Mspire-Simulator: LC-MS Shotgun Proteomic Simulator for Creating Realistic Gold Standard Data

Andrew B. Noyce¹, Rob Smith², James Dalgleish¹, Ryan M. Taylor¹, K C. Erb¹, Nozomu Okuda¹ and
John T. Prince¹*

¹ Department of Biochemistry, Brigham Young University, Provo UT.

² Department of Computer Science, Brigham Young University, Provo UT.

KEYWORDS: Simulation, simulator, model, mspire, mass spectroscopy, proteomics.

ABSTRACT: The most important step in any quantitative proteomic pipeline is feature detection (aka peak picking). However, generating quality hand-annotated data sets to validate the algorithms, especially for lower abundance peaks, is nearly impossible. An alternative for creating gold standard data is to simulate it with features closely mimicking real data. We present Mspire-Simulator, a free, open source shotgun proteomic simulator that goes beyond previous simulation attempts by generating LC-MS features with realistic m/z and intensity variance along with other noise components. It also includes machine learned models for retention time and peak intensity prediction and a genetic algorithm to custom fit model parameters for experimental data sets. We show that these methods are applicable to data from three different mass spectrometers, including two fundamentally different types, and show visually and analytically that simulated peaks are nearly indistinguishable from actual data.

1
Researchers can use simulated data to rigorously test quantitation software, and proteomic researchers may benefit from overlaying simulated data on actual data sets. [https://github.com/princelab/mspire-simulator](https://github.com/princelab/mspire-simulator)

INTRODUCTION

A single liquid chromatography-mass spectrometry (LC-MS) run is inherently capable of quantifying upwards of 100,000 peptides\(^1\). Unfortunately, in a typical analysis the majority of this data is discarded due to difficulties in identifying and accurately picking chromatographic peaks, especially those of lower abundance. Increasing the accuracy of peak picking results in the detection of more features that can be compared across runs. More accurate peak picking can also influence mass estimates and therefore yield an increase in the number and quality of identifications\(^2\). It ultimately simplifies cross-run comparisons of feature abundances and increases the overall accuracy of those quantitative comparisons. In other words, peak picking quality influences the entire downstream analysis.

For these reasons, it is undoubted that the most important step of a proteomic workflow is feature detection, for which many algorithms exist\(^2,3-4\). However, very little has been done to fully test or compare the performance of these algorithms. In large part, this is due to the challenging nature of creating gold-standard data. Fully annotating actual complex proteomic data sets, or even small portions, is extremely time consuming, difficult, error prone, and subjective. Because MS/MS annotation is rare for small peaks and because they have intensities near the signal to noise threshold, accurate human annotation of small peaks in a complex sample is very likely impossible.

Simulation is routinely used in related fields when gold standard data is difficult to come by (e.g., systems biology network simulation)\(^6\) or the cost of performing each experiment is high (simulated ion movement in MS fields)\(^5\). For quantitative mass spectrometry, an attractive alternative to using hand-
labeled data sets is to simulate actual data using noise parameters derived from experimental data. An ideal simulator would generate data sets where all aspects of the data are known, the various noise components are adjustable, and the peak characteristics conform to those found in biologically derived data sets. Such data sets would be invaluable for comparing algorithms because accuracy can be comprehensively and quickly ascertained programmatically. The speed of this feedback will also aid in the creation of new, more sophisticated algorithms.

Because simulators can produce fully defined peaks of any size, data sets produced by simulation are particularly well-suited to test algorithms for their ability to detect and accurately quantify small LC peaks. Small peaks are highly desirable targets for identification and quantitation because: 1) seminal biological events may occur at low quantities (e.g., upstream signal transduction) 2) a change in state to low quantity may be as significant as an increase in quantity 3) post-translational modifications may manifest themselves as a drop in the unmodified peptide’s concentration 4) lower abundance peaks constitute the majority of peaks in an LC-MS run and these are inaccessible by current MS/MS regimes. By quantifying low abundance peaks, proteomic and individual protein coverage may improve, and intra-protein variation can be tracked.

