Bioinformatics, 32(10), 2016, 1518–1526
doi: 10.1093/bioinformatics/btw023
Advance Access Publication Date: 21 January 2016
Original Paper

Systems biology

MS-REDUCE: an ultrafast technique for reduction of big mass spectrometry data for high-throughput processing

Muaaz Gul Awan[1] **and Fahad Saeed**[1,2,*]

[1]Department of Electrical and Computer Engineering and [2]Department of Computer Science, Western Michigan University, Kalamazoo, MI 49008, USA

*To whom correspondence should be addressed.

Associate Editor: Jonathan Wren

Received on 27 October 2015; revised on 28 December 2015; accepted on 12 January 2016

Abstract

Motivation: Modern proteomics studies utilize high-throughput mass spectrometers which can produce data at an astonishing rate. These big mass spectrometry (MS) datasets can easily reach peta-scale level creating storage and analytic problems for large-scale systems biology studies. Each spectrum consists of thousands of peaks which have to be processed to deduce the peptide. However, only a small percentage of peaks in a spectrum are useful for peptide deduction as most of the peaks are either noise or not useful for a given spectrum. This redundant processing of *non-useful* peaks is a bottleneck for streaming high-throughput processing of big MS data. One way to reduce the amount of computation required in a high-throughput environment is to eliminate non-useful peaks. Existing noise removing algorithms are limited in their data-reduction capability and are compute intensive making them unsuitable for big data and high-throughput environments. In this paper we introduce a novel low-complexity technique based on classification, quantization and sampling of MS peaks.

Results: We present a novel data-reductive strategy for analysis of Big MS data. Our algorithm, called MS-REDUCE, is capable of eliminating noisy peaks as well as peaks that do not contribute to peptide deduction *before* any peptide deduction is attempted. Our experiments have shown up to 100× speed up over existing state of the art noise elimination algorithms while maintaining comparable high quality matches. Using our approach we were able to process a million spectra in just under an hour on a moderate server.

Availability and implementation: The developed tool and strategy has been made available to wider proteomics and parallel computing community and the code can be found at https://github.com/pcdslab/MSREDUCE

Contact: fahad.saeed@wmich.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Mass spectrometry (MS) is an analytical chemistry technique which is used for determining the type and amount of constituents of a mixture. MS has found its application in the field of biomedical research. Among all the applications of MS in biology and medicine (Finehout and Lee, 2003) protein identification and quantization has proved to be the most widely used. MS based proteomics (Aebersold and Mann, 2003) is very frequently used for profiling of exosomes (Pisitkun *et al.*, 2004), toxicological screening (K, 2013), evolutionary biology (Zhao *et al.*, 2012) and numerous other applications (Awan and Saeed, 2015). Wide variety of computational techniques such as estimation of false positive rates (Du *et al.*, 2008), protein quantification from large datasets (Hoffert *et al.*, 2006), phosphopeptide filtering (Jiang *et al.*, 2010), phosphorylation site assignments (Saeed *et al.*, 2013b), spectrum-to-peptide matching (Eng *et al.*, 1994), (Perkins *et al.*, 1999) and denovo peptide identification (Dancik *et al.*, 1999) are required to make this MS data useful.

With the introduction of modern mass spectrometers such as Thermo Orbitrap, thousands of spectra can be generated in just a single run of experiment (AS *et al.*, 2014). An MS2 spectrum consists of mass-to-charge ratio and associated intensities for each peak depicting their abundance in the sample under consideration. On an average total number of peaks for one spectrum may range up to 4000 (Awan and Saeed, 2015) and for 60k human proteins the number of distinct peaks that need to be compared is close to 240 million (assuming that there is no redundancy). This number is just for a single human proteome and with projects like Peptide Atlas the number of distinct human observations are close to 35 000 which makes the total number of peaks equal to 8.4×10^{12}. Note that this number does not include other species, distinct experimental conditions or novel post-translational modifications which exponentially increases the number of peaks that needs to be processed.

The current computational analysis techniques have not been designed for such massive datasets. The current peptide identification techniques (e.g. Sequest, Mascot; Eng *et al.*, 1994; Perkins *et al.*, 1999) assume that each peak that is encountered is useful in making peptide deductions. This leads to processing much more number of peaks than are necessary to make a peptide deduction (Awan and Saeed, 2015; Ding et al., 2009; Mujezinovic *et al.*, 2006; Saeed *et al.*, 2013a). The processing of peaks that are noise and/or do not contribute in deduction of peptides makes the processing of these large datasets time consuming. We assert that in order to process big MS data we should be able to eliminate noisy peaks and the peaks that do not contribute to peptide deduction *before* an in-depth analysis of the spectra. This will clearly result in faster processing of the MS/MS spectra and will save overhead for peptide searches by reducing the number of peaks to be analyzed. Only processing the peaks that are useful rather than performing intensive per-peak-computations will result in tremendous time and space-advantages. To the best of authors knowledge there is no algorithm available which can perform the noise removal function without performing an in-depth analysis on spectra. Further, we are not aware of any procedure that can eliminate non-noisy and yet non-essential peaks that do not contribute to peptide deduction.

In this paper we introduce a novel algorithm, called MS-REDUCE, for ultrafast reduction of MS/MS data in pre-processing stage. The proposed algorithm is a low-complexity procedure based on random sampling, approximate classification and quantization making it highly scalable with increasing number of spectra. Further, user defined reduction ratio makes it suitable for a variety and sizes of MS datasets. Our experiments show peptide deduction accuracy of up to 95% with reduction in the data size of up to 70%. Our results also indicate that we are able to process 1 000 000 spectra in under 1 h on a sequential machine making it highly efficient for big datasets. Comparable reduction tools took over 3 days for the same dataset on a similar machine.

