BIOINFORMATICS

Vol. 22 no. 14 2006, pages e307–e313
doi:10.1093/bioinformatics/btl226

Peptide sequence tag-based blind identification of post-translational modifications with point process model

Chunmei Liu[1,*], Bo Yan[2], Yinglei Song[1], Ying Xu[2] and Liming Cai[1,*]

[1]Department of Computer Science, University of Georgia, Athens, GA 30602 and [2]Department of Biochemistry and Molecular Biology, University of Georgia, Athens, GA 30602

ABSTRACT

An important but difficult problem in proteomics is the identification of post-translational modifications (PTMs) in a protein. In general, the process of PTM identification by aligning experimental spectra with theoretical spectra from peptides in a peptide database is very time consuming and may lead to high false positive rate. In this paper, we introduce a new approach that is both efficient and effective for blind PTM identification. Our work consists of the following phases. First, we develop a novel tree decomposition based algorithm that can efficiently generate peptide sequence tags (PSTs) from an *extended spectrum graph*. Sequence tags are selected from all maximum weighted antisymmetric paths in the graph and their reliabilities are evaluated with a score function. An efficient deterministic finite automaton (DFA) based model is then developed to search a peptide database for candidate peptides by using the generated sequence tags. Finally, a point process model—an efficient blind search approach for PTM identification, is applied to report the correct peptide and PTMs if there are any. Our tests on 2657 experimental tandem mass spectra and 2620 experimental spectra with one artificially added PTM show that, in addition to high efficiency, our *ab-initio* sequence tag selection algorithm achieves better or comparable accuracy to other approaches. Database search results show that the sequence tags of lengths 3 and 4 filter out more than 98.3% and 99.8% peptides respectively when applied to a yeast peptide database. With the dramatically reduced search space, the point process model achieves significant improvement in accuracy as well.

Availability: The software is available upon request.

Contact: {chunmei,cai}@cs.uga.edu

1 INTRODUCTION

It is a challenging problem to determine the amino acid sequence of a protein peptide from a tandem mass spectrum. The problem becomes more difficult when the spectrum contains post-translational modifications (PTMs). Existing computational methodologies for solving this problem can be classified into two major categories: database search based approaches and *de novo* peptide sequencing. Database search based tools such as SEQUEST (Eng *et al*., 1994) and Mascot (Perkins *et al*., 1999) compare a query spectrum with spectra from peptide sequences in a database and output those with high correlation scores as sequencing candidates. When the query spectrum contains PTMs, it becomes very difficult to select the correct peptide sequence since calculation becomes prohibitively slow, due to the enumeration and scoring of all possible modifications for each peptide from the database. In contrast, *de novo* sequencing methods (Chen *et al*., 2001; Dancik *et al*., 1999; Fernandez *et al*., 1995; Han *et al*., 2005; Hines *et al*., 1992; Liu *et al*., 2006; Ma *et al*., 2003; Searle *et al*., 2004; Taylor *et al*., 2001; Yan *et al*., 2005) aim to infer a peptide sequence from its spectrum directly without looking up a protein database. However, the accuracy of *de novo* sequencing is highly sensitive to the quality of the input spectrum. Usually it cannot infer a full length peptide sequence due to missing peaks, which consequently limits its application in practice.

Most of existing approaches (Perkins *et al*., 1999; Tanner *et al*., 2005; Wilkins *et al*., 1999; Yates *et al*., 1995) for identifying PTMs assume a limited set of modification types. These modification types can be modeled with pseudo amino acids; approaches developed for spectra free of PTMs can thus be directly applied to those with PTMs. However, spectra with unknown types of modifications may be erroneously processed with this method. Recently, a few approaches have been proposed for blind PTM identification (Tsur *et al*., 2005; Yan *et al*., 2006). In particular, (Tsur *et al*., 2005) proposes a dynamic programming algorithm to solve this problem. Alternatively, (Yan *et al*., 2006) introduces a point process model to process a spectrum, in which all possible optimal alignments between two spectra are obtained feasibly by computing the correlation of their corresponding processes. Both approaches are effective and able to detect unknown types of modifications. However, due to the large size of the search space, the optimal spectral alignment may be very time consuming and both approaches may suffer high false positive rate and computing inefficiency.

Recently, the idea of database filtration based on peptide sequence tags has been introduced to speed up peptide database search (Frank *et al*., 2005a; Tabb *et al*., 2003). For example, GutenTag (Tabb *et al*., 2003), which is based on a fragmentation model, generates many short sequence tags that are possibly contained in the peptide for a spectrum and compares the spectrum with ones from peptide sequences in a database that contain at least one of the selected tags. PepNovo (Frank *et al*., 2005a, b) evaluates the reliability of sequence tags on *de novo* sequencing results with a machine learning based approach and uses high reliable sequence tags to filter out most of the peptide sequences in a peptide database. With the reduced search space, the correct peptides can thus be

*To whom correspondence should be addressed.

identified efficiently with the conventional database search methods. Apparently, the methodology of combing *de novo* peptide sequencing and database search can dramatically improve the efficiency of peptide identification without sacrificing too much sensitivity. It thus may represent a promising approach for rapid and reliable peptide identification. However, the presence of PTMs significantly increases the difficulty of both *de novo* sequencing and database search. It is unclear what performance of these tools could be for generating correct peptide sequence tags and further finding out the correct PTMs through database search, in the presence of PTMs.

