	Whole plot																			
	Condition <sub>I</sub>							Condition <sub>II</sub>												
	Subject <sub>I</sub>		Subject <sub>II</sub>		...			Subject <sub>J</sub>		...			Subject <sub>(I-1)J+1</sub>		Subject <sub>(I-1)J+2</sub>		...			Subject <sub>U</sub>
Subplot	Run <sub>1</sub>	Run <sub>2</sub>	Run <sub>3</sub>	Run <sub>4</sub>	Run <sub>5</sub>	Run <sub>6</sub>	...	Run <sub>JK</sub>	Run <sub>JK-1</sub>	Run <sub>JK</sub>	...	Run <sub>(I-1)JK+1</sub>	Run <sub>(I-1)JK+2</sub>	Run <sub>(I-1)JK+3</sub>	Run <sub>(I-1)JK+4</sub>	Run <sub>(I-1)JK+5</sub>	...	Run <sub>UK-2</sub>	Run <sub>UK-1</sub>	Run <sub>UK</sub>
Feature <sub>1</sub>	y	y	y	y	y	y	...	y	y	y	...	y	y	y	y	y	...	y	y	y
Feature <sub>2</sub>	y	y	y	y	y	NA	...	y	y	y	...	y	y	y	y	y	...	y	y	y
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Feature <sub>L</sub>	y	Cens	y	Cens	Cens	y	...	Cens	y	y	...	NA	y	y	y	y	...	y	Cens	y

	Whole plot		Subplot
$y_{ijkl} = \mu + \text{Condition}_i + \text{Subject}(\text{Condition})_{j(i)} +$		$\text{Run}_{ijk}$	$+ \text{Feature}_l + \text{Run} \times \text{Feature}_{ijkl}$
	Whole-plot biological variation	Whole-plot technical variation	
where	$\sum_{i=1}^I \text{Condition}_i = 0, \sum_{j=1}^L \text{Feature}_l = 0$		
	$\text{Subject}(\text{Condition})_{j(i)} \stackrel{\text{iid}}{\sim} \mathcal{N}(0, \sigma_{\text{Subject}}^2)$		
	$\text{Run}_{ijk} = \psi_{ijk} \stackrel{\text{iid}}{\sim} \mathcal{N}(0, \sigma_\psi^2)$		
	$\text{Run} \times \text{Feature}_{ijkl} = \epsilon_{ijkl} \stackrel{\text{iid}}{\sim} \mathcal{N}(0, \sigma_\epsilon^2)$		
			Subplot error

**C** Step 1 : Run-level subplot summarization

Impute censored intensities with Accelerated Failure Time model																					
$y_{ijkl} = \mu + Run_{ijk} + Feature_l + \epsilon_{ijkl}$ , where $\sum_{ijk} Run_{ijk} = 0$ , $\sum_l Feature_l = 0$ , $\epsilon_{ijkl} \stackrel{iid}{\sim} \mathcal{N}(0, \sigma_e^2)$																					
	Run <sub>1</sub>	Run <sub>2</sub>	Run <sub>3</sub>	Run <sub>4</sub>	Run <sub>5</sub>	Run <sub>6</sub>	...	Run <sub>JK-2</sub>	Run <sub>JK-1</sub>	Run <sub>JK</sub>	...	Run <sub>(1-1)JK+1</sub>	Run <sub>(1-1)JK+2</sub>	Run <sub>(1-1)JK+3</sub>	Run <sub>(1-1)JK+4</sub>	Run <sub>(1-1)JK+5</sub>	Run <sub>(1-1)JK+6</sub>	...	Run <sub>UK-2</sub>	Run <sub>UK-1</sub>	Run <sub>UK</sub>
Feature <sub>1</sub>	y	y	y	y	y	y	...	y	y	y	...	y	y	y	y	y	y	...	y	y	y
Feature <sub>2</sub>	y	y	y	y	y	NA	...	y	y	y	...	y	y	y	y	y	y	...	y	y	y
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Feature <sub>L</sub>	y	$\hat{y}_{imp}$	y	$\hat{y}_{imp}$	$\hat{y}_{imp}$	y	...	$\hat{y}_{imp}$	y	y	...	NA	y	y	y	y	y	...	y	$\hat{y}_{imp}$	y

↓

TMP : Parameter estimation by robust method

$y_{ijkl} = \mu + Run_{ijk} + Feature_l + \epsilon_{ijkl}$ , where  
 $median_{ijk}(Run_{ijk}) = 0$ ,  $median_l(Feature_l) = 0$ , and  $median_{ijk}(\epsilon_{ijkl}) = median_l(\epsilon_{ijkl}) = 0$

	Run <sub>1</sub>	Run <sub>2</sub>	Run <sub>3</sub>	Run <sub>4</sub>	Run <sub>5</sub>	Run <sub>6</sub>	...	Run <sub>JK-2</sub>	Run <sub>JK-1</sub>	Run <sub>JK</sub>	...	Run <sub>(1-1)JK+1</sub>	Run <sub>(1-1)JK+2</sub>	Run <sub>(1-1)JK+3</sub>	Run <sub>(1-1)JK+4</sub>	Run <sub>(1-1)JK+5</sub>	Run <sub>(1-1)JK+6</sub>	...	Run <sub>UK-2</sub>	Run <sub>UK-1</sub>	Run <sub>UK</sub>
Summarized	$\hat{y}$	$\hat{y}$	$\hat{y}$	$\hat{y}$	$\hat{y}$	$\hat{y}$	...	$\hat{y}$	$\hat{y}$	$\hat{y}$	...	$\hat{y}$	$\hat{y}$	$\hat{y}$	$\hat{y}$	$\hat{y}$	$\hat{y}$	...	$\hat{y}$	$\hat{y}$	$\hat{y}$

#### Step 2 : Whole plot modeling and inference

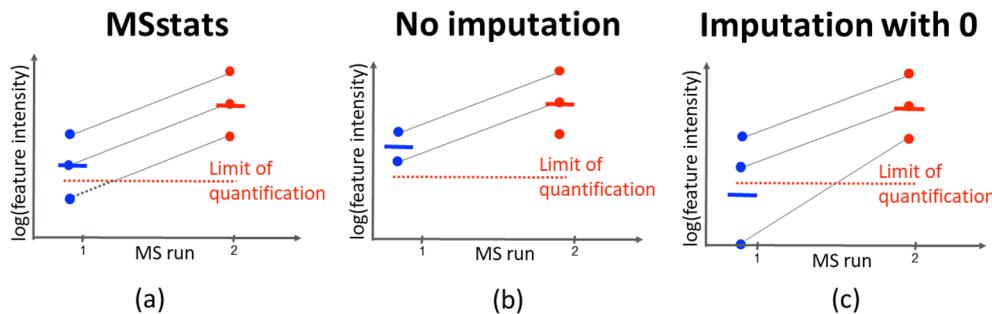
Step 2: Model performance and inference											
$Z_{ijk} = \mu + Condition_i + Subject(Condition)_{j(i)} + \psi_{ijk}$ , where											
$\sum_i Condition_i = 0$ , $Subject(Condition)_{j(i)} \stackrel{iid}{\sim} \mathcal{N}(0, \sigma_{Subject}^2)$ , $\psi_{ijk} \stackrel{iid}{\sim} \mathcal{N}(0, \sigma_\psi^2)$											
	Condition <sub>i</sub>				...			Condition <sub>j</sub>			
	Subject <sub>1</sub>		Subject <sub>2</sub>		...			Subject <sub>1</sub>		Subject <sub>2</sub>	
	Run <sub>1</sub>	Run <sub>2</sub>	Run <sub>3</sub>	Run <sub>4</sub>	Run <sub>5</sub>	Run <sub>6</sub>	...	Run <sub>JK-2</sub>	Run <sub>JK-1</sub>	Run <sub>JK</sub>	...
Summarized	ŷ	ŷ	ŷ	ŷ	ŷ	ŷ	...	ŷ	ŷ	ŷ	...

