MSDASH: MASS SPECTROMETRY DATABASE AND SEARCH

Zhan Wu*

Department of Computer Science, University of Western Ontario,
London, Ontario N6A 5B8, Canada
**Email: zwu47@csd.uwo.ca*

Gilles Lajoie

Department of Biochemistry, University of Western Ontario,
London, Ontario N6A 5B8, Canada
Email: glajoie@uwo.ca

Bin Ma

Department of Computer Science, University of Western Ontario,
London, Ontario N6A 5B8, Canada
Email: bma@csd.uwo.ca

Along with the wide application of mass spectrometry in proteomics, more and more mass spectrometry data are becoming publicly available. Several public mass spectrometry data repositories have been built on the Internet. However, most of these repositories are devoid of effective searching methods. In this paper we describe a new mass spectrometry data library, and a novel method to efficiently index and search in the library for spectra that are similar to a query spectrum. A public online server have been set up and demonstrated outstanding speed and scalability of our methods.

Together with the mass spectrometry library, our searching method can improve the protein identification confidence by comparing a spectrum with the ones that are already characterized in the database. The searching method can also be used alone to cluster the similar spectra in a mass spectrometry dataset together, in order to to improve the speed and accuracy of the protein identification or quantification.

1. INTRODUCTION

Mass spectrometry has become the standard high-throughput method for protein identification, and more recently, for protein quantification [1, 2]. In a typical protein identification experiment using mass spectrometry, proteins are enzymatically digested to peptides, and the tandem mass (MS/MS) spectra of the peptides are measured using a tandem mass spectrometer. The limitation of the current experiment procedure results in spectra that are difficult to interpret due to poor fragmentation and contaminations from chemical noise.

Many software programs have been developed to identify the sequence of a peptide from its MS/MS spectrum. All these programs more or less depend on a model to predict a theoretical spectrum of a given peptide sequence. By using either a search in a protein database, or by constructing a sequence from scratch, a peptide that gives the best match between the predicted and the experimental spectra is deduced. The approach using a protein database is referred to as database search [3–6], and the construction from scratch is called *de novo* sequencing [7, 8].

The prediction of the theoretical spectrum is a difficult task, partially because the mobile proton model [9] for the peptide fragmentation is not a quantitative model. Limited success was achieved in predicting the theoretical spectrum on a specific mass spectrometer type with a fixed parameter setting [10]. However, in order to do the data analysis in a high-throughput manner, most of the software programs use over-simplified models. Normally these programs expect good y-ion series and/or b-ion series to be observed in order to confidently identify the peptide sequence.

This creates the following situation. Some peptides with certain sequences do not produce good

[*] Corresponding author.

y-ion or b-ion series and therefore cannot be confidently identified by high-throughput experiments and software. The imperfect spectra are often due to the inherent nature of the peptides. That is, very similar spectra will be produced if the experiment is repeated under similar conditions. In a typical dataset, these imperfect spectra are mixed together with other low-quality spectra that are contaminated by chemical noise. This further complicates the data interpretation. According to our experience and the literature [11], a typical MS/MS dataset can contain as many as 80% of tandem mass spectra that are not characterized by current software.

In spectrometry analysis there have been another approach for spectrum interpretation, which matches the spectrum with a library of confidently characterized spectra (called Annotated Spectrum Library). Such an approach does not need to predict the spectrum from a peptide sequence and can potentially interpret more spectra with higher confidence. However, the huge number of possible peptide sequences and the lack of an efficient matching method make this approach computationally expensive for peptide identification. X! Hunter [12] is the only search engine that adopted this approach in peptide identification. By limiting the search in only the consensus spectra of confidently identified peptides of certain organisms, X! Hunter managed to perform the search relatively efficiently.

In this paper we propose to extend this Annotated Spectrum Library approach a step further. We compare a spectrum with all of the publicly-available spectra, annotated or not, and find the similar spectra. This makes the computation even more expensive, but has the advantages as discussed below.

Two situations may arise when matches are found:

(1) There are one or more previously-characterized spectra that match the current one. The current spectrum can use the previous characterization. This is the same as the Annotated Spectrum Library approach. Note that the previous characterization might have been done under a better experimental condition (such as a simpler protein mixture, more abundant sample, or better instrument).

