A Dynamic Programming Approach to De Novo Peptide Sequencing via Tandem Mass Spectrometry

Ting Chen *
Department of Genetics
Harvard Medical School
Boston, MA 02115, USA

Ming-Yang Kao
Department of Computer Science
Yale University
New Haven, CT 06520, USA

Matthew Tepel John Rush George M. Church †
Department of Genetics
Harvard Medical School
Boston, MA 02115, USA

*Current address: Department of Mathematics, University of Southern California, Los Angeles, CA 90089 USA. Email: tingchen@hto.usc.edu.

†To whom the correspondence should be addressed: church@arep.med.harvard.edu.

Abstract

Tandem mass spectrometry fragments a large number of molecules of the same peptide sequence into charged molecules of prefix and suffix peptide subsequences, and then measures mass/charge ratios of these ions. The *de novo peptide sequencing* problem is to reconstruct the peptide sequence from a given tandem mass spectral data of k ions. By implicitly transforming the spectral data into an *NC-spectrum graph* $G = (V, E)$ where $|V| = 2k+2$, we can solve this problem in $O(|V||E|)$ time and $O(|V|^2)$ space using dynamic programming. For an ideal noise-free spectrum with only b- and y-ions, we improve the algorithm to $O(|V| + |E|)$ time and $O(|V|)$ space. Our approach can be further used to discover a modified amino acid in $O(|V||E|)$ time. The algorithms have been implemented and tested on experimental data.

1 Introduction

The determination of the amino acid sequence of a protein is an important step toward quantifying this protein and solving its structure and function. Conventional sequencing methods (Wilkins *et al.*, 1997) cleave proteins into peptides and then sequence the peptides individually using Edman degradation or ladder sequencing by mass spectrometry or tandem mass spectrometry (McLafferty *et al.*, 1999). Among such methods, tandem mass spectrometry combined with high-performance liquid chromatography(HPLC) has been widely used as follows. A large number of molecules of the same but unknown peptide sequence are separated using HPLCs and a mass analyzer such as a Finnigan LCQ ESI-MS/MS mass spectrometer. They are ionized and fragmented by collision-induced dissociation. All the resulting ions are measured by the mass spectrometer for mass/charge ratios. In the process of collision-induced dissociation, a peptide bond at a random position is broken, and each molecule is fragmented into two *complementary* ions, typically an N-terminal ion called *b-ion* and a C-terminal ion called *y-ion*.

Figure 1 shows the fragmentation of a doubly charged peptide sequence of n amino acids ($\mathtt{NHHCHR_1CO} - \cdots - \mathtt{NHCHR_iCO} - \cdots - \mathtt{NHCHR_nCOOH}$). The ith peptide bond is broken and the peptide is fragmented into an N-terminal ion which corresponds to a charged prefix subsequence ($\mathtt{NHHCHR_1CO} - \cdots - \mathtt{NHCHR_iCO^+}$), and a C-terminal ion which corresponds to a charged suffix subsequence ($\mathtt{NHHCHR_{i+1}CO} - \cdots - \mathtt{NHCHR_n^+COOH}$). These two ions are *complementary* because joining them determines the original peptide sequence. This dissociation process fragments a large number of molecules of the same peptide sequence, and ideally, the resulting ions contain all possible prefix subsequences and suffix subsequences. Table 1 shows all the resulting b-ions and y-ions from the dissociation of a peptide ($\mathtt{R_1} - \mathtt{R_2} - \mathtt{R_3}$). These ions display a spectrum in the mass spectrometer, and each appears at the position of its mass because it carries a $+1$ charge. All the prefix (or suffix) subsequences form a sequence ladder where two adjacent sequences differ by one amino acid, and indeed, in the tandem mass spectrum, the mass difference between two adjacent b-ions (or y-ions) equals the mass of that amino acid. Figure 2 shows a hypothetical tandem mass spectrum of all the ions (including the parent ions) of a peptide SWR, and the ladders formed by the b-ions and the y-ions.

We define an *ideal* tandem mass spectrum to be noise-free and contain only b- and y-ions, and every mass peak has the same height (or abundance). The interpretation of an ideal spectrum only deals with the following two factors: (1) it is unknown whether a mass peak (of some ion) corresponds to a prefix or a suffix subsequence; (2) some ions may be lost in the experiments and the corresponding mass peaks disappear in the spectrum. The *ideal de novo peptide sequencing problem* takes an input of a subset of prefix and suffix masses of an unknown target peptide sequence P and asks for a peptide sequence Q such that a subset of its prefixes and suffixes gives the same input masses. Note that as expected, Q may or may not be the same as P, depending on the input data and the quality.

In practice, noise and other factors can affect a tandem mass spectrum. An ion may display two or three different mass peaks because of the distribution of two isotopic carbons, $\mathtt{C^{12}}$ and $\mathtt{C^{13}}$, in the molecules. An ion may lose a water or an ammonia molecule and displays a different mass peak from its normal one. The fragmentation may result in some other ion types such as a- and z-ions. Every mass peak displays a height that is proportional to the number of molecules of such an ion type. Therefore, the *de novo peptide sequencing problem* is that given a defined correlation function, asks to find a peptide sequence whose hypothetical prefix and

suffix masses are optimally correlated to a tandem mass spectrum.

A special case of the peptide sequencing problem is the amino acid modification. An amino acid at an unknown location on the target peptide sequence is modified and its mass is changed. This modification appears in every molecule of this peptide, and all the ions containing the modified amino acid display different mass peaks from the unmodified ions. Finding this modified amino acid is of great interest in biology because modifications are usually associated with protein functions.

Several computer programs such as SEQUEST (Eng *et al.*, 1994), Mascot (Perkins *et al.*, 1999), and ProteinProspector(Clauser *et al.*, 1999), have been designed to interpret the tandem mass spectral data. A typical program like SEQUEST correlates peptide sequences in a protein database with the tandem mass spectrum. Peptide sequences in a database of over 300,000 proteins are converted into hypothetical tandem mass spectra, which are matched against the target spectrum using some correlation functions. The sequences with top correlation scores are reported. This approach gives an accurate identification, but cannot handle the peptides that are not in the database. Pruning techniques have been applied in some program to screen the peptides before matching the database but at the cost of reduced accuracy.