**Figure 4.** Two-step estimation and inference procedure for the linear mixed-effects model in *MSstats* v4.0, Step 4 of Figure 2. (a) Overview of an example group comparison design. The whole plot and subplot subsections are highlighted. Observed values are indicated as *y*, missing values as NA, and censored values as Cens. (b) The full linear model with the whole plot and subplot sections highlighted. c) The two-step modeling procedure of *MSstats* v4.0. First feature level data are summarized into a single value per run using Tukey's Median Polish. Next a linear mixed effects model is fit using the summarized values.

case of a label-free experiment with group comparison design. A detailed explanation of the overall modeling and analysis workflow, and its extension to experiments with complex designs, can be found in [Supporting Information Section 1](#). This includes extensions to experiments with technical replicates ([Supporting Information Section 1.2](#)), time course designs ([Supporting Information Section 1.3](#)), paired designs ([Supporting Information Section 1.4](#)), and experiments with reference peptide designs ([Supporting Information Section](#)

**1.5).** Finally, a detailed method comparison between *MSstats* v2.0 and v4.0 is available in [Supporting Information Section 2](#).

Figure 4(a) illustrates the structure of the data for one protein in a label-free experiment with a group comparison design and technical replicates. The experiment has  $i = 1, \dots, I$  conditions, e.g., healthy and disease. Each condition is represented by  $j = 1, \dots, J$  subjects, i.e., distinct biological replicates (e.g., patients, mice, etc). Subjects are main experimental units, and in this special case of group



**Figure 5.** Missing value imputation in *MSstats* v4.0. The plot illustrates a single protein with three features measured over two MS runs. The MS runs are shown on the *x*-axis and the log feature intensity is on the *y*-axis. In the first MS run, colored blue, one of the feature points falls below the limit of quantification. (a) *MSstats* imputation. (b) No imputation. (c). Imputation with a small constant.

comparison designs subjects are *nested* within conditions (i.e., each condition is represented by different biological subjects). Furthermore, each subject sample is profiled in  $k = 1, \dots, K$  mass spectrometry *Runs*. In practice the number of biological and technical replicates varies across conditions and subjects.

In each run the protein is represented by  $l = 1, \dots, L$  spectral features. The features are peptide ions in DDA experiments, combinations of peptide ions and transitions in SRM experiments, and combinations of peptide ions and fragments in DIA experiments. For the purposes of this manuscript we do not distinguish transitions or fragments generated by a same or different peptides. Each feature in each run is quantified by its *Intensity*,  $y_{ijkl}$  (defined as peak area, peak height at apex, or any other measure used by a data processing tool), that are  $\log_2$  transformed and normalized. Such layouts are known in statistical literature as split-plot experimental designs.<sup>41</sup> More details can be found in Supporting Information Section 1.1.

Figure 4(b) shows a classical split-plot linear mixed effects model of a label-free group comparison experiment with both biological and technical replicates, reflecting the sources of variation in Figure 4(a). In the special case of a balanced experiment with no missing values, these sources of variation are estimated from the analysis of variance (ANOVA) table (Supporting Information Section 1). Unfortunately, the ANOVA decomposition does not hold in experiments with unbalanced designs, censored, and outlying values. An alternative approach to estimating the sources of variation is *restricted maximum likelihood* (REML). Unfortunately, in models with many terms such as in Figure 4(b), REML-based estimates can be inaccurate, especially when some sources of variation are close to zero.

As a solution, *MSstats* v4.0 separates the estimation procedure into two simpler steps, namely ANOVA-style summarization in the subplot, and REML estimation in the whole plot as shown in Figure 4(c), as shown in Step 4 of Figure 2. At the whole plot level the summarized data structure has fewer irregularities, and the model has fewer terms. This results in a more stable REML-based estimation. Below we detailed the two-step modeling workflow of the estimation procedure.

**Modeling Step 1: Subplot Summarization. Missing Value Imputation.** As an option, *MSstats* v4.0 imputes censored peak intensities (i.e., intensities assumed missing for reasons of low abundance) before subplot summarization. The peak intensities are considered censored if their values are below the cutoff in eq 1 or marked as "NA" (except in the case of Skyline, and in the case of reference peptides and

standards). In presence of censored observations, the observed  $\log_2$ -intensities  $y_{ijkl}$  are viewed as  $y_{ijkl} = \max(y_{ijkl}, m_{ijkl})$ , where  $m_{ijkl}$  is the minimum threshold, i.e. the lowest quantifiable  $\log_2$ -intensity for that feature. Here we assume that the threshold is feature-specific but constant across the runs, i.e.,  $m_{ijkl} = m_b$  and estimate it by the smallest observed  $\log_2$ -intensity of the feature. We define an indicator of whether the peak was detected and quantified as

$$\delta_{ijkl} = \begin{cases} 1 & \text{if } m_{ijkl} \leq y_{ijkl}, \text{ i.e., observed} \\ 0 & \text{if } m_{ijkl} > y_{ijkl}, \text{ i.e., censored} \end{cases} \quad (2)$$

For the imputation, *MSstats* v4.0 relies on an accelerated failure time (AFT) model<sup>42,43</sup>

$$y_{ijkl} = \mu + Run_{ijk} + Feature_l + \sigma_e x_{ijkl}^{iid} \quad \text{where } x_{ijkl} \sim N(0, 1), \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K Run_{ijk} = 0, \sum_{l=1}^L Feature_l = 0 \quad (3)$$

The parameters  $\mu$ ,  $Run_{ijk}$ ,  $Feature_l$ , and  $\sigma$  in eq 3 are estimated by maximizing the product of the likelihoods of the observed and censored peaks

$$L(\mu, Run_{ijk}, Feature_l, \sigma_e | y_{ijkl}) = \prod_{i,j,k,l} f(y_{ijkl})^{\delta_{ijkl}} \times \prod_{i,j,k,l} F(y_{ijkl})^{1-\delta_{ijkl}} \quad (4)$$

where  $f$  is the probability density function and  $F$  is the cumulative density function of the Normal distribution with expected value  $\mu + Run_{ijk} + Feature_l$  and variance  $\sigma_e^2$ . The imputed  $\log_2$ -intensities are

$$\hat{y}_{ijkl,imp} = \hat{\mu} + \widehat{Run}_{ijk} + \widehat{Feature}_l \quad (5)$$

Therefore, feature imputation is only possible for feature  $y_{ijk}$  in  $Run_{ijk}$  if there is an observed value for the feature in another run and if there is an observed value from another feature in  $Run_{ijk}$ . In particular, features are not imputed if the protein is entirely missing in a run.

Figure 5 visualizes the imputation, and contrasts it to other simpler methods. The imputation in eq 4 leverages information from the noncensored values of the feature and the other features in the protein, and relies on the assumption of parallel profiles of features from a same protein between the runs. In contrast, avoiding imputation underestimated the difference

between the MS runs, while imputing with a small constant overestimated the difference.

*MSstats* v4.0's imputation of censored values excels for proteins with many features, whose information can be leveraged by the statistical model. In situations with low feature counts, such as when modeling peptides instead of proteins, there may not be enough information to perform imputation and the imputation may not be reliable. As with all imputation methods, *MSstats* imputation relies on the underlying assumption it is making (that values are missing for reasons of low abundance). The performance of *MSstats* missing value imputation can be assessed by looking at the feature-level data after summarization (obtained using the *dataProcess()* function). We recommended that users inspect the modeling assumptions and the resulting imputed values. If the assumptions are violated, for instance if the values are missing at random, the imputation may be biased, and it is best to omit this option.

**Robust Summarization.** We consider an additive model similar to eq 3

$$y_{ijkl} = \mu + Run_{ijk} + Feature_l + error_{ijkl}, \text{ where} \\ \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K Run_{ijk} = 0, \sum_{l=1}^L Feature_l = 0 \quad (6)$$

where  $\mu$  is the median log<sub>2</sub>-intensity across features and runs, the medians of  $Run_{ijk}$ ,  $Feature_l$ , and  $error_{ijkl}$  are centered at 0, and the errors are independent. The difference from eq 3 is that  $y_{ijkl}$  are now both observed and imputed log<sub>2</sub>-intensities, and the parameters of this model are estimated with robust Tukey's Median Polish (TMP)<sup>38</sup> that accounts for outlying observations. TMP iteratively subtracts medians of each row (*Feature*) and column (*Run*) from  $y_{ijkl}$  until there is no change. The values remaining in the table after these operations are the residuals of the fit. The run-level summaries  $z_{ijk} = \hat{\mu} + \widehat{Run}_{ijk}$  are obtained by subtracting the residuals from  $y_{ijkl}$  and summing the resulting values in the run.