(2) There are several uncharacterized spectra that match the current one. This implies that these spectra are unlikely random noises and deserve further examination by more optimized experiments or more extensive computation. In addition, because the MS/MS spectra of the same peptide on different instruments may produce slightly different spectra, the combination of these similar but not identical spectra (and their associated information such as organisms and experimental conditions) will reveal more information about the peptide than any single spectrum would. As a result, the chance of successful peptide identification will be greatly increased.

Clearly, this strategy comes with the price of increased computational complexity. A method based on locality sensitive hashing was proposed to speed up the matching [13]. The method first clusters the database spectra into clusters according locality sensitive hashing. Then a query spectrum is compared only with the clusters that are "neighbors" of the query spectrum. This avoided the one-against-all comparison between the query spectrum and the database. The method provides good trade off between the sensitivity and the speed of the search. However, this method is complicated and the 100 times theoretical speed up factor claimed in the paper would be diminished when implemented in a real system.

In this paper we propose a new algorithm for speeding up the spectrum matching in a large MS/MS database, based on a novel "thumbnail" idea. The method is simple and easy to implement. Written in Java, our algorithm achieves an average matching speed of comparing one million pairs of spectra per second on a single CPU. When the precursor ion mass of an MS/MS spectrum is known (which is usually the case in peptide identification), we can use the precursor ion mass to pre-select the database spectra for the matching. Depending on the mass accuracy of the data, this further improves the speed by hundreds to thousands of times, resulting in a final speed of searching one spectrum in 10^8-10^{10} spectra per second. We believe such a speed should be sufficient for most applications nowadays. Our method is also very memory-friendly: the index of each spectrum requires only 8 bytes in the main memory. This drastically reduces the memory usage, because keeping a spectrum in memory will usually require thousands of bytes. The method can also be easily parallelized. All these properties enables the inexpensive implementation of a real system.

In the past several years, more and more mass spectrometry data have been made publicly available. Among many available mass spectrometry data repositories, some popular ones are Open Proteomics Database [14], Peptide Atlas [15–17], and Sashimi Repository [18]. They provide great testing data for the research of new data analysis software. However, none of these data repositories supports efficient searching of similar spectra. This makes the repositories to be no much more than well-organized FTP sites, and the data in them are not fully utilized in the analysis of a newly measured dataset. In this paper we introduce our new public mass spectrometry data library. Equipped with our efficient searching method, the library allows the user to query with an MS/MS spectrum and efficiently retrieve all the similar spectra (together with their annotations if there are any) in the library.

Our efficient searching method can also be used without a spectrum database. The method can be used to cluster similar spectra in one or a few datasets together. This not only speeds up the subsequent data analysis by removing redundancies, but also improves the peptide identification confidence by gathering information from different MS/MS scans together (possibly from repeated experiments under slightly different conditions).

The rest of the paper is organized as follows: Section 2 defines the mass spectrometry terms used in the paper. All definitions are standard and can be skipped by a reader who is familiar with this area. Section 3 introduces our fast searching algorithm. Section 4 introduces the implementation of the public data library. The speed and the sensitivity of our searching method is demonstrated in Section 5.

2. TERMS AND NOTATIONS

An *MS/MS spectrum* of a peptide contains a list of peaks. Each signal *peak* is caused by some fragment ions of the peptide, and can be encoded with two real values: the *m/z value* of the peak represents the mass to charge ratio of the fragment ions, and the *intensity* value of the peak indicating the abundance of the fragment ions. There are different types of fragment ions for a peptide, where the most important ones are the *y-ions* and *b-ions*. The *precursor mass* of a spectrum is the mass of the whole peptide. The mass unit for m/z and precursor mass is *dalton*. Peptides are obtained by digesting proteins using enzymes and the most commonly used enzyme is *trypsin*. The resulting peptides using trypsin are called *tryptic peptides* and typical tryptic peptides range from 500 to 3000 daltons. A mass spectrometer measures m/z and precursor mass with small errors. For this reason mass errors are allowed in spectrum matching. The maximum error allowed for matching two m/z values is called the *mass error tolerance*. Typical mass error tolerance ranges from ±0.01 dalton to ±1 dalton depending on the types of the spectrometer.

3. SEARCHING METHOD

The main idea of our searching method is an efficient filtration that efficiently rejects the apparently unmatched spectra and keeps only the possible matches for further examination using more time-consuming but more accurate criteria. This filtering method is a common practice to speed up approximate pattern matching. A good filtration method should reject as many false matches as possible (to maximize the selectivity), while keeping as many true matches as possible (to maximize the sensitivity).

