

# PepSOM: An Algorithm for Peptide Identification by Tandem Mass Spectrometry Based on SOM

Kang Ning

ningkang@comp.nus.edu.sg

Hoong Kee Ng

nghoongk@comp.nus.edu.sg

Hon Wai Leong

leonghw@comp.nus.edu.sg

Department of Computer Science, School of Computing, National University of Singapore, 3 Science Drive 2, 117543, Singapore

## Abstract

Peptide identification by tandem mass spectrometry is both an important and challenging problem in proteomics. At present, huge amount of spectrum data are generated by high throughput mass spectrometers at a very fast pace, but algorithms to analyze these spectra are either too slow, not accurate enough, or only gives partial sequences or sequence tags. In this paper, we emphasize on the *balance* between identification completeness and efficiency with reasonable accuracy for peptide identification by tandem mass spectrum. Our method works by converting spectra to vectors in high-dimensional space, and subsequently use self-organizing map (SOM) and multi-point range query (MPRQ) algorithm as a coarse filter reduce the number of candidates to achieve efficient and accurate database search. Experiments show that our algorithm is both fast and accurate in peptide identification.

**Keywords:** peptide identification, tandem mass spectrometry, self organizing map

## 1 Introduction

Peptide identification by tandem mass spectrometry is a challenging problem in proteomics. Current high throughput mass spectrometers have generated a huge amount of spectra, and the analysis of these spectra should keep pace. Fast algorithms for peptide identification are crucial for such analysis.

Unfortunately, the process of analyzing these spectrum data is still slow and not accurate. Approaches for peptide identification can be categorized into database search algorithms [6, 8, 18] and de novo algorithms [4, 7, 13, 24]. The former are suitable for known peptide sequences that already exist in the database. However, they generally do not perform well for novel peptide sequences not already available in database. For such peptide sequences, the de novo algorithms are the method of choice. De novo algorithms interpret peptide sequences from spectrum data purely by analyzing the intensity and correlation of the peaks in the spectrum data.

In the peptide identification problem, database search usually return the peptide sequences that match the parent mass of the spectrum. However, the accuracy depends on the quality of the database, and the process is slow (usually a few minutes). Typical analyses of an LC/LC/MS/MS experimental dataset using the popular BioWorks program by ThermoFinnigan with a single processor take several hours for computation (e.g. 30,000 scans against the *Escherichia coli* database). The de novo algorithm can find tags with high accuracy [3, 8], and the process is fast (always within 1 minute) but tags are usually not complete sequences for the spectra. Hence, how to achieve a *balance between identification completeness and efficiency yet manage reasonable accuracy* for peptide identification by tandem mass spectrum is an important consideration. Recently there are some research interests on this issue with solutions that combine database search with de novo techniques [8, 17, 22].

With InsPecT [8], the idea is to generate a set of highly accurate tags from spectrum data, and then use these tags to search for peptide sequences in database. The accuracy of this algorithm depends heavily on the quality of the tags. Because it uses automata to search for peptide sequences, for a batch of spectrum data, the process can be very quick (about 10 ms per spectrum).

Recently, coarse and fine filtering methods commonly associated with database search techniques are introduced for peptide identification [21]. The spectra are mapped to vectors, and using a metric space indexing algorithm, initial candidates for later fine filtering were produced. A variant of share peaks count (SPC) scoring function was used to compute the similarity among spectra. The coarse filtering can reduce the number of candidates to about 0.5% of the database and for fine filtering, a Bayesian scoring scheme is applied on candidate spectra to more accurately identify peptide sequences.

In this paper, we propose a novel peptide identification algorithm in which candidate peptide sequences are first selected from database by self-organizing map (SOM) [10] and multiple point range query (MPRQ) techniques [15, 16], and then we score and rank (fine filter) these peptide sequences by comparing their theoretical spectrum with the experimental spectrum. Since the candidates are essentially found by database search algorithm, all of the candidates in database that are similar (defined as a similarity metric which is a parameter that can be adjusted; more in Section 2.2) to the experimental spectrum are retrieved. By doing so, the completeness and efficiency are achieved, with reasonable accuracy attained.

In Section 2, we will formulate the problem at hand and subsequently describe our proposed algorithm, PepSOM. In Section 3, experiment settings and experimental results are presented, compared and analyzed. Section 4 concludes this paper with future work.

## 2 Computational Model and Algorithm

In this section, we first give some necessary definitions, and describe the concept of binning of peaks. Next, we describe SOM and MPRQ, two techniques used in database search for identifying peptides. Finally, we describe our peptide identification algorithm, PepSOM.

