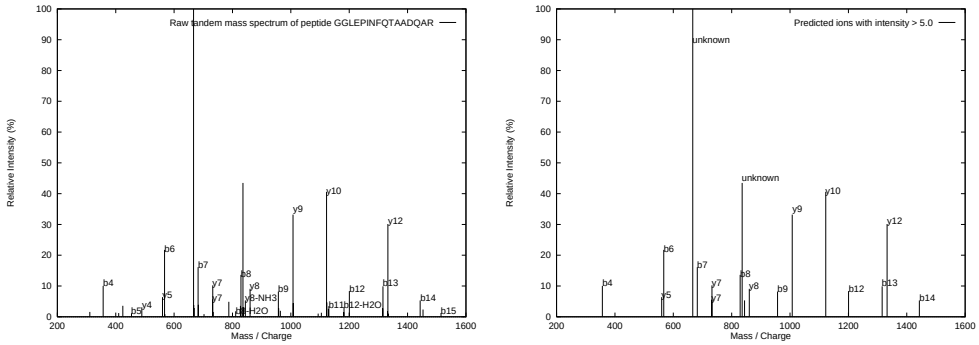

Figure 2: Raw tandem mass spectrum and predicted ions of the Chicken Ovalbumin peptide GGLEPINFQTAADQAR.

References

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

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

[3] Eng, J.K. & McCormack, A.L. & 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.

[4] Dancik, V. & Addona, T.A. & Clauser, K.R. & Vath, J.E. & Pevzner, P.A. (1999). De Novo Peptide Sequencing via Tandem Mass Spectrometry: A Graph-Theoretical Approach. *Proceedings of the 3rd Annual International Conference on Computational Molecular Biology.*

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

Lemma 11 *Given $G(V, E)$,* Algorithm Compute-Q *computes the table Q in $O(|V||E|)$ time.*

Proof. The correctness proof is similar to that for Lemma 3. For every j, Steps 3a and 3b visit every edge of G at most once. So the total time is $O(|V||E|)$. □

Theorem 12 *Given $G(V, E)$, a feasible solution can be found in $O(|V||E|)$ time and $O(|V|^2)$ space.*

Proof. Algorithm Compute-Q computes Q in $O(|V||E|)$ time and $O(|V|^2)$ space. For every i and j, calculate $Q(i, j) + s(x_i, y_j)$ if $Q(i, j) > 0$ and $E(x_i, y_j) = 1$. Let $Q(p, q) + s(x_p, y_q)$ be the maximum value. All edges can be reconstructed by backtracking $Q(p, q)$ in $O(|V||E|)$ time. The total cost is $O(|V||E|)$ time and $O(|V|^2)$ space. □

5 Experimental results

We have presented fast algorithms for several problems of reconstructing peptide sequences from a tandem mass spectral data with loss of ions. This section reports experimental studies which focus on the case where b-ions lose a water or ammonia molecule and there are isotopic varieties for an ion. It is rare or occurs with a low intensity that y-ions lose a water or ammonia molecule, b-ions lose two water or ammonia molecules, or there exist other types of ions such as a-ions. Thus we treat these rare occurrence as noise and apply Algorithm Compute-Q to skip them automatically.

Isotopic ions come from isotopic carbons of C^{12} and 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 intensities reflect the binomial distribution between C12 and C13. This distribution can be used for identification. Isotopic ions can be merged to the ion of the highest intensity or replaced by a new mass.

It is very common for a b-ion to lose a water or ammonia molecule. In the construction of NC-spectrum graph, we add three types of edges whose lengths equal the masses of a water molecule, amino acids minus one water, and amino acids plus one water. We use Algorithm Compute-Q and keep a track of the net number of waters in Steps 3a and 3b, since a feasible solution should have a net of zero water. We have implemented this algorithm 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 2 shows our prediction results. The ions labeled in the spectrum were identified successfully. We use resolution 1.0 dalton and relative intensity threshold 5.0 in our program. More experimental results will be shown in the full version of this paper.

