Fast De novo Peptide Sequencing and Spectral Alignment via Tree Decomposition

Chunmei Liu, Yinglei Song, Bo Yan, Ying Xu, and Liming Cai

Pacific Symposium on Biocomputing 11:255-266(2006)

FAST *DE NOVO* PEPTIDE SEQUENCING AND SPECTRAL ALIGNMENT VIA TREE DECOMPOSITION

CHUNMEI LIU[1,*] YINGLEI SONG[1], BO YAN[2], YING XU[2], LIMING CAI[1,*]
[1] *Department of Computer Science and* [2] *Department of Biochemistry and Molecular Biology*
University of Georgia, Athens GA 30602, USA

De novo sequencing and spectral alignment are computationally important for the prediction of new protein peptides via tandem mass spectrometry (MS/MS). Both approaches are established upon the problem of finding the longest antisymmetric path on formulated graphs. The problem is of high computational complexity and the prediction accuracy is compromised when given spectra involve noisy data, missing mass peaks, or post translational modifications (PTMs) and mutations. This paper introduces a graphical mechanism to describe relationships among mass peaks that, through graph tree decomposition, yields linear and quadratic time algorithms for optimal *de novo* sequencing and spectral alignment respectively. Our test results show that, in addition to high efficiency, the new algorithms can achieve desired prediction accuracy on spectra containing noisy peaks and PTMs while allowing the presence of both b-ions and y-ions.

1. Introduction

Tandem mass spectrometry (MS/MS) has been extensively used in proteomics to identify and analyze proteins[2,3,9]. In this method, molecules of a protein can be cleaved into short peptide sequences by enzymes. Amino acids in these peptides are then determined and combined to obtain the sequence of the protein. To sequence a peptide, sequences with the same amino acids are fragmented into charged prefix and suffix subsequences (ions) and their mass/charge ratios can be measured by a mass spectrometer. In a theoretical MS/MS spectrum, there are usually two types of ions present: b-ions associated with N-terminals and y-ions with C-terminals. Ideally, fragmentation may occur at any position along the peptide backbone and we thus expect to be capable of inferring the amino acids a peptide contains from its MS/MS spectrum and the masses of single amino acids.

[*] Corresponding authors: {chunmei, cai}@cs.uga.edu

However, difficulty may arise when we intend to identify the ion types for mass peaks. In addition, experimental spectra are usually incomplete and contain noisy peaks. Therefore, the *de novo* sequencing of a peptide solely from its spectrum remains a challenging task[5,6].

A number of algorithms have been developed for the *de novo* sequencing problem. An early developed algorithm[12] generates all amino acid sequences and the corresponding theoretical spectra to be compared with the experimental spectrum. Since, to find out the best match an exponential number of spectra may need to be generated, the algorithm is not efficient. Prefix pruning approaches have been developed to speed up the search by restricting it to sequences whose prefixes match the spectrum well[13,17,18]. However, heuristic pruning may adversely affect the sequencing accuracy while the computation time may remain expensive. Recently, based on the notion of spectrum graph[6], the *de novo* sequencing problem has been reduced to finding the longest (or maximum scored) antisymmetric path in directed graphs[2,6,7,8,15]. However, a straightforward path-finding algorithm may yield undesired paths containing multiple vertices associated with complementary ions. This issue was resolved later with a linear time dynamic programming algorithm[5] that ensures the path found to be antisymmetric. However, it requires quadratic time to discover one modified amino acid and more time to deal with additional noisy peaks.

Comparing and evaluating the similarity between two spectra are often used in database search for peptide identification[10]. Traditional methods for computing the similarity identify the shared mass peaks between two spectra and use the count as a measure of the similarity. More recently, spectral alignment was proposed as a new method for evaluating spectral similarity; it proves useful for identifying related spectra in the presence of post translational modifications (PTMs) and mutations[9]. In particular, based on finding the longest (k-shift) path in alignment graph, a spectral alignment algorithm can align two spectra of n peaks in time $O(n^2k)$, where k is the maximum number of peak shifts resulting from PTMs[9]. However, the algorithm considers only b-ions or y-ions but not both. To consider both ion types, a dynamic programming algorithm in the same spirit as that for *de novo* sequencing[5] is possible. But it would require a computation time that is polynomial of a much higher degree.