Consider an experimental mass spectrum  $S = \{p_1, p_2, \dots, p_n\}$  of maximum charge  $\alpha$  that is produced by an MS/MS (tandem MS) experiment on a peptide  $\rho = (a_1 a_2 \dots a_l)$ , where  $a_j$  is the  $j^{\text{th}}$  amino acid in the sequence. The parent mass of the peptide  $\rho$  is given by  $M = m(\rho) = \sum_{j=1}^l m(a_j)$ . Consider a peptide prefix fragment  $\rho_k = (a_1 a_2 \dots a_k)$ , for  $k \leq n$ , that has mass  $m(\rho_k) = \sum_{j=1}^k m(a_j)$ . Suffix masses are defined similarly. Then, the set of all possible prefixes and suffixes of a peptide forms the “full ladder” of the peptide. We always express a fragment mass in experimental spectrum using its PRM (prefix residue mass) representation, which is the mass of the prefix fragment. For suffix fragments ( $y$ -ions), we use its corresponding prefix fragment. Mathematically, for a fragment  $q$  with mass  $m(q)$ , we define  $PRM(q) = m(q)$  if  $q$  is a prefix fragment ( $\{b\text{-ion}\}$ ); and we define  $PRM(q) = M - m(q)$  if  $q$  is a suffix fragment ( $\{y\text{-ion}\}$ ). Let  $TS_0(\rho) = \{m(\rho_1), m(\rho_2), \dots, m(\rho_n)\}$  to be the set of all possible (*uncharged*) prefix fragment masses of the peptide  $\rho$ . A peak in the experimental spectrum  $S$  then corresponds to the detection of some charged prefix or suffix peptide fragment that results from peptide fragmentation in the mass spectrometer. Each peak  $p_i$  in the experimental spectrum  $S$  is described by its *intensity*( $p_i$ ) and *mass-to-charge ratio*  $mz(p_i)$ . However fragmentation is usually not very clean and other types of fragments occur. Noise and contaminants can also cause a peak in the experimental spectrum. In peptide sequencing, we are given an experimental spectrum with true peaks and noise and the problem is to try to determine the original peptide  $\rho$  that produced the spectrum.

**Theoretical Spectrum:** To theoretically characterize a multi-charge spectrum of a known peptide  $\rho$ , we consider the set of all *possible* true peaks that correspond to prefix fragments (N-terminal ions) and suffix fragments (C-terminal ions). Each peak  $p$  can be characterized by the ion-type, that is specified by  $(z, t, h) \in (\Delta_z \times \Delta_t \times \Delta_h)$ , where  $z$  is the charge of the ion,  $t$  is the basic ion-type, and  $h$

is the neutral loss incurred by the ion. In this paper, we restrict our attention to the set of ion-types  $\Delta = (\Delta_z \times \Delta_t \times \Delta_h)$ , where  $\Delta_z = \{1, 2, \dots, \alpha\}$ ,  $\Delta_t = \{a\text{-ion}, b\text{-ion}, y\text{-ion}\}$  and  $\Delta_h = \{\emptyset, -\text{H}_2\text{O}, -\text{NH}_3\}$ . The  $(z, t, h)$ -ion of the peptide fragment  $q$  (prefix or suffix fragment) will produce an observed peak  $p_i$  in the experimental spectrum  $S$  that has a mass-to-charge ratio of  $mz(p)$ , that can be computed using a shifting function, *Shift*, defined as follows:

$$m(q) = \text{Shift}(p_i, (z, t, h)) = mz(p_i) \cdot z + (\delta(t) + \delta(h)) - (z - 1)$$

where  $\delta(t)$  and  $\delta(h)$  are the mass differences associated with the ion-type  $t$  and the neutral loss  $h$ , respectively. We say that peak  $p_i$  is a *support peak* for the fragment  $q$  and *has ion-type*  $(z, t, h)$ .

We define the *theoretical spectrum*  $TS_\alpha^\alpha(\rho)$  for  $\rho$  for *maximum charge*  $\alpha$  to be the set of all *possible* observed peaks that may be present in an experimental spectrum for the peptide  $\rho$  with maximum charge  $\alpha$ . More precisely,  $TS_\alpha^\alpha(\rho) = \{p : p \text{ is an observed peak for the } (z, t, h)\text{-ion of peptide prefix fragment } \rho_k, \text{ for all } (z, t, h) \in \Delta \text{ and } k = 1, \dots, n\}$ .

## 2.1 Binning of Peaks

The very first step of PepSOM is to convert spectra in database to high-dimensional vectors of same dimension in vector space. This is related to binning of the peaks in spectrum. The binning idea was used in [19] for mass spectrum alignment. In [19], the peaks of the spectrum were packed into many bins, and the spectrum was translated into sequences comprising 0's and 1's. We used similar method for binning, except that our binning results are sequences of real numbers.

The important parameters for binning include the size of the bins, the interpretation of supporting peaks (bins), as well as the peaks (bins) intensity. Given the properly set values of tolerances, binning can preserve the accuracies, while decreasing the computational cost greatly, especially for noisy spectra. Here we give a simple lemma about binning and identification accuracies.

**Lemma 1.** Given the mass range  $m_{bin}$  for bin, and mass tolerance of  $m_t$  without binning, if we increase tolerance to  $m_t^* = m_{bin} + m_t$  after binning, then the binning will not miss any possible amino acids interpretations.