Many proteomic workflows allow users to examine their experimentally derived fragmentation spectra alongside a representation of the theoretically matched spectrum (i.e., an MS/MS fragmentation view). With a simulated LC/MS data set in hand, a somewhat analogous view could be generated for the user where simulated MS1 data is layered on top of actual data. This view would encourage a researcher to examine their MS1 output in order to reconcile what they can observe with what they expect to observe. A peak that went unidentified may still be present, and researchers would then know where to look within their MS1 data. Alternatively, a peak that should have been present may be absent
prompting researchers to simulate data with conjectured post-translational modifications in an effort to locate the modified peak. Simulated data sets have the potential to augment the traditional proteomics workflow in which researchers often neglect to thoroughly examine their MS1 data.

While previous MS1 proteomic simulators⁷,⁸ have been created, a simulator that mimics the intensity and m/z variance found in real data sets is critical for testing peak-picking/quantitation algorithms. Here we present a full featured LC/MS shotgun proteomics simulator, Mspire-Simulator, that generates peptide peaks with realistic m/z and intensity variance and elution profiles. Machine learning is used to generate peaks with a realistic retention time distribution as well as peak heights reflecting peptide ionization efficiency.

2 METHODS

Mspire-Simulator takes as its input FASTA files containing the protein sequences that are to be in the simulated run. Using one of 16 proteolytic enzymes and relevant digestion parameters each protein sequence is in-silico digested into peptides. Each peptide's charge, mass and theoretical spectrum, including the isotopic distribution, is calculated. These calculations are currently used to create centroided data. The simulator will be extended to create profile data in the future. Centroided data will be most useful initially because most analytical software deals with this type data. The simulator is implemented in the Ruby programming language and makes use of and extends the mspire⁹ (mass spectrometry proteomics in Ruby) library. It is available under the MIT license and works out of the box with sensible defaults. Customization to data from different machines is achieved through an included Ruby script which uses a genetic curve fitting algorithm. This script produces SVG files that visualize the fits as well as the necessary parameters for Mspire-Simulator to adapt its simulations.
The actual data used to create our default simulation model was obtained from our in-house LTQ-Orbitrap mass spectrometer coupled with reverse phase liquid chromatography using nanospray ionization. The data is derived from an LC-MS shotgun proteomic run of complex Human Embryonic Kidney (HEK-293T) cells. We used a Waters Nano Acuity column (15cm long). ‘A’ solvent used was 95:5 water to acetonitrile and 0.1% formic acid and ‘B’ solvent was acetonitrile and 0.1% formic acid. Gradient was formed by 5% - 60% solvent mix over 70% of the run. This data along with all files produced and used is deposited in ProteomeCommons.org Tranche with the following hash key: plQ/C4OZT43Kbhz1Sq1UVrfzuqKNytwHEIwYK7cjZjF4rBo9t6tNGjyZDTrEmkZwK/7y8pY/YTbYA UoucKsYHoZy+jAAAAAAAB0g==

Except the MM14 data which is already available at http://msbi.ipb-halle.de/msbi/centwave/, and the Orbitrap-Velos data which is available upon request.

Orbitrap-Velos data was generously provided by the Christine Vogel lab and was from a ubiquitin pulldown from Saccharomyces Cerevisiae (Eksigent NanoFlow Plus, LC gradient 2-90% acetonitrile over 4.5 hrs at flow rate 400nL/min). MM14 data is from the Bruker MicrOTOF-Q instrument, is described by Tautenhahn et al. and details can be found in that publication.\textsuperscript{10}

3 RESULTS

Mspire-Simulator models elution, variance in intensity and variance in the mass to charge ratio (m/z) and predicts retention times and intensities for peptides. We follow the convention of Cappadona et al. and refer to a peptide feature as the full chromatographic profile of a peptide (at a given charge state) and a peptide peak as an individual isotopic component of a feature\textsuperscript{11}. Fig. 1 outlines the overall process of simulation, and we consider each component in turn.

3.1 RETENTION TIME & INTENSITY PREDICTION

5
A peptide is first situated along the retention time axis (see Fig. 1A). Both retention time and intensity are predicted for each peptide using a machine learned model that was built in WEKA\textsuperscript{12}. We used the M5Rules\textsuperscript{13} algorithm for retention time prediction and the M5P\textsuperscript{14} algorithm for intensity prediction, both of which gave the best correlation coefficients for our test data, \(\sim 0.96\) and \(\sim 0.74\) respectively using the internal WEKA ten-fold cross validation technique. The test data can be downloaded using the above hash. These prediction models were trained on in house data which contained amino acid counts, average m/z value, the charge state, mass, retention time, and a binned intensity value for 1484 peptides. The intensity values were binned into ten bins based on magnitude ranges. This allowed for a better prediction of intensities. The user may replace the default models with custom/better ones in order to mimic other configurations and machines. For each peptide a single retention time and intensity is predicted; these values are used as starting points from which the retention times and intensities of all the centroids related to a particular peptide are generated. The retention times are coerced into times that conform to a user specified sampling rate (e.g. one scan per two seconds). Elution profiles are generated by sampling from the normal distributions of parameters t and f.