An alternative approach (Dancik *et al.*, 1999 and Taylor and Johnson, 1997) is *de novo peptide sequencing*. Some candidate peptide sequences are extracted from the spectral data before they are validated in the database. First, the spectral data is transformed to a directed acyclic graph, called a *spectrum graph*, where (1) a node corresponds to a mass peak and an edge, labeled by some amino acids, connects two nodes that differ by the total mass of the amino acids in the label; (2) a mass peak is transformed into several nodes in the graph, and each node represents a possible prefix subsequence (ion) for the peak. Then, an algorithm is called to find the highest-scoring path in the graph or all paths with scores higher than some threshold. The concatenation of edge labels in a path gives one or multiple candidate peptide sequences. However, the well-known algorithms (Cormen *et al.*, 1990) for finding the longest path tend to include multiple nodes associated with the same mass peak. This interprets a mass peak with multiple ions of a peptide sequence, which is rare in practice. This paper provides efficient sequencing algorithms for a general interpretation of the data by restricting a path to contain at most one node for each mass peak.

For this purpose, we introduce the notion of an *NC-spectrum graph* $G = (V, E)$ for a given tandem mass spectrum, where $V = 2k+2$ and k is the number of mass peaks in the spectrum. In conjunction with this graph, we develop a dynamic programming approach to obtain the following results for previously open problems:

- The de novo peptide sequencing problem can be solved in $O(|V||E|)$ time and $O(|V|^2)$ space, and in $O(|V| + |E|)$ time and $O(|V|)$ space if the given spectrum is ideal.
- A modified amino acid can be found in $O(|V||E|)$ time.

Our paper is organized as follows. Section 2 formally defines the NC-spectrum graph and the peptide sequencing problem. Section 3 describes the dynamic programming algorithms for the peptide sequencing problem for three kinds of spectra: ideal spectra, noisy spectra and spectra with a modified amino acid. Section 4 reports the implementation and testing of our algorithms on experimental data. Section 5 mentions further research.

2 Spectrum graphs and the peptide sequencing problem

An amino acid unit in a peptide is called a *residue*. In forming the peptide bonds, an ionized amino acid molecule loses an Oxygen and two Hydrogens, so the mass of a residue is approximately 18 Daltons less than the mass of an ionized amino acid molecule. The structures of both molecules are shown in Figure 3. In this paper, we use the amino acid mass referring to the residue mass.

Given the mass W of a target peptide sequence P, k ions $I_1, \ldots, I_k$ of P, and the masses $w_1, \ldots, w_k$ of these ions, we create an *NC-spectrum graph* $G = (V, E)$ as follows.

For each I_j, it is unknown whether it is an N-terminal ion or a C-terminal ion. If I_j is a C-terminal ion, it has a complementary N-terminal ion, denoted as I_j^c, with a mass of $W-(w_j-2)$, where the 2-Dalton mass is from the two extra Hydrogens of the y-ion shown in Figure 1. Therefore, we create a pair of nodes N_j and C_j to represent I_j and I_j^c, one of which must be an N-terminal ion. We also create two auxiliary nodes N_0 and C_0 to represent the zero mass and the total mass of all amino acids of P respectively. Let $V = \{N_0, N_1, ..., N_k, C_0, C_1, ..., C_k\}$. Each node $x \in V$, is placed at a real line, and its coordinate $\mathrm{cord}(x)$ is the total mass of its amino acids, i.e.,

$$\mathrm{cord}(x) = \begin{cases} 0 & x = N_0; \\ W - 18 & x = C_0; \\ w_j - 1 & x = N_j \quad \text{for } j = 1, \ldots, k; \\ W - w_j + 1 & x = C_j \quad \text{for } j = 1, \ldots, k. \end{cases}$$

This coordinate scheme is adopted for the following reasons. An N-terminal b-ion has an extra Hydrogen (approximately 1 Dalton), so $\mathrm{cord}(N_j) = w_j - 1$ and $\mathrm{cord}(C_j) = (W-(w_j-2))-1 = W-w_j+1$; and the full peptide sequence of P has two extra Hydrogens and one extra Oxygen (approximately 16 Daltons), so $\mathrm{cord}(C_0) = W - 18$. If $\mathrm{cord}(N_i) = \mathrm{cord}(C_j)$ for some i and j, I_i and I_j are complementary: one of them corresponds to a prefix sequence and another corresponds to the complementary suffix sequence. In the spectrum graph, they are merged into one pair of nodes. We say that N_j and C_j are *derived* from I_j. For convenience, for x and $y \in V$, if $\mathrm{cord}(x) < \mathrm{cord}(y)$, then we say $x < y$.

The edges of G are specified as follows. For x and $y \in V$, there is a directed edge from x to y, denoted by (x, y) and $E(x, y) = 1$, if the following conditions are satisfied: (1) x and y are not derived from the same I_j; (2) $x < y$; and (3) $\mathrm{cord}(y) - \mathrm{cord}(x)$ equals the total mass of some amino acids. Figure 4 shows a tandem mass spectrum and its corresponding NC-spectrum graph. In Figure 4, the path $N_0 - C_2 - N_1 - C_0$ that contains exactly one of every pair of complementary nodes derived from the same ion corresponds to the original peptide sequence `SWR`.

Since G is a directed graph along a line and all edges point to the right on the real line, we list the nodes from left to right according to their coordinates as $x_0, x_1, \ldots, x_k, y_k, \ldots, y_1, y_0$, where x_i and y_i, $1 \le i \le k$, are complementary. In practice, a tandem mass spectrum may contain noise such as mass peaks of other types of ions from the same peptide, mass peaks of ions from other peptides, and mass peaks of unknown ions. A general way to deal with these situations is to use a pre-defined edge (and node) scoring function $s(\cdot)$ such that nodes corresponding to high peaks and edges labeled with single amino acid receive higher scores. We define the score of a path to be the sum of the scores of the edges (and the nodes) on the path. Therefore,

Definition 1 The peptide sequencing problem *is that given an NC-spectrum graph* $G = (V, E)$ *and an edge scoring function* $s(\cdot)$*, asks for a maximum score path from* x_0 *to* y_0*, such that at most one of* x_j *and* y_j *for every* $1 \leq j \leq k$ *is on the path.*

If the peptide sequence is known, we can identify the nodes of G corresponding to the prefix subsequences of this peptide. These nodes form a directed path from x_0 to y_0. Generally the mass of a prefix subsequence does not equal the mass of any suffix subsequence, so the path contains at most one of x_j and y_j for each $j > 0$. On the other hand, a satisfying directed path from x_0 to y_0 contains observed prefix subsequences. If each edge on the path is labeled with some amino acids, we can visit the edges on the path from left to right, and concatenate these amino acids to form one or multiple peptide sequences that display the tandem mass spectrum. If an appropriate scoring function is given, finding the maximum score path is equivalent to finding a peptide sequence that is optimally correlated to the spectrum.