**Proof.** For any two peaks  $p_i$  and  $p_j$  with masses of  $m(p_i)$  and  $m(p_j)$  respectively, and some amino acid with mass  $m(AA_k)$ , suppose  $||m(p_i) - m(p_j)| - m(AA_k)| \leq m_t$ , there is an amino acid interpretation. Suppose after binning, their respective bins have the peak  $p_i^*$  and  $p_j^*$ . Then  $||m(p_i^*) - m(p_j^*)| - |m(p_i) - m(p_j)|| \leq m_{bin}$ . It follows that  $||m(p_i^*) - m(p_j^*)| - m(AA_k)| \leq m_{bin} + m_t$ . Given tolerance  $m_t^* = m_{bin} + m_t$  after binning, it is obvious that  $||m(p_i^*) - m(p_j^*)| - m(AA_k)| \leq m_t^*$ . Thus, the same amino acid interpretation is not missed. ■

Since the masses of amino acids are at least of 1.0 Da difference (except for I and L, and Q and K, which based on masses, cannot be distinguished by any de novo peptide identification algorithms currently), the proper value of mass tolerance  $m_t^*$  is set to be 0.5 Da, and the mass range of bin  $m_{bin}$  is set to be 0.25 Da.

In our binning process, for each bin, we have selected the peak with the highest intensity in this bin, and remove all the other peaks in the bin. After binning, we observe that there are a certain number of bins with very low intensity, and to identify the putative noisy bins, we have to score every bin. Based on domain knowledge, the important parameters for scoring should include peak intensity, the number of support peaks and mass errors.

To score every peak  $p_i$  within every bin in spectrum: Define  $N_{\text{support}}(p_i)$  as the number of  $p_j$  ( $p_j \neq p_i$ ), where  $PRM(p_j) = PRM(p_i)$ . Define the intensity function as  $f_{\text{intensity}}(p_i) = \max(0.01, \log_{10}(\text{intensity}(p_i)))$ , so that  $f_{\text{intensity}}(p_i) > 0$ . Let  $L$  be the total number of incoming and outgoing edges for  $p_i$ , and  $a_j$  be the amino acid for the edge  $(p_i, p_j)$  (or  $(p_j, p_i)$ ). Then  $\sum ||(PRM(p_j) - PRM(p_i)) - m(a_j)|/L$  is the average mass error for  $p_i$ . To avoid "divide-by-zero" errors, we define error function as  $f_{\text{error}}(p_i) = \max(0.05, \sum ||(PRM(p_j) - PRM(p_i)) - m(a_j)|/L)$ . This ensures that

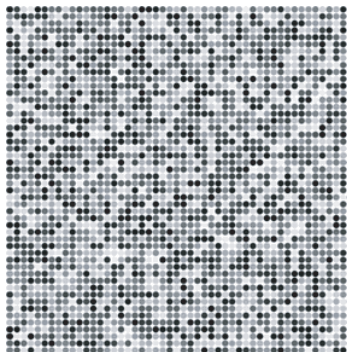


Figure 1: In this example of SOM generated from spectra, each spectrum is represented by a grayscale dot. Notice that neighboring dots have mutually similar shades of gray.

$f_{\text{error}}(p_i) \geq 0.05$ , a reasonably small error value. Then the *score*  $p_i$  is defined as

$$\text{score}(p_i) = \frac{N_{\text{support}}(p_i) + f_{\text{intensity}}(p_i)}{f_{\text{error}}(p_i)}$$

Based on the analysis of the scores of peaks in the spectrum (details not shown here), the lowest 20% bins in scores ranking, or those bins with scores less than 1% of the highest ones are filtered out. With the process of binning, many noisy peaks are also removed from spectrum. Therefore, later processes can be even more accurate (Lemma 1 shows that there is no loss of accuracy) as well as more efficient because less peaks are considered.

## 2.2 SOM and Multiple Point Range Query

SOM is a method for unsupervised learning, based on a grid of artificial neurons whose weights are adapted to match input vectors in a training set. In the training process, a SOM (map) is built and the neural network organizes itself using a competitive process. The SOM usually consists of a two-dimensional regular grid of nodes. The node whose weights are closest to the input vector, termed the best-matching or winner node, is updated to be more similar to it while the winner's neighbors are also updated (to a smaller extent) to be more similar to the input vector. As a result, when a SOM is trained over a few thousand epochs, it gradually evolves into clusters whose data (sequences) are characterized by their similarity. Therefore, it is very suitable for analysis of the similarities among sequences and is widely used. Increasingly, SOM is used as an efficient and powerful tool for analyzing and extracting a wide range of biological information as well as for gene prediction [1, 2, 14].

For spectrum data, each node represents an observation of the spectrum (converted to vector), and the distance between nodes represent their similarities. For a visual illustration, we give an example of SOM with 995 spectra (ISB test dataset, which we will describe in Section 3.1) on a 50×50 grid. Figure 1 illustrates the *relationship* among these spectra. Observe that some of the spectra (black dots) are clustered together and are hard to distinguish. Many spectra are surrounded by gray dots representing similar vectors (updated by SOM algorithm during training phase but not representing any spectrum in particular). It follows that spectra similarities are represented by neighborhoods of the points on SOM.