3.2 FEATURE SHAPE

Peptide features are modeled along the m/z axis (Fig. 1A) by predicting the charge states and isotope distribution of a peptide. For charge state prediction, a user specifies a pH, after which a standard iterative procedure is used to determine the ratio of charge states that would be observed (e.g. for the peptide DRVYIHPF at a pH of 2.0, 29.045\% of this peptide would have a charge of +2 and 70.959\% a charge of +3). We label this parameter “ionized pH” to indicate that it represents the acidity of the peptide as it enters the mass spectrometer and not necessarily in LC buffer. Isotope distributions are calculated by FFT convolution\textsuperscript{15}.
The elution profile (Fig. 1B) is produced by function composition of a dynamic standard deviation with a Gaussian function. The standard deviation ($\sigma$) is based on the relative position along the elution curve:

$$\sigma = tx + f$$

where $x$ is the relative retention time index from the starting retention time of the feature, $t$ is the tailing factor and $f$ influences the shape at the front of the profile. The elution profile is then given by substitution of $\sigma$ into a Gaussian function:

$$i = he^{-12x - \mu \sigma^2}$$

where $i$ is the intensity at that point in the elution, $\mu$ is where the apex of the curve is located, $x$ is as above, and $h$ is an initial height factor which determines the maximum height of the peak and is the same for each peptide. Thus, $i$ is a generalized intensity which is later modified by a variance model and predicted intensity values as mentioned above. This produces a skewed elution profile that fits peaks derived from a wide variety of elution conditions as shown for data on LTQ-Orbitrap (Fig. 2A), a Quadrupole Time-of-Flight (Qq-TOF) (Fig. 2B) and an Orbitrap-Velos (Fig 2C).

For the mass spectrometer types we examined there was a global relationship between the intensity of a peak and the variance of its measurement. We observed larger intensity variance in more intense features and thus also nearer the apex of an eluting peak (Fig. 1C,D). An inverse exponential function captures this relationship:

$$\sigma = m*(1-e^{-c*i}) + d$$
Where $\sigma$ defines the standard deviation in intensity given the intensity value, $i$. The $c$, $d$, and $m$ parameters represent experimentally derived constants that can be used to fine-tune the function for different mass spectrometers or run conditions (Fig. 3A,B). $\sigma$ is then composited into a Gaussian function for each peak, again like above, and the ideal intensity is modified by drawing stochastically from this distribution. Our intensity variance model adequately mimics real data. When compared to actual data we observe a RMSD of 0.9051 (Fig. 3A,B).

The m/z variance is a function of intensity and therefore may vary between peptide features, and also within each elution profile contained in a feature (Fig. 1E,F). This is modeled by the following function:

$$\sigma = m \cdot i - y$$

Where $\sigma$ is the standard deviation, $i$ is the relative intensity of the feature at that point in its elution profile and $y$ is another experimentally derived constant that can be fit to different data types. The standard deviation function is composited with a Gaussian function, similar to the above elution functions, and is then randomly sampled from to give the quantity of m/z variance in either direction. The m/z variance model produces realistic results based on the comparison of simulated m/z variance to actual m/z variance (Fig. 3C,D).

Feature shape is further modeled by given protein abundances. The abundances can be specified in the FASTA file header by a '#' symbol followed by a value representing the percentage of that protein in the sample. If no abundances are given, equal molarity is assumed. These values are then used to modify the total area under the function that determines feature shape by a simple scaling procedure.

3.3 DROPS & NOISE
At certain retention times, in real LC-MS runs, entire scans where very few if any peaks are observed are referred to as “drops” (e.g. PeptideAtlas accession PAe000142 contributed by S. Markey). Our model also simulates drops at random retention times by a specified percentage of the total run time. This elevates the realism in the simulation and adds another dimension of control when using simulated data to test analytical software.

The simulator has the ability to add white noise to the spectra based on density and intensity factors specified by the user. The higher the density factor, the more white noise there is in each spectrum. Intensities are pulled from a flat random distribution that varies between a maximum and minimum value given by the user. These parameters, along with the option to turn off the white noise completely, give the user complete control for testing purposes.