6 Further results

Our approach is very robust and efficient for the single-peptide case. However, our algorithm can be generalized to handle multiple-peptide cases as well.

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 $N(j+1, j) = 1$.

Lemma 9 *Given $G(V, E)$,* Algorithm Compute-N *computes the table N in $O(|V|^2)$ time.*

Proof. Similar to Lemma 3. □

Theorem 10 *Given $G(V, E)$, all possible amino acid modifications can be found in $O(|V||E|)$ time and $O(|V|^2)$ space.*

Proof. Let M and N be two tables for G computed from Lemma 3 and 9. Without loss of generality, let the modification be between two consecutive N-terminal ions, corresponding to nodes x_i and x_j, for $0 \leq i < j \leq k$ and $E(x_i, x_j) = 0$. By adding an edge (x_i, x_j) to G, we create a feasible solution S that contains (x_i, x_j). If $i+1 < j$, $y_{i+1} \in S$, and thus $M(i, i+1) = 1$ and $N(j, i+1) = 1$. There are $O(k^2)$ possible combinations of i and j, and checking all of them takes $O(|V|^2)$ time. If $i + 1 = j$, S must contain an edge (y_q, y_p), $q > j$ and $i > p$, which skips over y_i and y_j. S can be found if $E(y_q, y_p) = 1$ and $M(i, p) = 1$ and $N(j, q) = 1$. There are at most $O(|E|)$ edges, and they can be examined in $O(|E|)$ time. Checking $O(|V|)$ possible i costs $O(|V||E|)$ time. The total cost is $O(|V||E|)$ time and $O(|V|^2)$ space. □

4.2 Using scoring functions

In practice, a tandem mass spectral data 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. To deal with this situation, let $s()$ be a pre-defined edge scoring function, and let $G(V, E)$ be a weighted NC-spectrum graph. We re-define the *peptide sequencing problem*: given G, asks for a maximum weight path from x_0 to y_0, such that at most one of x_j and y_j for $j = 1, \ldots, k$ is on the path. This path is also called a *feasible solution* to G.

If $s() = 1$ for all edges, the maximum weight directed path becomes the longest path in G. Let $Q(i, j)$ be a two dimension table with $0 \leq i, j \leq k$. $Q(i, j) > 0$ 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 belongs to $L \cup R$ for every $p \in [0, i] \cup [0, j]$. Let $Q(i, j)$ be the maximum total weight of L and R pairs. Let $Q(i, j) = 0$ otherwise.

Algorithm Compute-Q(G)

1. Initialize $Q(0, 0) = 1$ and $Q(i, j) = 0$ for all $i \neq 0$ or $j \neq 0$;
2. For $j = 1$ to k
3. For $i = 0$ 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 7 *Assume that $G(V, E)$ is given.*

1. *A feasible solution can be found in $O(|V| + |E|)$ time and $O(|V|)$ space.*
2. *All feasible solutions 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 6, lce and dia can be computed in $O(|V| + |E|)$ time and $O(|V|)$ space. By Lemma 5 the last row and the last column of M can be reconstructed from lce and dia in $O(|V|)$ time. By Theorem 4, a feasible solution can be found in $O(|E|)$ time. Therefore, finding a feasible solution takes $O(|V| + |E|)$ time and $O(|V|)$ space.

Statement 2. Similar to the proof of Statement 3 in Theorem 4, finding an additional feasible solution takes $O(|V|)$ time and $O(|V|)$ space. Finding n solutions takes $O(n|V| + |E|)$ time and $O(n|V|)$ space. □

A feasible solution of G implies $|E| = O(|V|^2)$ edges by transitive relation. However, in practice, a threshold is usually set for the length of an edge, so the number of edges in G could be much smaller than $O(|V|^2)$, and equal $O(|V|)$ sometimes. An $O(|V|)$-time algorithm is much faster than an $O(|V|^2)$-time algorithm.