The general idea behind an efficient algorithm for implementing MPRQ is to perform only one pass of the R-tree while simultaneously processing multiple query points (transformed from experimental spectrum). The R-tree is widely used as a data structure for indexing 2D points. Each node of an R-tree is represented by a minimum bounding rectangle (MBR) that bounds the location of its children (of smaller MBRs) until the leaf level in which the actual 2D points are stored. At each MBR node  $R$

---

```

PepSOM(DB, ES, d)
// input:  peptide database DB, expt spectra ES, similarity d
// output: candidates results C
begin
  TS ← bin all peaks of putative peptides in DB;
  V1 ← GenerateVectors(TS);
  som_map ← TrainSOM(V1);           // SOM training
  2d_map ← MapSOM(som_map, V1); // map of (x,y)-coords
  ES ← bin all peaks of ES;         // bin ES if not previously done so
  V2 ← GenerateVectors(ES);
  Q ← MapSOM(som_map, V2);        // obtain multi points query set
  C ← MPRQ(2d_map, Q, d);          // obtain candidates set C from MPRQ query
  return C;
end;

```

---

Figure 2: Algorithm for PepSOM uses SOM and MPRQ for coarse filtering.

in the R-tree, the MPRQ algorithm processes all the children of  $R$  against all the query points. MPRQ takes only  $O(\log_B n + k)$  time, using bulkloaded R-trees (such as STR [12]) which has a bounded height of  $O(\log n)$ , where  $m$  is the number of query points,  $n$  is the total number of points in the plane,  $B$  is the disk block size, and  $k$  is the number of results found. The key observation is that when the search proceeds down the R-tree, the number of query points to be processed at each node also decreases rapidly (since the MBR is much smaller). We refer the readers to [15, 16] for more details.

Once the theoretical spectra for the peptide sequences in the database are mapped as 2D points on a SOM, we transform the query (experimental) spectra into query points in 2D plane and proceed to query. At this point, it is possible to use many experimental spectra as the query, which translates to multiple points in 2D plane as the input for MPRQ algorithm. Experiments showed that a large input (many points) *does not* increase the overall query time by a lot. This phenomenon is due to the intelligent pruning rules embedded within the MPRQ algorithm. Apart from a set of query points, the MPRQ algorithm also accepts as input a parameter  $d$  that controls the radius of the search distance. The larger the value of  $d$ , the more results will be returned. MPRQ can efficiently process the multiple input points *simultaneously* with respect to  $d$  and the MBRs during query, effectively performing multi-spectra similarity (which is adjustable) search on database of known peptides.

### 2.3 Novel Algorithm

We propose a novel peptide identification algorithm in which candidate peptide sequences are first selected from database by SOM [10] and the MPRQ [15, 16] algorithm, and then fine-filter these candidate peptides by comparing their theoretical spectrum with experimental spectrum by SPC. The theoretical spectra are binned to reduce noise and the number of peaks in consideration. Then they are converted to high-dimensional vectors and trained with SOM algorithm to obtain a SOM (map). Each spectrum is then matched with the SOM map to obtain its best-matching node (expressed in (x,y)-coordinates) which forms the basis input map for the MPRQ algorithm to perform a single, efficient query. The experimental spectra are prepared similarly (binned, vectorized, matched; albeit without the training) and the resulting coordinates form the input points for the MPRQ query. Figure 2 shows PepSOM as a course filtering step.

Upon retrieving the candidate peptides, they are compared to experimental spectrum by share peaks count (SPC). The SPC score is computed as the number of shared peaks between experimental

spectrum and theoretical spectrum of candidates (within tolerance). The flow of PepSOM is illustrated in Figure 3.

Although SOM has been used before to predict genes, this is the first attempt of its kind to combine SOM with spatial database search for peptide identification. Many efficient algorithms exist for spatial database search in orthogonal 2D grids or hierarchical data structures. SOM is useful because we believe that it satisfies the condition that the distance on the map reflects the similarity of peptides.

## 3 Experiments

### 3.1 Experiment Settings and Datasets

The experiments were performed on a PC with 3.0 GHz CPU and 1.0 GB memory, running Linux system. PepSOM was implemented in C++ and Perl. SOM\_PAK [11] was the SOM implementation that we used. We had selected two database search algorithms, Sequest [6] and InsPecT [8]; as well as two de novo algorithms with freely available implementations, Lutefisk [23] and PepNovo [7], for comparison and analysis. The best results (results with first rank) given by these algorithms were used for analysis. We treated Sequest result with cross-correlation score (Xcorr) above 2.5 as ground truth.

Spectrum datasets were obtained from Open Proteomics Database (<http://apropos.icmb.utexas.edu/OPD>) [20], PeptideAtlas database [5] and Institute for Systems Biology (ISB) [9]. We will refer to these datasets as OPD, PeptideAtlas and ISB datasets in the remainder of this paper. The three datasets chosen are of vastly different sizes to examine the issue of scalability of PepSOM as well as other algorithms.

For OPD, the spectrum dataset used was opd00001\_ECOLI, *Escherichia coli* spectra 021112.EcoliSol 37.1 (000). The spectra were obtained from *E. coli* HMS 174 (DE3) cell, which is grown in LB medium until  $\sim 0.6$  abs (OD 600). The spectra were generated by the ThermoFinnigan ESI-Ion Trap “Dexa XP Plus” and the sequences for these spectra were validated by Sequest algorithm [6]. There are 3,903 spectra in total – of which 1573, 1165 and 1165 have parent charge  $\alpha = 1, 2$  and  $3$ , respectively. We had chosen all of the 202 spectra that were identified with Xcorr above 2.5.