Even if the mass of a prefix subsequence coincidently equals the mass of a suffix subsequence, which means the directed path contains both x_j and y_j, we can remove either x_j or y_j from the path and form a new path corresponding to multiple peptide sequences which contain the real sequence. We call such a directed path *a feasible reconstruction* of P or *a feasible solution* of G.

To construct the edges of G, we use a mass array $\mathcal{A}$, which takes an input of mass m, and returns 1 if m equals the total mass of some amino acids; and 0 otherwise. Let h be the maximum mass under construction. Let δ be the measurement precision for mass. Then,

Theorem 1 *Assume that we are given the maximum mass* h *and the mass precision* δ*.*

1. *The mass array* $\mathcal{A}$ *can be constructed in* $O(\frac{h}{\delta})$ *time.*
2. *Given a spectrum of* k *mass peaks,* G *can be constructed in* $O(k^2)$ *time.*

Proof. These statements are proved as follows.

Statement 1. Given a mass m, $0 < m \leq h$, $\mathcal{A}[m] = 1$ if and only if m equals one amino acid mass, or there exists an amino acid mass $r < m$ such that $\mathcal{A}[m-r] = 1$. If $\mathcal{A}$ is computed in the order from $\mathcal{A}[0]$ to $\mathcal{A}[\frac{h}{\delta}]$, each entry can be determined in constant time since there are only 20 amino acids and all the previous entries have been determined. The total time is $O(\frac{h}{\delta})$.

Statement 2. For any two nodes v_i and v_j of G, we create an edge for v_i and v_j, $E(v_i, v_j) = 1$, if and only if $0 < \text{cord}(v_j) - \text{cord}(v_i) < h$ and $\mathcal{A}[\text{cord}(v_j) - \text{cord}(v_i)] = 1$. There are totally $O(k^2)$ pairs of nodes. With $\mathcal{A}$, G can be constructed in $O(k^2)$ time. □

In current practice, $\delta = 0.2$ Dalton, and $h = 400$ Daltons, roughly the total mass of four amino acids. The efficiency of our algorithm will allow biologists to consider much larger h and much smaller δ.

3 Algorithms for peptide sequencing

An ideal tandem mass spectrum is noise-free and contains only b- and y-ions, and every mass peak has the same height. This section starts with algorithms for ideal spectra in Section 3.1 and Section 3.2, and then describes algorithms for noisy spectra in Section 3.3 and spectra with a modified amino acid in Section 3.4.

3.1 Algorithm for ideal peptide sequencing

Given an ideal spectrum, we want to find a peptide sequence such that every mass peak of the spectrum matches with some b- or y-ion of the peptide. Therefore,

Definition 2 The ideal peptide sequencing problem *is equivalent to the problem which, given* $G = (V, E)$, *asks for a directed path from* x_0 *to* y_0 *which contains exactly one of* x_j *and* y_j *for each* $j > 0$.

We list the nodes of G from left to right as $x_0, x_1, \ldots, x_k, y_k, \ldots, y_1, y_0$. Let $M(i, j)$ be a two-dimension matrix with $0 \leq i, j \leq k$. Let $M(i, j) = 1$ if and only if in G, there is a path L from x_0 to x_i and a path R from y_j to y_0, such that $L \cup R$ contains exactly one of x_p and y_p for every $p \in [1, i] \cup [1, j]$. Denote the two paths $L \cup R$ as the LR paths for $M(i, j) = 1$. Let $M(i, j) = 0$ otherwise. Table 2 shows the matrix M for the NC-spectrum graph in Figure 4.

Algorithm Compute-M(G)
1. Initialize $M(0, 0) = 1$ and $M(i, j) = 0$ for all $i \neq 0$ or $j \neq 0$;
2. Compute $M(1, 0)$ and $M(0, 1)$;
3. For $j = 2$ to k
4. For $i = 0$ to $j - 2$
   (a) if $M(i, j-1) = 1$ and $E(x_i, x_j) = 1$, then $M(j, j-1) = 1$;
   (b) if $M(i, j-1) = 1$ and $E(y_j, y_{j-1}) = 1$, then $M(i, j) = 1$;
   (c) if $M(j-1, i) = 1$ and $E(x_{j-1}, x_j) = 1$, then $M(j, i) = 1$;
   (d) if $M(j-1, i) = 1$ and $E(y_j, y_i) = 1$, then $M(j-1, j) = 1$.

Theorem 2 *The following statements hold.*

1. *Given* $G = (V, E)$, Algorithm Compute-M *computes the matrix* M *in* $O(|V|^2)$ *time.*
2. *Given* $G = (V, E)$ *and* M, *a feasible solution of* G *can be found in* $O(|V|)$ *time.*
3. *Given* $G = (V, E)$, *a feasible solution of* G *can be found in* $O(|V|^2)$ *time and* $O(|V|^2)$ *space.*
4. *Given* $G = (V, E)$, *all feasible solutions of* G *can be found in* $O(|V|^2 + n|V|)$ *time and* $O(|V|^2 + n|V|)$ *space, where* n *is the number of solutions.*

Proof. These statements are proved as follows.

Statement 1. Without loss of generality, assume that $i < j$ and $M(i, j) = 1$. By definition, either x_{j-1} or y_{j-1} (but not both) must be on the LR paths for $M(i, j) = 1$, and there exists a node y_p such that $E(y_j, y_p) = 1$ and $M(i, p) = 1$. Thus either $i = j - 1$ or $p = j - 1$, corresponding to Steps 4(b) and 4(d) respectively in the algorithm. A similar analysis holds for $M(j, i) = 1$ and $i < j$ in Steps 4(a) and 4(c). Therefore, every entry in M is correctly computed in the algorithm. Note that $|V| = 2k + 2$ and Steps 4(a), 4(b), 4(c), and 4(d) take $O(1)$ time, and thus the total time is $O(|V|^2)$.