2 Literature review

Spectral pre-processing has become an essential part of the MS based proteomics in recent years. Most of the spectral pre-processing techniques have a common objective i.e. to improve the reliability of the peptide to spectral matches assigned by a peptide search engine such as Sequest or Mascot. Some of the pre-processing methods that allow better identification of peptides include spectral clustering (Saeed *et al.*, 2013a), noise reduction in spectra (Ding et al., 2009), quality assessment of spectra (Bern *et al.*, 2004) and precursor charge determination (Wu *et al.*, 2008). The prime objective of these techniques is to reduce the noise level in spectra which leads to better identification of peptide using standard search engines. It is also shown by several studies that reduction in data can also speedup the process of peptide identification in peptide search engines (Ding et al., 2009). Below we will emphasize on the application of the existing work for reduction of Big Mass Spectrometry data. Note that we are not aware of any method that allows elimination of peaks that are not noise *and* may not contribute to peptide identification; with or without significant processing of the data.

In the literature several noise reducing or spectral denoising algorithms are available. These algorithms identify the noisy peaks in a spectrum, depending upon the approach each algorithm uses these peaks are then either removed or their intensity is decreased to a certain value. Mujezinovic *et al.* (2006) presented the MS Cleaner software for removing the unwanted peaks from the spectra to facilitate the peptide search engines. Their technique provided an added advantage of data reduction. They made use of numerical analysis and signal detection approach to form four different algorithms. Each algorithm looked for multiply charged ions, isotopic clusters of peaks, periodic background noise and detection of non-interpretable spectra. However, these methods are deemed to be too compute-intensive to be used as a big data pre-processing application especially for high-throughput put environments e.g. the authors report compute time per spectrum of 0.25s while treating 53 944 spectra. Their results show a total reduction of 15% to 39% in raw data. The same authors presented an upgrade of MS Cleaner software, a version 2.0 in 2010 (Mujezinovic *et al.*, 2010). The improved software employs a new algorithm for screening the interpretable spectra. It detects the peptide ladder sequence using a fixed number of most intense peaks from each spectrum. With this upgrade they claim to have reduced the data to up to 80%. Time per spectrum for newer version has been stated about 0.02–0.08 s per spectrum depending upon the dataset used.

The method presented in Ding *et al.* (2009) consists of two steps. In first; a peak intensity adjustment takes place based upon scores obtained from five different features. In the later stage a morphological reconstruction filter is employed to remove the noisy peaks based upon their adjusted intensity in the previous stage. This algorithm is able to reduce up to 69% of data but is extremely compute intensive to be used for high-throughput or parallel processing. Our experiments show a computational time of around 3 days for 1 000 000 spectra. Two other similar algorithms can be found in Zhang *et al.* (2008) and Gentzel *et al.* (2003). Like previously discussed algorithms they also suffer from huge number of per-peak calculations. The implementation of Gentzel et al. (2003) takes approximately 1.7 s per spectrum and will take hours to process a million spectra.

A quality assessment technique for spectra has been presented in Lin *et al.* (2012). The authors estimate the probability of a spectrum being a high quality one by treating this problem as a constraint optimization problem. Their results show that a total of 63–74% of low quality spectra were removed while losing 9–10% of high quality spectra in the process. In Na and Paek (2007) a new feature has been introduced for assessment of spectral quality which is based on cumulative intensity normalization. The results show a removal of about 60% spectra with a loss of losing 2% of high quality spectra. Some other spectral quality assessment algorithms have been presented in Tabb *et al.* (2003), Bern *et al.* (2004), Purvine *et al.* (2004) and Ding *et al.* (2011). All of these algorithms take different approaches towards assessing the spectra. However, most of the approaches are compute intensive which makes them impractical and ineffective for evaluation of big datasets.

3 Proposed MS-REDUCE algorithm

In this paper we present a highly efficient dimensionality reduction technique which allows massive reduction in number of peaks per spectra and in turn decrease the overall amount of data that needs to be processed. Our work builds upon a random sampling strategy that we presented earlier (Awan and Saeed, 2015). The proposed algorithm, apart from being more accurate than previous strategies, has very low-computational complexity which makes it ideal for big data computations. Our classification and sampling strategy allows us to determine useful peaks *before* any peptide deduction calculations. Also in each stage calculations are performed on only a handful of peaks from each spectrum regardless of the size of individual spectrum. This makes processing for each spectrum a constant time operation resulting in linear-time algorithm. Here we formally introduce the problem. Notations will be introduced and defined wherever they occur first throughout the paper.

Definition 1: Let there be N number of spectra $S = \{s_1, s_2, \ldots, s_N\}$. If length of spectrum s_i is l_i then each spectrum can be represented as a series of peaks i.e. $s_i = \{p_1, p_2, p_3, \ldots, p_l\}$. Where p is a peak in a spectrum.

Definition 2: If s'_i denotes a spectrum after being processed by MS-REDUCE and the size of the processed spectrum be l'_i then R is the reduction factor such that $R = (l'_i/l_i) * 100$ for each spectrum.

Each spectrum s in S needs to be reduced to obtain s' such that both s and s' correspond to the same peptide with a high confidence value. Note that there may be cases where s and s' do not correspond to a same peptide, in that case if the peptide match for s' has a confidence value better than the threshold value to qualify for a high confidence hit then that counts as a correct hit. For example a raw spectrum s might correspond wrongly to a peptide A but after being reduced using MS-REDUCE its noise level may get lowered and the reduced spectrum s' may correspond correctly to another peptide B or vice versa. The correctness of match is determined using quality assessment method discussed in Section 5.3.

MS-REDUCE exploits the fact that about 90% of peaks in a spectrum are noisy or are not required for peptide deduction (Mujezinovic *et al.*, 2010). The sampling technique is dependent on the level of noise and intensity variation in a given spectrum. The algorithm comprises of a three stage pipeline. Each spectrum streams throught it while discarding the peaks that cannot pass through the last stage. The three stages of the pipeline are (i) Spectral Classification, (ii) Peak Quantization and (iii) Weighted Random Sampling. Figure 1 shows the proposed three stage pipeline for MS-REDUCE algorithm.

The spectral classification module is the first stage in the pipeline of MS-REDUCE. The main objective of this module is to determine an estimate of a spectrum's noise level. To this end, we present a novel metric, called Spectral Intensity Spread that allows us to bring about an approximate classification of spectra according to their noise level. The Intensity Spread of a spectrum roughly estimates how diverse the intensities of different peaks are. The module makes this plain assumption that larger the value for Intensity Spread, more noisy the spectrum is Wells *et al.* (2011).