Our searching method consists of the following steps: First, the database spectra are preprocessed and the major peaks of each spectrum are stored in a relational database. Then a “thumbnail” is computed for each spectrum and put in a computer's main memory. For each spectrum, this thumbnail is a 64-bit integer. The filtration is done using this 64-bit integer. Lastly, the spectra passing the filtration will be retrieved from the relational database for examination using a more accurate scoring function, and outputs are generated.

These steps are described in more details in the following subsections.

3.1. Spectrum Preparation

For each spectrum in the library, data preprocessing is needed to prepare the spectrum for the fast matching. First, due to the random measurement error of the instrument, multiple copies of the identical ion can cause a cluster of adjacent peaks with very small difference in their m/z values. These adjacent peaks need to be centroided together to form only one peak. Any standard centroiding method can be used in this step.

After centroiding, each spectrum can still possibly contain hundreds to thousands of peaks. A large portion of these peaks are very weak and should be regarded as noise. Keeping them in the comparison will not only reduce the speed, but also add errors to the scoring function for comparing two spectra. For a typical tryptic peptide of length 15, there are only 28 y-ions and b-ions that are the most useful for peptide identification. Therefore, for the purpose of spectrum comparison, it is safe to only examine the strongest 50 peaks of each centroided spectrum. In our method, the strongest 50 peaks of each centroided spectrum are selected and added into a relational database as a BLOB for fast retrieval. This greatly reduces the spectrum complexity with only negligible loss in the accuracy of the scoring function.

3.2. Thumbnail of a Spectrum and Rapid Filtration

We propose here a novel "thumbnail" idea for fast filtration. Basically, a thumbnail of a spectrum is a bit array where each bit indicates whether the spectrum contains a strong peak at some given mass value. Then the comparing of two spectra can be done in a rapid way by a bitwise-and operation on their thumbnail, and counting the number of 1s in the result.

More precisely, let $[0, K-1] = \{0, 1, \ldots, K-1\}$ and $h : R^+ \to [0, K-1]$ be a hash function that maps the positive numbers to integers between 0 and $K-1$. Let S be a spectrum with peaks at m/z values $x_1, x_2, \ldots, x_m$. We denote $mz(S) = \{x_1, x_2, \ldots, x_m\}$. Then the thumbnail of S is defined as

$$h(S) = \{i \,|\, \text{there is } x_j \text{ such that } h(x_j) = i\}.$$

In a computer, $h(S)$ can be equivalently represented by a length-K bit array T such that $T[i] = 1$ if and only if $i \in h(S)$. We will sometimes call T as the thumbnail of S too.

Lemma 3.1. *Let h be a hash function. Let S_1 and S_2 be two spectra of length m. Suppose S_2 is a random spectrum independent to S_1 and h, then*

$$\begin{aligned} &Pr(|h(S_1) \cap h(S_2)| > (1+\delta)m^2/K) \\ &< \left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^{m^2/K}. \end{aligned}$$

Proof. Because S_1 has at most m peaks, $|h(S_1)| \leq m$. For any $x \in mz(S_2)$, the probability that $h(x) \in h(S_1)$ is therefore at most $p = m/K$. The expected number of peaks from S_2 that are mapped into $h(S_1)$ is then at most $mp = m^2/K$. By using Chernoff's bound [19] straightforwardly,

$$\begin{aligned} &Pr(|h(S_1) \cap h(S_2)| > (1+\delta)m^2/K) \\ &\leq Pr((1+\delta)m^2/K \text{ peaks of } S_2 \text{ mapped in } S_1) \\ &< \left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^{m^2/K} \qquad \square \end{aligned}$$

When $\delta \geq 0$, $\frac{e^\delta}{(1+\delta)^{(1+\delta)}}$ is a monotonically decreasing function that approaches 0 rapidly when δ increases. As a result, by selecting proper $t = \lfloor (1+\delta)m^2/K \rfloor$, m, and K, we can let $Pr(|h(S_1) \cap h(S_2)| > t)$ become very small. For example, when $m = 20$, $K = 128$, and $t = 12$, $Pr(|h(S_1) \cap h(S_2)| > t) < 1.74 \times 10^{-4}$ according to Lemma 3.1. We note that the bound given in Lemma 3.1 is not tight and the real probability is much lower than this. This suggests that we can use the size of $h(S_1) \cap h(S_2)$, i.e., the intersection of the thumbnails of the query spectrum and the database spectrum, to filter out the random spectra. In order to be useful, this filter should not reject the correct matches. This is guaranteed by Lemma 3.2.