Statement 2. Note that $|V| = 2k + 2$. Without loss of generality, assume that a feasible solution S contains node x_k. Then there exists some $j < k$, such that edge $(x_k, y_j) \in S$ and $M(k, j) = 1$. Therefore, we search the non-zero entries in the last row of M and find a j that satisfies both $M(k, j) = 1$ and $E(x_k, y_j) = 1$. This takes $O(|V|)$ time. With $M(k, j) = 1$,

we backtrack M to search the next edge of S as follows. If $j = k-1$, the search starts from $i = k-2$ to 0 until both $E(x_i, x_k) = 1$ and $M(i,j) = 1$ are satisfied; otherwise $j < k-1$, and then $E(x_{k-1}, x_k) = 1$ and $M(k-1, j) = 1$. We repeat this process to find every edge of S. Similar process holds for feasible solutions that contain node y_k. Using a common data structure such as link lists or a two-dimension matrix, this algorithm visits every node of G at most once in the order from x_k to x_0 and from y_k to y_0 at a total cost of $O(|V|)$ time.

Statement 3. We compute M by means of Statement 1 and find a feasible solution by means of Statement 2. The total cost is $O(|V|^2)$ time and $O(|V|^2)$ space.

Statement 4. The proof is similar to that of Statement 2. For feasible solutions that contain node x_k, we search every j that satisfies both $M(k,j) = 1$ and $E(x_k, y_j) = 1$, and each j corresponds to different feasible solutions. For every $M(k,j) = 1$, we backtrack M to search the next edges as follows. If $j = k-1$, the search starts from $i = k-2$ to 0 to find every i that satisfies both $E(x_i, x_k) = 1$ and $M(i,j) = 1$; otherwise $j < k-1$, and then $E(x_{k-1}, x_k) = 1$ and $M(k-1, j) = 1$. Every edge found in this process corresponds to different feasible solutions. We repeat this process to find all feasible solutions that contain node x_k. Similar process holds for feasible solutions that contain node y_k. Finding one feasible solution costs $O(|V|)$ time and $O(|V|)$ space because the algorithm visits every node of G at most once for each solution. Computing M and finding n solutions cost $O(|V|^2 + n|V|)$ time and $O(|V|^2 + n|V|)$ space in total. □

3.2 An improved algorithm for ideal peptide sequencing

To improve the time and space complexities in Theorem 2, we encode M into two linear arrays. Define an edge (x_i, y_j) with $0 \le i, j \le k$ to be a *cross edge*, and an edge (x_i, x_j) or (y_j, y_i) with $0 \le i < j \le k$ to be an *inside edge*. Let $\mathrm{lce}(z)$ be the length of the longest consecutive inside edges starting from node z; i.e.,

$$\begin{cases} \mathrm{lce}(x_i) = j - i & \text{if } E(x_i, x_{i+1}) = \ldots = E(x_{j-1}, x_j) = 1 \text{ and } (j = k \text{ or } E(x_j, x_{j+1}) = 0); \\ \mathrm{lce}(y_j) = j - i & \text{if } E(y_j, y_{j-1}) = \ldots = E(y_{i+1}, y_i) = 1 \text{ and } (i = 0 \text{ or } E(y_i, y_{i-1}) = 0). \end{cases}$$

Let $\mathrm{dia}(z)$ be two diagonals in M, where

$$\begin{cases} \mathrm{dia}(x_j) = M(j, j-1) & \text{for } 0 < j \le k; \\ \mathrm{dia}(y_j) = M(j-1, j) & \text{for } 0 < j \le k; \\ \mathrm{dia}(x_0) = \mathrm{dia}(y_0) = 1. \end{cases}$$

Lemma 3 *Given* $\mathrm{lce}(\cdot)$ *and* $\mathrm{dia}(\cdot)$*, any entry of* M *can be computed in* $O(1)$ *time..*

Proof. Without loss of generality, let the $M(i,j)$ be the entry we want to compute where $0 \le i < j \le k$. If $i = j-1$, $M(i,j) = \mathrm{dia}(y_j)$ as defined; otherwise $i < j-1$ and $M(i,j) = 1$ if and only if $M(i, i+1) = 1$ and $E(y_j, y_{j-1}) = \ldots = E(y_{i+2}, y_{i+1}) = 1$, which is equivalent to $\mathrm{dia}(y_{i+1}) = 1$ and $\mathrm{lce}(y_j) \ge j - i - 1$. Thus both cases can be solved in $O(1)$ time. □

Lemma 4 *Given* $G = (V, E)$*,* $\mathrm{lce}(\cdot)$ *and* $\mathrm{dia}(\cdot)$ *can be computed in* $O(|V| + |E|)$ *time.*

Proof. We retrieve consecutive edges starting from y_k, y_{k-1}, ..., until the first y_p with $p \le k$ and $E(y_p, y_{p-1}) = 0$. Then we can fill $\mathrm{lce}(y_k) = k - p$, $\mathrm{lce}(y_{k-1}) = k - p - 1$, ..., and $\mathrm{lce}(y_p) = 0$ immediately. Next, we start a new retrieving and filling process from y_{p-1}, and

repeat this until y_0 is visited. Eventually we retrieve $O(k)$ consecutive edges. A similar process can be applied to x. Using a common graph data structure such link lists, a consecutive edge can be retrieved in constant time, and thus $\mathrm{lce}(\cdot)$ can be computed in $O(|V|)$ time.

By definition, $\mathrm{dia}(x_j) = M(j, j-1) = 1$ if and only if there exists some i with $0 \le i < j-1$, $M(i, j-1) = 1$ and $E(x_i, x_j) = 1$. If we have computed $\mathrm{dia}(x_0), \ldots, \mathrm{dia}(x_{j-1})$ and $\mathrm{dia}(y_{j-1}), \ldots, \mathrm{dia}(y_0)$, then $M(i, j-1)$ can be computed in constant time by means of the proof in Lemma 3. To find the x_i for $E(x_i, x_j) = 1$, we can visit every inside edge that ends at x_j. Thus $\mathrm{dia}(x_j)$ can be computed and so can $\mathrm{dia}(y_j)$. Therefore the computation of $\mathrm{dia}(\cdot)$ visits every inside edge exactly once, and the total time is $O(|V| + |E|)$. □

Theorem 5 *Assume that $G = (V, E)$ is given.*

1. *A feasible solution of G can be found in $O(|V| + |E|)$ time and $O(|V|)$ space.*
2. *All feasible solutions of G can be found in $O(n|V| + |E|)$ time and $O(n|V|)$ space, where n is the number of solutions.*

Proof. These statements are proved as follows.