In this paper, we introduce an *ab initio* approach to sequence tag selection, which when further combines with the point process model (Yan *et al*., 2006) yields an efficient and accurate method for blind PTM identification. We have observed from our previous work (Liu *et al*., 2006) that the sequence tags can be selected from the maximum weighted antisymmetric path in a spectrum graph. Due to missing peaks or the shift of peaks in a spectrum that contains PTMs, a *de novo* sequencing algorithm may not be able to find a fully connected antisymmetric path that explains the spectrum. Nevertheless, it is possible to find all maximum weighted antisymmetric paths between certain pairs of vertices in the spectrum graph to obtain partial knowledge of the amino acid sequence of the spectrum. To efficiently implement this idea, we propose a novel tree decomposition based algorithm that can efficiently and effectively find all maximum weighted antisymmetric paths in a spectrum graph. We use the notion of *extended spectrum graph* that contains additional edges to describe the relationships between pairs of *complementary* vertices. Such a graph can deal with spectra with the presence of both b-ions and y-ions and ensure the antisymmetric property of the paths.

The algorithm has two major components. The fundamental component computes the maximum weighted antisymmetric paths connecting each pair of vertices contained in each tree node from a tree decomposition of the spectrum graph. Different tree decompositions are then generated from the fundamental component to find all maximum weighted antisymmetric paths between certain pairs of vertices. The time complexity of the algorithm is $O(6^t n(n+m))$, where t is the tree width of the tree decomposition and is usually small, n is the number of peaks in the spectrum, and m is the number of maximum weighted antisymmetric paths. Sequence tags (Frank *et al*., 2005a, b) are then selected from all maximum weighted antisymmetric paths and their reliabilities are evaluated with a score function.

We implemented our algorithm and applied it to PTM identifications. We first generated sequence tags from 2657 experimental yeast spectra downloaded from the Open Proteomics Database (OPD) (Prince *et al*., 2004). We compared the accuracy of the sequence tags with those generated by the popular tool PepNovo. Our experiments shows that our *ab initio* tag generation algorithm is significantly faster than PepNovo with comparable accuracies. We then manually added PTMs to 2620 spectra from the same data set and used our program to generate sequence tags and filter a yeast peptide database with a deterministic finite automaton (DFA) based model. The point process blind search model was then applied to the selected candidate peptides to identify the PTMs. Our experiments on the spectra with PTMs show that, compared with the results without database filtration, this combined approach can achieve significantly improved accuracy with 10 times and 80 times of speedups using the filtration of sequence tags of lengths 3 and 4 respectively.

2 MODELS AND ALGORITHMS

2.1 Extended spectrum graph and sequence tag selection problem

Although a spectrum may contain a few different types of ions, there are two mostly common ion types: N-terminal ions and C-terminal ions. For simplicity, we use b-ions and y-ions to represent them respectively. We assume $S = \{s_1, s_2, \ldots, s_m\}$ to be an experimental spectrum with complementary ions added if they are missing in the original experimental spectra. The possible mass values for the partial peptide for a peak s_i in the spectrum S form a set $V_i = \{s_i + \delta_1, s_i + \delta_2, \ldots, s_i + \delta_k\}$, where δ_k is the mass offset of ion i in the form of ion type k. Each of the mass values in V_i can be represented with a graph vertex and a vertex set $V = \{v_0\} \cup \cup_{i=1}^{m} V_i \cup \{v_n\}$ can thus be generated for S, where v_0 and u_n are two additional vertices with zero mass and the parent peptide mass respectively. A *spectrum graph* (Dancik *et al*., 1999) can be constructed upon V by connecting a directed edge from u to v if the mass difference between them is the mass of a single amino acid and the mass of u is less than that of v. u is an *in-neighbor* of vertex v and v is an *out-neighbor* of vertex u.

Based on a stochastic model for ions and peaks in a spectrum, vertices and edges in a spectrum graph can be assigned weights. Traditional approaches for *de novo* sequencing determine the amino acid sequence of a peptide by finding the maximum weighted path in the spectrum graph that connects v_0 and v_n. However, since a valid sequencing path only contains either b-ions or y-ions, it is necessary to identify pairs of vertices that cannot appear in the same sequencing path. A pair of vertices are *complementary* if a sequencing path can contain at most one of them. A path in a spectrum graph is *antisymmetric* if it contains at most one vertex from each pair of complementary vertices. A valid sequencing path is thus the maximum weighted antisymmetric path that connects v_0 and v_n. To address this issue, in addition to the directed edges in a spectrum graph, we also connect complementary vertices in the spectrum graph with undirected edges, yielding an *extended spectrum graph* (Liu *et al*., 2006). We show later in the paper that these undirected edges are important to ensure the antisymmetry of the paths found by our algorithm. Figure 1(a) through (c) show the spectrum of a short peptide and an *de novo* antisymmetric sequencing path contained in the corresponding extended spectrum graph.