In this paper, we introduce a graphical mechanism to describe related mass peaks in spectra. In particular, the peaks associated with complementary ions are linked with non-directed edges, yielding extended spectrum graphs and extended alignment graphs. Such graphs demonstrate small tree width t (usually $t \leq 6$) for real mass spectra, so a very efficient algorithm for finding the longest antisymmetric path can be devised based on the tree decompositions of these graphs. In particular, the resulting new algorithms for *de novo* sequencing and spectral alignment run in time $O(6^t n)$ and $O(6^t n^2)$ respectively. Based on the notion of tree decomposition, the antisymmetry of complementary vertices can be efficiently ensured on the found path; both ion types can be simultaneously considered by our algorithms. In addition, using the graphical mechanism, vertices for peaks with similar masses can be easily related and considered simultaneously in the tree decomposition-based dynamic programming algorithm. This allows vertices for noisy peaks to be eliminated from the found path.

We have implemented the algorithms and tested their performance on both simulated spectra and real experimental ones with noisy peaks. Our algorithm is able to identify the correct peptide sequences from all the tested spectra with noisy peaks in a few seconds. In particular, the algorithm achieves more than 96% accuracy on spectra in which the number of noisy peaks is the same as that of others. In addition, we used the algorithm to identify PTMs of amino acids based on spectra generated *in silico*. Test results for spectral alignment demonstrated that the algorithm can identify all PTMs accurately in a few seconds.

2. Models and Algorithms

2.1. *Problem Description*

Since theoretically, any ion has its complementary ion contained in the same spectrum[16], we assume the MS/MS spectrum S of a peptide P be a set of mass peaks $\{x_1, x_2, \cdots, x_{2k}\}$, where $x_i > x_j$ for $i > j$. For any mass peak x_i in S, there exists a mass peak x_{2k+1-i} *complementary* to x_i and the sum of their mass values is the total mass M of P. One of x_i and x_{2k+1-i} is a b-ion and the other is a y-ion. A *spectrum graph* $G = (V_s, E_s)$ can be constructed from the mass peaks in S. Specifically, vertex $v_i \in V_s$ represents x_i and, in addition to the mass peaks in S, vertices *source* v_0 and *sink* v_{2k+1} are included in G with virtual mass values 0 and M respectively. Directed edge $(v_i, v_j) \in E_s$ if the mass value difference $x_j - x_i$ is the mass of a single amino acid. Sequencing a peptide from its spectrum thus corresponds to finding the longest antisymmetric directed path from the source v_0 to the sink v_{2k+1}. A path is *antisymmetric* if it includes at most one of the complementary vertices v_i and v_{2k+1-i}, for all $i = 1, \ldots, k$. An *extended spectrum graph* can be obtained from a spectrum graph by