3.4 MERGING OVERLAPPING PEAKS

As a final processing step, overlapping peaks are detected and merged. This is accomplished by using a ppm range to define whether two peaks are sufficiently close to be overlapping or not. The intensities of the peaks to be merged are summed and the new m/z value is calculated by a weighted average of the original m/z values weighted by the intensities of the respective peaks. We use ¼ of the m/z variance in ppm to define the range that we use to detect overlapping peaks, and this parameter is adjustable by the user.

3.5 MS/MS

Theoretical fragmentation spectra are produced by generating fragment ion formulas for all possible cleavages and calculating the mass for each ion, at the predicted charge states. The ion types are configurable, and masses can be average or monoisotopic. The fragmentation spectra are produced by the MS-fragmenter gem, freely available from Rubygems.org.
3.6 MODIFICATIONS

Mspire-Simulator has the ability to add modifications to specified residues and termini. Modifications are read in by the user specifying a modification ID from the PSI-MOD.obo and which residue/terminus to apply it too. These modifications are then used in the calculation of each spectrum. Since there will always be modifications in peptide samples this is an important part of simulation.

3.7 OUTPUT

The simulated run is written to an mzML file that can be visualized with any mzML file viewer. The mzML format is the standard *de jure* and is quickly becoming the standard *de facto* for mass spectrometer data. Cross-platform converters like Proteowizard\textsuperscript{16} can convert mzML into mzXML\textsuperscript{20} or other formats. Alternatively, the code base itself could easily be extended to directly output other representations of the data as well. The program also creates an SQLite, XML or CSV file which contains information on all of the data in the simulated run which can then be used to validate peak picking and quantitation software.

3.8 PARAMETER FITTING AUTOMATION

Simulating data from different mass-spectrometers and operating conditions requires some customization of noise and variance parameters. We developed a genetic algorithm to discover parameters from actual data. Fig. 4 demonstrates the automatic fitting of Orbitrap Velos data, and it works equally well on the many peaks and instrument types we have tested.

3.9 USING THE SIMULATOR TO ASSESS QUANTITATION PERFORMANCE

The lack of quantitative comparison of data processing and wet lab protocol is due in large part to the daunting task of obtaining labeled data\textsuperscript{17,18}. The size and complexity of MS data sets precludes obtaining labeled data without a significant outlay of resources. Mspire-Simulator provides a facile method for
generating any quantity of labeled simulated data. As a case study, consider Smith et al, where Mspire-Simulator data used in conjunction with hand labeled real data allowed the use of qualitative metrics to evaluate the accuracy of a peak summarization, a data processing step in non-chromatographic studies.

DISCUSSION & CONCLUSION

Mspire-Simulator succeeds at creating highly realistic LC-MS peptide features as demonstrated by the comparison between actual and simulated data shown in Fig. 5. Under macro- and microscopic inspection, analytically and visually, the two features are virtually indistinguishable (Table 1). An entire simulated run of Bovine Serum Albumin (BSA) (Fig. 6) shows the similarity between the simulated data and what is commonly observed in performing an actual BSA digest during quality control runs. Mspire-Simulator can also produce highly complex runs (Fig. 7) as well as simulate data from different mass-spectrometers (Fig. 2-3).

With Mspire-Simulator’s abilities, layering simulated data next to or on top of actual data, visually or analytically, could become standard practice in proteomics—much the way MS/MS spectra are layered onto predicted b and y ion series to identify potential database matches. By comparing actual data with the model and then refining the model, a feedback loop is created that has utility not only in affirming what is known but in pointing out what is missing. Is an expected peptide missing because it has been modified? Are changes in the ratios of charge states indicative of pH or electrospray voltage aberrations? These and other aspects of a run can now be queried, and this process will inevitably result in more complete, more refined models of shotgun proteomics.

Refinements to Mspire-Simulator will focus initially on technical aspects of a LC-MS proteomic experiment. These include: a more explicit model of a peptide’s ionization efficiency, the pH of a
solution as buffer concentrations change and as influenced by the electrospray process; the relative rates of tryptic digestion as a function of adjacent amino acid residues; profile data simulation; exploring the relationship between variance parameters and m/z and retention time; and improvements in peak merging. Future efforts will be devoted to these refinements.