For most of the spectra that contain PTMs, an antisymmetric path that connects v_0 and v_n may not exist in each of the corresponding spectrum graphs. As an example, Figure 1(b)(d) show a shift of peaks and the spectrum graph of the peptide with a PTM on one of its amino acids. However, we observe that parts of the amino acid sequence of the peptide can be obtained from maximum weighted antisymmetric paths between certain pairs of vertices. A path P in a spectrum graph is *maximum weighted ntisymmetric* if it satisfies the following constraints:

(1) P is antisymmetric,

(2) if u, v are the two ends of the path, any antisymmetric path P_1 that connects u and v has a weight no larger than that of P,

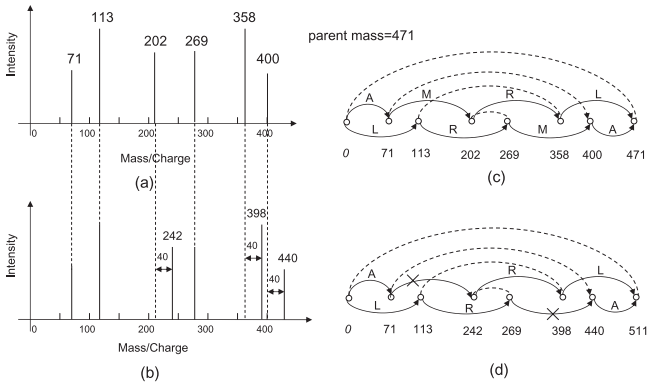

Fig. 1. (a) A tandem mass spectrum for a short peptide AMRL; for simplicity, only b and y-ions are included. (b) the spectrum for the same peptide, but with a PTM on amino acid M. (c) The extended spectrum graph for the spectrum in (a) and a longest antisymmetric sequencing path; dashed undirected edges connect complementary vertices. (d) The extended spectrum graph for the spectrum in (c), the original antisymmetric path for sequencing is disconnected due to the modification.

(3) there does not exist an antisymmetric path P_2 in the graph such that $P \subset P_2$.

2.2 Tree decompositions and path finding

DEFINITION 2.1 (Robertson *et al*., 1986) *Let $G = (V, E)$ be a graph, where V is the set of vertices in G, E denotes the set of edges in G. Pair (T, X) is a tree decomposition of graph italic G if it satisfies the following conditions*:

(1) *$T = (I, F)$ defines a tree, the sets of vertices and edges in T are I and F respectively*,

(2) *$X = \{X_i \mid i \in I, X_i \subseteq V\}$, and $\forall\, u \in V, \exists\, i \in I$ such that $u \in X_i$*,

(3) *$\forall (u, v) \in E, \exists\, i \in I$ such that $u \in X_i$ and $v \in X_i$*,

(4) *$\forall i, j, k \in I$, if k is on the path that connects i and j in tree T, then $X_i \cap X_j \subseteq X_k$.*

The tree width of the tree decomposition (T, X) is defined as $max_{i \in I} |X_i| - 1$. The tree width of the graph G is the minimum tree width over all possible tree decompositions of G.

As shown in Figure 2(a)(b), tree decomposition provides a new topological view on a graph. Based on a tree decomposition of a graph, many NP-hard optimization problems can be efficiently solved with a generic dynamic programming framework (Arnborg *et al*., 1989). In this framework, partial optimal solutions on subgraphs induced by vertices contained in subtrees can be extended and combined to obtain optimal solutions for larger subgraphs. In particular, partial optimal solutions can be combined with an exhaustive search performed only on vertices contained in a single tree node. The computation time needed by such a dynamic programming approach is thus dominantly determined by the tree width of the tree decomposition. Our testing results on 2657 experimental spectra show that the tree widths of extended spectrum graphs are generally around 5, which is sufficiently small for designing an efficient algorithm based on this framework.

The path-finding algorithm is based on the tree decompositions of the extended spectrum graph. The core part of the algorithm

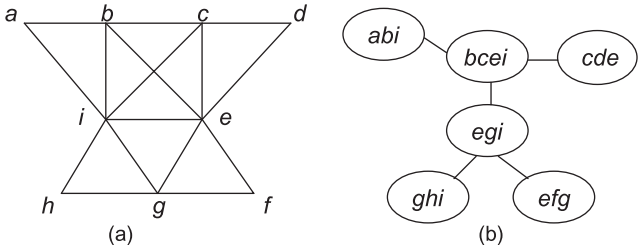

Fig. 2. (a) An example of a graph. (b) A tree decomposition for the graph in (a).

consists of two major components. In particular, in a given tree decomposition, the fundamental component finds the maximum weighted antisymmetric paths that connects each pair of the vertices contained in each tree node. To find all maximum weighted antisymmetric paths connecting certain pairs of vertices in the graph, the algorithm generates different tree decompositions and applies the procedure of the fundamental component to each of them. The overall time complexity of the algorithm is $O(6^t n(n+m))$, where t is the tree width, n is the number of vertices in the spectrum graph, and m is the number of maximum weighted antisymmetric paths in the spectrum graph.