4 Algorithms for noisy data

4.1 Amino acid modification

Amino acid modifications are related to protein functions. For example, kinase proteins are active when phosphorylated but inactive when dephosphorylated. Although there are a few hundred known modifications, a peptide rarely has two or more modified amino acids. This section discusses how to find the position of a modified amino acid from a tandem mass spectral data. Assume that the modified mass is not equal to the total mass of any number of amino acids; otherwise, it is theoretically impossible to determine an amino acid modification from tandem mass spectral data.

Lemma 8 *The 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.*

Proof. Similar to Lemma 1. □

Let $G(V, E)$ be a NC-spectrum graph with nodes from left to right, $x_0, x_1, \ldots, x_k, y_k, \ldots, y_1, y_0$. Let $N(i, j)$ be a two dimension table 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.

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

which takes $O(|V|)$ time. With $M(k,j) = 1$, we backtrack M to search the next edge of S: if $j = k-1$, the search starts from x_{k-2} to x_0 until both $E(x_i, x_k) = 1$ and $M(i,j) = 1$ are satisfied; if $j < k-1$, then $E(x_{k-1}, x_k) = 1$ and $M(k-1, j) = 1$. We repeat this process to find every edge of S. The process visits every node of G at most once in a decreasing order from x_k to x_0 and from y_k to y_0. The total cost is $O(|V|)$ time.

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

Statement 3. Similar to the proof shown in Statement 1, we can find all the feasible solutions by backtracking M, and each feasible solution costs $O(|V|)$ time and $O(|V|)$ space. Computing M and finding n solutions cost $O(|V|^2 + n|V|)$ time and $O(|V|^2 + n|V|)$ space in total. □

3.2 Improved Algorithm

To improve the time and space complexities in Theorem 4, we encode M into two linear arrays. An edge (x_i, y_j) with $0 \le i, j \le k$ is a *cross edge*, and an edge (x_i, x_j) or an edge (y_j, y_i) with $0 \le i < j \le k$ is an *inside edge*. Let $\mathrm{lce}(z)$ be the length of the longest consecutive inside edges starting from node z; i.e.,

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

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

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

Lemma 5 *Given* lce *and* dia*, any entry of M can be determined in $O(1)$ time.*

Proof. Without loss of generality, let the $M(i,j)$ be the entry we want to determine and $0 \le i < j \le k$. If $i = j - 1$, $M(i,j) = \mathrm{dia}(y_j)$; otherwise, by definition $M(i,j) = 1$ if and only if $M(i, i+1) = 1$ and $E(y_j, y_{j-1}) = \ldots = E(y_{i+2}, y_{i+1}) = 1$, equivalent to $\mathrm{dia}(y_{i+1}) = 1$ and $\mathrm{lce}(y_j) \ge j - i - 1$. Both cases can be solved in constant time. □

Lemma 6 *Given $G(V, E)$,* lce *and* dia *can be computed in $O(|V| + |E|)$ time.*

Proof. For $0 < p \le k$, we examine consecutive edges starting from y_p and extend to y_{p-d} with no further edge after it. Then we can fill $\mathrm{lce}(y_p) = d, \mathrm{lce}(y_{p-1}) = d-1, \ldots, \mathrm{lce}(y_{p-d}) = 0$ immediately. The next time, we exam y_{p-d-1} and repeat extending and filling. If we start from $p = k$ and repeat this process, eventually we visit $O(k)$ edges. Similar result holds for x. Since a consecutive edge can be retrieved in constant time, lce can be computed in $O(|V|)$ time.

By definition, $\mathrm{dia}(x_j) = 1$ if and only if for some i with $0 \le i < j - 1$, $M(i, j-1) = 1$ and $E(x_i, x_j) = 1$. If we have computed $\mathrm{dia}(x_0), \ldots, \mathrm{dia}(x_{j-1}), \mathrm{dia}(y_{j-1}), \ldots, \mathrm{dia}(y_0)$, then $M(i, j-1)$ can be determined in constant time by Lemma 5. Then $\mathrm{dia}(x_j)$ can be determined by visiting every inside edge that ends at x_j. Since every edge of G is visited at most once, dia can be computed in $O(|V| + |E|)$ time. □

3 Algorithms for peptide sequencing

3.1 Dynamic programming

We list the nodes of G from left to right: $x_0, x_1, \ldots, x_k, y_k, \ldots, y_1, y_0$. Let $M(i,j)$ be a two dimension table 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 $p \in [0,i] \cup [0,j]$. Let $M(i,j) = 0$ otherwise.

