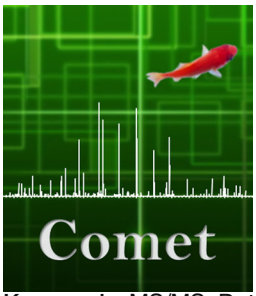
J. Am. Soc. Mass Spectrom. (2015) 26:1865–1874
DOI: 10.1007/s13361-015-1179-x

FOCUS: 20 YEAR ANNIVERSARY OF SEQUEST: RESEARCH ARTICLE

# A Deeper Look into Comet—Implementation and Features

Jimmy K. Eng,[1] Michael R. Hoopmann,[2] Tahmina A. Jahan,[1] Jarrett D. Egertson,[1] William S. Noble,[1] Michael J. MacCoss[1]

[1]Department of Genome Sciences, University of Washington, Seattle, WA, USA
[2]Institute for Systems Biology, Seattle, WA, USA




**Abstract.** The Comet database search software was initially released as an open source project in late 2012. Prior to that, Comet existed as the University of Washington's academic version of the SEQUEST database search tool. Despite its availability and widespread use over the years, some details about its implementation have not been previously disseminated or are not well understood. We address a few of these details in depth and highlight new features available in the latest release. Comet is freely available for download at http://comet-ms.sourceforge.net or it can be accessed as a component of a number of larger software projects into which it has been incorporated.

**Key words:** MS/MS, Database search, Comet

Received: 23 February 2015/Revised: 22 April 2015/Accepted: 27 April 2015/Published Online: 27 June 2015

## Introduction

In a seminal paper published in 1994, the ability to sequence peptides by searching uninterpreted tandem mass spectra (MS/MS) against protein sequence databases was disseminated to the proteomics community [1]. Now over 20 years later, MS/MS database searching has become arguably the most commonly applied computational proteomics analysis method in practice. Not surprisingly, a number of novel MS/MS database search tools have been developed over the years [2] but the SEQUEST algorithm continues to be widely used.

SEQUEST was originally developed at the University of Washington and commercially licensed to the Thermo Electron Corporation. For a number of years through the 1990s, SEQUEST existed in two forms: the academic version developed at University of Washington and the commercial version distributed by Thermo. Over the years, SEQUEST-like tools have expanded to include a commercial version from Sage-N Research (Sorcerer [3]), other academic versions developed at the University of Washington (Crux [4], Tide [5]), Scripps Research (ProLuCID [6]), and Dartmouth College (macroSEQUEST [7],Tempest [8]) among others. In 2012, the University of Washington's version of SEQUEST was released as an open source project and renamed Comet [9]. This article describes in detail some of the features and optimizations in the latest version of the Comet software tool.

High resolution MS/MS data are more common these days because of improvements in instrumentation with Orbitrap (Fourier transform) and Time-of-Flight analyzers. With advances in instrumentation, the ability to generate high resolution MS/MS spectra at a fast acquisition rate makes such data much more ubiquitous. While accurate MS/MS fragmentation spectra allow for more stringent identifications due to the significantly increased selectivity of matching fragmentation peaks with tight mass tolerances, such data poses a challenge to the Comet algorithm, and SEQUEST before it, with respect to how the data is represented internally. With spectra stored as discrete arrays of numbers, where the array index represents the mass and the array value at that index representing the intensity, a high resolution spectrum requires a lot of memory to be stored in this array data format because of the large number of small mass buckets or bins. A detailed description of what such mass bins represent, how optimal bin sizes were determined for low and high resolution data, and two different mechanisms for addressing memory use for high resolution data will be presented in this paper.

In 1995, the second paper to be published on the SEQUEST algorithm described the ability to search for post-translational modifications [10]. Being able to identify modified residues not contained in the protein sequences stored in sequence databases is a powerful method, with numerous applications for biological insight. For example, Swaney et al. were able to identify

**Electronic supplementary material** The online version of this article (doi:10.1007/s13361-015-1179-x) contains supplementary material, which is available to authorized users.

*Correspondence to:* Jimmy K. Eng; *e-mail:* engj@uw.edu

over 2,000 phosphorylation sites co-occurring with over 2,000 ubiquitylation sites in *S. cerevisiae*, allowing the investigation of how phosphorylation can be regulated by ubiquitylation [11]. In a different application, the ability to search for post-translational modifications enabled Chavez et al. to identify cross-linked proteins in living human cells, demonstrating the ability to make direct topological measurements and provide evidence for novel protein–protein interactions [12]. In order to provide researchers with more flexibility in how post-translational modifications can be analyzed, the most recent release of Comet incorporates additional new options for modification analysis, which will be described in detail below with usage examples.

## Materials and Methods

Comet is written in C++ and developed on both Linux and Windows operating systems. Comet incorporates the MSToolkit file parsing library [13] to read mass spectral data in various formats. Multi-threading is implemented using POSIX threads on Linux and Windows native threads on Windows. The computer configuration used for all analyses and benchmarks is a dual Intel Xeon E5-2470 2.4 GHz CPU (eight total physical cores) with 64 GB RAM, running Red Hat Enterprise Linux Server 6.5. False discovery rates and q-value calculations are based on ordering results by Comet's E-value score and then computing, for a concatenated target-decoy search, the ratio of the number of accepted decoy matches divided by the number of accepted target matches at a given score threshold. Mass spectral data files used in the analysis presented here were downloaded from the PRoteomics IDEntification (PRIDE) [14] repository or the Stem Cell Omics Repository (SCOR) [15].

## Results and Discussion

### *Sparse Matrix Representation of Spectra*

The core scoring algorithm in SEQUEST is the cross-correlation score or xcorr. In the 1994 manuscript, the cross-correlation score was calculated by performing Fourier transforms on both the experimental spectrum and the theoretical spectrum, multiplying one transform by the complex conjugate of the other transform, and performing an inverse Fourier transform. This mathematical operation generated the full correlation spectrum from which the cross-correlation score was derived. In 2008, a method to calculate the cross-correlation score in an efficient manner was published [16], where each experimental spectrum was preprocessed in a way that allows the cross-correlation score to be calculated by simply summing up processed intensity values at each theoretical fragment ion mass location. This optimization enabled the cross-correlation score to be applied to scoring all peptides instead of just the 500 best candidate peptides in the original implementation, enabling E-value [9] and *p*-value [17, 18] calculations based on the cross-correlation score distribution. Performance comparisons of Comet with other search engines can be found in [1] and [2] and a comparison of Comet versus SEQUEST cross-correlation scores is presented below.