**Modeling Step 2: Whole Plot Modeling and Inference.** At the whole plot level, the linear mixed-effects model is shown in Step 2 of Figure 4(c). In the simple case of a group comparison experiment with biological replicates and no technical replicates, the model is substituted with

$$z_{ij} = \mu + Condition_i + \psi_{ij}, \text{ where } \sum_{i=1}^I Condition_i \\ = 0, \psi_{ij} \stackrel{iid}{\sim} N(0, \sigma_\psi^2) \quad (7)$$

where  $\sigma_\psi^2$  represents a combination of the biological and the technological variation. The same model is fit for controlled mixtures with technical replicates and no biological replicates, in which case  $\sigma_\psi^2$  is the technological variation. Extensions to complex designs are in (Supporting Information Section 1).

**MSstats v4.0 Provides a Flexible Framework for Comparing Conditions.** *MSstats* v4.0 implements a flexible framework for model-based pairwise comparisons between conditions, as well as any linear combinations of conditions such as averaging of multiple groups. To compare the conditions, we first estimate their expected values. The expected value for condition  $i$  is defined as  $\mu_i = \mu + Condition_i$ , where  $\mu$  and  $Condition_i$  are parameters of the model in Eq. 7. The parameters are estimated from the experimental data using

restricted maximum likelihood (REML) (Supplementary Sec. 2.1).

Next we spell out the comparison of interest, which involves two or more expected values.<sup>44,45</sup> Eq. 8 defines a linear combination  $l$  of the expected values  $\mu_i$ .

$$L = \sum_{i=1}^r c_i \mu_i \text{ where } \sum_{i=1}^r c_i = 0 \quad (8)$$

For example, a pairwise comparison between the expected values of Conditions 1 and 2 is expressed with coefficients  $C = (1, -1, 0, 0)$ . As another example, a comparison between the average of the expected values of Conditions 1 and 2 versus the average of the expected values of Conditions 3 and 4 is expressed with coefficients  $C = (1/2, 1/2, -1/2, -1/2)$ . The special cases of linear combinations, where the coefficients  $c_i$  sum to zero, are called contrasts. *MSstats* v4.0 can estimate any linear combination of the expected values, not just a contrast. To test the null hypothesis  $H_0: L = 0$  against the alternative  $H_a: L \neq 0$ , the estimate of the linear combination  $\hat{L}$  and its standard error are combined into a *t*-statistic

$$t = \frac{\hat{L}}{SE(\hat{L})} \quad (9)$$

which is then compared against the student distribution with appropriate degrees of freedom to determine the *p*-value.<sup>44</sup>

We now illustrate this model-based inference in the special case of an experiment with  $I$  conditions and  $J$  biological replicates per condition in a balanced design, the model Figure 4(c), and a pairwise comparison between Conditions 1 and 2. The model expresses the expected values  $\mu_1, \mu_2, \mu_3, \mu_4$  of the four conditions. In the special case of balanced designs and simpler ANOVA-based estimation, the model-based estimate of an expected values matches its sample averages, i.e.,  $\hat{\mu}_i = \bar{z}_i$ .<sup>44</sup>

Since the contrast coefficients of interest are  $C = (1, -1, 0, 0)$ ,  $\hat{L} = \bar{z}_1 - \bar{z}_2$  is the model-based estimate of the log<sub>2</sub>-fold change between the conditions. In this special case, the standard error of  $\hat{L}$  is estimated as

$$SE(\hat{L}) = \sqrt{\frac{\hat{\sigma}_\psi^2}{IJ}} \quad (10)$$

where  $\hat{\sigma}_\psi^2$  is the ANOVA-based estimate of the error variance in Figure 4(c). Finally, in this special case, the *t*-statistic in eq 9 is compared against the Student distribution with  $I(J - 1)$  degrees of freedom to determine the *p*-value.

## Implementation

**New Coding Strategies Improve Memory and Time Computational Complexity.** *MSstats* v2.0 and v3.0 exclusively used base R<sup>46</sup> verbs with parts of code written with tidyverse.<sup>47</sup> *MSstats* v4.0 implements all operations on tabular data with data.table<sup>48</sup> verbs, which ensures strong performance for large data sets that fit in memory.<sup>49</sup> In particular, whenever possible, *MSstats* v4.0 relies on grouped tabular operations rather than iterative procedures. Moreover, parts of the code, including summarization and postprocessing of fitted statistical models, were rewritten with C++ using the Rcpp interface<sup>50</sup> to improve execution time. Performance of logging was improved by the addition of log4r backend,<sup>51</sup> while performance of parameter validity checks relies on the checkmate package.<sup>52</sup>

*MSstats* v4.0 is highly modularized. Code related to preprocessing raw outputs of signal processing tools was

**Table 5.** Use of Computational Resources by MSstats v2.0, v3.0, and v4.0 on Each Data Set<sup>a</sup>

Data Set	File Size	Processing Time [s]			Mem. Total [MB]			Mem. Sum Max [MB]		
		v2	v3	v4	v2	v3	v4	v2	v3	v4
1: Controlled Mixture-DDA–MaxQuant	200 MB	831.56s	231.77s	164.38s	16,583.60	40,315.09	13,542.00	109.26	147.24	20.21
2: Controlled Mixture–DIA–Spectronaut	1.04 GB	3,895.67s	1,428.82s	963.18s	20,726.46	17,920.83	7,090.18	291.09	243.60	13.76
3: Controlled Mixture–DDA–Skyline	63 MB	243.52s	97.10s	42.35s	20,341.49	15,176.89	6,042.41	100.64	111.61	20.54
4: Mouse–DDA–MaxQuant	257 MB	1,076.97s	636.60s	338.98s	16,948.49	24,507.43	10,665.79	297.54	335.40	16.17
5: <i>S. cerevisiae</i> –DIA–Skyline	315 MB	4,411.80s	2,258.34s	1,897.62s	85,845.08	26,531.06	18,125.68	412.43	234.18	29.83

<sup>a</sup>“Processing time” is the time in seconds the data sets took to run. MSstats versions 2.0, 3.0, and 4.0 were each measured 5 times per data set, and the mean processing time was reported. Because there was no converter in version 2.0, we applied data conversion to the experiments before measuring processing time. Measurement was done using the microbenchmark package.<sup>53</sup> “Mem. total [MB]” is the total memory allocation for summarization and group comparison across controlled mixtures and biological experiments. “sum of maximum allocation” is the sum of maximum allocation for summarization and group comparison across all data sets. MSstats v2.0, v3.0, and v4.0 were run once for each data set and memory allocation was reported for *dataProcess* and *groupComparison* step. Here, we report total allocation for both steps of the workflow and sum of maximum allocation in both steps. Measurement was done using the lineprof package.<sup>54</sup>

moved to the *MSstatsConvert* package, and converters in *MSstats* rely on functions from that package. This backend can be reused to ensure high performance of new and custom converters. Moreover, updated *MSstats* provides a set of alternative functions to the existing *dataProcess()* and *groupComparison()*. With these functions, each step of the *MSstats* workflow is implemented in a separate function (such as *MSstatsNormalize()* for normalization, *MSstatsSummarize()* for summarization) which serve as backend for established functions, but can be used independently. This alternative workflow improves both time and memory management, as follows. First, it is possible to reduce the amount of repeated operations when replicating the analysis. For example, new analysis may start with normalized data without the need to rerun operations done by the *dataProcess()* function before normalization. Second, this workflow was designed with parallel execution in mind. Summarization (particularly with imputation) and model fitting are the most time- and memory-consuming parts of *MSstats* workflow. Both operations are done separately for each protein in a data set. Thus, operations that use all data are implemented with *data.table* verbs for maximum performance, while per-protein operations are provided in self-contained functions (such as *MSstatsSummarizeSingleTMR()*) that can be run in parallel. Due to multiplicity of architectures for parallel computations and differences in parallelization options across operating systems, we do not provide parallel versions of *MSstats* function. Instead, the current version of the package provides tools which allow users to take advantage of their own parallel infrastructure. Moreover, these building blocks of *MSstats* workflow are easily reusable across packages that implement data analysis methods for different experimental workflows.