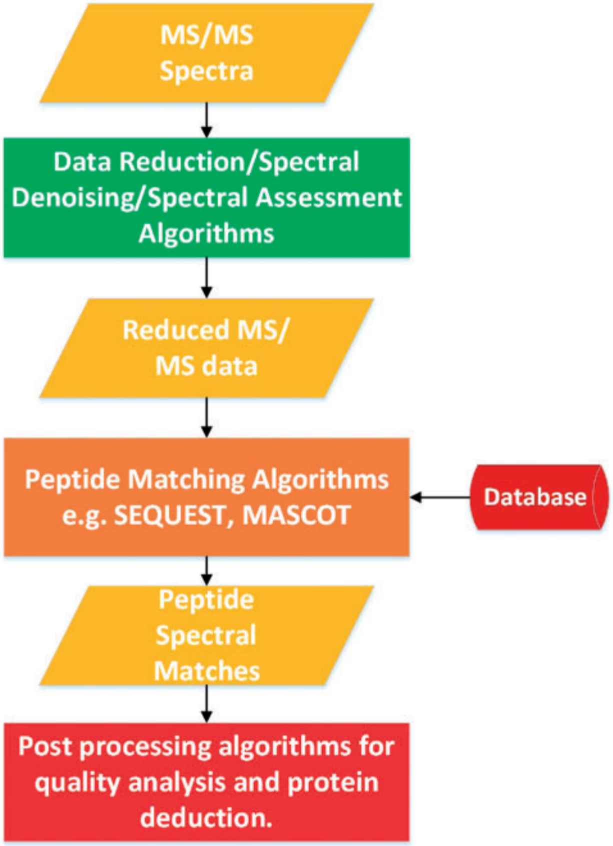
MS/MS
Spectra
Data Reduction/Spectral Denoising/Spectral Assessment Algorithms
Reduced MS/ MS data
Peptide Matching Algorithms e.g. SEQUEST, MASCOT
Database
Peptide Spectral Matches
Post processing algorithms for quality analysis and protein deduction.

Fig. 1. Figure showing the pipeline for MS-REDUCE algorithm

Once a spectrum has been assigned a class based on its estimated noise level, it is sent forward to the Spectral Quantization module. Here the spectrum is quantized into several levels along the intensity axis. The number of quantization levels depends on the class of spectra that was assigned in previous stage. A noisier spectrum is quantized into larger number of quanta. This module distributes the peaks into different groups based upon their intensity levels thus making it much simpler and faster to access peaks based on their intensity levels.

The quantized spectra is sent into the last module, where possible signal peaks are retained using random peak sampling on the quanta. The number of peaks to be retained are calculated based on the user defined reduction factor R. Weighted sampling rates are calculated for each quantum such that the sum of peaks gathered from each level equals to the percentage of peaks required. Here sampling rate is defined as the percentage of peaks to be retained in one quantization level. We give details of each module below.

3.1 Spectral classification

Most pre-processing algorithms process the spectra without any regards to the quality of spectra i.e. spectra with better signal to noise ratio are processed in the same way spectra with poor S/N ratio. This results in a wastage of resources as a lot of redundant work is performed for the spectra already having higher S/N ratio. This module takes care of this issue by classifying spectra on the basis of approximate noise content in them.

3.1.1 Intensity spread

The classification is performed by comparing each spectrums Intensity Spread with the Average Intensity Spread of the dataset.

More formally:

Definition 3: Let N be the total number of spectra in set S then $S = \{s_1, s_2, s_3, \ldots, s_n\}$ here $\mathrm{s_i}$ represents one spectrum. Then the intensity spread for spectrum s_i can be calculated as:

$$V_i = \mathrm{Max10Avg}(s_i) - \mathrm{Min10Avg}(s_i) \qquad (1)$$

where V_i is the Intensity spread of the spectrum i and $\mathrm{Max10Avg}(s_i)$ and $\mathrm{Min10Avg}(s_i)$ present the average of ten most and least intense peaks of the spectrum respectively.

Similarly Average Intensity Spread for a dataset can be calculated as:

$$V_{\mathrm{avg}} = \frac{\sum_{i=1}^{N}(\mathrm{Max10Avg}(s_i) - \mathrm{Min10Avg}(s_i))}{N} \qquad (2)$$

where

V_{avg} = Average Intensity Spread

N = number of spectra in set S

For each incoming spectrum the Intensity Spread value is calculated. As seen in Eq. (2), this calculation requires only twenty peaks from each spectrum regardless of its size.

3.1.2 Classification

Spectra are classified in four different classes depending upon how much above or below the V_{avg} their value of V lies. Details regarding the choice of number of classes can be found in Section 4.1 of supplementary materials. Classes are named in increasing numerical order; higher classes contain spectra with larger value of V and vice versa. Threshold values for V to be assigned to a particular class are determined based on each dataset's V_{avg}. More formally the threshold values for each class can be defined as follows:

Definition 4: Let x denote a class then for $x = \{1, 2, 3\}$

$$S_x = \{s_i | (x-1) * \frac{1}{4} * V_{avg} \le V_i \le x * \frac{1}{4} * V_{\mathrm{avg}}\} \qquad (3)$$

and for $x = \{4\}$:

$$S_x = \{s_i | \frac{3}{4} * V_{\mathrm{avg}} \le V_i\} \qquad (4)$$

where S_x = Class x containing spectra assigned to it.

It can be seen in Figure 2 how spectrum 1 and 2 have very different range of intensities yet they have similar spectra spread hence have been assigned the same class. Algorithm 1 in supplementary materials presents a pseudo code for the classification module.

3.2 Spectral quantization

In our proposed algorithm quantization of spectra takes place along the intensity axis. The intensity of a peak is simply compared with the upper and lower level of a quanta, if it lies within the limit, the peak is assigned to that quantum. This process provides us with different bins, each containing peaks of intensities within a specific range. The advantage of quantization is exploited in the following step, where useful peaks are just picked out from their quanta and added to the final reduced spectrum. Thus preventing the need of performing per peak computations.