2.2.1 The Fundamental Component The algorithm arbitrarily selects a tree node as the root of a tree decomposition and maintains a dynamic programming table for each tree node. It then proceeds from leaves of the tree to the root to fill in all the dynamic programming tables. The table for each tree node stores the weight of the partial maximum weighted antisymmetric path connecting each pair of vertices in the tree node.

For a tree node with t vertices, the dynamic programming table contains $2t + 1$ columns, of which the first t columns store the *selection* of each vertex in the node to form a subpath and the other $t - 1$ columns are used to store the *connection state* between each pair of consecutive selected vertices in the tree node. Two additional columns V and L store the *valid bit* and the maximum weight of the partial antisymmetric path associated with a combination of selections and connection states in the same table entry respectively.

The selection value of a vertex in a tree node is 1 if it is selected to be in the partial optimal path and 0 otherwise. The value of a connection state could be one of the integers in set $\{0, 1, \ldots, l\}$, where l is the number of children of the tree node. The connection state for a pair of consecutive selected vertices in the tree node is 0 if they are contiguous in the path and is i ($i > 0$) if the vertices on the path between the pair of vertices are covered by the subtree rooted at the ith child. The number of possible combinations of selections and connection states can thus be up to $(2(l + 1))^t$. However, since we can remove tree nodes with more than two children by generating extra tree nodes, the table for a tree node with t vertices may contain up to 6^t entries. The valid bit for a given entry is set to be 1 if there exists a partial antisymmetric path that follows the combination of selections and connection states in the entry.

To determine an entry in the table for a leaf node, the algorithm exhaustively enumerates and directly computes the validity and the maximum path weight for every possible combination of

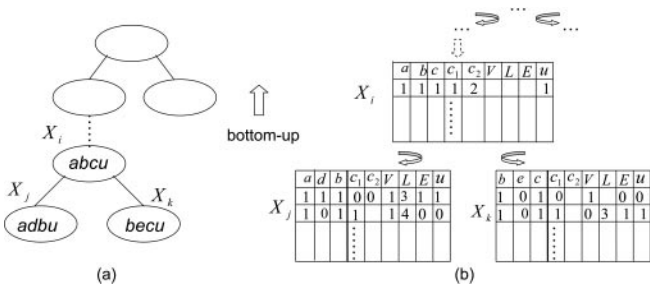

Fig. 3. A tree decomposition and its corresponding dynamic programming tables. The algorithm follows a bottom-up fashion starting with the leaf tree nodes. When computing the dynamic programming tables for an internal node X_i, the tables of its child nodes X_j and X_k need to be queried to compute the validity (V), the maximum path weight (L) and the extendibility (E) of a given entry in the table for X_i; vertex u is added to each tree bag to compute all the maximum weighted antisymmetric paths that start with it.

selections and connection states for vertices in the node. For an internal node, the algorithm refers to the tables of its children to determine the validity and the maximum path weight for each of its table entry. Figure 3 provides an example for computing the table entries for an internal node X_i. The computation time needed by the algorithm is $O(6^t n)$, where t is the tree width of the tree decomposition and n is the number of vertices in the graph. Due to space constraint, we refer the reader to our previous paper (Liu *et al*., 2006) for details.

2.2.2 Finding all maximum weighted antisymmetric paths The fundamental component can only compute the maximum antisymmetric paths between vertices that are included in at least one tree node. Further processing is thus needed to find all maximum weighted antisymmetric paths in the spectrum graph. For each vertex u in the spectrum graph, a new tree decomposition can be constructed by including u in all the tree nodes in the original tree decomposition. n different tree decompositions are thus generated.

To guarantee that the paths found by the algorithm satisfy the constraints of being maximum weighted antisymmetric, we further modify the previously described fundamental component. Specifically, as shown in Figure 3, one additional column E is added to each dynamic programming table to indicate whether the corresponding path can be extended to form an antisymmetric path with a larger weight by adding one of the in-neighbors of u to the path. This bit is set to be 1 if such an extension exists and 0 otherwise. The algorithm also sets the E bit to be 0 if the corresponding path for an entry does not start with u. This property for a given entry can be obtained by combining the E bits of its descendent entries with a direct inspection on vertices that are not included in the descendent entries. In view of the fact that the number of in-neighbors of u is bounded by 20, the aggregate computation time for this additional checking is $O(6^t n)$.