Statement 1. By Lemma 4, $\mathrm{lce}(\cdot)$ and $\mathrm{dia}(\cdot)$ can be computed in $O(|V| + |E|)$ time and $O(|V|)$ space. By Lemma 3, the last row and the last column of M can be reconstructed from $\mathrm{lce}(\cdot)$ and $\mathrm{dia}(\cdot)$ in $O(|V|)$ time. By Theorem 2 and Lemma 3, a feasible solution of G can be found in $O(|E|)$ time. Therefore, finding a feasible solution takes $O(|V| + |E|)$ time and $O(|V|)$ space.

Statement 2. The proof is similar to the proof of Statement 4 in Theorem 2. Finding an additional feasible solution takes $O(|V|)$ time and $O(|V|)$ space. Thus finding n solutions takes $O(n|V| + |E|)$ time and $O(n|V|)$ space. □

A feasible solution of G is a path of $k+1$ nodes and k edges, and therefore there must exist an edge between any two nodes on the path by the edge transitive relation. This implies that there are at least $(k+1)k/2$ or $O(|V|^2)$ edges in the graph. However, in practice, a threshold is usually set for the maximum length (mass) of an edge, so the number of edges in G could be much smaller than $O(|V|^2)$ and may actually equal $O(|V|)$ sometimes. Thus Theorem 5 actually finds a feasible solution in linear time for a sparse graph G.

3.3 Algorithm for peptide sequencing

In practice, a tandem mass spectrum contains noise and other types of ions. This section describes an algorithm for the peptide sequencing problem (Definition 1). We first compute an NC-spectrum graph G from this spectrum. Let $s(\cdot)$ be the edge scoring function for G. Let $Q(i, j)$ be a two-dimension matrix with $0 \le i, j \le k$. $Q(i, j) > 0$ if and only if in G, there is a path L from x_0 to x_i and a path R from y_j to y_0, such that at most one of x_p and y_p is in $L \cup R$ for every $p \in [1, i] \cup [1, j]$; $Q(i, j) = 0$ otherwise. If $Q(i, j) > 0$, $Q(i, j) = \max_{L,R}\{s(L) + s(R)\}$, the maximum score among all L and R pairs. Table 4 shows the matrix Q for the NC-spectrum graph in Figure 4 using a scoring function $s(e) = 1$ for every edge $e \in G$.

Algorithm Compute-Q(G)
1. Initialize $Q(i, j) = 0$ for all $0 \le i, j \le k$;
2. For $j = 1$ to k
3. If $E(y_j, y_0) = 1$, then $Q(0, j) = \max\{Q(0, j), s(y_j, y_0)\}$;

4. If $E(x_0, x_j) = 1$, then $Q(j, 0) = \max\{Q(j, 0), s(x_0, x_j)\}$;
5. For $i = 1$ to $j - 1$
   (a) For every $E(y_j, y_p) = 1$ and $Q(i, p) > 0$, $Q(i, j) = \max\{Q(i, j), Q(i, p) + s(y_j, y_p)\}$;
   (b) For every $E(x_p, x_j) = 1$ and $Q(p, i) > 0$, $Q(j, i) = \max\{Q(j, i), Q(p, i) + s(x_p, x_j)\}$.

Theorem 6 *The following statements hold.*

1. *Given $G = (V, E)$,* Algorithm Compute-Q *computes the matrix Q in $O(|V||E|)$ time.*
2. *Given $G = (V, E)$, a feasible solution of G can be found in $O(|V||E|)$ time and $O(|V|^2)$ space.*

Proof. These statements are proved as follows.

Statement 1. Let L and R be the maximum score paths that correspond to $Q(i, j) > 0$ for $i < j$. By definition, after removing node y_j from R, $L \cup R - \{y_j\}$ contains at most one of x_q and y_q for all $1 \leq q \leq j - 1$. Let $(y_j, y_p) \in R$ such that $Q(i, j) = Q(i, p) + s(y_j, y_p)$ corresponding to Steps 3 and 5(a) in the algorithm. A similar analysis holds for $Q(j, i) = 1$ and $i < j$ in Steps 4 and 5(b). The loop at Step 2 uses the previously computed maximum scores $Q(0, j-1), \ldots, Q(j-1, j-1)$ and $Q(j-1, 0), \ldots, Q(j-1, j-1)$ to fill up the maximum scores in $Q(0, j), \ldots, Q(j, j)$ and $Q(j, 0), \ldots, Q(j, j)$. Thus every entry in Q is correctly computed in a correct order. For every j, Steps 5(a) and 5(b) visit every edge of G at most once, so the total time is $O(|V||E|)$.

Statement 2. Algorithm Compute-Q computes Q in $O(|V||E|)$ time and $O(|V|^2)$ space. For every i and j, if $Q(i, j) > 0$ and $E(x_i, y_j) = 1$, we compute the sum $Q(i, j) + s(x_i, y_j)$. Let $Q(p, q) + s(x_p, y_q)$ be the maximum value, and we can backtrack $Q(p, q)$ to find all the edges of the feasible solution. The total cost is $O(|V||E|)$ time and $O(|V|^2)$ space. □

3.4 Algorithm for one-amino acid modification

Amino acid modifications are related to protein functions. There are a few hundred known modifications. For example, some proteins are active when some amino acid is phosphorylated but inactive when it is dephosphorylated. In most experiments, a protein is digested into multiple peptides, and most peptides have at most one modified amino acid. This section discusses how to find one modified amino acid from a tandem mass spectrum. For the simplicity of our explanation, we assume that a given tandem mass spectrum is ideal. The methodology works for a noisy spectrum too.

We make two assumptions about the modification: (1) the modified mass is unknown and is not equal to the total mass of any number of amino acids; otherwise, it is information-theoretically impossible to detect an amino acid modification from tandem mass spectral data; (2) there is no feasible reconstruction for the given spectral data because a modification is rare if there is a feasible solution.

Definition 3 The one-amino acid modification problem *is equivalent to the problem which, given $G = (V, E)$, asks for two nodes v_i and v_j, such that $E(v_i, v_j) = 0$ but adding the edge (v_i, v_j) to G creates a feasible solution that contains this edge.*

Suppose the peptide sequence and the position of the modification are given. The modified mass can be determined by the difference between the experimentally measured peptide mass

and the un-modified mass. Thus, in the NC-spectrum graph G, we can identify the nodes corresponding to the prefix subsequences, among which there are only one pair of adjacent nodes v_i and v_j, such that $E(v_i, v_j) = 0$ and node v_j contains the modified amino acid. By adding the edge (v_i, v_j) to G, these nodes form a directed path from x_0 to y_0. This path is a feasible solution.