Spectra from PeptideAtlas database [5] were also selected. The spectrum dataset A8\_IP were obtained from Human Erythroleukemia K562 cell line. Electrospray ionization source of an LCQ Classic ion trap mass spectrometer (ThermoElectron, San Jose, CA) was used, and DTA files were generated from the MS/MS spectra using TurboSequest. The dataset consists of a total of 1,564 spectra, in which there are 782 and 782 spectra for parent charge  $\alpha = 2$  and  $3$ , respectively. We had chosen all of the 44 spectra that were identified with Xcorr above 2.5.

The ISB dataset was generated using an ESI source from a mixture of 18 proteins, obtained from ion trap mass spectrometry, and consists of spectra of up to charge 3. The ISB dataset was of low quality, having between 200-700 peaks each and an average of 400 peaks. The entire dataset consists of a total of 37,044 spectra. We had chosen all of the 995 spectra that were identified with Xcorr above 2.5.

The databases that we used were theoretical spectrum generated from the respective protein sequences dataset. Specifically, *E. coli* K12 protein sequences for OPD datasets, IPI HUMAN protein sequences for PeptideAtlas dataset and human plus control protein mixture for ISB dataset. As the number of protein sequences were very large for PeptideAtlas (60,090) and ISB (88,374) datasets, we used only the protein sequences corresponding to spectra identified with Xcorr above 2.5 (our ground truth set). However, the sizes of databases were still very large because of many fragmentations.

The parameters for the generation of databases, the test datasets and theoretical spectra are shown in Table 1. Additionally, we use a search distance radius  $d = 0.25$  as the MPRQ parameter.

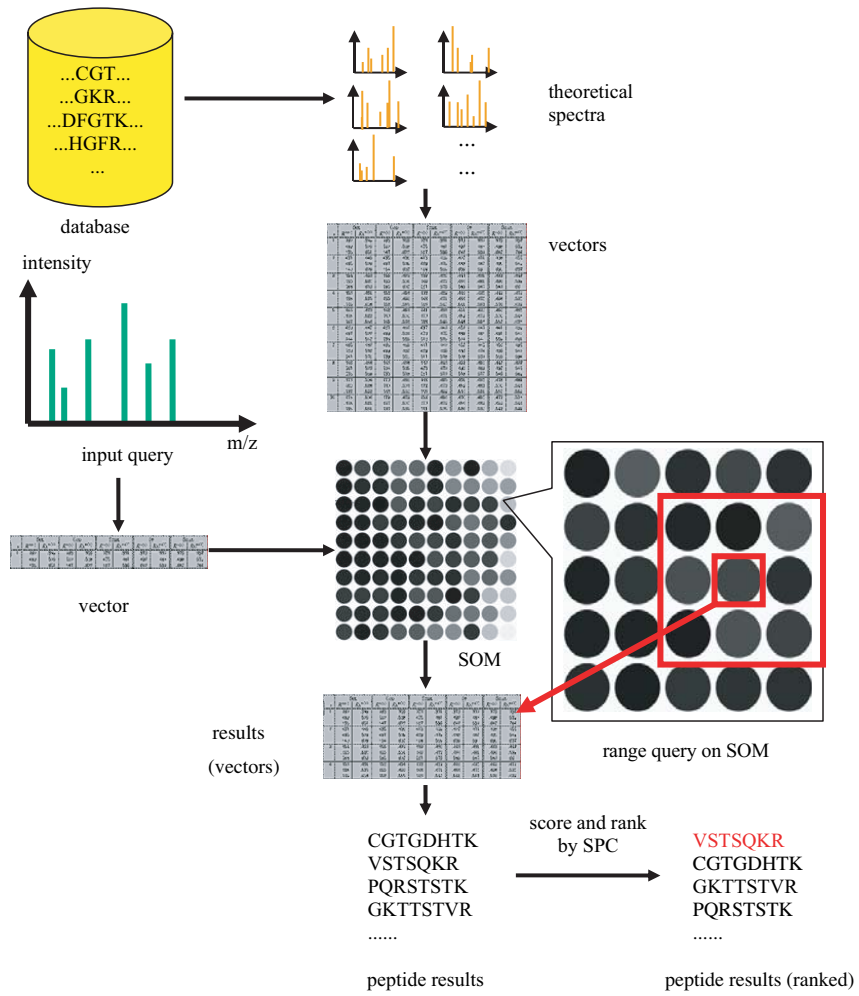


Figure 3: Diagram for the peptide identification with PepSOM.

Table 1: Parameters for the generation of databases and theoretical spectra.