Lemma 3.2. *Let h be a hash function. Let S_1 and S_2 be two length-m spectra. Suppose S_2 is such that $|mz(S_1) \cap mz(S_2)| = n$, and the hash function h is independent to S_1 and S_2. Then for any $\delta > 0$,*

$$\begin{aligned} &Pr(|h(S_1) \cap h(S_2)| \leq t) \\ &\leq \binom{K}{t} \times (t/K)^n \qquad (1) \\ &\leq \frac{1}{\sqrt{2\pi t}} \times e^t \times (t/K)^{n-t} \qquad (2) \end{aligned}$$

Proof. Denote $X = mz(S_1) \cap mz(S_2)$. Then $|X| = n$ and the number of possible mappings from X to $[0..K-1]$ is K^n.

Clearly, for any $x \in X$, $h(x) \in h(S_1) \cap h(S_2)$. Therefore, if $|h(S_1) \cap h(S_2)| \le t$, then all the n values in X need to be mapped into a size-t subset of $[0, K-1]$. There are $\binom{K}{t}$ such subsets. For each of them, there are t^n possible ways to map X to it. Hence, the total number of possible mappings that satisfies $|h(S_1) \cap h(S_2)| \le t$ is upper bounded by $\binom{K}{t} \times t^n$. Consequently,

$$\begin{aligned}
& Pr(|h(S_1) \cap h(S_2)| \le t) \\
\le & \binom{K}{t} \times t^n \times K^{-n} \\
\le & \frac{K^t}{t!} \times (t/K)^n \\
\le & \frac{K^t}{\sqrt{2\pi t} \times (t/e)^t} \times (t/K)^n \qquad (3) \\
= & \frac{1}{\sqrt{2\pi t}} \times e^t \times (t/K)^{n-t}
\end{aligned}$$

The inequation (3) is true because of Stirling's formula [20]. □

From (2) it is clear that when t is much smaller than n and K larger than n, the probability becomes very small. For example, when $m = 20$, $K = 128$, and $t = 12$ as before and $n = 19$, $Pr(|h(S_1) \cap h(S_2)| \le t) \le 1.2 \times 10^{-3}$ according to (1). Again, the bound proved in Lemma 3.2 is not tight and the real probability is much lower than this.

From the above two examples, we can see that by choosing a right threshold t for the size of the intersection thumbnail, Lemma 3.1 guarantees that a random S_2 can be rejected by the threshold with high probability; whereas Lemma 3.2 guarantees that a spectrum S_2 similar to S_1 can pass the threshold with high probability.

In our implementation of this filtering method, we use $m = 20$, $K = 64$, and $t = 12$. Given a query spectrum S_1, a spectrum S_2 passes the filtration if and only if $h(S_1) \cap h(S_2)$ contains more than t elements. By randomly sampling one million spectrum pairs, the probability that a random spectrum can pass the filtration is estimated and shown in Figure 1. In particular, when $t = 12$, this probability is only 0.000166.

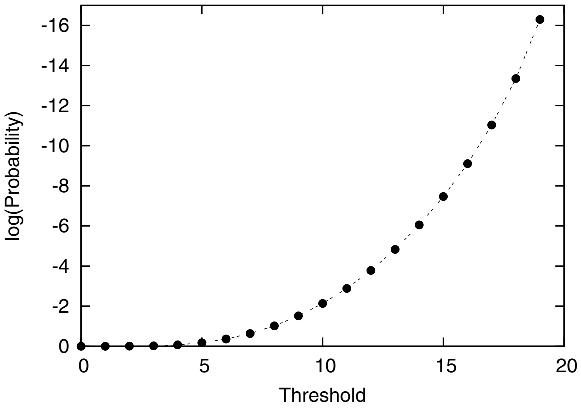

Fig. 1. When $m = 20, K = 64$, the base-10 logarithm of the probability that a random spectrum passes the filtration with threshold t. The x axis is the threshold. The y axis is the probability in logarithmic scale.

For a spectrum S_2 such that $|mz(S_1) \cap mz(S_2)| = n$, the sensitivity of the filtration, i.e., the probability that S_2 can pass the filtration is also estimated by random sampling and given in Figure 2.