Inherent in both forms of the cross-correlation calculation is the representation of the experimental spectrum as a discrete array where the array index represents the mass and the array value at that index represents the intensity of a peak at that mass. This data representation is depicted in Figure 1. In this example, an integer array named *spec* stores a digital representation of the mass spectrum where each array index is 1 Da wide and represents the corresponding integer mass-to-charge ($m/z$). The peak intensity is stored as the array value at that mass index. When more than one peak is present in a mass bin, the largest peak intensity is stored. For the fast cross-correlation calculation, this data representation is extremely efficient because the intensity lookup for each calculated fragment ion mass can be accomplished by reading the intensity value stored at the corresponding mass index using a direct lookup in the spectral array.

The array index for a spectrum does not need to be exactly 1 Da wide. In fact, the optimum mass bin width is 1.0005 for low resolution data such as that acquired on an ion trap detector, and we recommend a mass bin width of 0.02 for high resolution spectra. In Comet, this bin size setting is controlled by the parameter "fragment_bin_tol." For any given mass $m$ and bin width $w$, the appropriate array index $idx$ for any given mass is determined by the equation "$idx$=(int)($m/w$)", which simply defines the array index as the integer value of the mass divided by the bin width. This allows for the array representation of a spectrum at any arbitrary bin width value. For high resolution data, much narrower bin widths are necessary to take advantage of the high mass accuracy measurements on the fragment ion masses. But as the bin width $w$ is reduced from say 1.0 to 0.01, the corresponding spectral array grows 100-fold larger in size to accommodate the much smaller mass bins. Accordingly, the memory requirements to internally store the spectral data in this array format is increased 100-fold, making this representation untenable for the analysis of standard-sized LC-MS/MS runs on typical desktop computers.

To address the memory issue when small mass bins are used, a sparse matrix data representation was developed. In Comet version 2015.01, a new sparse matrix format was implemented that trades off some memory efficiency for increased speed and stability compared with the original sparse matrix implementation used in the previous releases. The sparse matrix option is invoked by setting the parameter "use_sparse_matrix=1". The motivation behind developing the sparse matrix option is that high resolution spectral peaks are sufficiently resolved such that hundreds of mass bins between peaks contain no intensity values when using a small mass bin size. These empty bins reserve large blocks of memory without contributing to the analysis. Many of the empty bins are removed using a two-dimensional sparse matrix to represent each spectrum. The first dimension

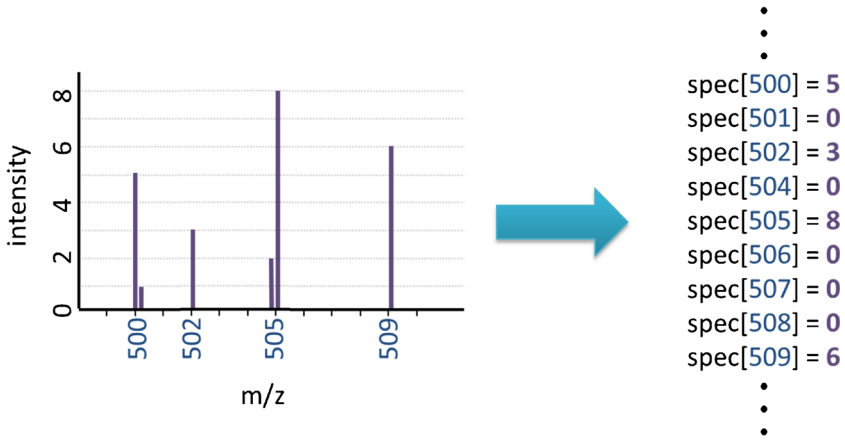



**Figure 1.** A depiction of the representation of a mass spectrum in an array format

divides a spectrum into broad segments, and the second dimension divides each segment into a series of bins sized according to the "fragment_bin_tol" parameter setting. One hundred bins per segment was determined to be optimal (see Supplemental Materials for supporting data). If all the bins in a segment do not contain any peak information, the entire segment is released and the memory is not used. See Figure 2 for a visual depiction of the sparse matrix data representation. During peak matching, a simple hash function converts each requested *m/z* value to the two-dimensional coordinates of the sparse matrix. If the segment requested by the first coordinate is *null*, indicating all bins would return a *null* value, the process moves to the next *m/z* value. If the segment is not *null*, the value in the bin represented by the second coordinate is returned. The overhead on the whole process occurs in two places: (1) a one-time overhead converting the full matrix to sparse matrix prior to searching, and (2) a small amount of additional time required to perform the hash operation for each m/z value being compared.

## *Spectrum Batches*

Despite the memory savings with the sparse matrix representation, there exist high resolution LC-MS/MS data sets that are still too large to fit into the memory of some desktop computers. The "spectrum_batch_size" parameter is a second option in Comet developed to address memory use; this search option can be used independently or in conjunction with "use_sparse_matrix." For a typical search, Comet loads all spectra into memory, performs all processing on the spectral data, and searches all spectra against all peptides in a single pass through the sequence database. This is the optimal approach to searching the data because the sequence database is parsed just once and fragment ions are calculated just once for each candidate peptide and scored against every relevant query spectrum. When a non-zero "spectrum_batch_size" is specified, Comet will iteratively load that many spectra at a time, perform a search through the database, report results, and repeat the process with the next batch of spectra until every input spectrum has been searched. By only loading a subset of