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 $M(j-1,j) = 1$.

Lemma 3 *Given $G(V,E)$,* Algorithm Compute-M *computes the table M in $O(|V|^2)$ time.*

Proof. Let L and R be the paths that correspond to $M(i,j) = 1$. If $i < j$, by definition, after removing node y_j from R, $L \cup R - \{y_j\}$ contains exactly one of x_q and y_q for every $q = 1, \ldots, j-1$. Let edge $(y_j, y_p) \in R$, thus $M(i,p) = 1$, and either $p = j-1$ or $i = j-1$, which correspond to Step 4b and Step 4d in algorithm. A similar analysis holds for the other cases. The loop under Step 4 uses previously computed $M(0,j-1), \ldots, M(j-1,j-1)$ and $M(j-1,0), \ldots, M(j-1,j-1)$ to fill up $M(0,j), \ldots, M(j,j)$ and $M(j,0), \ldots, M(j,j)$. Thus Compute-M computes M correctly. Note that $|V| = 2k+2$ and Steps 4a, 4b, 4c, and 4d take $O(1)$ time, so the total time is $O(|V|^2)$. □

Theorem 4 *The following statements hold.*

1. *Given $G(V,E)$ and M, a feasible solution can be found in $O(|V|)$ time;*
2. *Given $G(V,E)$, a feasible solution can be found in $O(|V|^2)$ time and $O(|V|^2)$ space;*
3. *Given $G(V,E)$, $O(|V|^2 + n|V|)$ time and $O(|V|^2 + n|V|)$ space, where n is the number of solutions.*

Proof. These statements are proved as follows.

Statement 1. Note that $|V| = 2k+2$. Without loss of generality, assume a feasible solution S contain node x_k. There exists some j, such that (x_k, y_j) is an edge in S, and thus $M(k,j) = 1$. Therefore, any feasible solution corresponds to a non-zero entry in the last row or the last column of M. Suppose we have found the j for $M(k,j) = 1$ and $E(x_k, y_j) = 1$,

This coordinate scheme is adopted for the following reasons. An N-terminal b-ion has an extra Hydrogen (approximately 1 dalton), so $\text{cord}(N_j) = w_j - 1$ and $\text{cord}(C_j) = (W - (w_j - 1)) - 1 = W - w_j$; and the full peptide sequence of P has two extra Hydrogens and one extra Oxygen (approximately 16 dalton), so $\text{cord}(C_0) = W - 18$. If $\text{cord}(N_i) = \text{cord}(C_j)$ for some i and j, I_i and I_j are two complementary ions, and thus are merged into one ion. We assume all the complementary ions are merged in our discussion. We say that N_j and C_j are *derived* from I_j. For convenience, for x and $y \in V$, if $\text{cord}(x) < \text{cord}(y)$, 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 $E(x, y) = 1$, if the following conditions are satisfied: (1) x and y are not derived from the same I_j; (2) $x < y$; and (3) $\text{cord}(y) - \text{cord}(x)$ equals the total mass of some amino acids.

Since G is a directed graph on 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: $x_0, x_1, \ldots, x_k, y_k, \ldots, y_1, y_0$.

Lemma 1 *The 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 $j > 0$.*

Proof. 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 exactly one of x_j and y_j for each $j > 0$.