3.2.1 Quantization levels

Number of quantization levels is chosen such that those spectra having wide Intensity Spread are quantized into larger number of levels while those having a narrow spread are processed using smaller number of quantization levels. Our in house experiments suggest that a spectrum with a smaller intensity spread yields no improvement if processed using larger number of quantization levels while increasing the processing time. In order to save time and space resources we use the smallest possible number of quantization levels necessary to perform the computation. Similarly the spectra with wider Intensity Spread needs more number of quantization levels to achieve similar accuracy. Classes 1, 2, 3 and 4 are assigned 5, 7, 9 and 11 levels of quantization respectively. These values have been chosen based upon an empirical study, details of which can be found in Section 4.2 of supplementary materials. The quantization process can be formally defined as:

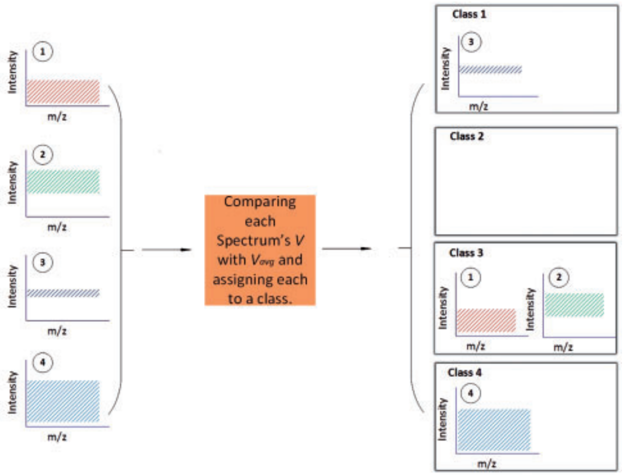
Intensity
m/z
1
2
3
4
Comparing each Spectrum's V with V_avg and assigning each to a class.
Class 1
Class 2
Class 3
Class 4

Fig. 2. Figure depicts a visual representation of classification stage. The shaded regions present the Spectral Spread (*V*), larger the shaded area larger the value of *V* and noisier the corresponding spectrum is considered

Definition 5: Let n_x be the maximum number of quantization levels for class x then we can have $n_1 = 5, n_2 = 7, n_3 = 9$ and $n_4 = 11$. q_{ij} represents the quantum j of spectrum i. Then following equations are calculated for each spectrum s_i, for each quantum j from 1 till n_x.

for $j < n_x$

$$q_{ij} = \{p | \frac{(j-1)}{n_x} * \mathrm{M10A}(s_i) \le ||p|| \le \frac{j}{n_x} * \mathrm{M10A}(s_i)\} \qquad (5)$$

for $j = n_x$

$$q_{ij} = \{p | \frac{(j-1)}{n_x} * \mathrm{M10A}(s_i) \le ||p||\} \qquad (6)$$

where j = quantization level under consideration

q_{ij} = jth quantization level of ith spectrum

n_x = number of quantization levels for class x

$||p||$ = intensity of peak p

$\mathrm{M10A}(s_i)$ = Average Intensity of 10 most intense peaks of s_i

Eqs. (5) and (6) are computed for each value of n_x ranging from 1 till n_x. The quantum number assigned to each peak represents certain characteristics e.g. quantum 1 is the lowest and it contains the least intense peaks, similarly the quantum number 11 would be the highest for class 4 spectra and would contain the most intense peaks. The quanta are equally spaced rather than being of irregular spread because about 90% of the data is redundant so the probability that

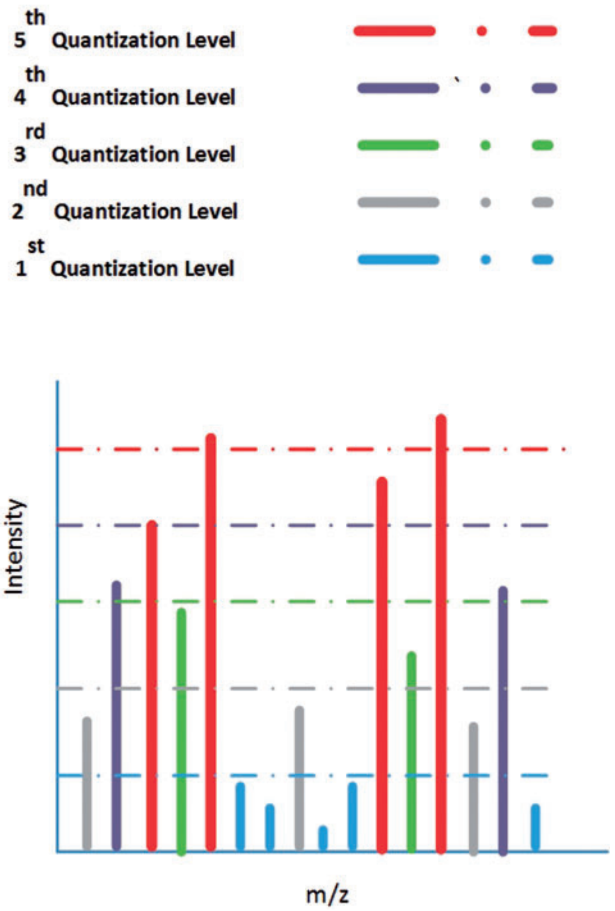

Fig. 3. Figure represents quantization of a class I spectrum with five different quantization levels. The red colored peaks are the most intense and belong to the fifth quantum while the light blue colored are the least intense and have been binned into the lowest quantum

any outlier (if any) will affect the quality of the hits is extremely small. Algorithm 2 in supplementary materials shows a pseudo code for this module.

3.3 Weighted random sampling

Rather than dealing with each peak, this step deals with quanta of peaks. Here this assumption is made that each peak within one quantization level has an equal probability of being a useful peak. Also because of the presence of more high intensity peaks, probability of finding a useful peak is greater in the higher quanta (Havilio *et al.*, 2003). In order to determine the number of peaks to be sampled from one quantum, sampling weights are determined as explained below (Fig. 4).