Based on the E bit of an entry, we are able to select the antisymmetric paths that start with u and cannot be extended with an in-neighbor of u. However, it is possible that some of the maximum antisymmetric paths can be extended from the other end of the path. We thus create an array S of size n and initialize all its elements to be zero. For any vertex v other than u in the spectrum graph, we can obtain $W(u, v)$, the weight of the maximum weighted antisymmetric path that connects u and v. For each out-neighbor v_t of v, we obtain $W(u, v, v_t)$, the weight of the maximum weighted antisymmetric path that passes through v and connects u and v_t. We then check whether $W(u, v, v_t)$ is equal to $W(u, v) + w(v, v_t)$ or not, where $w(v, v_t)$ is the weight of the edge (v, v_t). If it is the case for one out-neighbor of v, we set $S[v]$ to be 1, which suggests that the maximum antisymmetric path between u and v is extendable. The correctness of this operation is obvious since only in the case where the path is extendable, we can have one out-neighbor v_t of v such that $W(u, v, v_t) = W(u, v) + w(v, v_t)$. The aggregate time for this operation is again $O(6^t n)$. We then apply the tracing back procedure in the fundamental component to obtain all the maximum weighted antisymmetric paths starting with u. Based on the E bits and the array S, we can find all maximum weighted antisymmetric paths from the n tree decompositions and the total computation time is $O(6^t n(n + m))$, where m is the total number of maximum weighted antisymmetric paths. We thus have the following theorem.

THEOREM 2.1. *Given an extended spectrum graph $G = (V, E)$ and a tree decomposition of G with tree width t, all maximum weighted antisymmetric paths in G can be identified in time $O(6^t |V|(|V| + m))$, where m is the total number of maximum weighted antisymmetric paths in G.*

2.3 Reliability of sequence tags

We used the scoring scheme proposed in (Dancik *et al*., 1999) to assign weights to the vertices and edges in the extended spectrum graphs. The overall reliability of a sequence tag t_i was considered as a linear combination of normalized reliabilities $r_1(t_i)$ and $r_2(t_i)$ computed from the weights of the corresponding edges for t_i and an autocorrelation score developed in (Liu *et al*., 2005) respectively. In particular, the reliability $r(t_i)$ of sequence tag t_i is

$$r(t_i) = w_1 r_1(t_i) + w_2 r_2(t_i) \tag{1}$$

where $r_1(t_i)$ and $r_2(t_i)$ are computed with

$$r_1(t_i) = \frac{W(t_i)}{\sum_{l=1}^{q} W(t_l)} \tag{2}$$

$$r_2(t_i) = \frac{A(t_i)}{\sum_{l=1}^{q} A(t_l)} \tag{3}$$

where $W(t_i)$ is the sum of the weights of the edges that form t_i in the extended spectrum graph, q is the number of sequence tags, and $A(t_i)$ is an autocorrelation score computed with

$$A(t_i) = \sum_{k \in P(t_i)} I^*(k) I^*(n - k) \tag{4}$$

where $P(t_i)$ is the set of peaks that form t_i and $I*(k)$ and $I*(n - k)$ are adjusted intensities of complementary peaks k and $n - k$ in the spectrum. Both $r_1(t_i)$ and $r_2(t_i)$ are obtained by normalizing $W(t_i)$ and $A(t_i)$ over all sequence tags that are selected from the maximum weighted antisymmetric paths.

2.4 Database filtration with sequence tags

From the generated peptide sequence tags, we introduced a deterministic finite automaton (DFA) based model and used it to search a

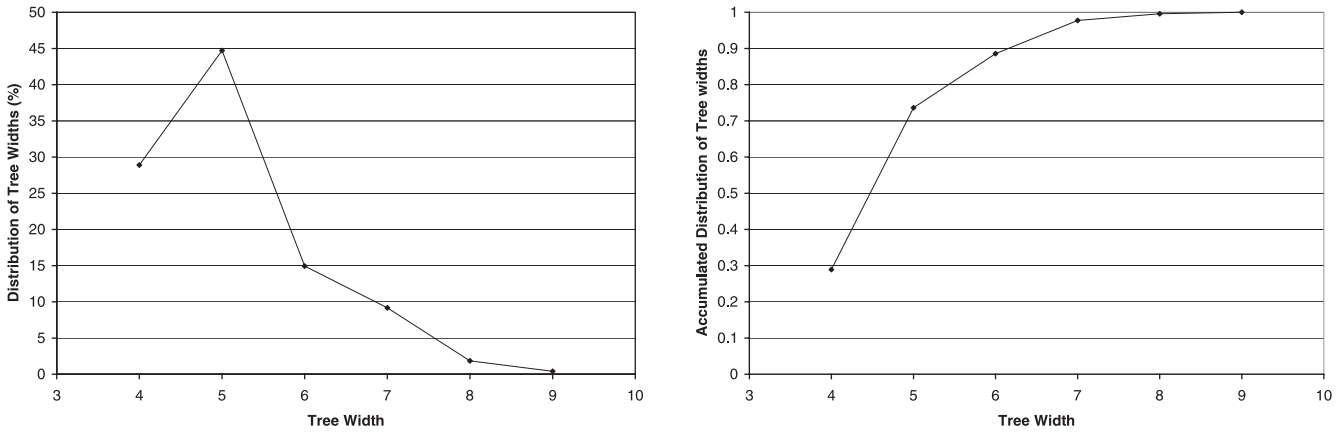

Fig. 4. The tree widths of the extended spectrum graphs for 2657 experimental spectra; left: the distribution of the tree widths, right: the cumulative distribution of the tree widths.