A satisfying directed path from x_0 to y_0 contains all prefix subsequences. Since each edge on the path corresponds to some amino acids, the concatenation of these amino acids forms the peptide sequences that display the tandem mass spectra. □

We call such a directed path *a feasible reconstruction* of P or *a feasible solution* of G. To construct G, we use a mass array $\mathcal{A}$, which takes an input of 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 2 *Assume 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. *With $\mathcal{A}$, 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 $\mathcal{A}[m - r] = 1$. Since there are only 20 possible amino acids, each entry from $\mathcal{A}[0]$ to $\mathcal{A}[\frac{h}{\delta}]$ can be determined in constant time using dynamic programming. 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 $0 < cord(v_j) - cord(v_i) < h$ and $\mathcal{A}[cord(v_j) - cord(v_i)] = 1$. There are $O(k^2)$ pairs of nodes. With $\mathcal{A}$, G can be constructed in $O(k^2)$ time. □

In practice, we choose $\delta = 0.01$ dalton, and $h = 3000$ dalton.

because the number of a protein's peptides grows quadratically with the length of the protein. Pruning techniques have been applied to screen the peptides before matching but at the cost of sacrificing accuracy.

An alternative approach [4] is *de novo peptide sequencing*. The peptide sequences are extracted from the spectral data before they are validated in the database. The spectral data is transformed to a directed acyclic graph, called a *spectrum graph*, where a node corresponds to a mass peak and an edge, labeled by some amino acids, connects two nodes differed by the total mass of the amino acids in the label. A mass peak is transformed into several nodes in the graph, and each node represents a possible prefix subsequence for the peak. Then, an algorithm is called to find a longest or highest-scoring path in the graph. The concatenation of edge labels in the path is the candidate peptide sequence. However, the well-known algorithms [5] for finding the longest path tend to include multiple nodes associated with the same mass peak. This interprets a mass peak with multiple ions, which is rare in practice. This paper addresses a general interpretation which restricts the path to contain at most one node for each mass peak.

In this paper, we introduce the notion of an *NC-spectrum graph* $G(V, E)$ for a given tandem mass spectra. In conjunction with this graph, we develop a dynamic programming approach to the following previously open problems:

- The de novo peptide sequencing problem can be solved in $O(|V| + |E|)$ time and $O(|V|)$ space for clean spectral data, and in $O(|V||E|)$ time and $O(|V|^2)$ space for noisy data.
- A modified amino acid can be found in $O(|V||E|)$ time. This is the first known algorithm to solve this problem.

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. Section 4 refines the algorithm for the data with a modified amino acid and other types of noise. Section 5 reports the implementation and testing of our algorithms on experimental data. Section 6 mentions further results.

2 Spectrum graph and peptide sequencing problem

Given the mass W of a target peptide sequence P, k ions I_1, ..., I_k of P, and the masses w_1, ..., w_k of these ions, we create a *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 $W - w_j$. Therefore, we create two complementary 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-length and full-length N-terminal ions of P. 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.,

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

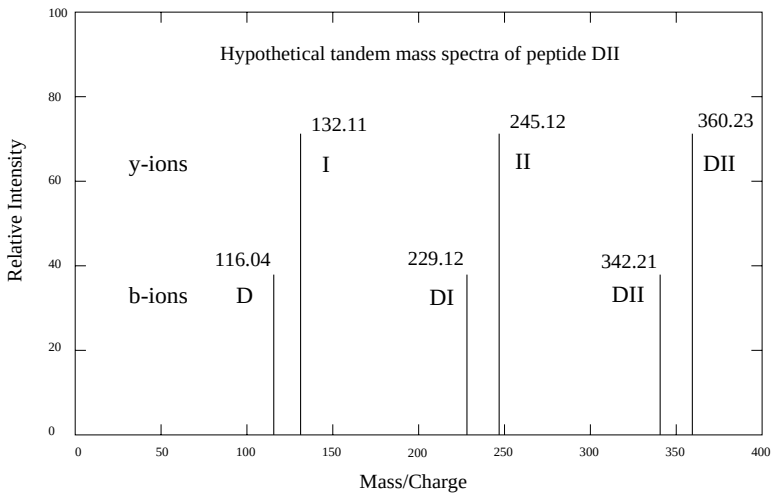

Figure 1: Hypothetical tandem mass spectra of peptide DII.