3.3.1 Weights calculation

First an estimate of number of peaks to be retained is calculated based upon the user defined reduction factor. Then a recursive method estimates the sampling weights for each quantum such that they satisfy the following equation:

$$\sum_{i=1}^{n_x} \left(\frac{x_i}{100} * q_i\right) = p' \tag{7}$$

where

$x_i =$ sampling rate for quantization level i

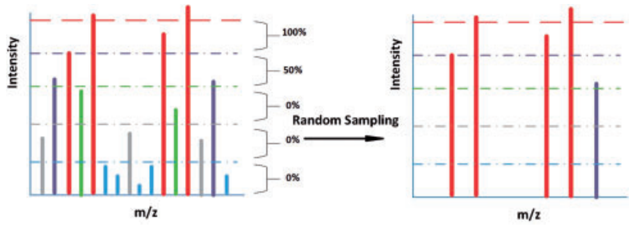

Fig. 4. Figure presents a visual representation of the random sampling module. In this figure the top most quantum is assigned a weight of 100% while the fourth quantum is assigned a weight of 50%. Peaks from all other quanta are discarded owing to their zero sampling rate

$q_i =$ ith quantization level
$||q_i|| =$ number of peaks at ith quantization level
$p' =$ number of peaks required to satisfy the reduction factor

Peaks are taken starting from the highest quantization level and continuing with lower levels until the required number of peaks is reached. If there are more peaks at a given quantization level than are needed to reach the required number of peaks, the sufficient peaks are chosen at random from that quantization level. Formally this can be presented by Eqs. (8) through (10):

case 1: $||q_{n_x}|| = p'$

$$x_i = \begin{cases} 100, & \text{if } i = n_x. \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

case 2: $||q_{n_x}|| > p'$

$$x_i = \begin{cases} \dfrac{||q_i|| - (||q_i|| - p')}{||q_i||}, & \text{if } i = n_x. \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

case 3: Default

$$x_i = \begin{cases} 100, & \text{if } p' - ||q_j|| > ||q_{j+1}||. \\ \dfrac{p' - \sum_{j=i+1}^{n_x} ||q_j||}{||q_i||}, & \text{otherwise.} \end{cases} \tag{10}$$

Figure 4 shows an example of weighted random sampling being performed on a class I spectrum. In the right half of the figure a reduced spectrum can be observed, it can be noticed that among the two peaks from fourth quantum only one appears in the final spectrum because of 50% sampling rate. This one peak is chosen totally at random.

4 Experimental datasets and method

We made use of 13 datasets to carry out performance and speed evaluation of MS-REDUCE algorithm. The details of the datasets have been provided in Supplementary Materials.

5 Performance evaluation

We carried out performance evaluation of MS-REDUCE in two phases. In first part we evaluate the time complexity and the speed up achieved in comparison to some of the existing algorithms. In the second part we perform the quality assessment experiments. This tests the quality of the peptide matches obtained after performing data reduction using MS-REDUCE. We also compare the quality assessment results of MS-REDUCE with the existing algorithms as well as investigate the improvement achieved above the previous random sampling approach (Awan and Saeed, 2015).

5.1 Time complexity

Time complexity of the algorithm can be formulated by observing the working of each module closely and summing up the individual complexities of the modules. Theoretical time complexity for MS-REDUCE comes out to be $O(N)$. The step by step calculations to obtain this result can be found in supplementary materials.

In order to verify this linear time complexity over datasets varying from conventionally sized to the modern big datasets we replicated UPS2 dataset several times to obtain datasets of desired sizes. We formed ten datasets with each subsequent set having 100 000 more spectra. The MS-REDUCE has been designed keeping in mind the challenges of big datasets from proteomics so it makes sense to use such huge datasets to perform time related experiments. For all the experiments discussed from here onwards we made use of a Linux based server with 24 CPUs, each operating at 1200 MHz.

Figure 5 shows the time taken by MS-REDUCE to process each datasets explained above. Currently the MS-REDUCE has been developed only as a single threaded program. To compensate for background tasks and other time delays we performed the experiment on each dataset about ten times and averaged the time taken. For these experiments we set the user defined reduction factor to 50 and 90.

It can be observed from Figure 5 that MS-REDUCE has a linear time complexity with respect to the number of spectra processed which is in agreement with our theoretical computational complexity. It can further be observed that a varying reduction factor does not significantly affect the running time efficiency of the algorithm. A reduction factor as described before determines the amount of data to be retained by the algorithm. It can be observed that the algorithm was able to retain its linear trend while being run with different values of Reduction Factor.

5.2 Speed comparison

We compared the processing speed of MS-REDUCE with the denoising algorithm presented in Ding *et al.* (2009). In order to compare the speed we define two metrics here. One is the conventional speed up calculation method while the other is *spectra per second* or *SPS*. Following equations describe both these metrics:

$$S = T_{\text{other}}/T_{\text{reduce}} \tag{11}$$

Where S is the speed up obtained, T_{other} is the processing time of other algorithm under consideration while T_{reduce} is the time taken by MS-REDUCE.

$$\text{SPS} = \text{Spectra}/\text{Time} \tag{12}$$

Eq. (11) presents the conventional way of calculating speed up and the Eq. (12) presents *spectra per second* metric. Larger number of spectra per seconds would mean a faster processing rate for the algorithm. Here we will refer the algorithm in Ding *et al.* (2009) as De-Noising Algorithm.

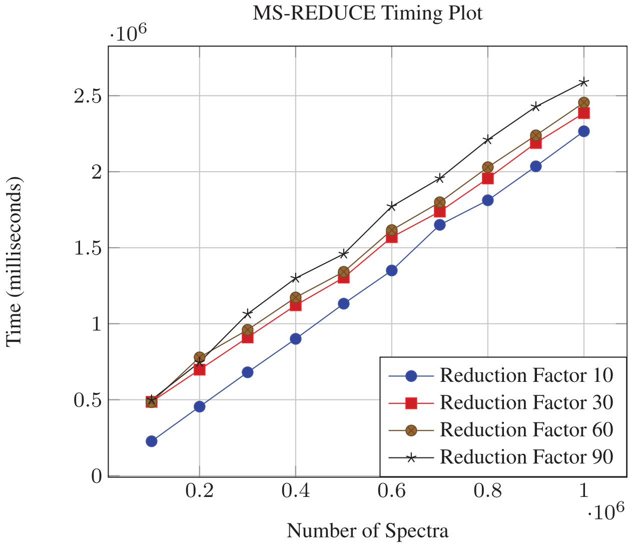

Fig. 5. Figure showing a graph between processing time of MS-REDUCE and the number of spectra processed at reduction factors of 10, 30, 60 and 90. The horizontal axis represents the number of spectra while the vertical axis represents time in milliseconds

5.2.1 Comparison with De-noising algorithm

Both the algorithms were operated in similar environments for this study. As it was explained before, De-Noising algorithm makes use of four different scoring techniques to perform peak adjustments and then undesirable peaks are filtered out using a morphological filter.