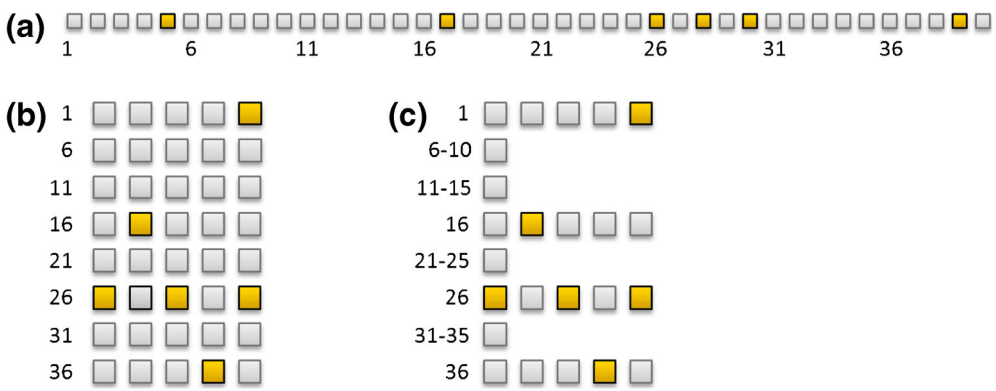



**Figure 2.** Illustration of the sparse matrix format. (**a**) Linear representation of a spectrum array. Each box represents a bin: gray boxes are empty and yellow boxes have a non-zero value. (**b**) Two-dimensional representation of the linear array. (**c**) The sparse matrix is made by freeing memory for each row (segment) where all bins are null. A single bin remains at the start of the row to indicate the null pointer representation of the segment. Panel (**c**) shows a 40% savings in memory usage. Actual savings vary for each spectrum depending on bin size, scan range, signal density, and frequency of empty bins

spectra at a time, the memory use is proportionally reduced. The tradeoff is an increase in search times because of the redundant passes through the sequence database and redundant fragment ion calculations of candidate peptides for each batch. To evaluate the effects of both the "use_sparse_matrix" and "spectrum_batch_size" parameters, a high-resolution LC-MS/MS run containing over 18,000 MS/MS spectra was searched against yeast and human sequence databases. Variable modification of 15.9949 on methionine was applied to the yeast search and an additional 79.966331 on serine, threonine, and tyrosine was applied to the human database search. Both low resolution and high resolution MS/MS search settings were evaluated. The "spectrum_batch_size" parameter was set to 3,000 when applied. See Supplemental Materials for the details on the search parameters, sequence databases, and raw file. The search times results, tabulated in Table 1, indicate that the new sparse matrix format is just as fast as the default array format. For the low resolution search settings, the memory use is minimal and is not a bottleneck for both the default array and sparse matrix formats. But for the high resolution search settings, the new sparse matrix format exhibits a significant 7- to 8-fold savings in memory use. Coupled with no degradation in search times, it is recommended that this new sparse matrix implementation always be applied going forward.

## *Search Speed and Memory Use Versus Spectrum Batch Size*

As mentioned previously, searches run most efficiently by loading as many MS/MS spectra at a time into the computer's memory. Figure 3 shows memory use and run times for a query set with 50,000 MS/MS spectra searched against a human sequence database with the following search parameters: four search threads, 20 ppm precursor tolerance with isotope error option on, full tryptic search allowing two missed cleavages, oxidized methionine variable modification, carbamidomethyl cysteine as a static modification, and using the sparse matrix option. Both high resolution and low resolution fragment ion search settings are evaluated. Actual memory use depends on a number of factors but the data presented in Figure 3 can be used as a guide for determining memory use as a function of the "spectrum_batch_size" parameter. The plots indicate that one should run searches with batches of at least 10,000 spectra at a time for optimum search speed. Going to even larger spectrum batch size values is helpful but additional gains in search speed plateau quickly.

Given the rapidly increasing data acquisition throughput of modern instrumentation, the ability to process much larger MS/MS datasets is a necessity for modern search engines. Comet is well suited to handle this use case given the two developments described here: the sparse matrix format for improved memory efficiency and batch searching to facilitate iteratively analyzing extremely large files. Comet will run well on any modern computer where search throughput is directly related to the CPU speed, core count, and, to a lesser extent, memory size. The fast cross-correlation score is a spectrum-specific analysis where every spectrum is processed independently of every other spectrum. So there are no inherent issues to searching extremely large queries beyond simply having to process more spectra. The benchmark run times shown in Figure 3, using four cores of a 2012-era Intel CPU, should assist users in defining computer/server configurations suitable for the processing throughput desired (e.g., double the CPU core count to double the search throughput).

## *Impact of the Fragment Bin Offset Parameter*

Comet has a parameter, "fragment_bin_offset," that is important but not well understood. The bin offset defines the alignment of the fragment mass bins relative to the fragment ion peaks. Finding the mass bin index with an offset is accomplished using the equation "$idx=(\text{int})((m/w)+offset)$" where *idx* is the array index, *m* is the mass, *w* is the bin width, and *offset* is the bin offset. The importance of the bin offset parameter is illustrated in Figure 4. In Figure 4a, all fragment ions from the MS/MS spectra of a high-resolution LC-MS/MS run were summed together and plotted. The figure displays the periodicity in fragment ion and peptide *m/z*, clearly showing discrete locations in the mass range where peaks exist and where there are no signals (i.e. forbidden zones [19]), where there are no peptide or fragment *m/z* at those values because of the elemental composition of amino acid masses. The smaller interleaved peaks are due to doubly charged ions. In Figure 4b, the blue lines depict the edges of the 1.0005 mass bins where the bin offset is set to 0. In the original implementation of SEQUEST, this is approximately where the bin edges lie as the concept of the bin offset did not exist in that version nor were high resolution MS/MS spectra common at that time. The bin edges at the blue lines in Figure 4b are at the least optimal locations

**Table 1.** Run times (in minutes:seconds) and memory use for a combination of default, sparse matrix, batch size, and fragment bin width options. Memory use is a function of the bin width setting plus the number of input spectra and does not vary with the sequence database size. Results indicate the new sparse matrix format performs as fast as the default array format with the added benefit of significant memory savings for high resolution search settings

| Database | Bin width | Default array | Sparse matrix | Default array + batch size | Sparse matrix + batch size |
|---|---|---|---|---|---|
| Yeast | 1.0005 | 1:56 | 1:59 | 2:32 | 2:32 |
| Yeast | 0.02 | 2:32 | 2:29 | 2:59 | 2:57 |
| Human | 1.0005 | 17:03 | 17:50 | 22:15 | 22:41 |
| Human | 0.02 | 18:56 | 18:41 | 23:27 | 22:23 |
| | 1.0005 | 1.2 GB | 1 GB | 0.3 GB | 0.2 GB |
| | 0.02 | 27 GB | 3.4 GB | 5.6 GB | 0.8 GB |