Parameters	Values		
	OPD	PeptideAtlas	ISB
No. of protein sequences	4,279	31	3,553
Total database size	494,049	9,421	1,248,212
Test dataset size	202	44	995
Fragments mass tolerance	0.5 Da		
Parent mass tolerance	1.0 Da		
Modifications	–		
Charge	+2, +3		
Ion type	a, b, y, -H <sub>2</sub> O, -NH <sub>3</sub>		
Missed cleavages	0		
Protease	Trypsin		
Mass range	0-5000 Da		

The following accuracy measures were used to compare the different algorithms:

$$\text{Sensitivity} = \frac{\#correct}{|\rho|}, \quad \text{Specificity} = \frac{\#correct}{|P|},$$

where  $\# correct$  is the “number of correctly identified amino acids”. The number of correctly identified amino acids is computed as the longest common subsequence (lcs) of the correct peptide sequence  $\rho$  and the identification results  $P$  of the algorithm. Sensitivity indicates the quality of the sequence results with respect to the correct peptide sequence – a high sensitivity being that the algorithm recovers a large portion of the correct peptide. For a fairer comparison with algorithms like PepNovo that only outputs the highest scoring tags (subsequences), we also use specificity measure, which measures the number of correct results.

### 3.2 Experimental Results

We first analyzed the quality of peptide sequences identified by PepSOM as candidates. These candidates would be tested against the experimental spectra (test size) to return the final results. Generally, the size of candidates set should be as small as possible (minimal false positives) yet able to yield the final results.

Table 2: Statistical results on the quality of candidates identification by PepSOM. For specificity and sensitivity, the results for “first-rank peptide/best-match peptide” are shown.

Datasets	Database Size	Test Size	No. of Complete Correct	Complete Correct Accuracy	Specificity	Sensitivity	Time (ms)
OPD	494,049	202	44	0.218	0.560/0.785	0.428/0.593	10.6
PeptideAtlas	9,421	44	10	0.227	0.334/0.377	0.445/0.637	10.5
ISB	1,248,212	995	116	0.117	0.529/0.895	0.680/0.726	10.8

Statistical results for all three selected peptide databases are shown in Table 2. The best ranked result (highest SPC) among the results we obtained using the test set is labeled as first-rank peptide. It represents the peptide with theoretical spectrum that share the highest SPC with the experimental spectra. Best-match peptide is the peptide from all candidates that match with “real” peptide with highest specificity (or sensitivity).

From Table 2, it is clear that both sensitivity and specificity for PepSOM is high. For example, in the OPD dataset, both sensitivity and specificity are higher than 0.55; as for the ISB dataset, the sensitivity is higher than 0.65. There are also a significant number (10% to 25%) of completely correct peptide identifications among top rank peptide sequences. There are also a significant number of completely correct peptide identifications. The time taken for peptide identification is also very small, and this is expected when using both SOM and MPRQ combined (more details will be provided in the next section). The average search time for each spectrum is approximately 11 ms. This is comparable to InsPecT (with average 10 ms search time per spectrum with default settings, but based on smaller database) which is one of the fastest database search algorithms because PepSOM is able to filter a small set of high quality candidates and yet keep the accuracy of the resulting set.

Next, we compared PepSOM with other well-known peptide identification algorithms, namely Sequest [6], Lutefisk [23], PepNovo [7] and InsPecT [8] among others. Recall that the Sequest algorithm provides the spectra identified with high Xcorr score ( $\geq 2.5$ ), therefore here we treated them as ground truth.

We can observe from Table 3 that both specificity and sensitivity of PepSOM are better than Lutefisk and PepNovo (both de novo algorithms), and they are comparable to InsPecT. Although

Table 3: Comparison of different algorithms on the accuracies of peptide identification. In each column, the ‘‘Specificity/Sensitivity’’ values are listed.

Datasets	Database Size	Test Size	Sequest	InsPecT	Lutefisk	PepNovo	PepSOM
OPD	494,049	202	1.0/1.0	0.592/0.556	0.129/0.008	0.252/0.200	0.560/0.428
PeptideAtlas	9,421	44	1.0/1.0	0.811/0.402	0.162/0.063	0.291/0.135	0.334/0.445
ISB	1,248,212	995	1.0/1.0	0.602/0.633	0.032/0.032	0.563/0.593	0.529/0.680

InsPecT has higher specificity, our results outperform InsPecT in sensitivity. Specifically, for the OPD dataset, both the algorithms have specificity and sensitivity of about 0.55. For the PeptideAtlas dataset, the specificity of our algorithm is much worse than that of InsPecT, but the sensitivity is about 10% better. For the ISB dataset, PepSOM has lower specificity than InsPecT, but the sensitivity value is higher.

From these experiments, we note that the results for PepSOM are at best preliminary because of the use of conventional SPC scoring. We believe that by implementing an improved scoring function (e.g. incorporating statistical analysis or reliable tags generated by de novo process), our results could be better. All in all, we can say that PepSOM performance is comparable to InsPecT in both accuracy and efficiency.

### 3.3 Efficiency

One of the most important features of PepSOM is that it is very fast. For batch processing of multiple spectra query, Table 2 and Table 4 show that it can perform peptide identification for large spectrum datasets ( $> 500$ ) in under 30 secs ( $500 \times 10.8\text{ms} = 5.4\text{secs}$ ).

Table 4: PepSOM-generated candidates size, average query size and coarse filtering rate.