Mspire-Simulator could also be extended with more sophisticated modeling of biological phenomenon. More rigorous post-translational modification or splice-variant prediction would alter the landscape of predicted peptides. Protein level enrichment could easily be added in, reflecting predictions about localization in a fractionated sample for instance. While biological questions are appropriately addressed after analysis of the raw data, it is nonetheless intriguing to consider mapping the biology as a simulated data set onto the raw data in an effort to generate putative identities for unanticipated peaks.

As simulated data becomes more sophisticated, we are aware of the possibility of its inappropriate use. The mzML file format is open and completely editable. As it currently stands, we see no way to prevent a simulated mzML file from being tampered with in order to be passed off as actual data. But, the problem is not as hopeless as it might seem at first glance: the mzML format encourages use of a file hash tag audit trail, so instrument produced data should always point back to a vendor produced raw data file. Deciding whether a file was simulated is now roughly equivalent to deciding whether a file was tampered with, and that is checking against a vendor produced raw data file. The potential for the fraudulent use of simulated data should serve, then, to encourage what researchers should be doing anyway: providing access to raw data and using audit trails. In any event, we suggest that the potential benefits of simulation software far outweigh the challenges presented by potential misuse.
Mspire-Simulator will be useful initially in testing and developing algorithms for peak picking and quantitation. Simulated data is not meant to replace testing on actual data, but to facilitate more rigorous testing of algorithms. Data may be simulated with a range of peak and noise characteristics, and strengths or flaws in algorithms uncovered. Mspire-Simulator will be especially useful in testing algorithms for their ability to accurately detect and quantify small peaks because the provenance of every centroid is known. Simulated data may ultimately facilitate workflows that find and quantitate an order of magnitude more peptides than is currently possible.
**Figure 1.** Overall process of simulation from theoretical spectrum to realistic peaks. The underscored 3d box in part B and E designates the specific peak shown in following parts. A: The theoretical spectrum calculated for a certain peptide. B: Ideal elution profiles are given to the spectrum. C, D: Intensity variance is calculated for each peak in the elution profile. E, F: Mass to charge variance is calculated for each peak in the elution profile.
Figure 2. Elution Peak Shape. Max intensity normalization was used in each case. The noisy gray line shows a peak from actual data. Dashed line shows the function that the simulator uses. The simulator’s model can be modified to fit many elution profiles present in real data. A: LTQ-Orbitrap. B: Orbitrap-Velos. C: Qq-TOF.
Figure 3. Intensity and m/z variance. Circles show simulated standard deviation variance and pluses show actual standard deviation variance. Max intensity normalization was applied in each case. The x axis is on a $\log_{10}$ scale. A-B: Standard deviation is calculated along intervals of ten peaks across the elution profiles. A: The simulator models this behavior from a LTQ-Orbitrap accurately with a small RMSD between actual and simulated of 0.9051. B: It can also model Qq-TOF data accurately; RMSD of 1.1167. C: Inset is one actual elution peak and the large image is four combined (RMSD = 0.1153). D: M/z variance from Qq-TOF data. C-D: This shows the general trend of measured m/z values varying more at low intensity signals and less at high intensity signals as well as the simulator’s ability to mimic this observation.
**Figure 4.** Demonstrating the visual output from the curve fitting program. Max intensity normalization was used for each. This is a fit of Orbitrap Velos data. The blue dots show the actual data and the red smooth lines represent the curve fit. This shows the ability to quickly generate parameters needed to simulate different types of data. This output took ~5 min.
Figure 5. Left side shows simulated features and right side shows the actual features. A: Visual comparison of LC-MS feature from the peptide: HLVDEPQNLIK (single letter code amino acids). See Table 1 for analytical comparison. B: Detail of a single elution profile showing m/z variance characteristics. Simulated m/z variance is very similar to actual (see Table 1; row 1-5).
**Figure 6.** A simulated BSA run (top left) is compared to an actual BSA run (top right). As can be seen, there are differences in retention times and intensities between the two runs indicating that refinements can be made to these two prediction models. A detail segment (bottom) of the simulated (left) and real (right) BSA runs show that for each there are labels not found in the other (red) while there are many that are found in both (black).
Figure 7. Birdseye view of a simulated complex human cell run. 50,000 peptides were taken from the human FASTA database and simulated in two charge states creating 100,000 features. The run was generated in ~31hrs on a single 2.50GHz core and used ~1.9Gb of RAM. White vertical lines represent dropped/lower signal and are intentionally included. The run demonstrates the simulator’s ability to generate highly complex runs. Purple peaks are the highest intensity, then red, yellow, and gray the lowest. Viewed in TOPPView²⁴.
Table 1. Statistics comparing the two features shown in Fig. 5. Normalized values were calculated by using a max intensity normalization.