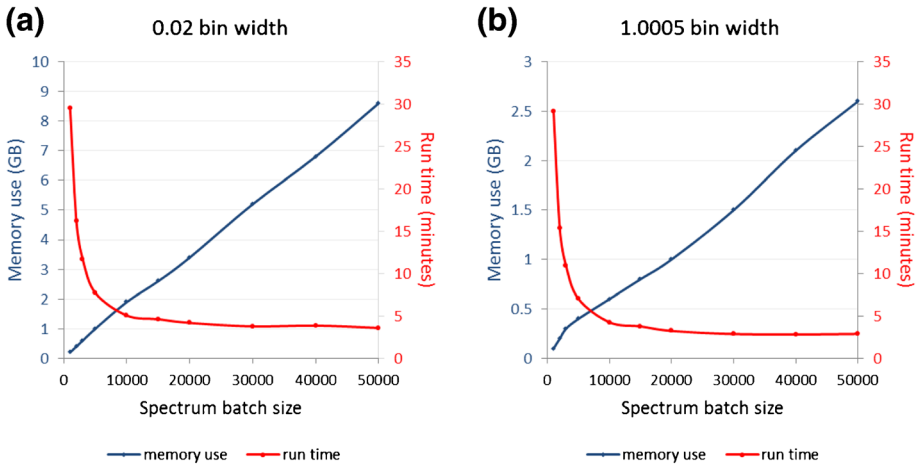



**Figure 3.** Run time and memory use as a function of spectrum batch size (with sparse matrix on). With both high resolution (**a**) and low resolution (**b**) MS/MS settings, run times improve dramatically as the spectrum batch size parameter is increased from 1,000 to 10,000. Memory use increases linearly as spectrum batch size increases

because they split the major peaks. Minor fluctuations in mass measurements, or instrumentation that is not properly calibrated, can result in a peak being assigned inconsistently to either bin. Figure 4c illustrates the location of what might be considered more optimal bin edges using a 0.5 offset where the major peaks are centered in each respective bin.

To demonstrate the effects of Comet's "fragment_bin_offset" parameter on search results, we analyzed two LTQ Orbitrap Velos runs (human ES cell lysates, see Supplemental Materials for details). A human sequence database was searched and decoy peptides were analyzed using the "decoy_search=1" parameter option, which generates on-the-fly decoy peptides and scores them in competition with the target peptides as if searching a concatenated target-decoy sequence database (see Supplemental Materials for details of the decoy peptide generation). The "fragment_bin_tol" parameter was set to 1.0005, and the bin offsets were varied from 0.0 to 0.9 in 0.1 increments. The results of this analysis are shown in Figure 5, in which the x-axis is q-value [20] and the y-axis is the number of target peptide-spectrum-matches. The best performing offset value is 0.4, whereas the 0.0, 0.1, and 0.2 offsets perform poorly. This analysis and others (data not shown) indicate that a setting of "fragment_bin_offset=0.4" consistently performs well when "fragment_bin_tol" is set to 1.0005. From a practical standpoint, users who do not want to tinker with optimizing parameters for each dataset should simply apply the following parameters for low resolution MS/MS data:

fragment_bin_tol = 1.0005
fragment_bin_offset = 0.4
theoretical_fragment_ion = 1

and for high resolution MS/MS data:

fragment_bin_tol = 0.02
fragment_bin_offset = 0.0
theoretical_fragment_ion = 0

As demonstrated in Figure 4, the choice of bin size and bin offset can make a big impact on the resulting spectral representation and search scores. Note that the bin width is related to but inherently different from a classic fragment mass tolerance setting. The effect of varying the bin width is not the same as varying a fragment mass tolerance using the same values. The bin width choice, along with the bin offset value, will define where the bin edges lie but this does not guarantee that they are centered on each fragment peak even if using values greater than the instrument mass accuracy. Small variations of the bin width will cause the bin edges to end up in suboptimal locations across many regions of the spectrum.

## *Theoretical Peaks Shape and Flanking Peaks*

The "theoretical_fragment_ion" parameter instructs Comet whether or not to include signal from the flanking bins in the cross-correlation calculation. In the original implementation of SEQUEST, fragment ions in the theoretical spectrum have reconstructed peaks with an intensity of 50 at the mass bin corresponding to the fragment ion mass and peaks of intensity 25 at the flanking bins. Adding the flanking peaks was meant to generate a peak shape that mimicked the wide peaks of the low resolution data at that time. The "theoretical_fragment_ion" parameter controls whether or not to incorporate these flanking peaks in the current cross-correlation calculation. With a wide bin size of 1.0005, adding signal from the flanking bins performs poorly compared with leaving off the flanking peaks. But with narrow bin widths, contributions from the flanking peaks do improve identification rates (data not shown).

## *Modification Options*

Searching for post-translational modifications is routinely applied in MS/MS analysis. Whether it is the addition of, nominally, 57 Da to cysteine to account for the chemical derivative