On the contrary, suppose adding an edge (v_i, v_j) to G creates a feasible solution that contains this edge. Edge (v_i, v_j) is labeled by α indicating a modified amino acid. If each edge on the path corresponds to one amino acid, we can visit the edges on the path from left to right, and concatenate these amino acids to form a peptide sequence that display the tandem mass spectrum. If some edge corresponds to multiple amino acids, we obtain more than one peptide sequences. With additional information such as a protein database or a modification database, we can predict the original amino acid(s) for α.

Let $G = (V, E)$ be an NC-spectrum graph with nodes from left to right as $x_0, \ldots, x_k, y_k, \ldots, y_0$. Let $N(i, j)$ be a two-dimension matrix with $0 \leq i, j \leq k$, where $N(i, j) = 1$ if and only if there is a path from x_i to y_j which contains exactly one of x_p and y_p for every $p \in [i, k] \cup [j, k]$. Let $N(i, j) = 0$ otherwise. Table 3 shows the matrix N for the NC-spectrum graph in Figure 4.

Algorithm Compute-N(G)
1. Initialize $N(i, j) = 0$ for all i and j;
2. Compute $N(k, k-1)$ and $N(k-1, k)$;
3. For $j = k - 2$ to 0
4. For $i = k$ to $j + 2$
   (a) if $N(i, j+1) = 1$ and $E(x_j, x_i) = 1$, then $N(j, j+1) = 1$;
   (b) if $N(i, j+1) = 1$ and $E(y_{j+1}, y_j) = 1$, then $N(i, j) = 1$;
   (c) if $N(j+1, i) = 1$ and $E(x_j, x_{j+1}) = 1$, then $N(j, i) = 1$;
   (d) if $N(j+1, i) = 1$ and $E(y_i, y_{j+1}) = 1$, then $N(j+1, j) = 1$.

Theorem 7 *The following statements hold.*

1. *Given $G = (V, E)$,* Algorithm Compute-N *computes the matrix N in $O(|V|^2)$ time.*
2. *Given $G = (V, E)$, all possible amino acid modifications can be found in $O(|V||E|)$ time and $O(|V|^2)$ space.*

Proof. These statements are proved as follows.

Statement 1. Let L and R be the paths that correspond to $N(i, j) = 1$ and $i > j$. By definition, after removing node y_j from R, $L \cup R - \{y_j\}$ contains exactly one of x_q and y_q for all $j + 1 \leq q \leq k$. Let $(y_p, y_j) \in R$, then $N(i, p) = 1$. Therefore, either $i = j + 1$ or $p = j + 1$, corresponding to Step 4(d) or 4(b) respectively in the algorithm. A similar analysis holds for $N(j, i) = 1$ and $i > j$ in Steps 4(a) and 4(c), and thus every entry in N is correctly computed in the algorithm. The loop at Step 3 uses previously computed $N(k, j+1), \ldots, N(j+1, j+1)$ and $N(j+1, k), \ldots, M(j+1, j+1)$ to fill up $N(k, j), \ldots, N(j, j)$ and $N(j, k), \ldots, N(j, j)$. Thus the algorithm computes N in a correct order. Note that $|V| = 2k + 2$ and Steps 4(a), 4(b), 4(c), and 4(d) take $O(1)$ time, and thus the total time is $O(|V|^2)$.

Statement 2. Let M and N be two matrices for G computed from Algorithm Compute-M and Algorithm Compute-N respectively at a total cost of $O(|V|^2)$ time and $O(|V|^2)$ space. Without loss of generality, let the modification be between two prefix nodes x_i and x_j with

$0 \leq i < j \leq k$ and $E(x_i, x_j) = 0$. All the prefix nodes to the right of x_j have the same mass offset from the normal locations because the corresponding sequences contain the modified amino acid. By adding a new edge (x_i, x_j) to G, we create a feasible solution S that contains this edge: (1) If $i+1 < j$, then $y_{i+1} \in S$, and thus $M(i, i+1) = 1$ and $N(j, i+1) = 1$. Finding all such x_i and x_j pairs takes $O(|V|^2)$ time because there are $O(k^2)$ possible combinations of i and j. (2) If $1 < i+1 = j < k$, then there exists an edge $(y_q, y_p) \in S$ and $q > j > i > p$, such that $E(y_q, y_p) = 1$ and $M(i, p) = 1$ and $N(j, q) = 1$. There are at most $O(|E|)$ edges that satisfy $E(y_q, y_p) = 1$, and checking $O(|V|)$ possible $i + 1 = j$ costs $O(|V||E|)$ time. (3) If $0 = i = j - 1$, then there exists an edge $(y_q, y_0) \in S$ and $q > j > i$, such that $E(y_q, y_0) = 1$ and $N(1, q) = 1$, which can be examined in $O(|V|)$ time. (4) If $i+1 = j = k$, then there exists an edge $(x_k, y_p) \in S$ and $j > i > p$, such that $E(x_k, y_p) = 1$ and $M(k-1, p) = 1$, which can be examined in $O(|V|)$ time. The case that the modification is between two prefix nodes x_k and y_j can be examined for $E(x_k, y_j) = 0$ and $M(k, j) = 1$ in $O(|V|)$ time. Thus the total complexity is $O(|V||E|)$ time and $O(|V|^2)$ space. □

Note that the condition in Theorem 7 does not require that all ions in the spectrum are observed. If some ions are lost but their complementary ions appear, G still contains all prefix and suffix nodes of the target sequence. Furthermore, if G does not contain all prefix and suffix nodes because of many missing ions, this algorithm still finds the position of the modification but the result is affected by the quality of the data.

4 Experimental results

We have presented algorithms for reconstructing peptide sequences from tandem mass spectral data with noise and loss of ions. This section reports experimental studies which focus on cases of b-ions losing a water or ammonia molecule and cases of isotopic varieties for an ion. We treat the rare occurrence such as y-ions losing a water or ammonia molecule, b-ions losing two water or ammonia molecules, and other types of ions, as noise and apply Algorithm Compute-Q to reconstruct peptide sequences.

Isotopic ions come from isotopic carbons of $\mathtt{C}^{12}$ and $\mathtt{C}^{13}$. An ion usually has a couple of isotopic forms, and the mass difference between two isotopic ions is generally one or two Daltons. Their abundance reflects the binomial distribution between $\mathtt{C}^{12}$ and $\mathtt{C}^{13}$. This distribution can be used for identification. Isotopic ions can be merged to one ion of either the highest intensity or a new mass.