**MSstatsBig Enables the Analysis of out of Memory Data Sets.** While the *data.table* backend and other updates in *MSstats* v4.0 ensure strong performance for in-memory data sets, handling data larger than memory is a challenge for the *MSstats* workflow. Different steps of the workflow require aggregation across different variables (such as Run and Protein), which makes simple batch-processing and complete reuse of existing code infeasible. Thus, motivated by large data sets, such as the ones generated with Spectronaut, we created *MSstatsBig* currently available on the *MSstats* GitHub page <https://github.com/Vitek-Lab/MSstatsBig>. The package implements a restricted (in terms of freedom of parameter

choice) version of *MSstats* workflow, assuming that feature-level input data do not fit in the memory but preprocessed data in *MSstats* format do. Since the *MSstats* preprocessing removes low-quality and redundant PSMs, and aggregates measurements repeated for a given Run and Feature, there are many opportunities for reducing data set. Further reduction in data size can be achieved by reducing redundancy in Run annotation or changing Protein/Feature labels.

The biggest challenge in processing out-of-memory data sets are grouped operations, as grouping does not necessarily correspond to physical partitioning of the data. Thus, we use SparklyR package to connect to Apache Spark database to process data. This allowed us to process data set as big as 200 GB.

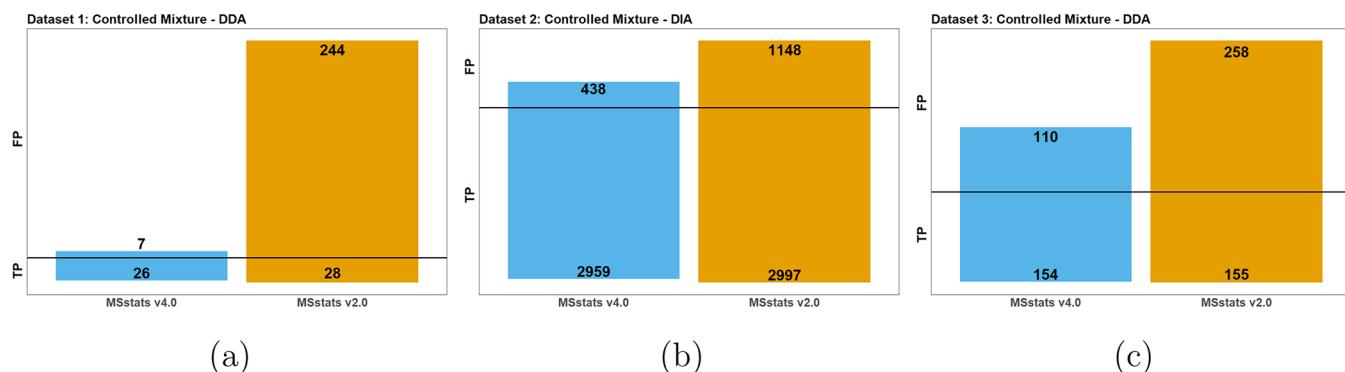
## EVALUATION

### Evaluation Criteria

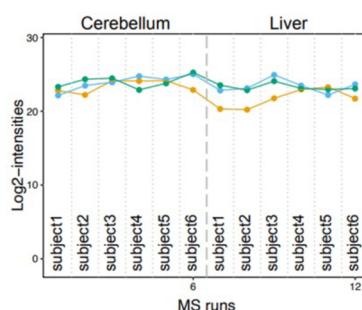
We compared the performance of *MSstats* v4.0, v3.0, and v2.0 in terms of computational time. To test computation speed, we first converted the data into the required format, using the corresponding converter for v4.0 and v3.0, and manually for v2.0, and ran the full statistical workflow for each version, using the *dataProcess()* and *groupComparison()* functions. The functions were each run 5 times for each version and each data set and the mean run time for the workflow was recorded.

We compared the statistical results of *MSstats* v2.0, *MSstats* v4.0, *MSqRob*, and *DEqMS* on the controlled mixtures and biological experiments. For *MSstats* v4.0, the raw files were input into the corresponding converters, and the default parameters were used for the *dataProcess()* and *groupComparison()* functions. For *MSqRob* the data were first manually converted into the required format, and then the workflow noted in the *MSqRob vignette* was followed with default parameters selected. For *DEqMS* the data were manually converted and the *vignette* workflow was generally followed. However, because the summarization function requires TMT labeled data, we manually summarized the feature level data using the log sum of features.

For statistical results, the versions were evaluated on the controlled mixtures (data sets 1, 2, and 3) in terms of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) differentially abundant proteins while controlling the FDR at 5%. The true positives were defined as



**Figure 6.** Controlled data sets 1, 2, and 3: Statistical analyses by *MSstats* v2.0 and v4.0. The true positives (TP) and false positives (FP) reported by each version. The FP are shown on the top, and the TP are shown at the bottom of each bar, with the black line showing 0. In all data sets, the TP numbers were similar between versions, while the FP were much lower when using v4.0. (a) Data set 1: Controlled Mixture–DDA–MaxQuant. (b) Data set 2: Controlled Mixture–DIA–Spectronaut. (c) Data set 3: Controlled Mixture–DDA–Skyline.



Model	Variance estimates		REML-based inference			
	ANOVA	Variance	Comparison: Liver-Cerebellum			
			Estimated log2FC	SE	DF	Pvalue
(a) Linear MIXED with FULL model v2.0	$\hat{\sigma}_{S2}^2=0.1738$ $\hat{\sigma}_{CS2}^2=-0.1653$ $\hat{\sigma}_2^2=0.7766$	$\hat{\sigma}_{S2}^2=0.1903$ $\hat{\sigma}_{CS2}^2<0.0001$ $\hat{\sigma}_2^2=0.6774$	-0.9984	0.2744	25	0.0012
(b) Linear MIXED with FULL model	$\hat{\sigma}_{S3}^2=0.1693$ $\hat{\sigma}_{CS3}^2=-0.1737$ $\hat{\sigma}_3^2=0.8017$	$\hat{\sigma}_{S3}^2=0.1857$ $\hat{\sigma}_{CS3}^2<0.0001$ $\hat{\sigma}_3^2=0.7052$	-0.9984	0.2799	27	0.0014
(c) Whole plot model v4.0	$\hat{\sigma}_r^2=0.2565$ $\hat{\sigma}_\zeta^2=0.0936$	$\hat{\sigma}_r^2=0.2565$ $\hat{\sigma}_\zeta^2=0.0936$	-0.9984	0.1766	5	0.0024

**Figure 7.** Data set 4: Mouse–DDA - MaxQuant, Protein O08547 In this example, the estimate  $\hat{\sigma}_{CS}^2$  of the Condition  $\times$  Subject was zero, and this severely undermined the estimation of degrees of freedom in *MSstats* v2.0 and in the full model v4.0. In contrast, the whole plot model in *MSstats* v4.0 was simpler, had fewer variance components, and the variance components were away from zero. This produced a smaller estimate of standard error(SE) of the pairwise comparison and the on-target degrees of freedom. After adjusting for multiple comparisons, such difference can affect the decision of differential abundance.

spike-in proteins with known changes in abundance. The true negatives were defined as spike-in proteins with no changes in abundance. Additionally, the positive predictive value (PPV)/empirical False Discovery Rate (eFDR) was calculated as described in eq 11.