Table 1 shows the results from timing experiments performed for comparing the time taken by the De-Noising Algorithm and MS-REDUCE. The columns two and three show the processing time for algorithms in milliseconds. The De-Noising algorithm takes almost three days to process 1 million spectra. Poor scalability of such algorithms with increasing size of the datasets renders them unsuitable for high-throughput environments. The table shows MS-REDUCE takes around 47 min to process a million spectra thus achieving an average speed up of 100.

5.3 Quality assessment

In this section we investigate the quality of the peptide matches obtained from spectra that have been processed by MS-REDUCE. First we present the quality improvements achieved over the previous technique and then we compare the results with the two similar algorithms described before.

Figure 6 presents procedure for assessing the quality of peptide matches obtained after the application of MS-REDUCE algorithm. The raw spectra are fed into the MS-REDUCE or any other algorithm under observation. The processed spectra are then sent to the Tide (Diament and Noble, 2011) search engine of Crux toolkit

Table 1. Speed achieved over the Denoising Algorithm

Spectra	T_{denoise} (m)	T_{reduce} (m)	Speed up
9.61×10^4	2.35×10^7	2.25×10^5	103
1.92×10^5	4.41×10^7	4.50×10^5	96
2.89×10^5	6.49×10^7	6.78×10^5	94
3.85×10^5	8.60×10^7	9.03×10^5	94
4.81×10^5	1.09×10^8	1.12×10^6	95
5.78×10^5	1.31×10^8	1.75×10^6	83
6.74×10^5	1.55×10^8	2.0×10^6	86
7.71×10^5	1.76×10^8	2.05×10^6	94
8.67×10^5	1.97×10^8	2.29×10^6	94
9.63×10^5	2.20×10^8	2.47×10^6	98
1.06×10^6	2.43×10^8	2.81×10^6	99

Table showing comparison between run runtimes of the De-Noising Algorithm denoted by $T_{denoise}$ and MS-REDUCE denoted by T_{reduce}.

(Park *et al.*, 2008). Tide provides with the peptide spectral matches (PSMs) and decoy peptide matches based on a decoy database. These two datasets are then sent to the post processing tool known as the percolator (Kall *et al.*, 2007). The percolator computes a statistical confidence value based upon the PSMs and the decoy database matches which serve as a false discovery rate (FDR) and assigns it to each PSM. We calculated the number of PSMs for same FDR threshold obtained by using the datasets which had been treated by the test algorithm. Using this information we were able to calculate a percentage of high quality PSMs obtained by the processed spectra with respect to the number of high quality PSMs obtained using the raw spectra. This experiment was repeated for FDR values of 1%, 3%, 5%, 7% and 9%. We are taking FDR of 5% as a nominal value, so in the following experiments because of limited space we will only be presenting the results for FDR of 5%.

5.3.1 Comparison with random sampling of peaks

We performed the above explained experiment on all the thirteen datasets which have been explained in the supplementary materials and plotted the results for each dataset. The results are for FDR value of 5% but the results are extendible to other FDR values. Figures 7 and 8 present the results for quality assessment experiments performed on MS-REDUCE and and the random peak sampling method (Awan and Saeed, 2015) using three HCD datasets and the UPS2 dataset. Results for remaining datasets can be found in supplementary materials (Figs 8, 9 and 10).

The graphs have been plotted by varying the value of reduction factor for MS-REDUCE and Sampling rate of random peak sampling approach from 10% to 90%. The 100% presents the untreated raw dataset. MS-REDUCE presents significant improvement over the random sampling approach. For some datasets percentage matches are nearing 90% with a data reduction rate of only 20%. The results are also shown to be consistent for a given fragmentation type (HCD or CID) with MS-REDUCE doing a bit better for HCD due to better S/N ratio for HCD datasets (Saeed *et al.*, 2013c).

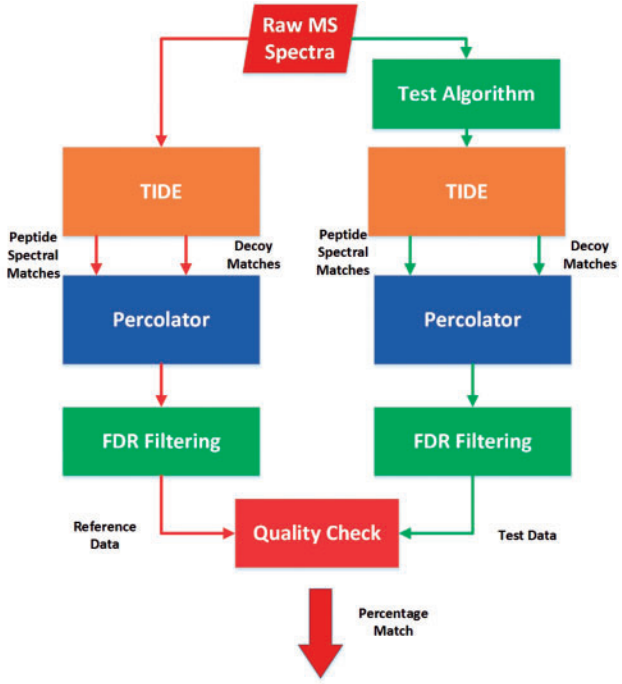

Fig. 6. Figure shows flow of quality assessment experiments. The Test Algorithm shown in top right corner is replaced by the algorithm under observation i.e. MS-REDUCE, MSCleaner 2.0 and Denoising Algorithm

5.3.2 Comparison with conventional algorithms

We also compared the quality of peptide matches for the data processed by MS-REDUCE with that processed by conventional noise reducing algorithms. The approach taken for these experiments was also the same as presented in Figure 6. Figure 9 shows quality assessment plots of De-Noising Algorithm, MSCleaner 2.0 and performance of MS-REDUCE at reduction factors of 30, 60 and 90. MS-REDUCE out performs MSCleaner 2.0 for all datasets except UPS2 while operating at nearly all the values of reduction factors. It out