It is very common for a b-ion to lose a water or ammonia molecule. In the construction of an NC-spectrum graph, we add two types of edges when (1) the distance between two nodes equals the total mass of some amino acids plus the mass of one water molecule, and (2) the distance between two nodes equals the total mass of some amino acids minus the mass of one water molecule. The first type includes the case that the distance equals the mass of exactly one water molecule. Therefore, a feasible path may contain edges of these two types, but the number of the first type edges should equal the number of the second type edges, so the net number of water molecules on the path equals zero. The scoring function for each edge is based on the abundance of two nodes and the error from a standard mass of some amino acids. We have implemented Algorithm Compute-Q and tested it on the data generated by the following process:

The Chicken Ovalbumin proteins were digested with trypsin in 100 mM ammonium

bicarbonate buffer pH 8 for 18 hours at $37°C$. Then 100 $\mu\ell$ are injected in acetonitrile into a reverse phase HPLC interfaced with a Finnigan LCQ ESI-MS/MS mass spectrometer. A 1% to 50% acetonitrile 0.1%TFA linear gradient was executed over 60 minutes.

Figure 5 shows one of our prediction results. The ions labeled in the spectrum were identified successfully. We use a resolution of 1.0 Dalton and a relative abundance threshold of 5.0 in our program.

5 Further research

We are working on a generalized scoring function which gives the best prediction, and the cases of multiple peptides.

6 References

Clauser, K.R. and Baker, P.R. and Burlingame, A.L. (1999). Role of Accurate Mass Measurement (+/- 10ppm) in Protein Identification Strategies Employing MS or MS/MS. *Analytical Chemistry.* Vol.71, 14:2871-.

Comen, T.H., Leiserson, C.E., Rivest, R.L. (1990). *Introduction to Algorithms.* (The MIT Press).

Dancik, V., Addona, T.A., Clauser, K.R., Vath, J.E. and Pevzner, P.A. (1999). De Novo Peptide Sequencing via Tandem Mass Spectrometry: A Graph-Theoretical Approach. *Journal of Computational Biology.* **6,** 327-342.

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

McLafferty, F.W., Fridriksson, E.K., Horn, D.M., Lewis, M.A. and Zubarev, R.A. (1999). Biomolecule Mass Spectrometry. *Science.* **284**: 1289-1290.

Perkins, D.N., Pappin D.J.C., Creasy, D.M. and Cottrell, J.S. (1999). Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis.* 20:3551-3567.

Taylor, J.A. and Johnson, R.S. (1997). Sequence Database Searches via *de Novo* Peptide Sequencing by Tandem Mass Spectrometry. Rapid Communications in Mass Spectrometry. **11**:1067-1075.

Wilkins, M.R., Williams, K.L., Appel, R.D. and Hochstrasser, D.F. (1997). *Proteome Research: New Frontiers in Functional Genomics.* (Springer-Verlag).

7 Tables

	B-ion Sequences
b_1	$(\mathtt{R}_1)^+$
b_2	$(\mathtt{R}_1 - \mathtt{R}_2)^+$

	Y-ion Sequences
y_2	$(\mathtt{R}_2 - \mathtt{R}_3)^+$
y_1	$(\mathtt{R}_3)^+$

Table 1: Ionization and fragmentation of peptide $(\mathtt{R}_1 - \mathtt{R}_2 - \mathtt{R}_3)$.

M	0	1	2
0	1	0	0
1	1	0	1
2	1	0	0

Table 2: Matrix M for the NC-spectrum graph in Figure 4.

N	2	1	0
2	0	1	0
1	1	0	1
0	1	1	0

Table 3: Matrix N for the NC-spectrum graph in Figure 4.

Q	0	1	2
0	0	0	0
1	1	0	2
2	2	0	0

Table 4: Matrix Q for the NC-spectrum graph in Figure 4.

8 Figures

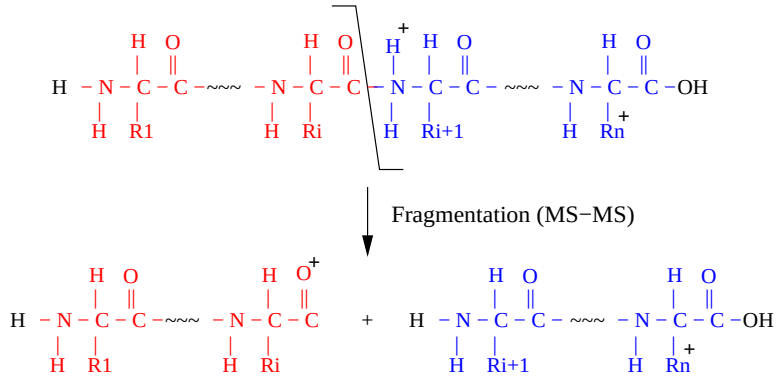

Figure 1: A doubly charged peptide molecule is fragmented into a b-ion and a y-ion.

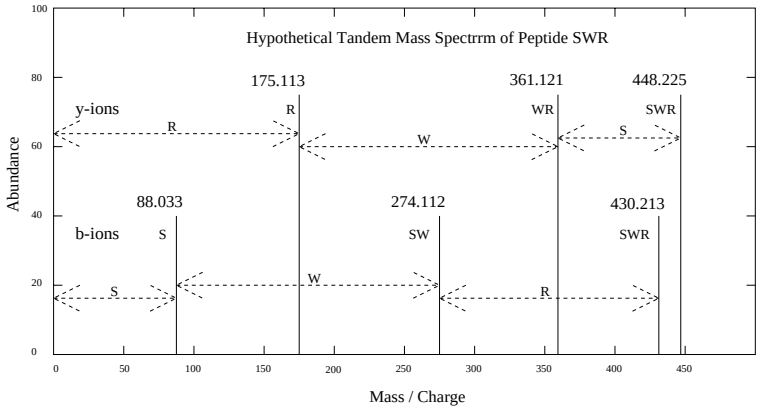

Figure 2: Hypothetical tandem mass spectrum of peptide SWR.