$$1 - \text{PPV} = \frac{FP}{TP + FP}$$

$$= \text{empirical False Discovery Rate (eFDR)} \quad (11)$$

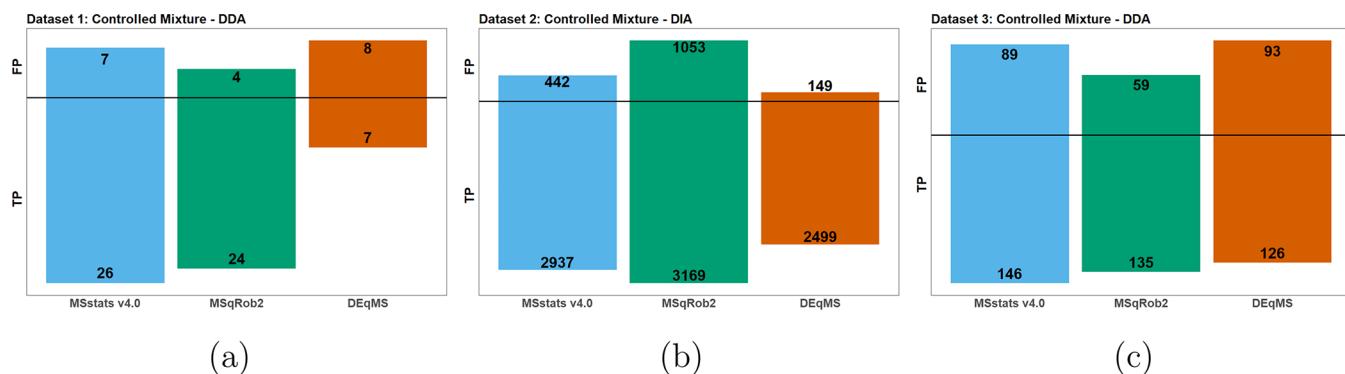
For the biological experiments (data sets 4 and 5) without known ground truth, we compared the lists of differentially abundant proteins produced by each version.

#### Evaluation

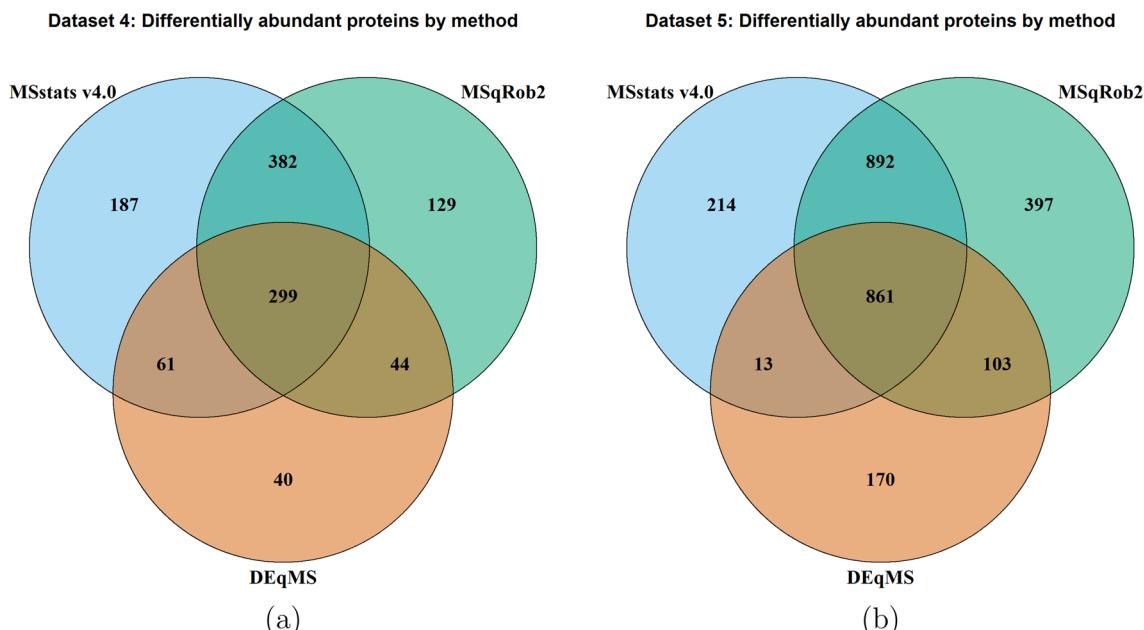
**MSstats v4.0 Reduced Processing Time and Memory Consumption Across All Data Sets.** Table 5 reports the computational resources used for *MSstats* versions 2.0, 3.0, and

4.0. *MSstats* v4.0 drastically reduced the mean processing time across all data sets (an average decrease of 74.66%). The processing time was dramatically reduced due to both changes in modeling strategy and refactoring of the code base. Memory allocation reports the line profiling for the complete statistical workflow (summarization and group comparison). Across all data sets both the total and maximum allocated memory decreased with version 4.0. The maximum allocation was stable despite different characteristic of the data (including data acquisition mode). The decrease in allocation ranged from about 10% to 65%, depending on a data set. This change was mainly due to technological and code improvements from v3 to v4. Table 5 shows the reduction in memory usage was inconsistent from v2.0 to v3.0, however there was a drastic reduction from v3.0 to v4.0.

**Statistical Methods in MSstats v4.0 Improved eFDR in Three Controlled Mixtures.** Figure 6 reports the results of



**Figure 8.** Controlled data sets 1, 2, and 3: Statistical analyses by *MSstats* v4.0, *MSqRob*, and *DEqMS*. The true positives (TP, below the horizontal line) and false positives (FP, above the horizontal line) reported by each tool. (a) Data set 1: Controlled Mixture–DDA–MaxQuant. The results between *MSstats* and *MSqRob* were similar, with *MSstats* being a more sensitive, detecting more true positives and false positives, while *DEqMS* lagging behind. (b) Data set 2: Controlled Mixture–DIA–Spectronaut. *DEqMS* was less sensitive than the other methods, missing many true positives but reporting few false positive. *MSstats* reported fewer false positives than *MSqRob* but also did not identify as many true positives. (c) Data set 3: Controlled Mixture–DDA–Skyline. All three methods were comparable. *MSstats* reported the most TP but also reported more FP than *MSqRob*.



**Figure 9.** Data sets 4 and 5: Statistical analyses by *MSstats* v4.0, *MSqRob*, and *DEqMS*. (a) Data set 4: Mouse–DDA–MaxQuant. *MSstats* and *MSqRob* show a large overlap of differentially abundant proteins, with *MSstats* reporting more differentially abundant proteins in total. *DEqMS* produced the least differentially abundant proteins, with most of those reported also reported by the other methods. (b) Data set 5: *S. cerevisiae*–DIA–Skyline. *MSstats* and *MSqRob* show a large overlap of differentially abundant proteins, however *MSqRob* reported more differentially abundant proteins in total than *MSstats*.

applying *MSstats* v2.0 and v4.0 to the three controlled mixtures. In all the data sets the number of TP were nearly equal, however the number of FP markedly decreased. Subsequently the eFDR in Data set 1 decreased from 89.7% in v2.0 to 20.6% in v4.0, the eFDR in Data set 2 decreased from 27.7% to 12.9%, and the eFDR in Data set 3 decreased from 62.45% to 41.67%. *MSstats* v2.0 overestimated the number of positives in all cases, resulting in a much higher eFDR.

***MSstats* v4.0 Improved Standard Error and Degrees of Freedom Calculation in Biological Experiment.** The difference in performance of the versions of *MSstats* was mainly caused by differences in statistical modeling and model fitting. *MSstats* v2.0 reported overly high degrees of freedom at the pairwise comparison stage, and as the result was overly

sensitive, and detected differentially abundant proteins even when there was no true change between conditions. The whole plot model used by *MSstats* v4.0 summarized all features into a single value per MS run prior to fitting the final model. This facilitated the estimation, and resulted in a more appropriate estimation of variability and degrees of freedom.