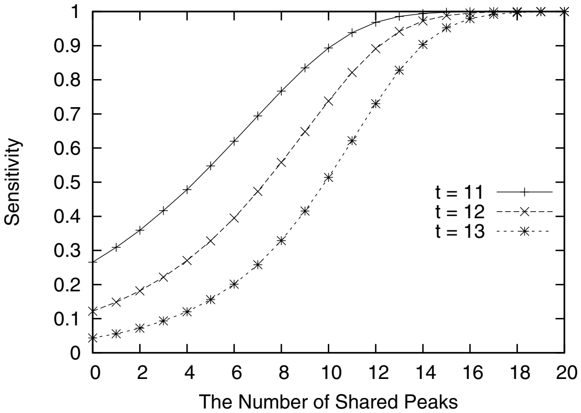

Fig. 2. When $m = 20, K = 64$, the sensitivity of the filtration for a similar spectrum sharing n out of the 20 peaks with the query.

In the discussion above, we assumed h to be a perfect hashing function. However, in reality we must consider the mass error tolerance. That is, two m/z values $x \in mz(S_1)$ and $y \in mz(S_2)$ match if $|x - y| \le \Delta$ for a predefined Δ. To allow such m/z values to be mapped together by h, we require $h(x) = h'(\lfloor x/(k \times \Delta) \rfloor)$ for a constant k and another hash function $h' : N \to [0, K-1]$. This way, when $|x - y| \le \Delta$, $\lfloor x/(k \times \Delta) \rfloor$ and $\lfloor y/(k \times \Delta) \rfloor$

are likely to be equal and therefore give the same hash value. In addition, if a value x is such that $\lfloor x/(k \times \Delta) \rfloor = \lfloor (x - \Delta)/(k \times \Delta) \rfloor + 1$, we add both $\lfloor x/(k \times \Delta) \rfloor$ and $\lfloor (x - \Delta)/(k \times \Delta) \rfloor$ into the thumbnail to further increase the sensitivity of the filtration. This change has a little affect to the selectivity as we will see in Section 5.

When $K = 64$, a thumbnail can be encoded with a 64-bit long int type. On a 64-bit computer, the intersection of two thumbnails can then be done by a single bitwise-and operation. The size of a thumbnail can be done by counting the number of 1s in the long integer, which can be done either by some very efficient programs [a] or by a single CPU instruction if such operation is supported by the CPU. In our system, a spectrum is considered to be divided into 64 segments, and the highest 20 segments of each spectrum are used to compute the thumbnail of the spectrum. The thumbnails of all the spectra in the database are pre-computed and loaded into the main memory of the computing servers. This requires $8N$ bytes of main memory if there are N spectra. When a search is performed, the thumbnail of the query spectrum is computed using the same hash function and then compared with each thumbnail using the above mentioned fast operations. Only those passing the filtration are further compared using the more accurate measurement described in the following subsection.

3.3. Spectrum Similarity

Once a spectrum passes the filtration, the strongest 50 peaks of the spectrum stored in the relational database are retrieved and compared against the strongest 50 peaks of the query spectrum. A similarity score is calculated as follows.

Let (x_i, h_i) be a peak with m/z value x_i and intensity h_i. Let $S_1 = \{(x_1, h_1), \ldots, (x_m, h_m)\}$ and $S_2 = \{(x'_1, h'_1), \ldots, (x'_m, h'_m)$. We assume the peaks in each spectrum are sorted in ascending order according to the m/z values. The peaks in the two spectra are compared using a merge-sort type of procedure to find all the pairs of peaks such that $|x_{i_k} - x'_{j_k}| \leq \Delta$ for $k = 1, \ldots, l$, where Δ is the mass error tolerance. Then the similarity score of the two spectra can be defined as

$$sc(S_1, S_2) = \frac{\sum_{k=1}^{l} h_{i_k} h'_{j_k}}{\sqrt{\sum_{i=1}^{m} h_i^2} \sqrt{\sum_{j=1}^{m} {h'}_j^2}}$$

According to our experience, many low-quality spectra in the library often contain one or few very strong peaks. If the above formula is used, then two spectra that share one strong peak may become very similar, despite the fact that their other peaks do not match each other. To reduce this risk, we convert the intensity of each peak to the logarithm of the intensity before the calculation given above.