yeast peptide database which consists of 670,000 tryptically digested peptides (allowed up to 2 missing cleavages). Each amino acid in the tags represents a state of the DFA. We added an additional state as the start state. The accept states are the states that correspond to the last amino acids in the tags. Upon reading the first amino acid in a peptide sequence in the database, the DFA transfers from the start state to an appropriate state that corresponds to the first amino acid in a tag. The DFA then transfers from that state to next appropriate state upon reading the following amino acid in the peptide sequence. The procedure continues until the end of the peptide sequence. The peptide reading always goes forwards in the entire procedure. However, a trie based model (Frank *et al*., 2005a) needs to go back to the head of the trie each time when a substring of tag length in the peptide sequence has been examined. It thus may need more computation time than our DFA based model.

2.5 PTM identification by point process blind search

We finally apply the point process model for peptide identification and PTM search (Yan *et al*., 2006) on the candidate peptides after filtration. This model is an efficient blind search approach that does not require a list of pre-specified PTMs as input in advance. The algorithm attempts to find a set of optimal mass shifts to maximize the spectral alignment. Through one round of cross-correlation calculation, it is able to obtain all possible mass shifts feasibly (naturally this includes the optimal mass shifts). The computation time is independent of the number of PTMs, which outperforms most of the existing PTM identification tools whose computation time grows exponentially with the number of PTMs.

3 EXPERIMENTS AND DISCUSSIONS

3.1 Datasets

We downloaded 2657 annotated yeast ion trap tandem mass spectra from the Open Proteomics Database (OPD) (Prince *et al*., 2004). These spectra were selected based on the criteria with +2 precursor ion and $Xcorr \geq 2.5$ without PTMs. All the experimental mass spectra were ion trap data having a relative low mass resolution. We ran a data preprocessing procedure as described in (Frank *et al*., 2005b) to remove isotopic peaks and tiny noise peaks. Due to the shortage of reliably annotated spectra with PTMs in public domain, we constructed 2620 modified ones from those spectra by artificially adding one PTM from a common PTM pool to each spectrum. The detailed procedure is referred to (Yan *et al*., 2006).

3.2 Tree widths for spectrum graphs

Computing the optimal tree decomposition for a given graph is an NP-hard problem (Arnborg *et al*., 1987). A few efficient heuristics (Bodlaender, 1991) have been developed to compute a tree decomposition with small tree width for certain types of graphs. We used a greedy fill-in heuristic (Bodlaender, 1991) to find tree decompositions for the spectrum graphs of the experimental spectra. Figure 4 shows the distribution of the tree widths of the 2657 spectrum graphs. It can be clearly seen from the figure that the tree widths of about 90% of the spectrum graphs are bounded by 6, which are sufficiently small for developing an efficient tree decomposition based algorithm.

3.3 Sequence tag generation

We used our program to generate sequence tags at different lengths on the two datasets. We also ran the public available program PepNovo on the same datasets to obtain sequence tags. We then compared the generated tags with the sequencing results by SEQUEST and obtained the percentages of correct tags at different lengths for both of our program and PepNovo. Table 1 lists the results of our experiments on the two datasets and the comparison with PepNovo at different tag lengths. Our approach achieves comparable performance to PepNovo and is more computationally efficient (over 10 times faster than PepNovo at all different tag lengths). More importantly, our approach is *ab-initio* and does not require a training data set as PepNovo does. We believe further improvements in accuracy can be achieved if a more sophisticated model to evaluate the reliabilities of generated sequence tags is applied in the future.

3.4 Blind PTM identification by database search

After candidate peptides are filtered out with the sequence tags, our point process based blind search model is applied to evaluate these candidate peptides for further peptide identification and PTM detection. The results on 2620 modified spectra are listed in Table 2. It can be seen that the sequence tags of lengths 3 and 4 are able to

Table 1. A comparison between the performance of our tag selection program and that of PepNovo at different tag lengths

	Tag length	Algorithm	$r = 1(\%)$	$r = 3(\%)$	$r = 5(\%)$	$r = 10(\%)$	$r = 25(\%)$	T (s)
a	3	Our program	75.8	89.1	94.6	96.9	98.1	0.33
		PepNovo	75.8	90.1	93.6	96.8	98.8	3.62
	4	Our program	65.3	80.5	88.7	93.6	96.4	0.34
		PepNovo	65.5	81.0	86.6	92.3	95.3	3.69
	5	Our program	56.4	72.8	78.3	85.1	89.8	0.33
		PepNovo	58.4	71.3	77.6	84.0	88.9	3.83
	6	Our program	50.2	62.3	66.9	76.6	82.4	0.34
		PepNovo	49.7	61.5	67.8	75.0	81.8	4.27
b	3	Our program	68.1	84.8	90.3	94.8	97.1	0.32
		PepNovo	62.8	83.7	89.7	94.9	97.8	3.59
	4	Our program	53.5	71.2	78.6	84.8	90.0	0.32
		PepNovo	51.1	71.7	79.3	85.8	91.4	3.64

(a) on 2657 experimentalspectra without PTMs and (b) on 2620 experimentalspectra with one artificially added PTM. Columns for $r = 1, 3, 5, 10, 25$ represent the percentagesof spectra that have at least onecorrect tag in top 1, 3, 5, 10, 25 tags generated by our programand PepNovo respectively; T is the average time in seconds used for generating sequence tags for one spectrum.