**Figure 7** illustrates a protein in Data set 4: Mouse - DDA - MaxQuant where REML estimation had a small impact on pairwise comparisons between conditions. The protein had a balanced design, and the models in *MSstats* v2.0 and v4.0 had identical theoretical ANOVA-based inference. In this example, the estimate  $\hat{\sigma}_{CS}^2$  of the Condition  $\times$  Subject was zero, and this severely undermined the estimation of degrees of freedom in *MSstats* v2.0 and in the full model v4.0. In contrast, the whole plot model in *MSstats* v4.0 was simpler, had fewer variance

components, and the variance components were away from zero. This produced a smaller estimate of standard error of the pairwise comparison, and the on-target degrees of freedom. After adjusting for multiple comparisons, such difference can affect the decision of differential abundance. x

**MSstats v4.0 Better Traded off False Positives and False Negatives and Had an Easier Use than MSqRob and DEqMS in Controlled Mixtures.** Figure 8 reports the results of the controlled mixtures using *MSstats* v4.0, *MSqRob*, and *DEqMS* in terms of TP and FP. In all data sets, *DEqMS* exhibited the poorest performance, mostly due to lack of dedicated data preprocessing functionality. Both *MSstats* v4.0 and *MSqRob* implement multiple data preprocessing steps directly into their workflows, and have dedicated protein summarization functions. In contrast, *DEqMS* required manual implementation of the majority of these steps. As the result, the feature-level measurements included low quality features, and the summarization was not as robust as *MSstats* and *MSqRob*'s summarization. This in turn produced worse statistical results.

*MSstats* v4.0 and *MSqRob* performed similarly on all controlled mixtures. In data sets 1 and 3 *MSstats* reported a higher number of TP and FP than *MSqRob*, whereas in data set 2 *MSqRob* reported more TP and FP. While the statistical results were similar, the application of *MSstats* to the data sets was much more straightforward. The output of MaxQuant and Spectronaut could be directly converted into *MSstats* format using the dedicated converters, whereas the data for *MSqRob* had to be manually converted. The data preprocessing and summarization for *MSstats* was applied using one function (*dataProcess()*) with all options laid out as function parameters. In comparison, *MSqRob* required each preprocessing step to be performed separately, with some processing steps, such as filtering nonzero intensities, requiring manual implementation. Additionally, in data set 1 *MSqRob* reported uninformative errors due to features entirely missing in some conditions. These features had to be manually filtered out in order for the summarization function to complete. In contrast, *MSstats* automatically took care of these features and required no extra work or debugging by the user.

**MSstats v4.0 Identified New Differentially Abundant Proteins As Compared to MSqRob and DEqMS in Biological Experiments.** Figure 9 shows the number of differentially abundant proteins reported by each method and their overlap for biological experiments in this manuscript. Unlike the controlled mixtures, the experiments contain biological variation, and represent repeated measures design. Consistently with the previous section, *MSstats* v4.0 and *MSqRob* reported the highest number of the differentially abundant proteins. In Data set 4, *MSstats* reported 929 differentially abundant proteins, while *MSqRob* reported 854. In Data set 5 *MSstats* reported 1980 differentially abundant proteins and *MSqRob* reported 2253. These two methods had a large overlap in both data sets. In Data set 4 there was an overlap of 681 differentially abundant proteins, 51% of all differential proteins being reported by any method. In Data set 5, there was an overlap of 1753, 66% of all differential proteins. While the results between these methods were similar, *MSstats* was much easier to apply. It required only 8 lines of code, calling data converter and wrapping all the data preprocessing functionality. *MSqRob* on the other hand was much more complicated to apply, taking 60 lines of code to complete the analysis. Additionally, since the experiments had repeated

measures design the models in *MSqRob* had to be entered manually, which can be challenging for users with limited statistical expertise.

*DEqMS* reported the least number of differential proteins, only 444 in Data set 4 and 1147 in Data set 5 (less than half of those reported by *MSstats* and *MSqRob*). The majority of proteins reported by *DEqMS* were also reported by the other methods. As described previously, this is most likely due to the lack of data preprocessing by the *DEqMS* workflow.

#### MSstatsBig Enabled Analysis of out-of-Memory Data.

We preprocessed both out of memory simulated data sets using *MSstatsBig* functionalities: *cleanBigSpectronaut()* function that performs initial data reduction and saves intermediate result which can be processed with standard *MSstatsConvert* tools or using the *BigSpectronauttoMSstatsFormat()* function from *MSstatsBig*. Execution time was measured using *system.time* function from base R. Table 6 summarizes the results.

**Table 6. Use of Computational Resources in out-of-Memory Analyses with *MSstats* v4.0<sup>a</sup>**

Number of Copies	File Size	Elapsed Time [s]		
		Processing	Cleaning	Total
15	17.3 GB	210.75	439.75	650.50
30	34.6 GB	380.54	1351.24	1731.78

<sup>a</sup>Processing time for two data sets created by merging copies of the Controlled Mixture–DIA. Computations were done a laptop with 16 GB RAM and 2.3 GHz CPU with 4 cores.

Processing step refers to the *cleanBigSpectronaut()* function while Cleaning step is implemented by *BigSpectronauttoMSstatsFormat()*. We used the Arrow<sup>SS</sup> package backend for this benchmark. *MSstatsBig* package also supports cleaning data with dplyr and sparklyr backends.

## DISCUSSION

The manuscript describes a substantial update to the core package of the *MSstats* Bioconductor family of packages. The updates improved the usability of the implementation, the statistical methodology, and the computational resource requirements. The package now includes converters which directly integrate it with the output of multiple spectral processing tools used for analyte identification and quantification. The converters greatly increase the usability of the package, making it much easier for users to begin their analysis, without having to go through a lengthy manual conversion. The new statistical methods are applicable to a broader range of experimental designs, and improved the accuracy of the statistical inference as compared to the previous version. The new implementation drastically improved both the computational speed and memory usage of the package. The users can now process experiments of size that was out of reach for older versions of *MSstats*.

In comparisons with *MSqRob* and *DEqMS*, *MSstats* v4.0 performed favorably, while being easier to use and implement. The existing methods relied on manual implementation by the user for upstream data processing, and focused mainly on the final statistical model. In contrast, *MSstats* v4.0 encompassed the entire data analysis pipeline, from the output of spectral processing tools, to upstream data processing, and statistical analysis. It included a wide range of additional analysis options, such as plotting functionality to visualize the results of data analysis, without depending on other tools. The broader scope

of MSstats v4.0 eased the analysis of all data sets in the manuscript, requiring only a few lines of code to go from raw data to group comparison and high quality statistical analysis.

Overall, we believe that MSstats v4.0 is a strong contribution to reproducible mass spectrometry-based proteomic research, taking a user-first approach to MS experimental analysis and producing accurate statistical results in a straightforward, easy to apply, implementation.

## ■ ASSOCIATED CONTENT

### SI Supporting Information

The Supporting Information is available free of charge at <https://pubs.acs.org/doi/10.1021/acs.jproteome.2c00834>.

Data structure in a quantitative proteomic experiment; example group comparison design; full linear mixed-effects model for one protein; analysis of variance for the model; ANOVA-based pairwise comparison of conditions; example time course design; full linear mixed-effects model for one protein; analysis of variance for the model; ANOVA-based pairwise comparison of conditions; example paired experimental design; data structure for one protein, from an experiment with labeled reference peptides; full linear mixed-effects model for one protein; ANOVA-based inference for pairwise comparisons in balanced label-free designs; ANOVA-based inference for pairwise comparisons in time course or paired label-free designs; ANOVA-based inference for pairwise comparisons in balanced label-free designs with technical replicates only; example annotation file for the group comparison design; example annotation file for the time course design; and example annotation file for the paired design ([PDF](#))

## ■ AUTHOR INFORMATION

### Corresponding Authors

**Meena Choi** — *Khoury College of Computer Science, Northeastern University, Boston, Massachusetts 02115, United States*; Present Address: Microchemistry, Proteomics and Lipidomics, Genentech, South San Francisco, CA, USA 94080; Email: [choi.meena@gene.com](mailto:choi.meena@gene.com)

**Olga Vitek** — *Khoury College of Computer Science, Northeastern University, Boston, Massachusetts 02115, United States*; Email: [o.vitek@northeastern.edu](mailto:o.vitek@northeastern.edu)

### Authors

**Devon Kohler** — *Khoury College of Computer Science, Northeastern University, Boston, Massachusetts 02115, United States*; [orcid.org/0000-0001-7301-2596](#)