4. THE DATABASE SYSTEM

We have implemented the search method mentioned above in Java, together with a public spectra database server that allows public users to deposit data to the database and search for similar spectra in the database. The system is called MSDash and available online at http://ala.bin.csd.uwo.ca:8080/msdash.

The system consists a web server, a database sever and ten computing servers. Each sever has a single-core AMD Opteron CPU. The web server runs Apache Tomcat and the database server uses MySQL. As soon as the user submit the query MS data file, the web server will forward the task to the computing servers. After the computing servers finish the matching process, the matched list of mass spectra will be transferred back to the web server and displayed to the user.

Currently, some publicly available data downloaded from the Open Protein Database [14] and the Sashimi data repository [18] have been added to the database as test data. These include about 3.3 million tandem mass spectra. The raw data are stored in mzXML format on the hard drives of the servers. Fifty strongest peaks of each spectrum are stored in the MySQL database. And the twenty strongest peaks of each spectrum are used to generate a thumbnail and all of the thumbnails are loaded in the main memory of the computing servers.

[a] See http://infolab.stanford.edu/˜manku/bitcount/bitcount.html for some examples.

5. EXPERIMENTS RESULT

Figure 3 shows the average searching time for one spectrum with unknown precursor mass in our spectra database, at different database size using 10 CPUs. Clearly the searching time grows linearly to the size of the database, indicating the excellent scalability of our system. The time does not approach zero when the database size approaches zero. This is because of the overhead due to network communication for query submission and result display. Besides the overhead, the average search speed (indicated by the slope of the line) is approximately 10 million matches per second on 10 CPUs, i.e., 1 million matches per second on one CPU.

In our experiment testing for speed, we assumed that the precursor ion mass of a spectrum is unknown. Therefore, the query spectrum needs to be compared with every spectrum in the database. However, when the precursor ion mass is known, one only needs to compare it with the spectra in the database with similar precursor ion mass. So that the query spectrum only needs to match 10^{-2} to 10^{-4} of the database spectra depending on the precursor ion mass error tolerance. If this precursor filtration option is selected by the user, our system can search one query spectrum in a database with 10^{8} to 10^{10} spectra per second on a single CPU.

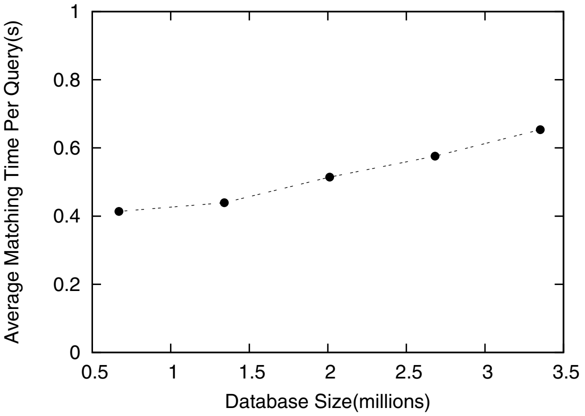

Fig. 3. Average searching time for searching one spectrum with unknown precursor ion mass in MSDash system with varying database size. Ten CPUs are used. The x axis shows the number of spectra in the database, and the y axis shows the searching time per query in seconds.

The thumbnail filtration contributed significantly to the speed of our system. Using our default threshold $t = 12$, Figure 4 shows the percentage of remaining spectra after the filtration. The figure shows that only 0.14% of the spectra in the database can pass the filtration and need further examination by the more time-consuming similarity function defined in Section 3.3 [b]. We note that because the hash function we use in the real system is not a perfect hash function and needs to be modified as described in Section 3.2, this percentage is higher than the estimation with simulation in Figure 1.

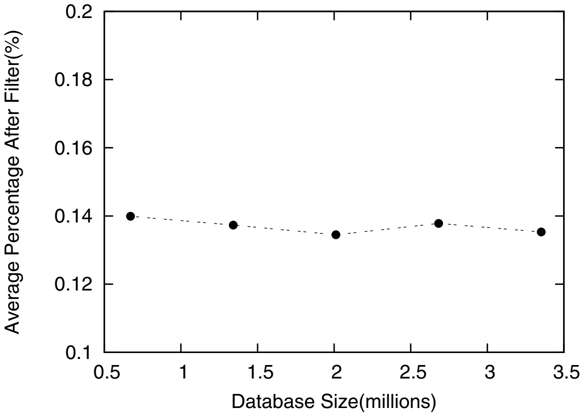

Fig. 4. The percentage of database spectra that pass the fast thumbnail filtration with $t = 12$. The y axis is the percentage and the x axis shows the number of spectra in the database.