subsequence ($\mathtt{NH_2CHR_1CO} - \cdots - \mathtt{NHCHR_iCO^+}$) and the C-terminal ion is a charged suffix subsequence ($\mathtt{NH_2CHR_{i+1}CO} - \cdots - \mathtt{NHCHR_nCOOH} + \mathtt{H^+}$). This process fragments many molecules of the same peptide, and the resulting ions contains almost all possible prefix subsequences and suffix subsequences. All these prefix (or suffix) subsequences form a sequence ladder where two adjacent sequences differ by one amino acid. In the tandem mass spectrum, an ion appears at the position of its mass because it carries a +1 charge. Figure 1 shows all the ions of peptide `DII` in a hypothetical tandem mass spectra. The interpretation of a real tandem mass spectra has to deal with the following two factors: (1) some ions may be lost in the experiment and the corresponding mass peaks disappear in the spectra; (2) it is unknown whether a peak corresponds to a prefix or a suffix subsequence. The *de novo peptide sequencing problem* takes an input of the masses of some N-terminal and C-terminal ions formed by a subset of suffixes and prefixes of a taret peptide sequence P, and asks for a peptide sequence Q such that a subset of its suffixes and prefixes gives the same input masses. (Remark. Q may or may not be the same as P, depending on the input data.)

In practice, other factors can also affect a tandem mass spectra. 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 ammonia molecule and display a different mass peak from its theoretical one. An amino acid at some unnkown location of the 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 the modified amino acid is of great interest to biologists because the modification is usually associated with protein functions.

Several computer programs have been designed to interpret the tandem mass spectral data. A popular approach [3] is to correlate peptide sequences in a protein database with the tandem mass spectra. Peptide sequences in the database are first converted into hypothetical tandem mass spectrum. Then these spectrum are marched against the target spectra using some correlation functions. The sequences with top scores are reported. This approach gives an accurate identification, but cannot handle the peptides that are not in the database. Also, it does not scale up very well with the length of a protein and the size of a protein database

A Dynamic Programming Approach to De Novo Peptide Sequencing via Tandem Mass Spectrometry (Extended Abstract)

Ting Chen * Ming-Yang Kao† Matthew Tepel‡ John Rush§
George M. Church¶

July 13, 1999

Abstract

A tandem mass spectrometer measures mass/charge ratios of ions after a peptide sequence is fragmented into charged prefix and suffix subsequences. The *de novo peptide sequencing* problem is to reconstruct the peptide sequence from a given tandem mass spectral data of k ions. By implicitly translating 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 $O(|V|)$ space using dynamic programming. Our approach can be further used to discover a modified amino acid in $O(|V||E|)$ time and to analyze other types of noisy data in $O(|V||E|)$ time. Our algorithms have been implemented and tested on the experimental data.

1 Introduction

The determination of the amino acid sequence of a protein is the first step toward solving the structure and the function of this protein. Conventional sequencing methods [1] cleave proteins into peptides and then sequence the peptides individually using Edman degradation or ladder sequencing by mass spectrometry or tandem mass spectrometry [2]. Among such methods, tandem mass spectrometry combined with microcolumn liquid chromatography has been widely used. After being selected from the liquid chromatographer and the mass analyzer, the molecules of the selected peptide are processed for collision-induced dissociation. For each molecule, a peptide bond at a random position is broken, and the molecule is fragmented into two *complementary* ions, typically an N-terminal b-ion and a C-terminal y-ion. For example, if the ith peptide bond of a peptide sequence of n amino acids ($\mathtt{NH_2CHR_1CO} - \mathtt{NHCHR_2CO} - \cdots - \mathtt{NHCHR_nCOOH}$) is broken, the N-terminal ion is a charged prefix

*Department of Genetics, Harvard Medical School, Boston, MA 02115, USA; tchen@salt2.med.harvard.edu. Supported by the Lipper Foundation.

†Department of Computer Science, Yale University, New Haven, CT 06520, USA; kao@cs.yale.edu. Supported in part by NSF Grant 9531028.

‡Department of Genetics, Harvard Medical School, Boston, MA 02115, USA

§Department of Genetics, Harvard Medical School, Boston, MA 02115, USA

¶Department of Genetics, Harvard Medical School, Boston, MA 02115, USA