Table 2. The accuracy of identifying PTMs from the modified spectra with selected tags of lengths 3 and 4

Tag length	Top 1	Top 2	Top 3	Top 4	Top 5	Filtration ratio	Time(s)
3	76.69	86.01	89.29	90.70	91.62	0.0167	263
4	74.98	80.77	81.71	82.17	84.40	0.0014	34
Without filtration	60.38	72.33	76.64	79.16	81.17	–	3843

The values at Top $i = 1, 2, 3, 4, 5$ represent the cumulative percentages of the search results capturing the original peptide sequences exactly in Top i; Filtration ratio is the ratio of the survived candidate peptides after tag filtration. Time is the total time in seconds used for the point-process blind search model to identify correct peptides and PTMs for all the 2620 experimental spectra. The last row is the results without sequence tag filtration.

filter out more than 98.3% and 99.8% peptides in the database respectively, which consequently speeds up the calculations dramatically. In addition, with the reduced search space and enriched signals of correct peptides, the accuracies of PTM identification by database search are significantly improved with both sequence tags of lengths 3 and 4. For example, with the filtration of tag length 3, approximately 77% and 92% of spectra are identified correctly as top 1 and within top 5 respectively, a significant improvement compared to the corresponding accuracies of 60% and 81% without database filtration. Increasing the tag length from 3 to 4 can further speed up the PTM identification by approximately 8 times. However, a slight drop in the identification accuracy is observed in this case due to the relative lower sensitivity of tag generation for tag length 4.

4 CONCLUSIONS

In this paper, we develop a novel tree decomposition based algorithm that can efficiently generate highly accurate sequence tags and conduct efficient PTM identification by combining sequence tag generation and database search. The algorithm models a spectrum with its corresponding extended spectrum graph and can find all maximum weighted antisymmetric paths in the spectrum graph with tree width t in time $O(6^t n(n+m))$, where n and m are the number of vertices and the number of maximum weighted antisymmetric paths in the graph, respectively. Sequence tags are then selected from all the maximum weighted antisymmetric paths. Our experiments show that this *ab-initio* approach can achieve accuracy comparable to that of PepNovo in a significantly reduced amount of computation time. More importantly, the sequence tags can be used to filter a peptide database effectively and thus enable the application of more accurate and sophisticated algorithms for PTM identification. In particular, we have built a rigid framework to conduct peptide identification and blind PTM search by combining high quality sequence tag generation and efficient database search. Experiments on spectra with PTMs show that this new approach can generate highly accurate sequence tags and significantly improve the accuracy of PTM identification by blind search.

ACKNOWLEDGEMENT

BY and YX's work was supported in part by National Science Foundation (NSF/DBI-0354771, NSF/ITR-IIS-0407204), and by a 'Distinguished Cancer Scholar' grant from the Georgia Cancer Coalition.

REFERENCES

Arnborg,S., Corneil,D. G. and Proskurowski,A. (1987) Complexity of finding embeddings in a *k*-tree. *SIAM Journal on Algebraic and Discrete Methods*, **8** (2), 277–284.

Arnborg,S. and Proskurowski,A. (1989) Linear time algorithms for NP-hard problems restricted to partial *k*-trees. *Discrete Applied Mathematics*, **23**, 11–24.

Bodlaender,H. L. (1991) Better algorithms for the pathwidth and treewidth of graphs. *Proceedings of the 18th international Colloquium on Automata, Languages and Programming*. Springer Verlag, Lecture Notes in Computer Science, **510,** 544–555.

Chen,T., Kao,M.Y., Tepel,M., Rush,J. and Church,G. M. (2001) A Dynamic Programming Approach to De Novo Peptide Sequencing via Tandem Mass Spectrometry. *Journal of Computational Biology*, **8** (3), 325–337.

Dancík,V., Addona,T.A., Clauser,K.R., Vath,J.E. and Pevzner,P.A. (1999) *De Novo* Peptide Sequencing via Tandem Mass Spectrometry. *Journal of Computational Biology*, **6** (3/4), 327–342.

Eng,J.K., McCormack,A.L. and Yates III,J.R. (1994) An Approach to Correlate Tandem Mass Spectral Data of Peptides with Amino Acid Sequences in A Protein Database. *Journal of the American Society of Mass Spectrometry*, **5** (11), 976–989.

Ferńandez de Cossío,J., Gonzales,J. and Besada,V. (1995) A computer program to aid the sequencing of peptides in collision-activated decomposition experiments. *CABIOS*, **11** (4), 427–434.