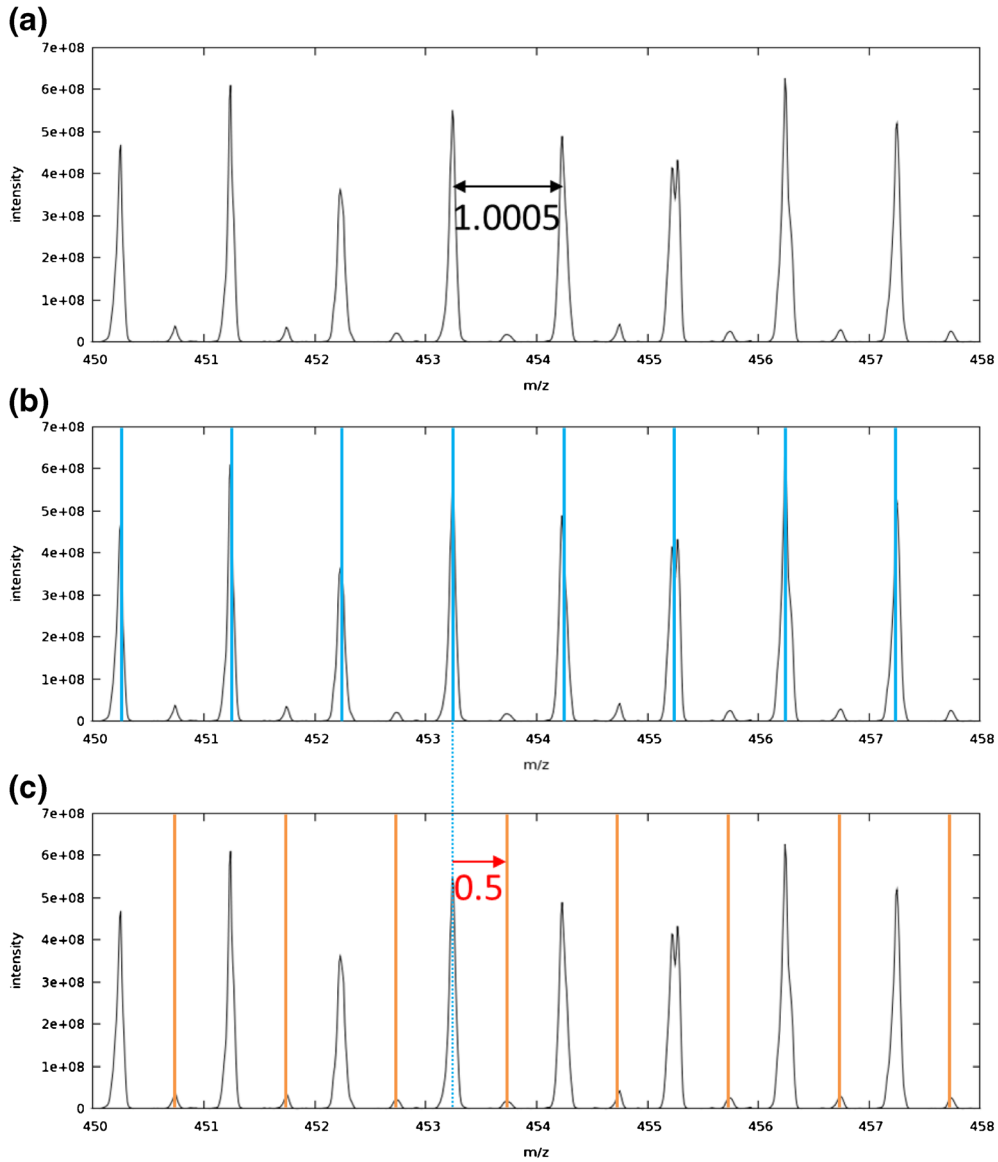



**Figure 4.** (**a**) Depicts the periodicity of peptide fragment peaks by summing all fragment ion masses from all MS/MS scans of a high-resolution LC-MS/MS run; (**b**) depicts where the bin edges are located at a 0 bin offset setting; (**c**) depicts bin edges where the major peaks are centered in the middle of each bin. This corresponds to a bin offset of 0.5, which reflects the starting edge of the bin beginning at 50% of the bin width. Note the small peaks in these plots are due to doubly charged ions

(carbamidomethylation) of the alkylation process with iodoacetamide, or the addition of 16 Da for methionine oxidation, which is common as a sample preparation artifact, the vast majority of database searches will include some form of modification analysis as default practice. Comet release 2014.02 introduced the ability to limit variable modifications to the N- or C-terminus of proteins, or a fixed number of amino acid positions from each terminus. For example, with extracellular proteins, a portion of the N-terminus may be cleaved off, exposing the new N-terminal residue to modification such as methionine cleavage and N-terminal acetylation [21]. The structure of each variable modification parameter in Comet is complex as documented in Figure 6. However, the complexity does allow one to perform more refined modification analysis. For example, to analyze protein N-terminal acetylation where the N-terminus of the protein may not contain the first few amino acids (up to 25 beyond the terminal residue) of the predicted gene product, the modification parameter would be set to:

variable_mod01 = 42.010565 K 0 1 25 0 0

In the current release, the terminal distance constraint option was extended to allow the terminal constraint to be applied to either the protein termini or peptide termini. This extension allows for the specification of a modification on particular residues only if they exist at the terminal position of each

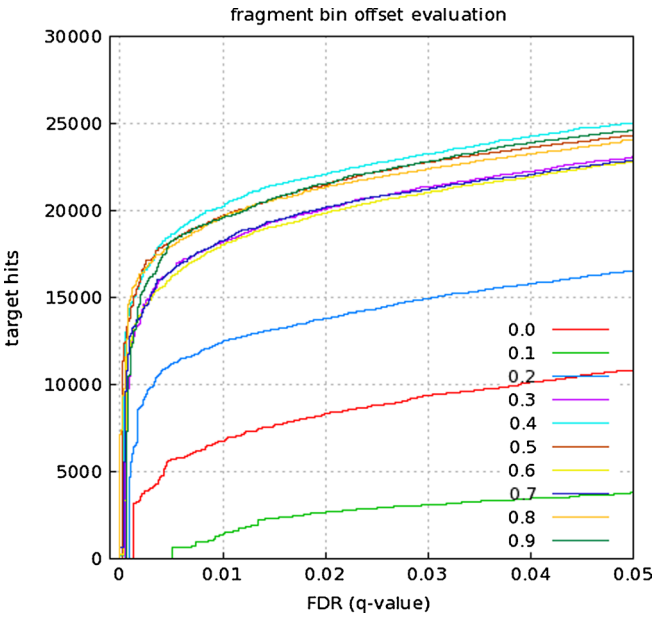



**Figure 5.** Plot of q-value versus number of target hits while varying the "fragment_bin_offset" parameter from 0.1 to 0.9 in 0.1 increments. Offset 0.4 give best identification performance whereas offsets 0.0, 0.1, and 0.2 perform much worse than the other offset settings

peptide (or within some number of amino acids from the termini). This enables Comet to now search for modifications such as pyroglutamate, also known as pyrrolidone carboxylate, which is a cyclic amino acid found on the N-termini of some proteins and peptides [22]. This modification is so common that early versions of X!Tandem [23] prior to version 2010.01.01.1 included it by default on all searches without a mechanism to turn it off. In Comet 2015.01, these modifications can now be specified using the parameter entries below. Note the '0' in the fifth field specifies that only the terminal residue can be modified by setting the terminal distance to 0; the '2' in the sixth field specifies that the terminal distance constraint applies to the N-terminal position of each peptide. So glutamine, glutamic acid, and carbamidomethylated cysteine can be modified only if they appear at the N-terminal position of each peptide using the following parameter settings:

variable_mod01 = −17.026549 QC 0 1 0 2 0
variable_mod02 = −18.010565 E 0 1 0 2 0

Additionally, Comet version 2015.01 includes two new options to force the analysis of modified peptides where only peptides that contain a variable modification will be analyzed. The "require_variable_mod" parameter will require that any analyzed peptide has to have a variable modification. Similarly, each individual variable modification parameter now has an extra seventh parameter field that can be used to require that specific modification be present in peptides that are analyzed. With these additional parameter options, any specific sets of variable modifications can be forced to be present or at least one of any of the variable modifications can be forced to be present. For example, the following parameters will search for phosphorylation and acetylation