Database	Database Size	Test Size	Candidates Size	Average Query Size	Coarse Filtering Rate
OPD	494,049	202	68,610	339.7	0.069%
PeptideAtlas	9,421	44	654	14.9	0.158%
ISB	1,248,212	995	101,443	102.0	0.008%

Traditional database search algorithms such as Sequest are much slower than PepSOM. Though de novo algorithms are usually faster than PepSOM, currently, they cannot generate results with comparable accuracy. In Table 4, the candidates size represent the combined total results from coarse filtering of the database using the experimental spectra (test size) as the input query points for the MPRQ algorithm. The average query size represents the average peptide sequence candidates for each spectrum (query point). Coarse filtering rate is computed by average query size over the original database size. We only need to compare each spectrum against the candidates identified for it by MPRQ. Therefore, the coarse filtering rate is very low. Compared to the tandem cosine coarse filter used in [21] which filters to around  $\sim 0.5\%$  of the database, it is obvious our method has a better filtering efficiency. This explains why PepSOM could achieve fast search time. From Figure 4 we see that the larger search distance radius  $d$  that we use, the larger the average query size (due to the increase of number of candidates), and the selection of  $d = 0.25$  is a compromise between efficiency and accuracy. Accuracy generally improves by a little with larger  $d$  but it is not significant.

For the calculation of processing time, note that preprocessing the peptide sequences in database by SOM are needed before database search, just as InsPecT needs preprocessing to transform the database to a trie data structure. Currently, the preprocessing time for PepSOM is a few hours for all the databases, the bulk of which is time taken to generate the coordinates of the best-matching node for all the peptides in the theoretical spectrum (the MapSOM step). The actual SOM training (the TrainSOM step) for our largest database, ISB, only takes about 15 mins while PeptideAtlas took less than 1 min to train.

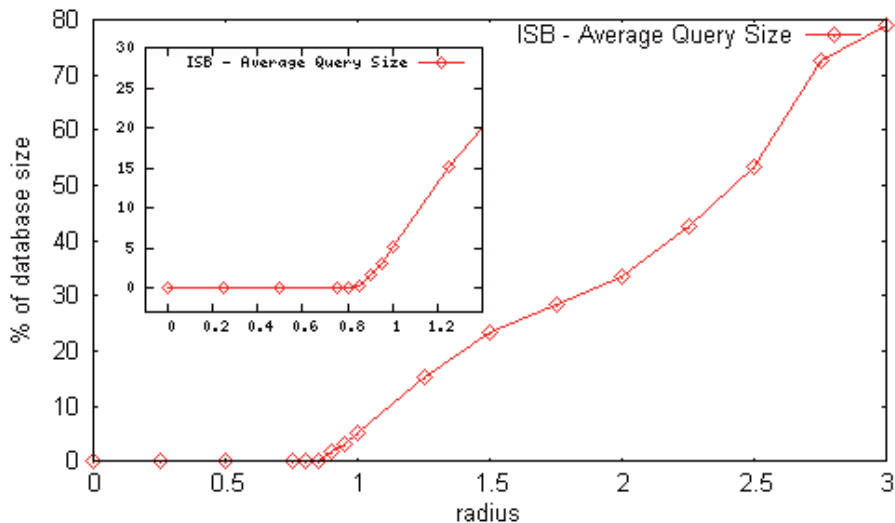


Figure 4: Average Query Size (search distance radius  $d$  vs. % of database size) for ISB dataset.

Like InsPecT, PepSOM also needs to preprocess the spectra and the search times for both are similar. As for main memory requirements, we observe that InsPecT, for the sake of efficiency, needs a great amount of memory to store the trie data structure. The huge size of sequences database is also an obstacle for us, but in our algorithm, we can fragment the database, and then transform each fragment by SOM on different workstations for parallel preprocessing. This is much more efficient, especially when done with a grid of workstations (details not shown). As the input for MPRQ is a 2D map derived from SOM-trained spectra, it can handle a large amount of points with ease typical of any general database system.

## 4 Conclusion

The peptide identification by tandem mass spectrometry is very important problem in proteomics. In this paper, we had focused on the balance between identification completeness and efficiency with reasonable accuracy for peptide identification by tandem mass spectrum.

We had proposed a new computational model that transforms spectrum similarity to similarity of vectors, and then to the neighborhood similarity of points on a 2D plane. Furthermore, we proposed the PepSOM algorithm that first selects from database of all putative peptide sequences, and transform them into vectors to be used for training by SOM and for querying by MPRQ, which together form a coarse filter for our approach. The resulting candidates are then fine-filtered by comparing their theoretical spectrum against experimental spectrum using SPC.

Experiments show that the accuracies (specificity and sensitivity of the results) of our algorithm are high. Many of our peptide identification results are identical with those identified by Sequest with high Xcorr score. These are better than or comparable with the results of presently available

most accurate database search algorithms (e.g. InsPecT). The algorithm is also efficient, especially for batch processing.

However, like other database search approaches, the accuracy of our algorithm is dependent on the completeness of spectra database to some extent. We believe that a better scoring function, or our algorithm combined with some other de novo processes, can better solve this problem. We are currently working on this.