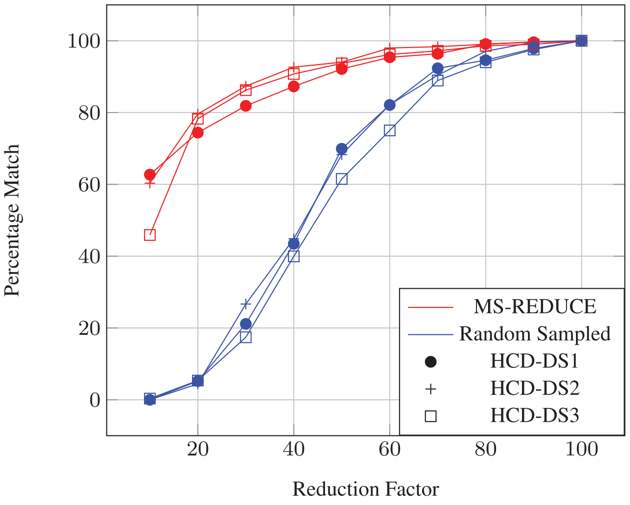

Fig. 7. Quality assessment plots for HCD-DS1, HCD-DS2 and HCD-DS3 datasets. The red colored plots represent the performance of MS-REDUCE while the blue colored plots represent the Random Sampled data. Three datasets have been differentiated using different symbols. The *x*-axis contains Reduction Factor i.e. amount of data retained. And the *y*-axis shows the percentage of accurate peptide hits obtained from the processed data

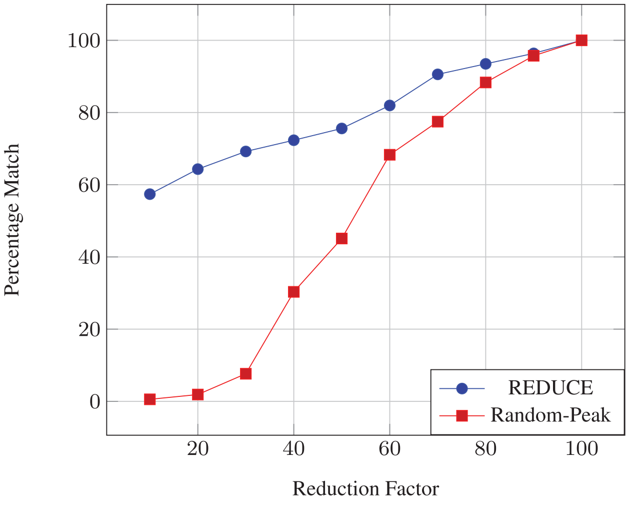

Fig. 8. Quality assessment plot for UPS2 dataset. The red plot represents the performance of MS-REDUCE while the blue plot represent the Random Sampled data. The *x*-axis contains Reduction Factor i.e. amount of data retained. And the *y*-axis shows the percentage of accurate peptide hits obtained from the processed data

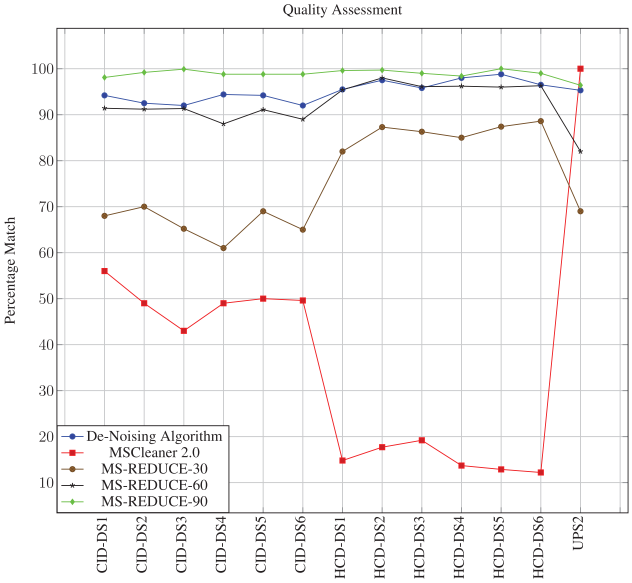

Fig. 9. Figure showing quality assessment plots for De-Noising Algorithm Algorithm, MSCleaner 2.0 and MS-REDUCE. Quality Assessment plots for different Reduction Factor of MS-REDUCE can be observed. In the legend a numerical value with MS-REDUCE represents its reduction factor. *X*-axis contain the labels for the experimental datasets while *Y*-axis represents the percentage of peptide matches obtained from each dataset after being processed by each algorithm

performs De-Noising Algorithm while operating around a reduction factor of 60. Note that other both algorithms are compute-intensive and take much longer time as compared to MS-REDUCE to produce comparable results as shown in Figure 9.

6 Conclusion

Analysis of high-throughput MS based proteomics data is an essential task in systems biology. Data from multiple experiments can scale from million to a billion spectra and this data volume can easily reach tera- to peta-byte level. The Big Data from modern mass spectrometers creates scaling problems for existing software designed for much smaller datasets. Although these algorithms are useful for interpretation of simple spectra, the search and match routine becomes computationally intractable for complex peptides. The big data volume that one gets from these high-throughput machines is enormous and low scalability of conventional tools cannot keep up with the rate of data generation. Hence dimensionality reduction techniques that can reduce the number of peaks that needs to be processed are essential for fast and efficient processing of MS data for system- wide studies.

In this paper we presented a novel dimensionality reduction technique, called MS-REDUCE, for pre-processing big MS datasets. To our knowledge, the proposed strategy is first attempt at data reduction of MS data for high-throughput environments. Our low-computational cost strategy is based on classification, quantization and sampling of MS data peaks. An approximate classification of spectra followed by a quantization step results in binning of peaks. Each quantum of a spectrum contains peaks within a particular intensity range. Then a random sampling step is performed on these bins to obtain the peaks which form the final reduced spectrum. Our strategy is linear in time complexity with increasing number of spectra which is confirmed by our experiments. We also show that MS-REDUCE can process up to a million spectra in 47 min as compared to the De-Noising Algorithm, which processes the same number of spectra in about 3 days. We performed rigorous testing of the algorithm using experimental datasets and compared its performance with two of the existing algorithms. The implemented software will be available for free academic use at the author's webpages.