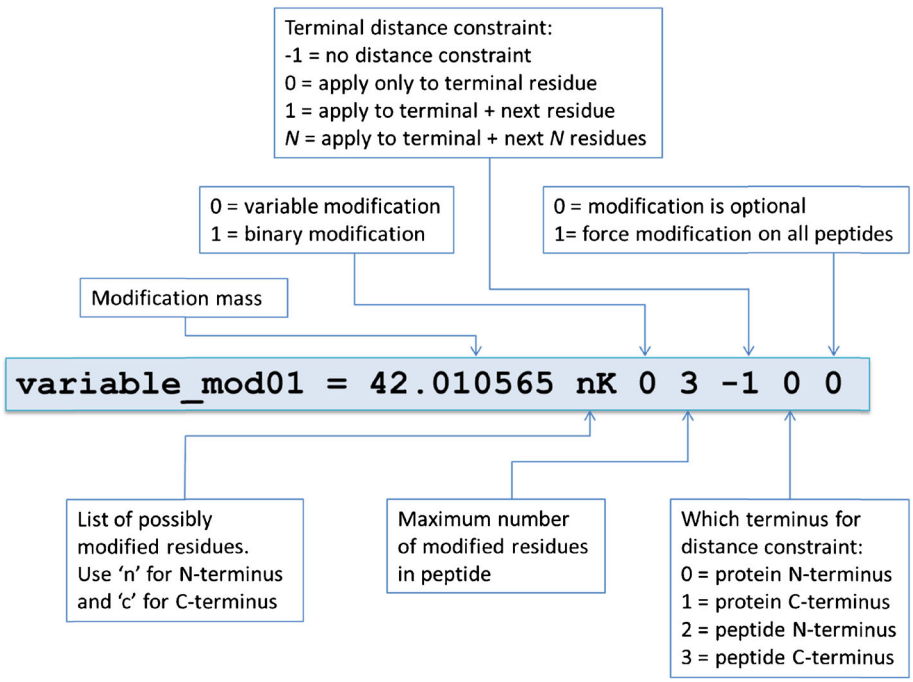



**Figure 6.** Format of the variable modifications parameter entry. There are seven fields that control how the variable modifications are defined and applied

and require that any analyzed peptide be modified with at least one of these modifications:

```
variable_mod01 = 79.966331 STY 0 3 -1 0 0
variable_mod02 = 42.010565 nK 0 3 -1 0 0
require_variable_mod = 1
```

Alternatively, to require all analyzed peptides to be phosphorylated but allow for the peptide to also be acetylated, can be accomplished using:

```
variable_mod01 = 79.966331 STY 0 3 -1 0 1
variable_mod02 = 42.010565 nK 0 3 -1 0 0
```

The current variable modification support allows for flexibility in how and where modifications are applied. Up to nine variable modifications can be specified, each of which can be applied to multiple residues, and more than one (actually up to nine) variable modification can be specified for the same amino acid. The concept of a binary modification, where all residues present in a peptide must be all modified or all unmodified, is currently supported on a per-modification parameter basis. However, binary modifications across modification parameters, such as heavy lysine and heavy arginine in a SILAC experiment requiring specification of different modification masses using separate variable modification parameters, is a feature that will be implemented in a future release. Additionally, N15 metabolic labeling currently requires two separate searches, one normal and one where all amino acid masses are statically modified to their N15 counterparts; a future release will support N15 light and heavy searches directly.

## *Comparison of Comet Versus SEQUEST Cross-Correlation Score*

Although they share a similar heritage, Comet and the University of Washington's (UW) academic version of SEQUEST have undergone nearly two decades of independent development from the commercial SEQUEST maintained by Thermo Scientific. A comparison of search scores depicted in Figure 7 displays the correlation between Comet versus UW SEQUEST (2012.01 rev. 6), Comet versus Thermo SEQUEST (Proteome Discoverer 1.2), and finally UW SEQUEST versus Thermo SEQUEST. The plots depict a pairwise distribution of cross-correlation scores where the same peptide is identified by each pair of tools. There is high correlation between Comet and UW SEQUEST cross-

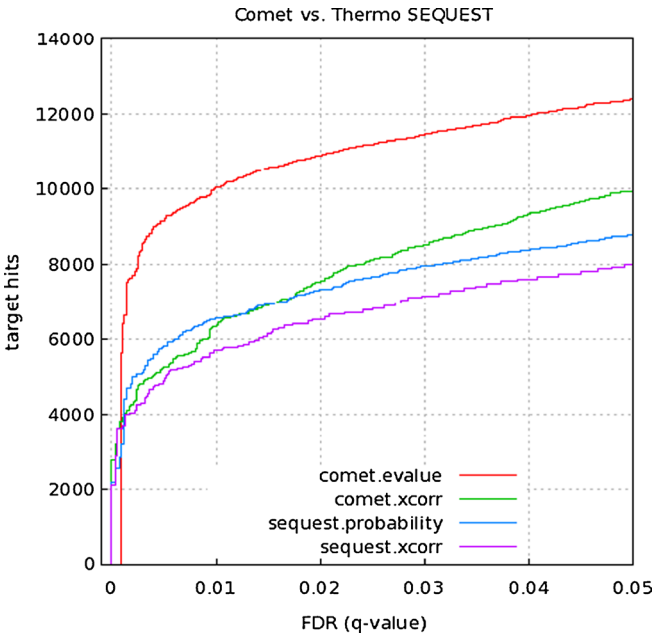



**Figure 8.** ROC plot of Comet and Thermo SEQUEST identification performance. Comet's expectation value and cross-correlation scores outperform the Thermo SEQUEST probability and cross-correlation score counterparts for this given dataset and search settings