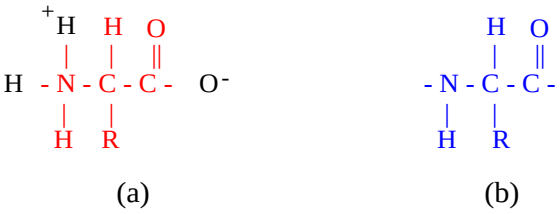

Figure 3: (a) An ionized amino acid molecule and (b) a residue.

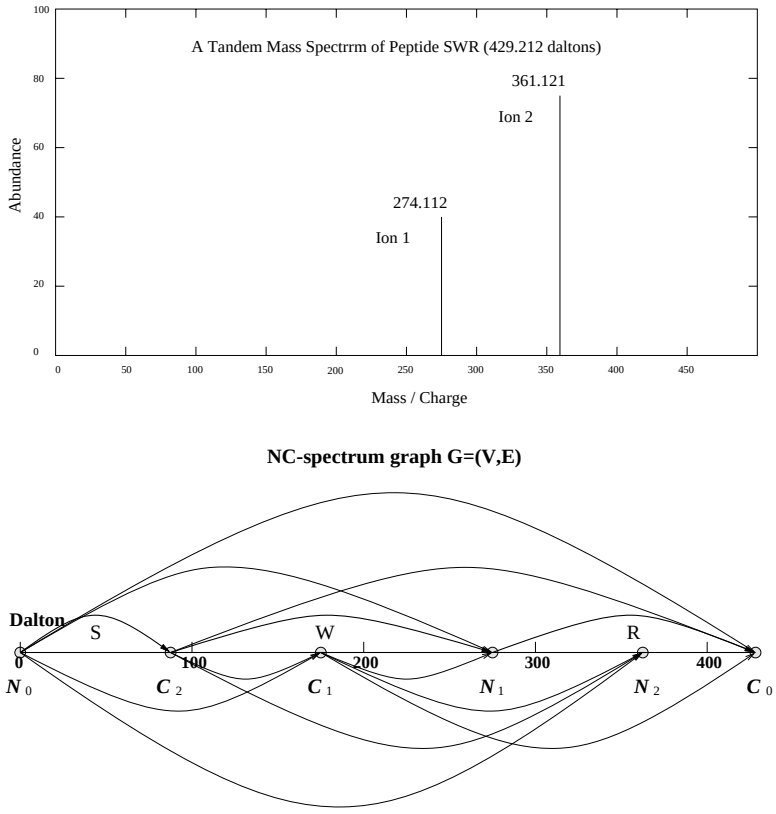

Figure 4: A tandem mass spectrum and its corresponding NC-spectrum graph.

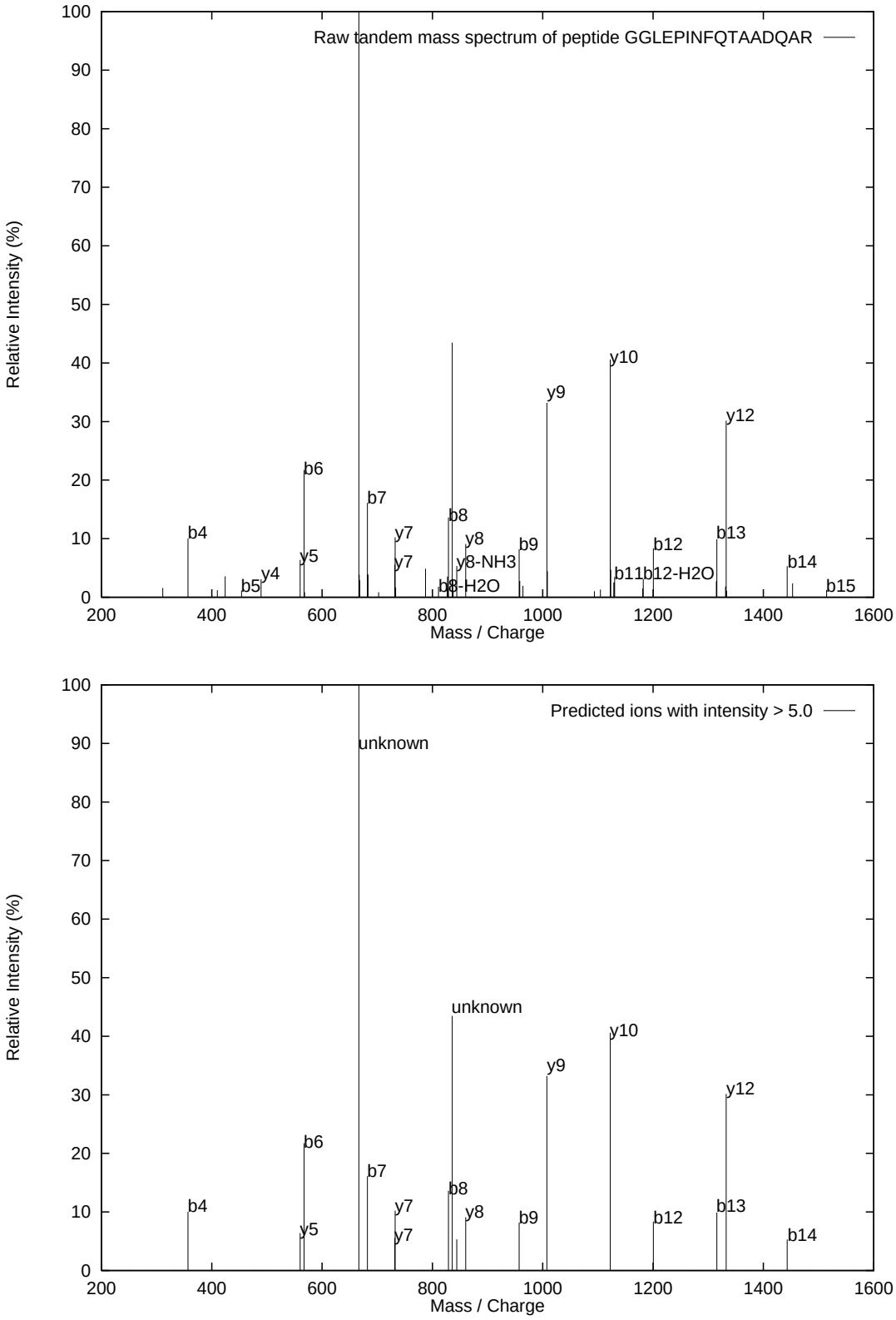

Figure 5: Raw tandem mass spectrum and predicted ions of the Chicken Ovalbumin peptide GGLEPINFQTAADQAR.