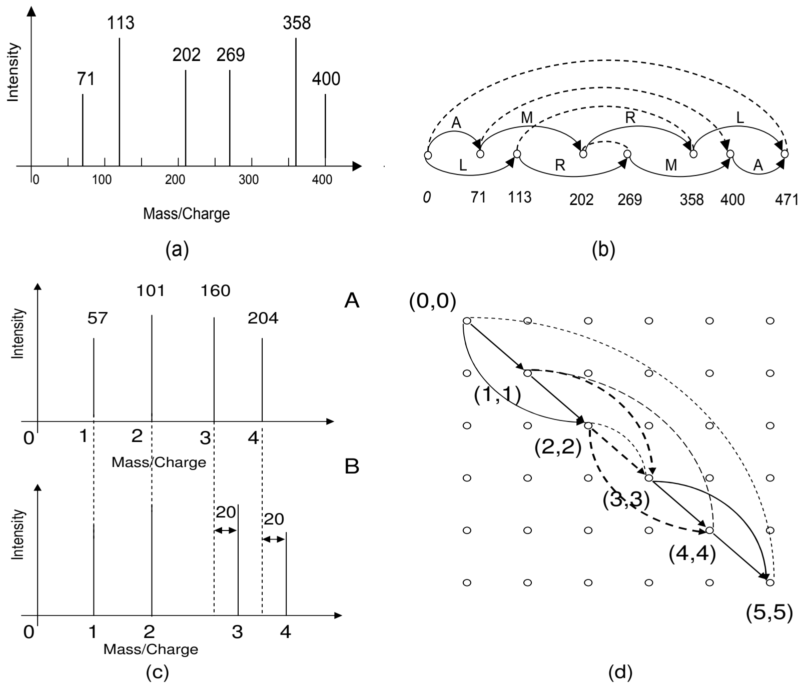


Figure 1. (a) The mass peaks in a tandem mass spectrum with total mass 471. (b) The corresponding extended spectrum graph, where dashed undirected edges connect complementary vertices. (c) The mass peaks in two tandem mass spectra A and B that are to be aligned. The total mass of spectrum A is 261. Mass peaks 3 and 4 in spectrum B have a shift of 20 in their mass values compared to those in spectrum A. (d) The alignment graph constructed based on the mass peaks in A and B. Solid and dashed directed edges represent real and virtual connections respectively; dashed non-directed edges connect complementary vertices; only edges along the diagonal vertices are drawn in the figure.

connecting all pairs of complementary vertices with non-directed edges. Figure 1(a)(b) provide an example for a spectrum and its corresponding extended spectrum graph.

An *alignment graph* $H = (V_a, E_a)$ can be constructed based on two spectra to be aligned. We assume the set of mass peaks for spectra A and B are $S_A = \{x_1, x_2, \cdots, x_{2k_1}\}$ and $S_B = \{y_1, y_2, \cdots, y_{2k_2}\}$ respectively. The set of vertices $V_a = (S_A \times S_B) \cup \{(x_0, y_0), (x_{2k_1+1}, y_{2k_2+1})\}$, where x_0, x_{2k_1+1} and y_0, y_{2k_2+1} are virtual mass peaks with zero and total peptide masses in spectra A and B respectively. (x_i, y_j) is connected to (x_k, y_l) with a *real directed edge* if $x_k > x_i$ and $x_k - x_i = y_l - y_j$. In addition to real directed edges, an *extended alignment graph* may also contain *virtual directed edges* and *non-directed edges.* (x_i, y_j) is connected to (x_k, y_l) with a virtual directed edge if $x_k > x_i$, $y_l > y_j$ and $|(y_l - y_j) - (x_k - x_i)| \leq \Delta_m$, where Δ_m is the maximum mass peak shift due to modified amino acids. Two vertices (x_i, y_j) and (x_k, y_l) are *complementary* if x_i and x_k or y_j and y_l are complementary mass peaks. Complementary vertices are connected with non-directed edges. The *source* and the *sink* in the graph are vertices (x_0, y_0) and (x_{2k_1+1}, y_{2k_2+1}) respectively. The similarity between spectrum A and B can thus be evaluated by finding in H the longest directed antisymmetric path that connects the source and the sink. Figure 1(c)(d) provides an example of two spectra and their extended alignment graph.

Weights on directed edges in the graph are masses. In practice, directed edges in an extended spectrum or alignment graph can be scored based on other experimental parameters. For example, Dancík *et al.*[6] proposed a stochastic edge scoring scheme where each mass peak in the spectrum is generated with a certain probability; the score of a directed edge can be evaluated based on the probabilities of its ends. The sequencing result with the maximum likelihood corresponds to the maximum scored antisymmetric path connecting the source and the sink in the graph.

2.2. *Tree Decomposition and Tree Width*

Definition 2.1. [11] *Let $G = (V, E)$ be a graph, where V is the set of vertices in G, E denotes the set of edges in G (E may contain both directed and non-directed edges). Pair (T, X) is a tree decomposition of graph 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) $X = \{X_i | i \in I, X_i \subseteq V\}$, 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.*

Figure 2(a)(b) shows that tree decomposition provides an alternative view over a graph where vertices are grouped into tree nodes according to

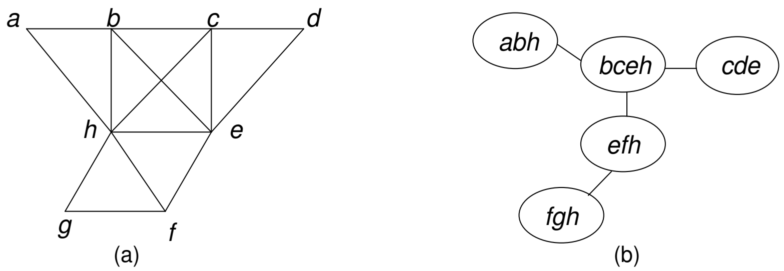


Figure 2. (a) An example of a graph. (b) A tree decomposition for the graph in (a).

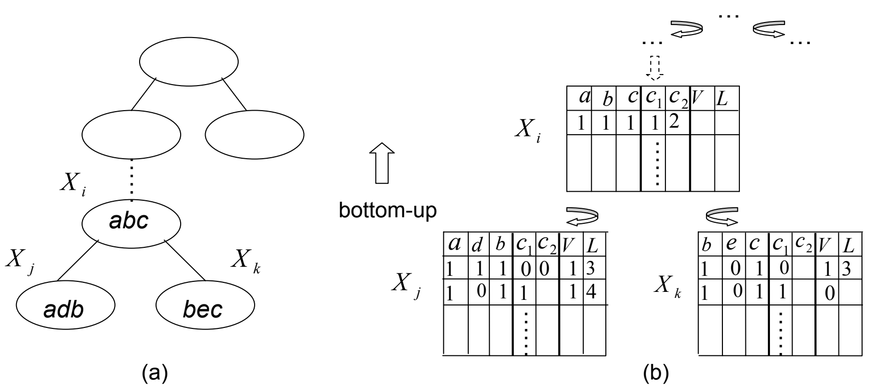
their topological relationships (represented by edges, arcs, etc). Our experiments on simulated and real spectra show that the tree width for extended spectrum graphs and extended alignment graphs are generally around 5. The property of having a small tree width makes it possible for us to develop very efficient algorithms for both problems based on the technique of tree decomposition, since partial optimal solutions on subgraphs induced by subtrees can be efficiently extended and combined with exhaustive enumeration restricted to vertices in a single tree node[1].

2.3. *The Path-finding Algorithm*

The algorithm selects a tree node that contains both the source and the sink as the root of a tree decomposition and maintains a dynamic programming table for each tree node. The algorithm follows a bottom-up fashion to fill the tables for all the tree nodes. The table in the root thus stores the length of the longest antisymmetric path connecting the source and the sink. The algorithm then follows a recursive tracing back procedure to find all the vertices in the path.

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. In addition, $t-1$ 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 largest length of the partial path associated with the 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 $\{0, 1, \cdots, l\}$, where l is the number of children of the tree node. The connection state for a pair of consecutive



a	b	c	c_1	c_2	V	L
1	1	1	1	2		
			⋮			

a	d	b	c_1	c_2	V	L
1	1	1	0	0	1	3
1	0	1	1		1	4
			⋮			

b	e	c	c_1	c_2	V	L
1	0	1	0		1	3
1	0	1	1		0	
			⋮			

Figure 3. A tree decomposition and its corresponding dynamic programming tables. The algorithm follows a bottom-up fashion starting with leave 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) and the largest path length (L) of a given entry in the table for X_i.

selected vertices in the tree node is 0 if they are contiguous in the path and is i ($i > 0$) if the vertices 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 the relative order of selected vertices in a partial path, the algorithm topologically sorts the vertices in each tree node.

To determine an entry in the table for a leaf node, the algorithm exhaustively enumerates and directly computes the validity and largest path length for every possible combination of 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 longest path length for each of its table entry. In particular, for a given entry, the algorithm obtains its selections of vertices and the corresponding connection states and then queries the table contained in each of the child nodes. All valid table entries whose selections of vertices and connection states do not contradict the given entry are queried and the one with the largest path length is selected as the *descendent entry* in the child. The algorithm sets an entry to be invalid if its selection of vertices violates the antisymmetric property or one of the child nodes contains no descendent entries. The largest path length for the entry is then computed by summing up the number of edges covered by the node itself and the largest path length for the descendent entry found in each of its child nodes.

Figure 3 provides an example of computing the table entries for an internal node X_i. Without loss of generality, we assume X_i has two child nodes X_j and X_k, and $X_i = \{a, b, c\}$, $X_j = \{a, d, b\}$ and $X_k = \{b, e, c\}$. To determine the V and L for the entry $\{(1, 1, 1), (1, 2)\}$ in the table for X_i, the algorithm needs to query both of the tables for X_j and X_k since the entry suggests that the vertices on the path between a and b are covered by the subtree rooted at X_j and those between b and c are covered by that rooted at X_k. To query the table for X_j, the algorithm only checks valid entries that select both a and b since $X_i \cap X_j = \{a, b\}$, thus the leading two entries in the table for X_j are checked by the algorithm. Similarly, since $X_i \cap X_k = \{b, c\}$, the algorithm only checks valid entries that select both b and c in the table for X_k.

In the last stage of the computation, the algorithm queries the table in the root node and considers those valid entries that select both the source and the sink and finds the one with the longest path length. The algorithm then follows an up-bottom tracing back procedure to recover the nodes selected to be present on the path. The path found by the algorithm is guaranteed to satisfy the antisymmetric property since, based on the definition of tree decomposition, any pair of complementary vertices is covered by at least one tree node. 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. The algorithm uses a greedy graph reduction technique to obtain a tree decomposition for a graph[4].

3. Experimental Results

We implemented the path-finding algorithm for both *de novo* sequencing and spectral alignment problems. The programs were tested on simulated and real MS/MS spectra. For *de novo* sequencing, we evaluated the performance of the program on simulated spectra that contain different amount of noise, and then analyzed real experimental MS/MS spectra. For spectral alignment, we generated simulated spectra for peptides with PTMs and identified modified amino acids.

3.1. *De Novo Sequencing*

To evaluate the performance of the program on spectra with different amount of noise, we obtained simulated tandem mass spectra for $100,000$ fully tryptic digested peptides of proteins in the Yeast genome. We then filtered out peptides of less than 5 and more than 24 amino acids. In addition to the mass peaks that result from the fragmentation of peptides, we incorporated noisy mass peaks into these simulated spectra and applied the program to obtain the peptide sequences from these noisy spectra. To simulate the noise generally present in real experimental spectra, noisy mass peaks were generated in groups and the differences of mass values for mass peaks in the same group were selected to be those of single amino acids or their combinations. Table 1 shows the performance of the program on spectra with different amount of noise. As we have expected, the tree widths of the spectrum graphs increase when more noisy peaks are inserted into the spectra. The program thus needs more computation time for analyzing a spectrum. In addition, a slight drop in sequencing accuracy is observed when an ideal spectrum is changed into a noisy one.

Table 1. The accuracy of the program on spectra with different amount of noise. N/S is the ratio of the number of noisy peaks to that of others in a spectrum. AC is the percentage of amino acids that are correctly identified by the program; PT(< 5), PT($= 5$) and PT(> 5) are percentages of spectrum graphs whose tree widths are less than 5, equal to 5 and greater than 5 respectively; CT is the average amount of time the program needs to analyze a spectrum.

N/S	AC (%)	PT(< 5) (%)	PT($= 5$) (%)	PT(> 5) (%)	CT(sec)
0.00	98.60	52.45	44.93	2.62	1.54
0.20	98.27	42.48	41.38	16.15	7.24
0.50	98.29	37.69	34.84	27.47	12.37
0.80	97.98	32.98	37.13	29.87	13.10
1.00	96.95	27.64	39.47	32.89	15.46

To evaluate the performance of the program on real experimental spectra, we downloaded 14 tandem mass spectra for peptides in *E. Coli* proteins from the Open Proteomics Database (OPD) and collected 3 experimental FT-ICR data from two different peptide sources. Before we applied the program to a spectrum, the mass peaks in the spectrum were preprocessed. Isotopic mass peaks and mass peaks with intensities less than 0.1 of the maximum intensity value were removed. In addition, a complementary ion was added back for each ion in the spectrum if it was missing. Table 2 shows sequencing results obtained with the program for each spectrum.

The program identified most amino acids correctly with few sequencing errors. However, the reason for these errors is clear. The program is unable to identify I from L and K from Q since the mass of I is equal to that of L, and the mass difference between K and Q is too small to be recognized by the program.

Table 2. The performance of the program on real experimental spectra. TW is the tree width of the spectrum graph; CT is the computation time of the program. Peptides in the first eighteen rows are from OPD. The first six are from EF-Tu protein, followed by seven from glutamate synthase, enolase, sodium-calcium antiporter, HU-2, ferric transport, and S5; the last two are from thioredoxin. The rest three rows are from experimental FT-ICR with the first two from horse myoglobin and the last one from BSA.

Real Sequence	Obtained Sequence	TW	CT(sec)
RAFDQIDNAPEEKA	RAFDQIDNAPEEQA	6	12.11
RPQFYFRT	RPQFYFRT	4	0.42
KVGEEVEIVGIKE	QVGEEVEIVGIKE	6	5.16
KMVVTLIHPIAMDDGLRF	KMVVTILHPIAMDDGIRF	5	7.30
RAGENVGVLLRG	RAGENVGVLLRG	6	13.09
KMVVTLIHPIAMDDGLRF	QMVVTIIHPIAMDDGLRF	5	6.85
KVVRTAIHALARMQHRG	KVVRTAIHAIARMQHRG	6	9.19
KFNQIGSLTETLAAIKM	KFNQIGSLTETLAAIQM	6	10.27
RKFATQYMNLFGIKQ	RKFATQYMNLFGIKK	6	10.15
KTQLIDVIAEKA	QTQIIDVIAEKA	5	5.09
KPVYSNGQAVKD	KPVYSNGQAVQD	5	2.53
KLNIDQNPGTAPKY	KINIDQNPGTAPQY	6	5.80
KNQTLALVSSRP	QNQTLALVSSRP	6	4.85
RVKSQAIEGLVKA	RVKSQAIEGLVQA	6	4.10
HGTVVLTALGGILK	HGTVVLTAIGGILQ	4	0.22
VEADIAGHGQEVLIR	VEADIAGHGQEVLLR	6	10.34
DAFLGSFLYEYSR	DAFLGSFLYEYSR	5	2.18

3.2. *Spectral Alignment*

As an application of spectral alignment, we used the program to identify modified amino acids on peptide sequences with PTMs. We generated pairs of spectra *in silico* for peptides and their modified sequences and perform a spectral alignment between each pair of spectra. The mass modifications can be identified from the longest antisymmetric path found by the program. We introduced two additional parameters, k and Δ, where k is the maximum number of modifications allowed in the peptide and Δ is the maximum amount of mass modification that may occur on a single amino acid. Based on the parameters k and Δ, the number of non-directed edges in the alignment graph can be significantly reduced. In particular, for spectra A and B with sets of mass peaks $S_A = \{x_1, x_2, \cdots, x_{2k_1}\}$ and $S_B = \{y_1, y_2, \cdots, y_{2k_2}\}$ respectively, (x_i, y_j) and $(x_{2k_1+1-i}, y_{j'})$ are connected with an non-directed edge if and only if $|y_{j'} - y_{2k_2+1-j}| \leq k\Delta$. In addition, Δ is used as the Δ_m defined in section 2.1 for adding virtual directed edges to the graph. In our experiment, the values of k and Δ were set to be 3 and 20.0 since, in practice, most of the peptides contain up to 3 modified amino acids. Table 3 shows the results we have obtained on identifying modified amino acids on pairs of spectra we generated *in silico*. The table shows that the tree width of an alignment graph ranges from 4 to 6 and the program is able to identify the modified amino acids accurately in a few seconds.

Table 3. The performance of the program on identifying modified amino acids using spectral alignment. DM is the number of modified amino acids identified by the program; TW is the tree width of the alignment graph; CT is the computation time in seconds. Modified amino acids are superscripted with asterisks.