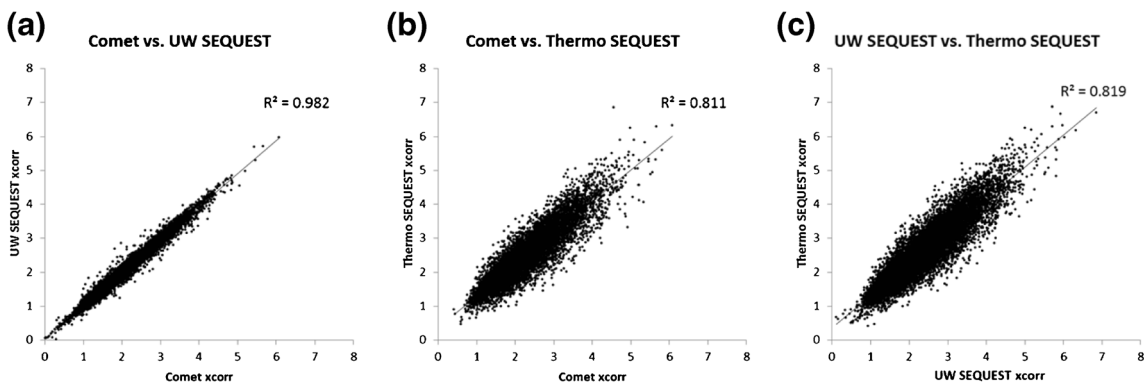



**Figure 7.** Pairwise distribution of cross-correlation scores between Comet (2015.01 rev. 1), academic SEQUEST (2012.01 rev. 6), and Thermo SEQUEST (Proteome Discoverer 1.2). Scatter plots of cross-correlation scores for those spectra that identify the same peptide in each pair of tools. Comet and the University of Washington's academic version of SEQUEST exhibit very high correlation as they shared the same code base in 2011. Both of these tools correlate a bit less with Thermo SEQUEST as that commercial tool diverged in the late 1990s

correlation scores (Figure 7a, $R^2$=0.98) as expected given that these tools were the same code base when the software was made open-sourced and renamed in 2011. The correlation between Comet versus Thermo SEQUEST (Figure 7b, $R^2$=0.81) and UW SEQUEST versus Thermo SEQUEST (Figure 7c, $R^2$=0.82) show that internal implementation details have diverged over the many years since these tools shared the same code. Without access to proprietary source code, it is difficult to pinpoint exact differences in the tools. The major differences are speculated to be changes to spectral processing and the implementation of bin offsets.

To compare search performance, an LTQ Orbitrap Velos file was searched against a human target-decoy database. A comparison of identification performance of Comet and Thermo SEQUEST is depicted in the receiver-operator-characteristic (ROC) plot displayed in Figure 8. Search performance using Comet's E-value (comet.evalue), cross-correlation score (comet.xcorr), Thermo SEQUEST probability score (sequest.probability), and cross-correlation score (sequest.xcorr) are plotted as FDR (q-value) versus the number of target hits. The data file, search parameters, and sequence database are documented in the Supplemental Materials. Comet's E-value significantly outperforms SEQUEST's probability score and Comet's cross-correlation score performs better than its counterpart. However, identification performance can vary significantly with search parameter settings so it is possible that Thermo SEQUEST is not being searched optimally in our hands with the applied parameters; this plot is a snapshot of identification performance for the search parameters applied. Additionally, newer versions of Thermo's commercial package exists, which may exhibit improved identification performance than that demonstrated here. Machine-learning post-processing tools that do not rely on a single search engine score will also mitigate the performance differences shown.

## Conclusion

Quite often, the internal details of database search algorithms are a mystery to those that use the tools daily, even those that are open sourced or have been in use for decades. While Comet stems from the academic version of SEQUEST that has existed for many years, it is still being actively developed and extended on a regular basis. Improvements include adding search features, optimizing the code for speed improvements, and tweaking the core identification routines. Changes in the current release of Comet include implementation of more flexible modification options, a new sparse matrix data structure, multi-threaded optimization, and better search progress reporting. mzXML, mzML, ms2, and native Thermo RAW files are supported input formats whereas pepXML, SQT, Percolator TSV, and text files are supported output formats. Since its initial release in 2012, Comet has had five subsequent major releases, has been directly downloaded well over 1,000 times, and is incorporated into a number of larger software projects. These include Crux [4], Chorus [24], PatternLab [25], ProHits [26], LabKey Server [27], PeptideShaker [28], MASSyPup [29], and the Trans-Proteomics Pipeline [30], among others. Users are encouraged to access Comet from within any one of these tools. Documentation and direct Comet download are available at http://comet-ms.sourceforge.net.

## Acknowledgments

The authors acknowledge support for this work NIH awards R01GM096306 (to W.S.N.) and P41GM103533 (to W.S.N. and M.J.M.). This work is supported in part by the University of Washington's Proteomics Resource (UWPR95794). The authors thank Mike Riffle and Vagisha Sharma for manuscript feedback. J.K.E. thanks Nathan D. Camp for in vivo support.