Frank,A., Tanner,S. and Pevzner,P. (2005) Peptide Sequence Tags for Fast Database Search in Mass-Spectrometry. *Journal of Proteome Research*, **4** (4), 1287–1295.

Frank,A. and Pevzner,P. (2005) PepNovo: De Novo Peptide Sequencing via Probabilistic Network Modeling. *Anal. Chem.*, **77** (4), 964–973.

Han,Y., Ma,B. and Zhang,K. (2005) SPIDER: Software for Protein Identification from Sequence Tags Containing Sequencing Error. *Journal of Bioinformatics and Computational Biology*, **3** (3), 697–716.

Hines,W.M., Falick,A.M., Burlingame,A.L. and Gibson,B.W. (1992) Pattern-based algorithm for peptide sequencing from tandem high energy collision-induced dissociation mass spectra. *J. Am. Soc. Mass. Spectrom.*, **3**, 326–336.

Liu,J., Ma,B. and Li,M. (2005) PRIMA: Peptide Robust Identification from MS/MS Spectra. *Proceedings of the Third Asia-Pacific Bioinformatics Conference*, 181–190.

Liu,C., Song,Y., Yan,B., Xu,Y. and Cai,L. (2006) Fast De Novo Peptide Sequencing and Spectral Alignment. *Proceedings of the Pacific Symposium on Biocomputing 2006*, 255–266.

Ma,B., Zhang,K., Hendrie,C., Liang,C., Li,M., Doherty-Kirby,A. and Lajoie,G. (2003) PEAKS: Powerful Software for Peptide De Novo Sequencing by Tandem Mass Spectrometry. *Rapid Communication in Mass Spectrometry*, **17** (20), 2337–2342.

Perkins,D.N., Pappin,D.J., Creasy,D.M. and Cottrell,J.S. (1999) Probability-based Protein Identification by Searching Sequence Databases Using Mass Spectrometry Data. *Electrophoresis*, **20** (18), 3551–3567.

Prince,J.T., Carlson,M.W., Wang,R., Lu,P. and Marcotte,E.M. (2004) The Need for a Public Proteomics Repository. *Nature Biotechnology*, **22** (4), 471–472.

Robertson,N. and Seymour,P.D. (1986) Graph Minors II. Algorithmic aspects of tree-width. *Journal of Algorithms*, **7**, 309–322.

Searle,B.C., Dasari,S., Turner,M., Reddy,A.P., Choi,D., Wilmarth,P.A., McCormack,A.L., David,L.L. and Nagalla,S.R. (2004) High-Throughput Identification of Proteins and Unanticipated Sequence Modifications Using a Mass-Based Alignment Algorithm for MS/MS De Novo Sequencing Results. *Anal. Chem*, **76** (8), 2220–30.

Tabb,D.L., Saraf,A. and Yates,J.R. (2003) GutenTag: High-Throughput Sequence Tagging via an Empirically Derived Fragmentation Model. *Analytical Chemistry*, **75**, 6415–6421.

Tanner,S., Shu,H., Frank,A., Wang,L.C., Zandi,E., Mumby,M., Pevzner,P.A. and Bafna,V. (2005) InsPecT: Identification of Posttranslationally Modified Peptides from Tandem Mass Spectra. *Analytical Chemistry*, **77** (14), 4626–4639.

Taylor,J.A. and Johnson,R.S. (2001) Implementation and uses of automated de novo peptide sequencing by tandem mass spectrometry. *Anal.Chem.*, **73** (11), 2594–2604.

Tsur,D., Tanner,S., Zandi,E., Bafna,V. and Pevzner,P. (2005) Identification of Post-translational Modifications by Blind Search of Mass Spectra. *Nature Biotechnology*, **23** (12), 1562–1567.

Wilkins,M.R., Gasteiger,E., Gooley,A.A., Herbert,B.R., Molloy,M.P., Binz,P.A., Ou,K., Sanchez,J.C., Bairoch,A., Williams,K.L. and Hochstrasser,D.F. (1999) High-throughput Mass Spectrometric Discovery of Protein Post-Translational Modifications. *Journal of Molecular Biology*, **289** (3), 645–657.

Yan,B., Pan,C., Olman,V.N., Hettich,R.L. and Xu,Y. (2005) A Graph-Theoretic Approach for the Separation of b and y Ions in Tandem Mass Spectrometry. *Bioinformatics*, **21** (5), 563–574.

Yan,B., Zhou,T., Wang,P., Liu,Z., Emanuele II,V.A., Olman,V. and Xu,Y. (2006) A Point-Process Model for Rapid Identification of Post-Translational Modifications. *Proceedings of 2006 Pacific Symposium on Biocomputing*, 327–338.

Yates III,J.R., Eng,J.K. and McCormack,A.L. (1995) Mining Genomes: Correlating Tandem Mass Spectra of Modified and Unmodified Peptides to Sequences in Nucleotide Databases. *Analytical Chemistry*, **67** (18), 3202–3210.