Figure 5 shows the average query time for searching real spectra according to different values of the threshold parameter t and the total number of spectra in the database is approximate 3.3 million. From Figure 5, we can find that the average speed per query is improved dramatically when the value of the threshold t is increased. Similarly, Figure 6 shows the average percentage of remaining spectra after the filtration with different threshold t and the same database size. The figure shows that the sensitivity is dropping quickly when the threshold t is increased. Clearly the threshold t can be used to control the trade-off between the speed and the sensitivity of the query. In our system, the default threshold t is

[b] This results in a speed up of 700 times, comparing to the 100 times speed up factor claimed in the paper [13].

12, but the user can change this value. The recommended value for the threshold t is from 11 to 13. In the following part we will examine the sensitivity when $t = 12$.

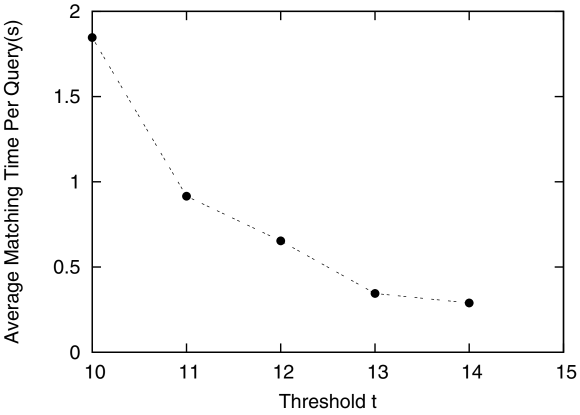

Fig. 5. Average searching time for searching one spectrum with unknown precursor ion mass in MSDash system with different threshold t. Ten CPUs are used. The Database size is 3.3 million. The x axis shows the threshold t, and the y axis shows the searching time per query in seconds.

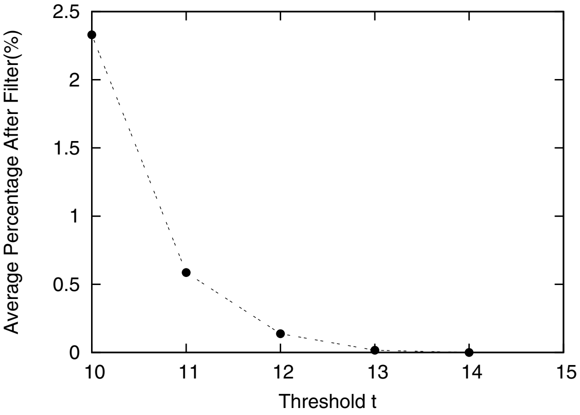

Fig. 6. The percentage of database spectra that pass the fast thumbnail filtration with different values of threshold t. The Database size is 3.3 million. The y axis is the percentage and the x axis shows the threshold t.

To test the sensitivity of our system, we selected 100 spectra from the database and randomly modified each of them 10 times. Then the modified spectra are searched in the database to see the percentage that the original spectra can be retrieved.

The random modifications are applied on both the intensities of peaks and the m/z values of the peaks. The m/z modification with probability p is done as follows: for *each* peak in the query spectrum, with probability p, the m/z value of the peak is replaced with another random value. The intensity modification with error range $\pm x\%$ is done as follows: for *each* peak in the query spectrum, add a uniformly random error between $-x\%$ to $x\%$ to the intensity of the peak. Table 1 shows the sensitivity of our method under different levels of modifications. From the table we can see that our method can find all real matches even if every peak's intensity is modified by up to ±30%, and 5% of the peaks are randomly moved around. Even when 20% of the peaks are randomly moved around, our method still keeps high sensitivity.

Table 1. Sensitivity under different levels of modifications.

	0%	±10%	±20%	±30%
$p = 0$	100%	100%	100%	100%
$p = 0.05$	100%	100%	100%	100%
$p = 0.1$	100%	99.8%	99.9%	99.8%
$p = 0.2$	98.3%	98.6%	98.5%	98.3%
$p = 0.3$	94.8%	93.0%	94.8%	93.6%

6. CONCLUSION

Based on a novel spectrum thumbnail concept, we introduced an efficient spectrum searching method. Comparing to other methods for the similar purpose, our strategy is not only significantly faster, but also much easier to implement. The method achieves a speed of searching one spectrum in one million spectra per second per CPU without knowing the precursor ion mass, or 10^8 to 10^{10} spectra per second per CPU if knowing the precursor ion mass. Our searching method has very high sensitivity. We have also implemented a public online server that allows users to deposit data to our database, and search a spectrum in the database. The server is available at http://ala.bin.csd.uwo.ca:8080/msdash.