**Mateusz Staniak** — *University of Wrocław, Wrocław 50-137, Poland*

**Tsung-Heng Tsai** — *Khoury College of Computer Science, Northeastern University, Boston, Massachusetts 02115, United States*; [orcid.org/0000-0001-5211-2985](#)

**Ting Huang** — *Khoury College of Computer Science, Northeastern University, Boston, Massachusetts 02115, United States*

**Nicholas Shulman** — *Department of Genome Sciences, University of Washington, Seattle, Washington 98195, United States*

**Oliver M. Bernhardt** — *Biognosys, Zürich 8952, Switzerland*

**Brendan X. MacLean** — *Department of Genome Sciences, University of Washington, Seattle, Washington 98195, United States*

**Alexey I. Nesvizhskii** — *Department of Pathology and Computational Medicine & Bioinformatics, University of Michigan, Ann Arbor, Michigan 48109, United States*  
**Lukas Reiter** — *Biognosys, Zürich 8952, Switzerland*; [orcid.org/0000-0001-5751-3139](#)

**Eduard Sabido** — *Center for Genomic Regulation, Barcelona Institute of Science and Technology, Barcelona 08003, Spain; Universitat Pompeu Fabra, Barcelona 08002, Spain*

Complete contact information is available at:  
<https://pubs.acs.org/10.1021/acs.jproteome.2c00834>

### Notes

The authors declare no competing financial interest.

## ■ ACKNOWLEDGMENTS

The authors would like to acknowledge David R. Spicariach and Sumedh Sankhe from OMNI Translational Medicine, Genentech, for their contribution to developing and testing the MSstatsBig package. This work was supported in part by NSF DBI-1759736 and the Chan-Zuckerberg Essential Open-Source Software Award to O.V, R24 GM141156 the Seattle Quant grant to N.S. and B.M., and grant 5R01GM094231 to A.N. The CRG/UPF Proteomics Unit is part of the Spanish Infrastructure for Omics Technologies (ICTS OmicsTech). We acknowledge “Secretaria d’Universitats i Recerca del Departament d’Economia i Coneixement de la Generalitat de Catalunya” (2017SGR595), the Spanish Ministry of Science and Innovation (PID2020-115092GB-I00), and the European Union’s Horizon 2020 research and innovation program (GA 823839; EPIC-XS).

## ■ REFERENCES

- (1) Rozanova, S.; Barkovits, K.; Nikолов, M.; Schmidt, C.; Urlaub, H.; Marcus, K. In *Quantitative Methods in Proteomics*, Marcus, K., Eisenacher, M., Sitek, B., Eds.; Springer US: New York, NY, 2021; p 85.
- (2) Zhang, F.; Ge, W.; Ruan, G.; Cai, X.; Guo, T. Data-Independent Acquisition Mass Spectrometry-Based Proteomics and Software Tools: A Glimpse in 2020. *Proteomics* **2020**, *20*, 1900276.
- (3) Vidova, V.; Spacil, Z. A review on mass spectrometry-based quantitative proteomics: Targeted and data independent acquisition. *Anal. Chim. Acta* **2017**, *964*, 7.
- (4) Gillet, L. C.; Leitner, A.; Aebersold, R. Mass Spectrometry Applied to Bottom-Up Proteomics: Entering the High-Throughput Era for Hypothesis Testing. *Annual Review of Analytical Chemistry* **2016**, *9*, 449.
- (5) Tyanova, S.; Temu, T.; Cox, J. The MaxQuant computational platform for mass spectrometry-based shotgun proteomics. *Nat. Protoc.* **2016**, *11*, 2301.
- (6) 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. Skyline: an open source document editor for creating and analyzing targeted proteomics experiments. *Bioinformatics* **2010**, *26*, 966.
- (7) 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. Extending the limits of quantitative proteome profiling with Data-Independent Acquisition and application to acetaminophen-treated three-dimensional liver microtissues. *Molecular & Cellular Proteomics* **2015**, *14*, 1400–1410.