<table>
<thead>
<tr>
<th>statistic</th>
<th>actual</th>
<th>simulated</th>
<th>difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>m/z variance isotope index 1 (Least Abundant)</td>
<td>0.094738989</td>
<td>0.21487649</td>
<td>0.120137498 ppm</td>
</tr>
<tr>
<td>m/z variance isotope index 2</td>
<td>0.070901966</td>
<td>0.13662707</td>
<td>0.065725100 ppm</td>
</tr>
<tr>
<td>m/z variance isotope index 3</td>
<td>0.239360653</td>
<td>0.37309639</td>
<td>0.133735740 ppm</td>
</tr>
<tr>
<td>m/z variance isotope index 4</td>
<td>0.246936346</td>
<td>0.20347133</td>
<td>0.043465011 ppm</td>
</tr>
<tr>
<td>m/z variance isotope index 5 (Most Abundant)</td>
<td>0.03203343</td>
<td>0.29622914</td>
<td>0.264195713 ppm</td>
</tr>
<tr>
<td>Intensity variance</td>
<td>26.3</td>
<td>25.8</td>
<td>0.52</td>
</tr>
<tr>
<td>Elution time interval length (seconds)</td>
<td>43.3845215</td>
<td>43.3596719</td>
<td>0.02484959</td>
</tr>
<tr>
<td>Normalized Intensity mean</td>
<td>21.9</td>
<td>21.35</td>
<td>0.55</td>
</tr>
<tr>
<td>Normalized Intensity median</td>
<td>8.2</td>
<td>8.081</td>
<td>0.119</td>
</tr>
<tr>
<td>Number of samples/peaks used (peaks &gt; 0.9 normalized abundance)</td>
<td>73</td>
<td>85</td>
<td>12</td>
</tr>
<tr>
<td>Number of Peaks in Quartile: 1</td>
<td>21</td>
<td>21</td>
<td>0</td>
</tr>
<tr>
<td>Number of Peaks in Quartile: 2</td>
<td>27</td>
<td>25</td>
<td>2</td>
</tr>
<tr>
<td>Number of Peaks in Quartile: 3</td>
<td>15</td>
<td>21</td>
<td>6</td>
</tr>
<tr>
<td>Number of Peaks in Quartile: 4</td>
<td>10</td>
<td>16</td>
<td>6</td>
</tr>
</tbody>
</table>

AUTHOR INFORMATION

Corresponding Author

*John T. Prince. Email: jtprince@gmail.com, Tel. 801-422-7123. Fax. 801-422-0153.

Author Contributions

Andrew B. Noyce wrote the main simulator code and drafted and edited the manuscript. Rob Smith assisted with the simulator design and validation. James Dalgliesh helped develop and optimize the machine learning components and helped with manuscript revisions. Ryan M. Taylor provided general insight and assistance and wrote the charge state prediction algorithm. KC Erb helped develop initial elution profile equations. Nozomu Okuda wrote initial mzML writer. John T. Prince instigated the
project, supervised, wrote the isotope distribution and final mzML writer code, and helped to draft and edit, the manuscript. All authors have given approval to the final version of the manuscript.

**Funding Sources**

Institutional startup funds from Brigham Young University and multiple BYU Undergraduate Research Awards. RS is funded by NSF GRF [DGE-0750759].

**ACKNOWLEDGMENT**

We would like to thank Brigham Young University for generous funding. We thank Christine Vogel for the Orbitrap-Velos data set, and we thank Adam Swensen for annotating BSA peaks for Figure 6.

**ABBREVIATIONS**

LC-MS, liquid chromatography mass spectrometry; m/z, mass to charge ratio; FFT, fast Fourier transform; BSA, bovine serum albumin.
REFERENCES

1. Michalski, A.; Cox, J.; Mann, M., More than 100,000 Detectable Peptide Species Elute in Single Shotgun Proteomics Runs but the Majority is Inaccessible to Data-Dependent LC–MS/MS. Journal of Proteome Research 2011, 10 (4), 1785-1793.


Abstract Graphic: