OPEN ACCESS Freely available online

PLOS ONE

A New Parameterized Algorithm for Rapid Peptide Sequencing

Yinglei Song*

School of Computer Science and Engineering, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu, China

Abstract

De novo sequencing is an important computational approach to determining the amino acid sequence of a peptide with tandem mass spectrometry (MS/MS). Most of the existing approaches use a graph model to describe a spectrum and the sequencing is performed by computing the longest antisymmetric path in the graph. The task is often computationally intensive since a given MS/MS spectrum often contains noisy data, missing mass peaks, or post translational modifications/mutations. This paper develops a new parameterized algorithm that can efficiently compute the longest antisymmetric partial path in an extended spectrum graph that is of bounded path width. Our testing results show that this algorithm can efficiently process experimental spectra and provide sequencing results of high accuracy.

Citation: Song Y (2014) A New Parameterized Algorithm for Rapid Peptide Sequencing. PLoS ONE 9(2): e87476. doi:10.1371/journal.pone.0087476

Editor: Yu Xue, Huazhong University of Science and Technology, China

Received September 9, 2013; **Accepted** December 22, 2013; **Published** February 14, 2014

Copyright: © 2014 Yinglei Song. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Funding: This work is fully supported by the new faculty start-up funding at Jiangsu University of Science and Technology, China. The funding number is 635301202. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The author has declared that no competing interests exist.

* E-mail: yingleisong@gmail.com

Introduction

In proteomics, tandem mass spectrometry (MS/MS) is an important experimental approach to the identification of proteins [2,3,19]. This approach uses enzymes to break molecules of a protein into short peptide sequences. The MS/MS spectrum for each such peptide can be obtained with experiments and the amino acids in these peptides can be determined by analyzing its MS/MS spectrum. The sequencing results of all these peptides can then be combined to obtain the amino acids sequence of the protein. To determine the amino acids sequence of a peptide, we fragment a number of peptides with the same amino acids sequence into charged prefix and suffix subsequences (ions) and measure their mass/charge ratios with a mass spectrometer. Each mass/charge ratio corresponds to a particular ion and forms a peak in the MS/MS spectrum of the peptide. The amino acids sequence of the peptide can be determined by analyzing the relationships among peaks in the spectrum.

In theory, an MS/MS spectrum contains two types of ions, which are b-ions associated with N-terminals of the peptide and y-ions associated with its C-terminals. In addition, we expect fragmentations to occur at all positions along the peptide backbone and the difference of the mass values of two consecutive peaks in the spectrum is the mass of a single amino acids residue. The amino acids sequence of a peptide can thus be determined by analyzing the consecutive peaks in the spectrum. However, the ion types of mass peaks in the spectrum are unknown and cannot be easily determined from the spectrum alone. In addition, some mass peaks are usually missing in a spectrum while some noisy peaks may appear due to multiple fragmentations of the same peptide. Due to these difficulties, the *de novo* sequencing of a peptide, which determines the amino acids sequence of a peptide solely from its MS/MS spectrum, remains challenging and additional research work is needed to make it practical [5,7].

So far, a large number of approaches have been developed for the *de novo* sequencing problem. The first algorithm for *de novo* sequencing is developed in [22]. The algorithm exhaustively enumerates all amino acid sequences of particular lengths and the experimental spectrum is compared with the theoretical spectra of these enumerated ones, the peptide associated with the best match is the sequencing result. The algorithm is not efficient since an exponential number of peptides and their theoretical spectra need to be generated. Later, prefix pruning approaches were developed to avoid exhaustive search. Specifically, prefix pruning approaches only search in these peptides whose prefixes match the experimental spectrum well and thus can significantly reduce the size of the search space [25,29,30]. However, applying heuristic pruning to reduce the search space may adversely affect the sequencing accuracy. Another type of approaches search a spectrum database to find the peptide whose spectrum is the closest to the experimental one. For example, SEQUEST [8] is an extensively used program for sequencing peptides. It searches a peptide spectrum database and evaluates the similarity between the spectrum of an unknown peptide and each spectrum in the database with a particular correlation function. Sequencing tools based on database search can provide sequencing results of high accuracy. However, they can only be applied to sequence those peptides whose spectra have been stored in a spectrum database.

To effectively analyze a spectrum, the concept of spectrum graph is proposed to model the relationships among the mass peaks and the *de novo* sequencing problem has been shown to be equivalent to finding the longest (or maximum scored) antisymmetric path in a spectrum graph [2,7,9,11,13,27]. Although the longest path in a directed acyclic graph can be efficiently computed with a topological sorting algorithm, the algorithm cannot be directly applied to a spectrum graph to compute the path due to the antisymmetric constraint. The constraint requires that vertices that represent complementary ions cannot appear

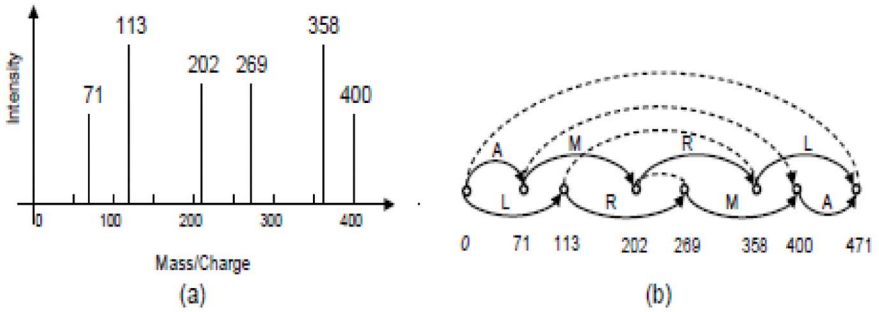

Figure 1. (a) The mass peaks in a tandem mass spectrum; (b) The corresponding extended spectrum graph.
doi:10.1371/journal.pone.0087476.g001

together in the path. In [5], a linear time dynamic programming algorithm is developed to ensure that the computed path satisfies the antisymmetric constraint. However, the algorithm needs quadratic computation time to determine one modified amino acid and more computation time to handle the additional noisy peaks that may appear in the spectrum.

In our previous work [17,18], we introduce a new graph model to describe related mass peaks in a spectrum. The peaks that represent complementary ions are joined with non-directed edges. We study the structure features of such graphs and show that their tree widths are usually small. We also show that, based on a graph tree decomposition of the graph, the longest antisymmetric path can be computed in time $O(6^t n)$, where t is the tree width of the graph and n is the number of peaks in the spectrum. When t is a small integer, the exponential term in the computational complexity is also a small integer and the computational efficiency of the algorithm is thus guaranteed. Testing results have shown that this algorithm can efficiently process both simulated and real spectra and generate sequencing results with high accuracy.

However, an antisymmetric path that connects the source and the sink in an extended spectrum graph may not exist when the peptide contains post translational modifications (PTMs) or some crucial mass peaks are missing in the spectrum. The algorithm developed in our previous work thus cannot be applied to analyze the spectra in these cases. Although a number of approaches have been developed to process the spectra that may contain missing mass peaks or PTMs [12,18,20], most of these approaches need to search a spectrum database and output the peptide whose

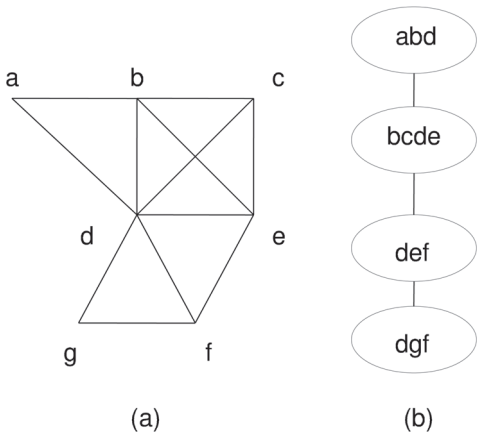

Figure 2. (a) An example of a graph; (b) A path decomposition for the graph in (a).
doi:10.1371/journal.pone.0087476.g002

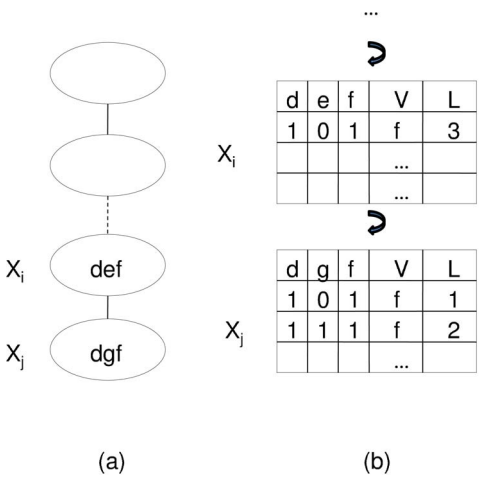

Figure 3. A path decomposition and its corresponding dynamic programming tables.
doi:10.1371/journal.pone.0087476.g003

spectrum has the highest similarity to the spectrum to be sequenced. A spectral alignment is then performed to obtain the sequencing result of the peptide. Spectral alignment is often time consuming and thus may adversely affect the computational efficiency of sequencing. In addition, these approaches cannot be used to sequence peptides whose spectra are not in the database.

In this paper, we develop new techniques that can significantly reduce the computation time needed to process extended spectrum graphs and generate sequencing results of high accuracy. This algorithm can efficiently process a spectrum even when it contains

Table 1. The distribution of path widths (PW) of extended spectrum graphs.

N/S	PW<5	PW = 5	PW>5
0.00	51.32%	42.23%	6.45%
0.20	39.34%	40.72%	19.94%
0.50	32.53%	30.26%	37.21%
0.80	27.45%	30.57%	42.18%
1.00	20.63%	30.31%	49.06%

doi:10.1371/journal.pone.0087476.t001

Table 2. The prediction accuracy of the program on simulated data.

N/S	PDS	TDS	NovoHMM	PepNovo
0.00	98.60%	98.60%	98.32%	98.46%
0.20	98.27%	98.27%	98.25%	98.31%
0.50	98.29%	98.29%	98.13%	98.23%
0.80	97.98%	97.98%	98.08%	98.12%
1.00	96.95%	96.95%	95.32%	97.03%

doi:10.1371/journal.pone.0087476.t002

missing mass peaks or PTMs. It generates the sequencing result by computing the longest antisymmetric partial path in the extended spectrum graph and analyzing this partial path.

The algorithm is based on the concept of path decomposition. We show that, given a path decomposition of an extended spectrum graph, the longest antisymmetric partial path in the graph can be computed in time $O(p^2 2^p n)$, where p is the path width of the path decomposition and n is the number of peaks in the spectrum. Our testing results show that the path width of an extended spectrum graph is only slightly larger than its tree width. This new algorithm thus can be significantly faster than the algorithm we have developed in our previous work. We have implemented this algorithm and compared its performance with that of a few other algorithms for *de novo* sequencing, including the algorithm developed in our previous work [17], PepNovo [10], and NovoHMM [11]. Our testing result shows that this new algorithm is significantly faster and can provide accurate sequencing results for experimental spectra.

Models and Algorithms

2.1 Problem Description

Ideally, ions contained in a spectrum form pairs, each pair contains an ion and its complementary one [28]. Given the MS/MS spectrum S of a peptide P, we use a set of mass peaks $\{p_1,p_2,...,p_{2k}\}$, where $p_i > p_j$ for $i > j$, to denote S. p_i and p_{2k+1-i} are complementary mass peaks and their mass values sum up to the total mass of peptide, which is denoted with M. One of the mass peaks in the same pair is a b-ion and the other one is a y-ion.

In a *spectrum graph* $G=(V_p,E_p)$, each vertex in V_p represents a mass peak in the spectrum and two vertices are joined with a directed edge if the difference of the mass values of their corresponding peaks is the mass of a single amino acid. Specifically, a directed edge from vertex v_i to v_j is created in G if the mass values p_i,p_j that correspond to v_i and v_j satisfy the condition that $p_j - p_i$ is the mass of a single amino acid. Two additional vertices are included in the spectrum graph to represent mass values 0 and M. These two vertices are source vertex v_0 and sink vertex v_{2k+1}. It is not difficult to see that the amino acids sequence of the peptide corresponds to a directed path that starts with the source vertex v_0 and ends with the sink vertex v_{2k+1}. The path must also be *antisymmetric* since all vertices in the path must be ions of the same type.

Given the spectrum graph of a peptide, its sequence of amino acids can thus be determined by computing the longest antisymmetric directed path from the source vertex v_0 to the sink vertex v_{2k+1}. In order to model the complementary relationships among mass peaks in the spectrum, we use a non-directed edge to join each pair of vertices that represent complementary mass peaks in the spectrum. Such a graph is an *extended spectrum graph*. Figure 1(a) and (b) provide an example of a spectrum and its extended spectrum graph.

Directed edges in an extended spectrum graph can be associated with weight values. The weight values can be computed with other experimental parameters. For example, in [7], a stochastic approach is developed to compute the weight values of all directed edges in the graph. The approach associates each mass peak in the spectrum with a certain probability; the weight value of a directed edge can be computed with the probability values of the vertices on its ends. The sequencing result can then be obtained by computing the antisymmetric path with the largest weight value, which corresponds to the path that is most likely to occur.

2.2 Path Decomposition and Path Width

Definition 1. *[21] 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 (P,X) is a path decomposition of graph G if it satisfies the following conditions:*

1. $P=(I,F)$ defines a tree, the sets of vertices and edges in P 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 P, then $X_i \cap X_j \subseteq X_k$.

Table 3. The computation time (secs) of the program on simulated data.

N/S	PDS	TDS	NovoHMM	PepNovo
0.00	0.06	0.78	3.23	0.08
0.20	0.28	3.67	3.79	0.31
0.50	0.53	6.27	7.32	0.67
0.80	0.42	6.64	9.57	0.73
1.00	0.66	7.83	11.46	0.85

doi:10.1371/journal.pone.0087476.t003

Table 4. The distribution of path widths of the extended spectrum graphs of 10000 real spectra.

PW<3	PW = 3	PW = 4	PW = 5	PW>5
5.33%	21.42%	36.79%	22.13%	14.33%

doi:10.1371/journal.pone.0087476.t004

Table 5. The average sequencing accuracy achieved by PDS, PepNovo and NovoHMM on each group.

	PDS	PepNovo	NovoHMM	TDS
Group 1	**96.7%**	90.7%	87.4%	85.3%
Group 2	**95.2%**	88.3%	82.6%	78.4%
Group 3	**90.4%**	86.4%	81.7%	74.2%
Group 4	86.2%	**90.9%**	87.8%	83.1%
Group 5	83.1%	**91.7%**	86.3%	84.6%

doi:10.1371/journal.pone.0087476.t005

Table 6. The average computation time (secs) needed to process the spectra in each group.

	PDS	PepNovo	NovoHMM	TDS
Group 1	0.04	0.06	3.27	0.07
Group 2	0.18	0.19	4.38	0.39
Group 3	1.35	2.23	5.66	4.76
Group 4	3.78	5.65	14.63	32.13
Group 5	12.29	6.75	16.79	127.35

doi:10.1371/journal.pone.0087476.t006

The path width of the path decomposition (P,X) is defined as $\max_i \{|X_i|-1\}$. *The path width of the graph G is the minimum path width over all possible path decompositions of G.*

Figure 2 (a) and (b) show that, path decomposition is a topological decomposition of a graph such that the global topology of the graph is separated from its local topology. Specifically, vertices that are included in the same path node represent the local topology of the graph and different path nodes are connected into a path that represents the global topology of the graph. Path decomposition is in fact a special case of tree decomposition, which has been extensively used to develop efficient algorithms for NP-hard problems in bioinformatics and theoretical computer science [1,4,15,16,17,18,23,24,25,26]. Similar to tree decomposition, path decomposition also provides an excellent framework for dynamic programming since partial optimal solutions on a subgraph induced by a subpath in the path decomposition can be efficiently extended and combined with exhaustive enumeration restricted to vertices in a single path node.

Our previous work has shown that the tree widths of most extended spectrum graphs are around 5. For example, experiments have shown that the tree widths of more than 97% spectrum graphs of ideal spectra generated *in silico* are less than 6 [17]. Based on tree decomposition, a dynamic programming algorithm has been developed to compute the longest antisymmetric path in time $O(6^t n)$, where t is the tree width of the extended spectrum graph and n is the number of mass peaks in the spectrum. The algorithm can thus efficiently process the majority of extended spectrum graphs since the tree widths of the majority of them are at most 6.

In the following sections, based on a path decomposition of a given extended spectrum graph, we develop a dynamic programming algorithm that can compute the longest anitsymmetric partial path between the source and the sink in time $O(p^2 2^p n)$, where p is the path width of the path decomposition. The algorithm is able to cope with missing mass peaks and PTMs and is significantly faster than the one we have developed in our previous work if the path width of an extended spectrum graph is only slightly larger than its tree width.

2.3 The Path-finding Algorithm

Definition 2. *Given an extended spectrum graph G = (V,E), a directed path P in G is an* antisymmetric path *if no two vertices in P are joined with a non-directed edge.*

Given an antisymmetric path that connects the source and the sink of an extended spectrum graph, the sequencing result can be immediately obtained since each edge in the path represents a single amino acid. The sequencing result with the maximum likelihood corresponds to the longest antisymmetric path in a spectrum. However, some crucial mass peaks may not appear in a spectrum due to experimental errors, an antisymmetric path that connects the source and the sink thus may not exist in an extended spectrum graph. In addition, the sequencing result cannot be obtained by computing such an antisymmetric path if the peptide to be sequenced contains PTMs. To cope with these cases, we need the notion of antisymmetric partial path as follows.

Definition 3. *Given an extended spectrum graph* $G=(V,E)$, *a set* $S=\{P_1,P_2,...,P_k\}$ *of directed paths in G is an antisymmetric partial path if the following conditions hold.*

1. Each directed path in S is antisymmetric;

2. All directed paths are mutually disjoint;

3. $\forall P_i,P_j \in S$, $(1 \le i < j \le k)$, *there does not exist vertices u, v, s, t such that* $u,v \in P_i, s,t \in P_j$ *and* $u<s<v$ *or* $s<u<t$.

The length *of S is the sum of the lengths of all its directed paths.*

We next present the details of a dynamic programming algorithm that can compute the longest antisymmetric partial path between the source and the sink in an extended spectrum graph. It is not difficult to see that such a path implies the sequencing result even in cases where some crucial mass peaks are missing in a spectrum or the peptide to be sequenced contains PTMs. We show that such a partial path can be computed based on a path decomposition of an extended spectrum graph $G=(V,E)$ in time $O(p^2 2^p n)$, where p is the path width of the path decomposition.

Without loss of generality, we assume each path node in the path decomposition contains both the source and the sink, since we can add both vertices to a path node if it is not the case. Although including the two vertices in each node may increase the path width by 2, we show later that the computational efficiency of the algorithm is not adversely affected by this. For each node in the path decomposition, we create and maintain a dynamic programming table. Starting with the node on the left end of the path decomposition, the algorithm determines the entries in the table of each node from left to right. Each entry in the table of a path node stores the largest length of the corresponding partial antisymmetric paths that are contained in the subgraph induced by vertices in the node itself and the nodes that are to the left of this node in the path decomposition. The length of the longest antisymmetric path that connects the source and the sink can thus be obtained by querying the table in the node on the right end of the path after the algorithm has filled all tables in the path decomposition. A recursive tracing back procedure can then be followed to determine the vertices in the antisymmetric partial path.

For a path node with p vertices, the dynamic programming table contains $p+2$ columns. Each vertex is associated with one column of the table and it stores the *selection status* of the vertex. A selection status value of 1 indicates that the vertex is included in the antisymmetric partial path and 0 indicates otherwise. One of the 2 remaining columns stores the largest length of the antisymmetric partial paths that are consistent with the selection status of vertices in the entry. We use L to denote this field. The other one stores the furthest vertex that can be reached from the source vertex with the corresponding antisymmetric partial path. We use F to denote this field and denote an entry e in the table with $<S,L,F>$, where S is the set of selection status of vertices in the node, L and F are the values of field L and field F respectively. Given an entry e, we use to $e(S)$,$e(L)$, and $e(F)$ to denote the values of its S,L and F respectively. A table may contain up to $(p+1)2^p$ entries since each vertex may have two different values for its selection status and V can have at most $p+1$ values. p of these$p+1$ values are for the vertices in the path node and the remaining one describes the case where the furthest vertex that can be reached from the source in the partial path is not included in the path node. Figure 3(a) and

(b) provide an example of the dynamic programming tables for a path decomposition of a graph.

The vertices contained in each path node are sorted in ascending order of their corresponding mass values. A vertex u is larger than v if the mass value of u is larger than that of v. We use $u>v$ or $v<u$ to denote this. The order of these vertices also determines the order they appear in the antisymmetric partial path that needs to be computed by the algorithm. The algorithm starts with the table in the left most node of the path decomposition. The possible combinations of the selection status values of all vertices in a leaf node and the values of F can be enumerated. For each such combination, the algorithm checks whether the following properties hold.

1. For each pair of vertices that represent complementary peaks, at most one of them is included in the path;

2. the vertex represented by the value of F is also included in the path;

3. the selected vertices form a directed path that connects the source to the vertex represented by the value of F;

4. none of the out-neighbors of the vertex represented by the value of F are included in the path;

5. the vertices selected in the combination induce a set of disjoint directed paths in G.

If it is the case, the combination is written into the table as an entry. The value of L in the entry can be computed by adding up the lengths of all disjoint directed paths induced by vertices that are selected in the combination.

After the table in the node that is on the left end of the path has been filled, the algorithm starts processing other nodes in the path decomposition. The algorithm processes these nodes from left to right and terminates when the table in the node that is on the right end of the path has been filled. Given an internal node X_i, we use X_j to denote the node that is to the immediate left of X_i in the path decomposition. To fill the table inX_i, the algorithm exhaustively enumerates all possible combinations of the selection status values of the vertices and the values of F in X_i.

For each such combination c, the algorithm checks whether the properties 1, 2, 4, and 5 that have been listed above hold. If all of them hold, the algorithm sets an initial value of 0 for the field L in c and proceeds to query the entries in the table of X_j to compute the values of other fields. For each entry e in the table of X_j, the algorithm checks whether the following properties hold.

1. $e(S)$ is consistent with c on vertices in $X_i \cap X_j$;

2. for any selected vertex $u \in X_i \cap X_j$, the number of vertices that are selected in c and e and joined to u with a directed edge that points to $u$is at most 1, the same holds for the number of vertices that are selected in c and e and joined to u with a directed edge that points from u to another vertex;

3. if $c(F) \in X_i$, $e(F) \in X_i \cap X_j$and $e(F)$ is connected to $c(F)$ with a directed path that goes from $e(F)$ to $c(F)$, each vertex in the path is selected in e or c;

4. if $c(F) \notin X_i$, $e(F) \notin X_i \cap X_j$;

5. there does not exist a vertex $v \in X_i$ such that $v \notin X_j$, v is selected in c and $e(F)$ is larger than v.

If all of the above properties hold, the algorithm proceeds to change the value of $c(L)$ if necessary. We use $L(X_i)$ to denote the sum of the lengths of the disjoint directed paths selected by c in X_i and $L(X_i \cap X_j)$ to denote the sum of the lengths of the disjoint directed paths selected by both c and e in $X_i \cap X_j$, the value of $c(L)$ is compared with $e(L)+L(X_i)-L(X_i \cap X_j)$, if the latter is found to be larger, the value of $c(L)$ is changed to the value of the latter. The above procedure is repeated until all entries in the table of have been processed. If the value of $c(L)$ is not zero, an entry e_c with the computed largest value of $c(L)$ is written into the table of X_i for c. An entry e in the table of X_j is *consistent* with e_c if e satisfies the properties that have been listed above.

After the tables of all nodes in the path decomposition have been filled, the algorithm checks the table in the node that is on the right end of the path and enumerates all entries in the table. From all these entries, the one that has the largest value of $c(L)$ corresponds to the longest antisymmetric partial path in G. The algorithm then follows a trace-back procedure to determine the vertices in this partial path. We next show the correctness of the algorithm.

Proposition 1. *Given two vertices u,v that are both included in an antisymmetric partial path and there is a directed edge from u to v, then the edge from u to v must be included in the path.*

Proof. Since the mass of a single amino acid cannot be the sum of the masses of a few other amino acids. The proposition immediately follows.

Based on Proposition 1, if two vertices are selected by an entry in a dynamic programming table and there is a directed edge between them, the edge must be included in the antisymmetric partial path. The length of an antisymmetric partial path can thus be computed from the vertices that are included in it.

Proposition 2. *Given a path node X_i, we use $t(X_i)$ to denote the dynamic programming table in X_i and $P(X_i)$ to denote the subgraph induced by vertices in X_i and nodes that are to the left of X_i in the path decomposition. For an entry e in $t(X_i)$, the length of the longest antisymmetric partial path that is consistent with e is $e(L)$.*

Proof. We show the proposition by induction. If X_i is the node that is on the left end of the path decomposition, $P(X_i)$ is the graph induced by vertices in X_i only. From the description of the algorithm, we know the value of $e(L)$ is the sum of the lengths of the disjoint paths induced by vertices selected in e. From proposition 1, the only partial path that is consistent with e is the set of disjoint paths induced by vertices selected in e, the value of $e(L)$ is thus the length of this partial path. The proposition thus holds for the node that is on the left end of the path decomposition.

Next, we assume that the proposition holds for $P(X_i)$. We use X_j to denote the node that is immediately to the right of X_i in the path decomposition. We show that the proposition also holds for $P(X_j)$. Given an entry $e \in t(X_j)$, we assume that antisymmetric partial path p is consistent with e and is the longest of all such paths. We need to show that the length of p is equal to $e(L)$.

We use S_e to denote the set of entries that are consistent with e in $t(X_i)$ and $p_w = p \cap P(X_i)$to denote the partial path formed by part of p that is in $P(X_i)$. From the definition, we can see that p_w is consistent with one of the entries in S_e, we use e_w to denote this entry. We claim that p_w is the longest antisymmetric partial path that is consistent with e_w in $P(X_i)$. Since if it is not the case, there exists a different antisymmetric partial path p_a that is consistent with e_w in $P(X_i)$ and is longer than p_w. We can then construct an antisymmetric partial path $p_a \cup (p \cap G(X_i))$. This partial path is consistent with and is longer than p since p_a is longer than p_w. This, however, is contradictory to the fact that p is the longest antisymmetric partial path consistent with e in $P(X_i)$. p_w is thus the longest antisymmetric partial path that is consistent with e_w in $P(X_i)$.

From the assumption of the induction, we know that the length of p_w is equal to the value of $e_w(L)$. From the method the algorithm uses to compute $e(L)$, we immediately obtain the following inequality:

$$e(L) \geq l(p_w) + L(X_i) - L(X_i \cap X_j) = l(p) \qquad (1)$$

where $l(p_w)$,$l(p)$ are the lengths of p_w,p respectively, $L(X_i)$ is the sum of the lengths of the disjoint directed paths selected by e in X_i and $L(X_i \cap X_j)$ is the sum of the lengths of the disjoint directed paths selected by both e_w and e in $X_i \cap X_j$.

On the other hand, we assume that the value of $e(L)$ is computed based on entry $e_f \in S_e$. From the induction assumption, there exists an antisymmetric partial path p_f in $P(X_j)$ that is consistent with e_f and its length is $e_f(L)$. $p_f \cup (p \cap G(X_i))$ is an antisymmetric partial path that is consistent with e in $P(X_i)$. From the assumption on p, we obtain:

$$l(p) \geq l(p_f \cup (p \cap G(X_i)) = e(L) \quad (2)$$

From (1) and (2), we have $l(p) = e(L)$. The proposition thus follows from the principle of induction.

Theorem 3. *Given an extended spectrum graph $G=(V,E)$ and a path decomposition of G, the longest antisymmetric partial path between the source and the sink can be computed in time $O(p^2 2^p n)$, where p is the path width of the path decomposition and n is the number of vertices in G.*

Proof. The correctness of the algorithm immediately follows from Proposition 2. The partial path found by the algorithm is guaranteed to be antisymmetric since, based on the definition of path decomposition, any pair of complementary vertices is covered by at least one path node, any violation of the antisymmetric property can be detected when the algorithm computes the entries in the dynamic programming table of that node. The dynamic programming table in each path node contains at most $(p+1)2^p$ entries. For each possible combination of the selection status values of vertices in a node and the value of field F, we need $O(p)$ time to check its validity. For an internal path node X_i, the algorithm needs to query the table in the node that is immediately to the left of X_i, the aggregated number of such queries that it needs to make to fill the table in X_i is $(p+1)2^p$, and each single query needs $O(p)$ computation time. Therefore, the amount of computation time needed to fill the table in one path node is at most $O(p^2 2^p)$. The total amount of computation time needed by the algorithm is thus at most $O(p^2 2^p n)$.

Although we include both the source vertex and the sink vertex into each path node in the path decomposition, the computational efficiency of the program is not adversely affected. In fact, both vertices must appear in the longest antisymmetric partial path that need to be computed and the selection status value of both vertices must be one for all entries in the dynamic programming table of any node in the path decomposition. The number of combinations the algorithm needs to enumerate and process thus does not increase.

In cases where the source and the sink are connected with an antisymmetric path, the algorithm is able to find the longest antisymmetric path that connects the source and the sink and the sequencing result can thus be obtained. In cases where the peptide to be sequenced contains PTMs or some crucial mass peaks are missing in the spectrum, such an antisymmetric path may not exist. However, the algorithm returns the longest antisymmetric partial path between the source and the sink in these cases and the sequencing result can be obtained from the partial path $S=\{P_1,P_2,...,P_k\}$ with the following steps.

1. Identify all vertices $v_1 < v_2 < v_3 < ... < v_{2k-2}$ in S such that v_{2i-1} $(1 \leq i \leq k)$ is the largest vertex in P_i and v_{2i} is the smallest vertex in P_{i+1}.

2. For each vertex pair (v_{2i-1},v_{2i}), compute the difference of the mass values of the two vertices. We use D_i to denote the difference.

3. We check whether D_i is the sum of the mass values of a few amino acids or not, if it is the case, we include these amino acids in the sequencing result.

4. If it is not the case, we check whether D_i is the sum of the mass values of a few amino acids and modified amino acids or not. If it is the case, we include them in the sequencing result.

5. If there exists a vertex pair that has not been processed, go back to step 2. Otherwise return.

Steps 3 and 4 in the above procedure can be implemented with a dynamic programming approach developed to solve the subset sum problem [6]. This approach needs time $O(D_i l)$, where m is the number of amino acids or modified amino acids whose mass values add up to D_i. Since $l \leq \frac{D_i}{m}$, where m is the minimum mass value of all 20 types of amino acids, both steps 3 and 4 can be completed in time $O(\frac{D_i^2}{m})$.

Experimental Results

We implemented the path-finding algorithm for the *de novo* sequencing problem. The program was tested on both 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.

3.1 On Simulated Data

To evaluate both the values of the tree widths and the path widths of extended spectrum graphs. We generated simulated tandem mass spectra for 100,000 fully tryptic digested peptides of proteins in the Yeast genome. We removed the peptides that contain less than 5 amino acids and more than 24 amino acids from the set of spectra. In order to obtain spectra that are similar to experimental ones, we create additional noisy mass peaks in these simulated spectra. These noisy mass peaks are generated in groups and the differences of mass values from mass peaks in the same group are those of single amino acids or their combinations. Table 1 shows the distribution of path widths of the extended spectrum graphs in the presence of different amount of noise. We use PW to denote the path width and N/S is the ratio of the number of noisy peaks to that of the real peaks in the spectrum. We can see from the table that the values of the path widths of the majority of the extended spectrum graphs are less than 6 for most of the extended spectrum graphs. In addition, the value of the path width increases when more noisy peaks appear in a spectrum.

We then use the program to process each extended spectrum graph and obtain the sequence of amino acids in the peptide for the longest antisymmetric path the program has found in the graph. We evaluate the accuracy of a sequencing result by the percentage of the amino acids that are correctly determined by the program. Tables 2 and 3 compare the sequencing accuracy and computation time of our program (PDS) with that of PepNovo [10], NovoHMM [10], a computer program that solves the *de novo* sequencing problem with a hidden markov model, and our previous work (TDS) [17] which can compute the longest antisymmetric path in an extended spectrum graph based on a tree decomposition of an extended spectrum graph. We can see from the tables that the program based on PDS is slightly faster than PepNovo and is significantly faster than both NovoHMM and TDS, since NovoHMM needs quadratic computation time and TDS needs $O(6^t n)$ time, where t is the tree width of the extended spectrum graph. Although the sequencing accuracy drops slightly when the amount of noise increases, all four programs achieve a sequencing accuracy above 95%. PDS can achieve the same sequencing accuracy as that of TDS since both

programs do the sequencing by computing the longest antisymmetric path in the extended spectrum graph.

3.2 On Experimental Spectra

To evaluate the sequencing accuracy and computational efficiency of the program on real experimental spectra, we obtained 10000 tandem mass spectra from the pep2pro organ-specific proteome map of *Arabidopsis thaliana* in the Proteomics Identification Database (PRIDE) [14]. A preprocessing step is used to process the mass peaks in a spectrum before the sequencing algorithms are used to process a spectrum. We remove Isotopic mass peaks and mass peaks with low intensity value (less than 0.1 of the maximum intensity value) from the spectrum during the preprocessing step. In addition, we check whether each ion has a complementary ion in the spectrum. If it is not the case, we create an ion that is complementary to it and include it in the spectrum.

Table 4 shows the distribution of the path widths of the extended spectrum graphs constructed from the 10000 experimental spectra. From the table, we are able to see that the path widths of more than 85% of the extended spectrum graphs are less than 6. This fact can guarantee the computational efficiency of our algorithm. Based on the values of the path widths of these extended spectrum graphs, we further divide the spectra into five groups. Group 1 contains spectra whose corresponding path widths are less than 3, group 2, 3, and 4 contain spectra whose corresponding path widths are equal to 3, 4, and 5 respectively, group 5 consists of spectra whose corresponding path widths are larger than 5.

For a spectrum, the sequencing accuracy is evaluated by computing the percentage of correctly recognized amino acids in the peptide. Table 4 shows the sequencing accuracy of PDS, PepNovo [10], NovoHMM [11], and TDS [17]. From Table 5, we are able to see that PDS can achieve significantly higher average sequencing accuracy than the other three sequencing tools in groups 1, 2, and 3. Since the extended spectrum graphs in these groups are sparser than those in groups 4 and 5, the probability for them to have missing mass peaks is significantly higher. As we have presented, PDS is able to detect missing mass peaks and thus can significantly improve the sequencing accuracy. On the other hand, PepNovo outperforms PDS, TDS, and NovoHMM in groups 4 and 5. Spectra in groups 4 and 5 contain a large number of mass peaks and the probability for them to have missing mass peaks is much lower. In addition, PepNovo uses a sophisticated stochastic network model to describe the relationships among mass peaks and is able to provide a more accurate description of the spectrum when a sufficient number of mass peaks are present in a spectrum. It is thus able to achieve higher sequencing accuracy than the other three sequencing tools.

Table 6 shows the average computation time needed by all four programs to process the extended spectrum graphs in each group. It can be seen from the Table that both PDS and PepNovo are significantly faster than both NovoHMM and TDS on groups 1, 2, 3, and 4, since the exponential term in the time complexity of our algorithm is a small integer and NovoHMM needs quadratic computation time. However, as we have observed in the Table, the computation time needed by PDS rises sharply when the path widths of extended spectrum graphs increase and is almost comparable to that needed by NovoHMM on spectra in group 5. In addition, TDS is the slowest of the four tools in group 5 since the exponential factor in the computation time of TDS is a large integer for spectra in this group. This fact suggests that when the path width of an extended spectrum graph is large, PepNovo should be chosen over the other three programs as the sequencing tool since it is computationally more efficient and also more accurate.

Conclusions

In this paper, we use the notion of extended spectrum graphs to model the relationships among mass peaks in an MS/MS spectrum. Based on graph path decomposition, we study the structural features of extended spectrum graphs and such features are exploited for the development of fast optimal algorithms for *de novo* peptide sequencing. We develop a dynamic programming algorithm that can efficiently compute the longest antisymmetric partial path in an extended spectrum graph that is of small path width. The sequencing result can be immediately obtained from the path returned by the algorithm. Our testing results show that this new algorithm is more accurate than NovoHMM and PepNovo on most of the tested experimental spectra and is also significantly faster than NovoHMM when the path width of the extended spectrum graph is less than 5.

So far, edges in an extended spectrum graph are all equally weighted and an edge-scoring scheme has not been introduced to model the probability [6] for an edge to be present in the longest antisymmetric path. The development of such a scoring scheme for edges in extended spectrum graphs constitutes an important aspect of our future work. In addition, the preprocessing stage of this approach needs to be refined to further improve the sequencing accuracy and the computational efficiency of this approach.

Author Contributions

Conceived and designed the experiments: YS. Analyzed the data: YS. Contributed reagents/materials/analysis tools: YS. Wrote the paper: YS.

References

1. Arnborg S, Proskurowski A (1989) Linear time algorithms for NP-hard problems restricted to partial *k*-trees. Discrete Applied Math 23: 11–24.
2. Bartels C (1990) Fast algorithm for peptide sequencing by mass spectrometry. Biomed. Environ. Mass Spectrom 19: 363–368.
3. Biemann K, Scoble HA (1987) Characterization of tandem mass spectrometry of structural modifications in proteins Science, 237: 992–998.
4. Bodlaender HL, Koster AMCA (2004) Safe separators for treewidth. Proc. of the 6th Workshop on Alg. Eng and Exp.70–94.
5. Chen T, Kao MY, Tepel M, Rush J, Church GM (2001) A Dynamic programming approach to de novo peptide sequencing via tandem mass spectrometry Journal of Computational Biology 8(3): 325–337.
6. Cormen TH, Leiserson CE, Rivest RL, Stein C (2001) Introduction to Algorithms, Second Edition, MIT Press
7. Danc'ık V, Addona TA, Clauser KR, Vath JE, Pevzner PA (1999) *De Novo* Peptide sequencing via tandem mass spectrometry Journal of Computational Biology 6(3/4): 327–342.
8. Eng JK, McCormack AL, Yates JR (1994) An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. J. Am. Soc. Mass. Spectrom 5: 976–989.
9. Fer'nandez DCJ, Gonzales J, Besada V (1995) A computer program to aid the sequencing of peptides in collision-activated decomposition experiments CABIOS 11(4): 427–434.
10. Fischer B, Roth V, Roos F, Grossmann J, Baginsky S, et al. (2005) NovoHMM: A Hidden Markov Model for the de novo peptide sequencing 77(22): 7265–7273.
11. Frank A, Pevzner P (2005) PepNovo: De novo peptide sequencing via probabilistic network modeling Analytical Chemistry 77(4): 964–973.
12. Frank A, Tanner S, Bafna V, Pevzner P (2005) Peptide sequence tags for database search in mass-spectrometry Journal of Proteome Research 4(4): 1287–1295.
13. Hines WM, Falick AM, Burlingame AL, Gibson BW (1992) Pattern-based algorithm for peptide sequencing from tandem high energy collision-induced dissociation mass spectra, J. Am. Soc. Mass. Spectrom 3: 326–336.

14. Jones P, Cote RC, Martens L, Quinn AF, Taylor CF, et al. (2005) PRIDE: a public repository of protein and peptide identifications for the proteomics community Nucleic Acids Research 34(1): D659–D663.
15. Liu C, Song Y (2009) Parameterized dominating set problem in chordal graphs: complexity and lower bound Journal of Combinatorial Optimization 18: 87–97.
16. Liu C, Song Y (2011) Parameterized complexity and inapproximability of dominating set problem in chordal and near chordal graphs Journal of Combinatorial Optimization 22(4): 684–698.
17. Liu C, Song Y, Yan B, Xu Y, Cai L (2006) Fast *de novo* peptide sequencing and spectral alignment via tree decomposition Pacific Symposium on Biocomputing pp. 255–266.
18. Liu C, Yan B, Song Y, Xu Y, Cai L (2006) Peptide sequence tag-based blind identification of post-translational modifications with point process model Bioinformatics 22(14): e307–e313.
19. Pevzner PA, Danc'ik V, Tang CL (2000) Mutation-tolerant protein identification by mass spectrometry Proceedings of The Fourth Annual International Conference on Computational Molecular Biology: 231–236.
20. Pevzner PA, Mulyukov A, Dancik V, Tang C (2001) Efficiency of database search for identification of mutated and modified proteins via mass spectrometry Genome Research 11: 290–299.
21. Robertson N, Seymour PD (1986) Graph Minors II. Algorithmic aspects of tree-width Journal of Algorithms 7: 309–322.
22. Sakurai T, Matsuo T, Matsuda H, Katakuse I (1984) Paas 3: a computer program to determine probable sequence of peptides from mass spectrometric data. Biomed. Mass Spectrom 11(8): 396–399.
23. Song Y, Liu C, Malmberg R, Pan F, Cai L (2005) Tree decomposition based fast search of RNA structures including pseudoknots Proceedings of 2005 IEEE Computational Systems Bioinformatics Conference, Stanford, California pp. 223–234.
24. Song Y, Liu C, Huang X, Malmberg RL, Xu Y, et al. (2006) Efficient parameterized algorithms for biopolymer structure-sequence alignment IEEE/ACM Transactions on Computational Biology and Bioinformatics 3(4): 423–432.
25. Song Y, Liu C, Malmberg RL, Cai L (2006) Phylogenetic network inference through efficient haplotyping Proceedings of 2006 International Workshop on Algorithms in Bioinformatics, Zurich, Switzerland pp. 68–79.
26. Siegel MM, Bauman N (1988) An efficient algorithm for sequencing peptides using fast atom bombardment mass spectral data. Biomed. Environ. Mass Spectrom pp. 15: 333–343.
27. Taylor JA, Johnson RS (1997) Sequence database searches via *de novo* peptide sequencing by tandem mass spectrometry. Rapid Commun. Mass Spectrom, 11: 1067–1075.
28. Yan B, Pan C, Olman VN, Hettich RL, Xu Y (2005) A graph-theoretic approach for the separation of b and y ions in tandem mass spectrometry Bioinformatics 21(5): 563–574.
29. Yates JR, Griffin PR, Hood LE, Zhou JX (1991) Computer aided interpretation of low energy ms/ms mass spectra of peptides Techniques in Protein Chemistry II, Academic Press pp. 477–485.
30. Zidarov D, Thibault P, Evans MJ, Bentrand MJ (1990) Determination of primary structure of peptides using fast atom bombardment mass spectrometry. Biomed. Environ. Mass Spectrom pp. 19: 13–16.