References

1. R Aebersold1, M Mann.Mass spectrometry-based proteomics. *Nature Journal* 2003; bf 422:198–20.
2. MA Baldwin. Protein Identification by Mass Spectrometry: Issues to be Considered. *Molecular Cellular Proteomics* 2004; **3**:1–9.

3. DN Perkins, DJC Pappin, DM Creasy, JS Cottrell. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis* 1999; **20(18)**:3551–3567.
4. JK Eng, AL McCormack, JR Yates. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *Journal of The American Society for Mass Spectrometry* 1994; **5**:976–989.
5. RE Moore, MK Young, TD Lee. Qscore: An Algorithm for Evaluating SEQUEST Database Search Results. *Journal of The American Society for Mass Spectrometry* 2003; **13(4)**:378–386.
6. R Craig, RC Beavis. TANDEM: matching proteins with mass spectra. *Bioinformatics* 2004; **20(9)**:1466–1467.
7. B Ma, K Zhang, C Liang. An Effective Algorithm for the Peptide De Novo Sequencing from MS/MS Spectrum. *Journal of Computer and System Sciences* 2005; **70(3)**:418–430.
8. A Frank, P Pevzner. PepNovo: De Novo Peptide Sequencing via Probabilistic Network Modeling. *Analytical Chemistry* 2005; **77(4)**:964–973.
9. VH Wysocki, G Tsaprailis, LL Smith, LA Breci. Mobile and localized protons: a framework for understanding peptide dissociation. *Journal of Mass Spectrometry* 2000; **35(12)**:1399–1406.
10. Z Zhang. Prediction of Low-Energy Collision-Induced Dissociation Spectra of Peptides. *Analytical Chemistry* 2004; **76(14)**:3908–3922.
11. A Keller, S Purvine, A Nesvizhskii, S Stolyar, DR Goodlett, E Kolker. Experimental Protein Mixture for Validating Tandem Mass Spectral Analysis. *OMICS* 2002; **6(2)**:207–212.
12. R Craig, JP Cortens, D Fenyo, RC Beavis. Using Annotated Peptide Mass Spectrum Libraries for Protein Identification. *Journal of Proteome Research* 2006; **5(8)**:1843–1849.
13. D Dutta, T Chen. Speeding up Tandem Mass Spectrometry Database Search: Metric Embeddings and Fast Near Neighbor Search. *Bioinformatics* 2007; **23(5)**:612–618.
14. JT Prince, MW Carlson, R Wang, P Lu, EM Marcotte. The need for a public proteomics repository. *Nature Biotechnoly* 2004; **22(4)**:471–472.
15. F Desiere, EW Deutsch, NL King, AI Nesvizhskii, P Mallick, J Eng, S Chen, J Eddes, SN Loevenich, R Aebersold. The PeptideAtlas Project. *Nucleic Acids Research* 2006; **34**:655–658.
16. EW Deutsch, JK Eng, H Zhang, NL King, AI Nesvizhskii, B Lin, H Lee, EC Yi, R Ossola, R Aebersold. Human Plasma PeptideAtlas. *Proteomics* 2005; **5(13)**:3497–3500.
17. F Desiere, EW Deutsch, AI Nesvizhskii, P Mallick, N King, JK Eng, A. Aderem, R Boyle, E Brunner, S Donohoe, N Fausto, E Hafen, L Hood, MG Katze, K Kennedy, F Kregenow, H Lee, B Lin, D Martin, J Ranish, DJ Rawlings, LE Samelson, Y Shiio, J Watts, B Wollscheid, ME Wright, W Yan, L Yang, E Yi, H Zhang, R Aebersold. Integration of Peptide Sequences Obtained by High-Throughput Mass Spectrometry with the Human Genome. *Genome Biology* 2004; **6**:R9.
18. http://sashimi.sourceforge.net/repository.html
19. H Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics* 1952; **23**:493–507.
20. W Feller. Stirling's Formula. *An Introduction to Probability Theory and Its Applications* 1968; **1**:50–53.