Peptide	Modified Peptide	DM	TW	CT(sec)
RAIKNLL	RAIK*NLL	1	4	0.04
FKMKRTQVFWKV	FK*MKRTQVFWK*V	2	6	2.43
MALPFQLLRQLGVA	M*ALPFQLLRQLGVA	1	4	0.12
AKYEGGL	AK*YEGGL	1	4	0.07
DFLIKRGV	DFLIK*RGV	1	5	0.73
PKDMILLFATTTTKF	PK*DMILLFATTTTK*F	2	6	2.31
LWEVKDRTAHS	LWEVK*DRTAHS	1	6	3.50
IGALKDKITMS	IGALK*DKITM*S	2	5	0.70
MAIVMGRLEVKAIS	MAIVMGRLEVK*AIS	1	4	0.12
FVPGQKNGIKGDLS	FVPGQK*NGIK*GDLS	2	4	0.04

4. Conclusions

We have extended the notions of MS/MS spectrum graphs and alignment graphs to include relationships among mass peaks such as complementarity and modification. Based on the notion of tree decomposition, such graphs have been exploited for the development of fast optimal algorithms for *de novo* peptide sequencing and spectral alignment. In addition to the efficiency, our work can accurately sequence peptides from noisy spectra and identify post translational modifications of amino acids while allowing the presence of both types of ions. In addition, we expect this approach can be extended to accurately infer partial sequence “tags” from a MS/MS spectrum [14], which can speed up the database search significantly.

Acknowledgement

BY and YX's work is supported in part by the US Department of Energy's Genomes to Life Program under project "Carbon Sequestration in Synechococcus sp: From Molecular Machines to Hierarchical Modeling" (http://www.genomes-to-life.org), by National Science Foundation (#NSF/DBI-0354771 #NSF/ITR-IIS-0407204) and by a "Distinguished Cancer Scholar" grant from Georgia Cancer Coalition.

References

1. S. Arnborg and A. Proskurowski, *Discrete Applied Math.*, 23: 11-24, 1989.
2. C. Bartels, *Biomed. Environ. Mass Spectrom.*, 19: 363-368, 1990.
3. K. Biemann and H.A. Scoble, *Science*, 237:992-998, 1987.
4. H. L. Bodlaender and A. M. C. A. Koster, *Proc. of the 6th Workshop on Alg. Eng. and Exp.*, 70-94, 2004.
5. T. Chen, M. Y. Kao, M. Tepel, J. Rush, and G. M. Church, *Journal of Computational Biology*, 8(3): 325-337, 2001.
6. V. Dancík, T. A. Addona, K. R. Clauser, J. E. Vath, and P. A. Pevzner, *Journal of Computational Biology*, 6(3/4): 327-342, 1999.
7. J. Ferńandez de Cossío, J. Gonzales, and V. Besada, *CABIOS* 11(4): 427-434, 1995.
8. W. M. Hines, A. M. Falick, A. L. Burlingame, and B. W. Gibson, *J. Am. Soc. Mass. Spectrom.*, 3: 326-336, 1992.
9. P. A. Pevzner, V. Dancík, and C. L. Tang, *Proceedings of The Fourth Annual International Conference on Computational Molecular Biology*: 231-236, 2000.
10. P. A. Pevzner, A. Mulyukov, V. Dancik, and C. Tang, *Genome Research*, 11:290-299, 2001.
11. N. Robertson and P. D. Seymour, *Journal of Algorithms*, 7: 309-322, 1986.
12. T. Sakurai, T. Matsuo, H. Matsuda, and I. Katakuse, *Biomed. Mass Spectrom.*, 11(8): 396-399, 1984.
13. M. M. Siegel and N. Bauman, *Biomed. Environ. Mass Spectrom.*, 15: 333-343, 1988.
14. D. Tabb, A. Saraf, and J. R. Yates, *Anal. Chem.*, 75: 6415-6421, 2003.
15. J. A. Taylor and R. S. Johnson, *Rapid Commun. Mass Spectrom.*, 11: 1067-1075, 1997.
16. B. Yan, C. Pan, V. N. Olman, R. L. Hettich, and Y. Xu, *Bioinformatics*, 21(5): 563-574, 2005.
17. J. R. Yates, P. R. Griffin, L. E. Hood, J. X. Zhou, *Techniques in Protein Chemistry II*, 477-485, Academic Press, 1991.
18. D. Zidarov, P. Thibault, M. J. Evans, and M. J. Bentrand, *Biomed. Environ. Mass Spectrom.*, 19: 13-16, 1990.