- (8) Nahnsen, S.; Bielow, C.; Reinert, K.; Kohlbacher, O. Tools for label-free peptide quantification. *Molecular & Cellular Proteomics* **2013**, *12*, S49.
- (9) Choi, M.; Eren-Dogu, Z. F.; Colangelo, C.; Cottrell, J.; Hoopmann, M. R.; Kapp, E. A.; Kim, S.; Lam, H.; Neubert, T. A.; Palmlad, M.; Phinney, B. S.; Weintraub, S. T.; MacLean, B.; Vitek, O. ABRF Proteome Informatics Research Group (iPRG) 2015 Study: Detection of Differentially Abundant Proteins in Label-Free Quantitative LC–MS/MS Experiments. *J. Proteome Res.* **2017**, *16*, 945–957.
- (10) Deutsch, E. W.; et al. The ProteomeXchange consortium in 2020: enabling ‘big data’ approaches in proteomics. *Nucleic Acids Res.* **2019**, *48*, D1145–D1152.
- (11) Nefedov, A. V.; Gilski, M. J.; Sadygov, R. G. Bioinformatics tools for mass spectrometry-based high-throughput quantitative proteomics platforms. *Current Proteomics* **2011**, *8*, 125–137.
- (12) Bukhman, Y. V.; Dharsee, M.; Ewing, R.; Chu, P.; Topaloglou, T.; Le Bihan, T.; Goh, T.; Duewel, H.; Stewart, I. I.; Wisniewski, J. R.; Ng, N. Design and analysis of quantitative differential proteomics investigations using LC–MS technology. *Journal of Bioinformatics and Computational Biology* **2008**, *6*, 107–123.
- (13) Goeminne, L. J.; Gevaert, K.; Clement, L. Peptide-level Robust Ridge Regression Improves Estimation, Sensitivity, and Specificity in Data-dependent Quantitative Label-free Shotgun Proteomics. *Molecular & Cellular Proteomics* **2016**, *15*, 657.
- (14) Sticker, A.; Goeminne, L.; Martens, L.; Clement, L. Robust Summarization and Inference in Proteome-wide Label-free Quantification. *Molecular & Cellular Proteomics* **2020**, *19*, 1209.
- (15) Goeminne, L. J. E.; Sticker, A.; Martens, L.; Gevaert, K.; Clement, L. MSqRob takes the missing hurdle: uniting intensity- and count-based proteomics. *Anal. Chem.* **2020**, *92*, 6278.
- (16) Zhu, Y.; Orre, L.; Tran, Y. Z.; Mermelekas, G.; Johansson, H.; Malyutina, A.; Anders, S.; Lehtio, J. DEqMS: A Method for Accurate Variance Estimation in Differential Protein Expression Analysis. *Molecular & Cellular Proteomics* **2020**, *19*, 1047.
- (17) Smyth, G. K. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology* **2004**, *3*, 1.
- (18) Smyth, G. K. In *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*; Gentleman, R., Carey, V., Huber, W., Irizarry, R., Dudoit, S., Eds.; Springer: New York, 2005; p 397.
- (19) Ritchie, M. E.; Phipson, B.; Wu, D.; Hu, Y.; Law, C. W.; Shi, W.; Smyth, G. K. limma powers differential expression analyses for RNA-seqencing and microarray studies. *Nucleic Acids Res.* **2015**, *43*, e47.
- (20) Ahlmann-Eltze, C. *proDA: Differential Abundance Analysis of Label-Free Mass Spectrometry Data*, R package version 1.12.0; 2022.
- (21) Stratton, K. G.; Webb-Robertson, B.-J. M.; McCue, L. A.; Stanfill, B.; Claborne, D.; Godinez, I.; Johansen, T.; Thompson, A. M.; Burnum-Johnson, K. E.; Waters, K. M.; Bramer, L. M. *pmartR*: Quality Control and Statistics for Mass Spectrometry-Based Biological Data. *J. Proteome Res.* **2019**, *18*, 1418.
- (22) Girden, E. R. *ANOVA: Repeated Measures*, Vol. 84; Sage, 1992.
- (23) Wieczorek, S.; Combes, F.; Lazar, C.; Giai Gianetto, Q.; Gatto, L.; Dorffer, A.; Hesse, A.-M.; Couté, Y.; Ferro, M.; Bruley, C.; Burger, T. DAPAR & ProStaR: software to perform statistical analyses in quantitative discovery proteomics. *Bioinformatics* **2017**, *33*, 135.
- (24) Gatto, L.; Vanderaa, C. *QFeatures: Quantitative features for mass spectrometry data*, R package version 1.8.0; 2022.
- (25) Hediyez-zadeh, S. *msImpute: Imputation of label-free mass spectrometry peptides*, R package version 1.8.0; 2022.
- (26) Huang, T.; Choi, M.; Tzouros, M.; Golling, S.; Pandya, N. J.; Banfai, B.; Dunkley, T.; Vitek, O. *MSstatsTMT*: statistical detection of differentially abundant proteins in experiments with isobaric labeling and multiple mixtures. *Molecular & Cellular Proteomics* **2020**, *19*, 1706.
- (27) Kohler, D.; Tsai, T.; Verschueren, E.; Huang, T.; Hinkle, T.; Phu, L.; Choi, M.; Vitek, O. *MSstatsPTM*: Statistical relative quantification of post-translational modifications in bottom-up mass spectrometry-based proteomics. *Molecular & Cellular Proteomics* **2023**, *22*, 100477.
- (28) Kohler, D.; Kaza, M.; Pasi, C.; Huang, T.; Staniak, M.; Mohandas, D.; Sabido, E.; Choi, M.; Vitek, O. *MSstatsShiny*: AGUI for Versatile, Scalable, and Reproducible Statistical Analyses of Quantitative Proteomic Experiments. *J. Proteome Res.* **2023**, *22*, 551–556.
- (29) Kong, A.; Leprevost, F.; Avtonomov, D.; Mellacheruvu, D.; Nesvizhskii, A. *MSFragger*: ultrafast and comprehensive peptide identification in mass spectrometry-based proteomics. *Nat. Methods* **2017**, *14*, 513–520.
- (30) Röst, H. L.; et al. OpenMS: a flexible open-source software platform for mass spectrometry data analysis. *Nat. Methods* **2016**, *13*, 741.
- (31) Tsou, C.-C.; Avtonomov, D.; Larsen, B.; Tucholska, M.; Choi, H.; Gingras, A.-C.; Nesvizhskii, A. I. *DIA-Umpire*: comprehensive computational framework for data-independent acquisition proteomics. *Nat. Methods* **2015**, *12*, 258.
- (32) 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. OpenSWATH enables automated, targeted analysis of data-independent acquisition MS data. *Nat. Biotechnol.* **2014**, *32*, 219.
- (33) Demichev, V.; Messner, C.; Vernardis, S.; Lilley, K.; Ralser, M. *DIA-NN*: neural networks and interference correction enable deep proteome coverage in high throughput. *Nat. Methods* **2020**, *17*, 41.
- (34) Navarro, P.; Kuharev, J.; Gillet, L. C.; Bernhardt, O. M.; MacLean, B. X.; 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. A multicenter study benchmarks software tools for label-free proteome quantification. *Nat. Biotechnol.* **2016**, *34*, 1130.
- (35) Chiva, C.; Ortega, M.; Sabidó, E. Influence of the digestion technique, protease, and missed cleavage peptides in protein quantitation. *J. Proteome Res.* **2014**, *13*, 3979–3986.
- (36) Meierhofer, D.; Halbach, M.; Sen, N. E.; Gispert, S.; Auburger, G. Ataxin-2 (Atxn2)-Knock-Out Mice Show Branched Chain Amino Acids and Fatty Acids Pathway Alterations. *Molecular & Cellular Proteomics* **2016**, *15*, 1728.
- (37) Selevsek, N.; Chang, C.-Y.; Gillet, L. C.; Navarro, P.; Bernhardt, O. M.; Reiter, L.; Cheng, L.-Y.; Vitek, O.; Aebersold, R. Reproducible and consistent quantification of the *Saccharomyces cerevisiae* proteome by SWATH-MS. *Mol. Cell. Proteomics* **2015**, *14*, 739–749.
- (38) Tukey, J. W. *Exploratory Data Analysis*; Addison-Wesley: New York, NY, 1977.
- (39) Oberg, A. L.; Vitek, O. Statistical design of quantitative mass spectrometry-based proteomic experiments. *J. Proteome Res.* **2009**, *8*, 2144.
- (40) Staniak, M.; Choi, M.; Huang, T.; Vitek, O. *MSstatsConvert: Import Data from Various Mass Spectrometry Signal Processing Tools to MSstats Format*, R package version 1.6.0; 2022.
- (41) Montgomery, D. C. *Design and Analysis of Experiments*, 8th ed.; John Wiley & Sons, Inc.: New Jersey, USA, 2013.
- (42) Wei, L. J. The accelerated failure time model: A useful alternative to the cox regression model in survival analysis. *Statistics in Medicine* **1992**, *11*, 1871.
- (43) Kalbfleisch, J.; Prentice, R. *The Statistical Analysis of Failure Time Data*, 2nd ed.; Wiley: New York, US, 2002.
- (44) Kutner, M.; Nachtsheim, C.; Neter, J.; Li, W. *Applied Linear Statistical Models*, 5th ed.; McGraw-Hill Irwin: Boston, MA, USA, 2005.
- (45) Clough, T.; Thaminy, S.; Ragg, S.; Aebersold, R.; Vitek, O. Statistical protein quantification and significance analysis in label-free LC–MS experiments with complex designs. *BMC Bioinformatics* **2012**, *13*, S6.
- (46) R Core Team. *R: A language and environment for statistical computing*; R Foundation for Statistical Computing: Vienna, Austria, 2022.

- (47) Wickham, H.; et al. Welcome to the tidyverse. *Journal of Open Source Software* **2019**, *4*, 1686.
- (48) Dowle, M.; Srinivasan, A. *data.table: Extension of 'data.frame'*, R package version 1.14.2; 2021.
- (49) H2O.ai. Database-like ops benchmark. *GitHub*. <https://h2oai.github.io/db-benchmark/> (accessed 11/24/2022).
- (50) Eddelbuettel, D.; François, R. Rcpp: seamless R and C++ integration. *Journal of Statistical Software* **2011**, *40*, 1–18.
- (51) White, J. M.; Jacobs, A. *log4r: A fast and lightweight logging system for R, based on 'log4j'*, R package version 0.4.2; 2021.
- (52) Lang, M. checkmate: Fast argument checks for defensive R programming. *R Journal* **2017**, *9*, 437.
- (53) Mersmann, O. *microbenchmark: Accurate Timing Functions*, R package version 1.4.9; 2021.
- (54) Wickham, H. *lineprof: An alternative display for line profiling information*, R package version 0.1.9001; 2022.
- (55) Richardson, N.; Cook, I.; Crane, N.; Dunnington, D.; François, R.; Keane, J.; Moldovan-Grünfeld, D.; Ooms, J.; *Apache Arrow arrow: Integration to 'Apache' 'Arrow'*, R package version 9.0.0; 2022.

## □ Recommended by ACS

### LFQ-Based Peptide and Protein Intensity Differential Expression Analysis

Mingze Bai, Yasset Perez-Riverol, et al.  
MAY 23, 2023  
JOURNAL OF PROTEOME RESEARCH

READ

### prolflqua: A Comprehensive R-Package for Proteomics Differential Expression Analysis

Witold E. Wolski, Christian Panse, et al.  
MARCH 20, 2023  
JOURNAL OF PROTEOME RESEARCH

READ

### Automated Proteomics Workflows for High-Throughput Library Generation and Biomarker Detection Using Data-Independent Acquisition

Selvam Paramasivan, Paweł Sadowski, et al.  
MAY 23, 2023  
JOURNAL OF PROTEOME RESEARCH

READ

### Evaluation of DDA Library-Free Strategies for Phosphoproteomics and Ubiquitinomics Data-Independent Acquisition Data

Chengwen Wen, Chuan-Qi Zhong, et al.  
MAY 31, 2023  
JOURNAL OF PROTEOME RESEARCH

READ

[Get More Suggestions >](#)