## References

1. Eng, J.K., McCormack, A.L., Yates III, J.R.: An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. J. Am. Soc. Mass Spectrom. **5**, 976–989 (1994)
2. Kapp, E., Schütz, F.: Overview of Tandem Mass Spectrometry (MS/MS) Database Search Algorithms. Curr. Protoc. Protein Sci. **49**, 25.2:25.2.1–25.2.19 (2007)
3. Lundgren, D.H., Martinez, H., Wright, M.E., Han, D.K.: Protein Identification Using Sorcerer 2 and SEQUEST. Curr. Protoc. Bioinformatics **28**, 13.3:13.3.1–13.3.21 (2009)
4. Park, C.Y., Klammer, A.A., Käll, L., MacCoss, M.J., Noble, W.S.: Rapid and accurate peptide identification from tandem mass spectra. J. Proteome Res. **7**, 3022–3027 (2008)
5. Diament, B.J., Noble, W.S.: Faster SEQUEST searching for peptide identification from tandem mass spectra. J. Proteome Res. **10**, 3871–3879 (2011)
6. Xu, T., Venable, J., Park, S.K., Cociorva, D., Lu, B., Liao, L., Wohlschlegel, J., Hewel, J., Yates III, J.R.: ProLuCID, a fast and sensitive tandem mass spectra-based protein identification program. Mol. Cell. Proteom. **5**, S174 (2006)
7. Faherty, B.K., Gerber, S.A.: MacroSEQUEST: efficient candidate-centric searching and high-resolution correlation analysis for large-scale proteomics data sets. Anal. Chem. **82**, 6821–6829 (2010)
8. Milloy, J.A., Faherty, B.K., Gerber, S.A.: Tempest: GPU-CPU computing for high-throughput database spectral matching. J. Proteome Res. **11**, 3581–3591 (2012)
9. Eng, J.K., Jahan, T.A., Hoopmann, M.R.: Comet: an open-source MS/MS sequence database search tool. Proteomics **13**, 22–24 (2013)
10. Yates III, J.R., Eng, J.K., McCormack, A.L., Schieltz, D.: Method to correlate tandem mass spectra of modified peptides to amino acid sequences in the protein database. Anal. Chem. **67**, 1426–1436 (1995)
11. Swaney, D.L., Beltrao, P., Starita, L., Guo, A., Rush, J., Fields, S.: Global analysis of phosphorylation and ubiquitylation cross-talk in protein degradation. Nat. Meth. **10**, 676–682 (2013)
12. Chavez, J.D., Weisbrod, C.R., Zheng, C., Eng, J.K., Bruce, J.E.: Protein interactions, post-translational modifications and topologies in human cells. Mol. Cell. Proteom. **12**, 1451–1467 (2013)
13. Available at: https://github.com/mhoopmann/mstoolkit. Accessed 1 April 2015. Subject: file parsing library.
14. Martens, L., Hermjakob, H., Jones, P., Adamski, M., Taylor, C., States, D., Gevaert, K., Vandekerckhove, J., Apweiler, R.: PRIDE: the proteomics identifications database. Proteomics **5**, 3537–3545 (2005)
15. Phanstiel, D.H., Brumbaugh, J., Wenger, C.D., Tian, S., Probasco, M.D., Bailey, D.J., Swaney, D.L., Tervo, M.A., Bolin, J.M., Ruotti, V., Stewart, R., Thomson, J.A., Coon, J.J.: Proteomic and phosphoproteomic comparison of human ES and iPS cells. Nat. Methods **8**, 821–827 (2011)
16. Eng, J.K., Fischer, B., Grossmann, J., Maccoss, M.J.: A fast SEQUEST cross correlation algorithm. J. Proteome Res. **7**, 4598–4602 (2008)

17. Howbert, J.J., Noble, W.S.: Computing Exact *p*-values for a Cross-correlation Shotgun Proteomics Score Function. Mol. Cell. Proteom. **13**, 2467–2479 (2014)
18. Klammer, A.A., Park, C.Y., Noble, W.S.: Statistical calibration of the SEQUEST XCorr function. J. Proteome Res. **8**, 2106–2113 (2009)
19. Nefedov, A.V., Mitra, I., Brasier, A.R., Sadygov, R.G.: Examining troughs in the mass distribution of all theoretically possible tryptic peptides. J. Proteome Res. **10**, 4150–4157 (2011)
20. Storey, J.D., Tibshirani, R.: Statistical significance for genomewide studies. Proc. Natl. Acad. Sci. U. S. A. **100**, 9440–9445 (2003)
21. Polevoda, B., Sherman, F.: N-terminal acetyltransferases and sequence requirements for n-terminal acetylation of eukaryotic proteins. J. Mol. Biol. **325**, 595–622 (2003)
22. Foreman, F.W.: The transformation of glutaminic acid into l-pyrrolidonecarboxylic acid in aqueous solution. Biochem. J. **8**, 481–493 (1914)
23. Fenyö, D., Eriksson, J., Beavis, R.: Mass Spectrometric Protein Identification Using the Global Proteome Machine. Methods Mol. Biol. **673**, 189–202 (2010)
24. Available at: https://chorusproject.org. Accessed 1 April 2015. Subject: data repository
25. Carvalho, P., Fischer, J., Chen, E., Yates, J., Barbosa, V.: PatternLab for proteomics: a tool for differential shotgun proteomics. BMC Bioinformat. **9**, 316 (2008)
26. Liu, G., Zhang, J., Larsen, B., Stark, C., Breitkreutz, A., Lin, Z.Y., Breitkreutz, B.J., Ding, Y., Colwill, K., Pasculescu, A., Pawson, T., Wrana, J.L., Nesvizhskii, A.I., Raught, B., Tyers, M., Gingras, A.C.: ProHits: integrated software for mass spectrometry-based interaction proteomics. Nat. Biotech. **28**, 1015–1017 (2010)
27. Rauch, A., Bellew, M., Eng, J., Fitzgibbon, M., Holzman, T., Hussey, P., Igra, M., Maclean, B., Lin, C.W., Detter, A., Fang, R., Faca, V., Gafken, P., Zhang, H., Whiteaker, J., States, D., Hanash, S., Paulovich, A., McIntosh, M.W.: Computational proteomics analysis system (CPAS): an extensible, open-source analytic system for evaluating and publishing proteomic data and high throughput biological experiments. J. Proteome Res. **5**, 112–121 (2005)
28. Vaudel, M., Burkhart, J.M., Zahedi, R.P., Oveland, E., Berven, F.S., Sickmann, A., Martens, L., Barsnes, H.: PeptideShaker enables reanalysis of MS-derived proteomics data sets. Nat. Biotech. **33**, 22–24 (2015)
29. Winkler, R.: MASSyPup—an 'Out of the Box' solution for the analysis of mass spectrometry data. J. Mass Spectrom. **49**, 37–42 (2014)
30. Keller, A., Eng, J., Zhang, N., Li, X.J., Aebersold, R.: A uniform proteomics MS/MS analysis platform utilizing open XML file formats. Mol. Syst. Biol. **1**, 2005.0017 (2005)