Funding

This work was partially funded by National Science Foundation grant NSF CCF-1464268.

Conflict of Interest: none declared.

References

Aebersold,R. and Mann,M. (2003) Mass spectrometry-based proteomics. *Nature*, **422**, 198–207.

Awan,M.G. and Saeed,F. (2015). On the sampling of big mass spectrometry data. In: *Proceedings of the 7th International Conference on Bioinformatics and Computational Biology*, BICOB pp. 143–148.

Bern,M. *et al.* (2004) Automatic quality assessment of peptide tandem mass spectra. *Bioinformatics*, **20**, i49–i54.

Dancik,V. *et al.* (1999) De novo peptide sequencing via tandem mass spectrometry. *J. Comput. Biol.*, **6**, 327–342.

Diament,B.J. and Noble,W.S. (2011) Faster sequest searching for peptide identification from tandem mass spectra. *J. Proteome Res.*, **10**, 3871–3879.

Ding,J. *et al.* (2009) A novel approach to denoising ion trap tandem mass spectra. *Proteome Sci.*, **7**.

Ding,J. *et al.* (2011) Svm-rfe based feature selection for tandem mass spectrum quality assessment. *Int. J. Data Min. Bioinf.*, **5**, 73–88.

Du,X. *et al.* (2008) Linear discriminant analysis-based estimation of the false discovery rate for phosphopeptide identifications. *J. Proteome Res.*, **7**, 2195–2203.

Eng,J.K. *et al.* (1994) 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.

Finehout,E. and Lee,K. (2003) An introduction to mass spectrometry applications in biological research. *Biochem. Mol. Biol. Educ.*, **32**, 93–100.

Gentzel,M. *et al.* (2003) Preprocessing of tandem mass spectrometric data to support automatic protein identification. *Proteomics*, **3**.

Havilio,M. *et al.* (2003) Intensity-based statistical scorer for tandem mass spectrometry. *Anal. Chem.*, **75**, 435–444.

Hebert,A.S. *et al.* (2014) The one hour yeast proteome. *Mol. Cell Proteomics*, **13**, 339–347.

Hoffert,J.D. *et al.* (2006) Quantitative phosphoproteomics of vasopressin-sensitive renal cells: regulation of aquaporin-2 phosphorylation at two sites. *Proc. Natl. Acad. Sci. USA*, **103**, 7159–7164.

Jiang,X. *et al.* (2010) Classification filtering strategy to improve the coverage and sensitivity of phosphoproteome analysis. *Anal. Chem.*, **82**, 6168–6175.

Kall,L. *et al.* (2007) Semi-supervised learning for peptide identification from shotgun proteomics datasets. *J. Proteome Res.*, **4**, 923–925.

Lin,W. *et al.* (2012) An unsupervised machine learning method for assessing quality of tandem mass spectra. *Proteome Sci.*, **10**, 1–8.

Linnet,K. (2013) Toxicological screening and quantitation using liquid chromatography/time-of-flight mass spectrometry. *J. Foren. Sci. Criminol.*, **1**, 1.

Mujezinovic,N. *et al.* (2006) Cleaning of raw peptide ms/ms spectra: Improved protein identification following deconvolution of multiply charged peaks, isotope clusters, and removal of background noise. *Proteome Sci.*, **6**, 5117–5131.

Mujezinovic,N. *et al.* (2010) Reducing the haystack to find the needle: improved protein identification after fast elimination of non-interpretable peptide ms/ms spectra and noise reduction. *BMC Genomics*, **11**, 1–8.

Na,S. and Paek,E. (2007) Quality assessment of tandem mass spectra based on cumulative intensity normalization. *J. Proteome Res.*, **5**.

Park,C.Y. *et al.* (2008) Rapid and accurate peptide identification from tandem mass spectra. *J. Proteome Res.*, **7**, 3022–3027.

Perkins,D.N. *et al*. (1999) Probabioity-based protein idenitification by searching sequence database using mass spectrometry data. *Electrophoresis*, **20**, 3551–3567.

Pisitkun,T. *et al*. (2004) Identification and proteomic profiling of exosomes in human urine. *Proc. Natl. Acad. Sci. USA*, **101**, 13368–13373.

Purvine,S. *et al*. (2004) Spectral quality assessment for high-throughput tandem mass spectrometry proteomics. *OMICS: J. Integr. Biol*., **8**, 255–265.

Saeed,F. *et al*. (2013a) Cams-rs: clustering algorithm for large-scale mass spectrometry data using restricted search space and intelligent random sampling. *IEEE/ACM Trans. Comput. Biol. Bioinf*., **11**, 128–141.

Saeed,F. *et al*. (2013b) An efficient dynamic programming algorithm for phosphorylation site assignment of large-scale mass spectrometry data. *IEEE Int. Conf. Bioinf. Biomed. Workshops (BIBMW)*, **7**, 618–625.

Saeed,F. *et al*. (2013c) Phossa: fast and accurate phosphorylation site assignment algorithm for mass spectrometry data. *Proteome Sci*., **11**, S14.

Tabb,D.L. *et al*. (2003) Similarity among tandem mass spectra from proteomic experiments: Detection, significance, and utility. *Anal. Chem*., **75**.

Wells, G. *et al*. (2011). Why use signal-to-noise as a measure of ms performance when it is often meaningless? Technical report, Agilent Technologies.

Wu,F.X. *et al*. (2008) An approach to assessing peptide mass spectral quality without prior information. *Int. J. Funct. Inf. Person. Med*., **1**, 140–155.

Zhang,J. *et al*., (2008) Peakselect: preprocessing tandem mass spectra for better peptide identification. *Rapid Commun. Mass Spectrom*., **22**, 1203–1212.

Zhao,B. *et al*. (2012) Cphos: a program to calculate and visualize evolutionarily conserved functional phosphorylation sites. *Proteomics*, **12**, 3299–3303.