## References

- [1] Abe, T., Sugawara, H., Kanaya, S., Kinouchi, M., and Ikemura, T., Self-Organizing Map (SOM) unveils and visualizes hidden sequence characteristics of a wide range of eukaryote genomes, *Gene*, 365:27–34, 2006.
- [2] Bertone, P. and Gerstein, M., Integrative data mining: the new direction in bioinformatics, *IEEE Eng. Med. Biol. Mag.*, 20:33–40, 2001.
- [3] Chong, K.F., Ning, K., and Leong, H.W., Characterisation of multi-charge mass spectra for peptide sequencing, *Proc. Asia Pacific Bioinformatics Conf.*, 2006.
- [4] Dancik, V., Addona, T., Clauser, K., Vath, J., and Pevzner, P., De novo protein sequencing via tandem mass-spectrometry, *J. Comp. Biol.*, 6:327–341, 1999.
- [5] Desiere, F., Deutsch, E.W., King, N.L., Nesvizhskii, A.I., Mallick, P., Eng, J., Chen, S., Eddes, J., Loevenich, S.N., and Aebersold, R., The PeptideAtlas Project, *Nucleic Acids Res.*, 34:D655–D658, 2006.
- [6] Eng, J.K., McCormack, A.L., and Yates, J.R., 3rd., An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database, *J. Am. Soc. Mass Spectr.*, 5:976–989, 1994.
- [7] Frank, A. and Pevzner, P., PepNovo: de novo peptide sequencing via probabilistic network modeling, *Anal. Chem.*, 77:964–973, 2005.
- [8] Frank, A., Tanner, S., and Pevzner, P., Peptide sequence tags for fast database search in mass spectrometry, *Proc. Research in Computational Molecular Biology*, 2005.
- [9] Keller, A., Purvine, S., Nesvizhskii, A.I., Stolyar, S., Goodlett, D.R., and Kolker, E., Experimental protein mixture for validating tandem mass spectral analysis, *OMICS*, 6:207–212, 2002.
- [10] Kohonen, T., *Self-Organizing Maps*, 3rd ed., Springer, 2001.
- [11] Kohonen, T., Hynninen, J., Kangas, J., and Laaksonen, J., SOM\_PAK: the self-organizing map program package, *Technical Report A31*, FIN-02150 Espoo, 1996.
- [12] Leutenegger, S.T., Lopez, M.A., and Edgington, J.M., STR: a simple and efficient algorithm for R-tree packing, *Proc. Int. Conf. on Data Engineering*, 1997.
- [13] Ma, B., Zhang, K., Hendrie, C., Liang, C., Li, M., Doherty-Kirby, A., and Lajoie, G., PEAKS: Powerful software for peptide de novo sequencing by MS/MS, *Rapid Commun. Mass Sp.*, 17:2337–2342, 2003.
- [14] Mahony, S., McInerney, J.O., Smith, T.J., and Golden, A., Gene prediction using the self-organizing map: automatic generation of multiple gene models, *BMC Bioinformatics*, 5–23, 2004.

- [15] Ng, H.K. and Leong, H.W., Path-based range query processing using sorted path and rectangle intersection approach, *Proc. Database Systems for Advanced Applications*, 184–189, 2004.
- [16] Ng, H.K., Leong, H.W., and Ho, N.L., Efficient algorithm for path-based range query in spatial databases, *Proc. Database Engineering and Applications Symposium*, 334–343, 2004.
- [17] Ning, K., Chong, K.F., and Leong, H.W., A database search algorithm for identification of peptides with multiple charges using tandem mass spectrometry, *Lect. Notes Bioinform.* 3916:2–13, 2006.
- [18] Perkins, D.N., Pappin, D.J.C., Creasy, D.M., and Cottrell, J.S., Probability-based protein identification by searching sequence databases using mass spectrometry data, *Electrophoresis*, 20:3551–3567, 1999.
- [19] Pevzner, P.A., Dancik, V., and Tang, C.L., Mutation-tolerant protein identification by mass-spectrometry, *Proc. Research in Computational Molecular Biology*, 231–236, 2000.
- [20] Prince, J.T., Carlson, M.W., Wang, R., Lu, P., and Marcotte, E.M., The need for a public proteomics repository, *Nat. Biotechnol.*, 22:471–472, 2004.
- [21] Ramakrishnan, S.R., Mao, R., Nakorchevskiy, A.A., Prince, J.T., Willard, W.S., Xu, W., Marcotte, E.M., and Miranker, D.P., A fast coarse filtering method for peptide identification by mass spectrometry, *Bioinformatics*, 22:1524–1531, 2006.
- [22] Tabb, D., Saraf, A., and Yates, J.R., 3rd, GutenTag: high-throughput sequence tagging via an empirically derived fragmentation model, *Anal. Chem.*, 75:6415–6421, 2003.
- [23] Taylor, J.A. and Johnson, R.S., Implementation and uses of automated de novo peptide sequencing by tandem mass spectrometry, *Anal. Chem.*, 73:2594–2604, 2001.
- [24] Taylor, J.A. and Johnson, R.S., Sequence database searches via de novo peptide sequencing by tandem mass spectrometry, *Rapid Comm. Mass Spectrom.*, 11:1067–1075, 1997.