Algorithms for Identifying Protein Cross-links via Tandem Mass Spectrometry

Ting Chen [*] Jake Jaffe [†] George M. Church [‡]

Abstract

Cross-linking technology combined with tandem mass spectrometry is a powerful method that provides a rapid solution to the discovery of protein-protein interactions and protein structures. We studied the problem of detecting the cross-linked peptides and cross-linked amino acids from tandem mass spectral data. Our method consists of two steps: the first step finds two protein subsequences whose mass sum equals a given mass measured from mass spectrometry; and the second step finds the best cross-linked amino acids in these two peptide sequences that are optimally correlated to a given tandem mass spectrum. We designed fast and space-efficient algorithms for these two steps, and implemented and tested them on real experimental data of cross-linked Hemoglobin proteins.

[*] Department of Mathematics, University of Southern California, Los Angeles, CA 90089-1113, USA. Email: tingchen@hto.usc.edu.

[†] Department of Molecular and Cellular Biology, Harvard University, Cambridge, MA 02138, USA. Email: jdjaffe@fas.harvard.edu.

[‡] Department of Genetics, Harvard Medical School, Boston, MA 02115, USA. Email: church@arep.med.harvard.edu.

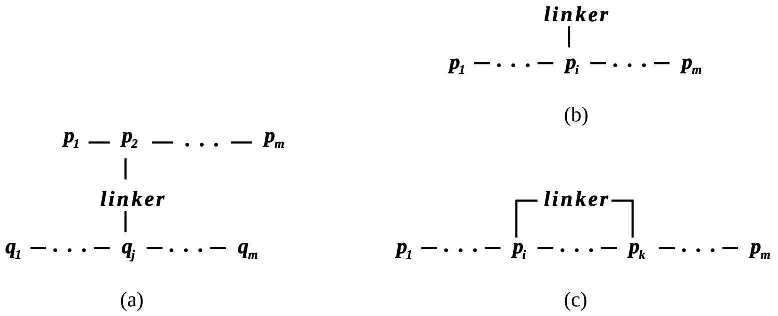
1 Introduction

In recent years, more and more genomes of model organisms have been sequenced. Using these genomic sequences, researchers have been focused on the identification of genes on the genome, the study of gene regulation and gene regulatory networks, the discovery of signal transduction pathways, the determination of protein structures, the detection of protein-protein, protein-DNA, and protein-metabolite interactions, and the elucidation of functions of genes and their protein products. A method which combines chemical cross-linking of proteins with mass spectrometry may be useful in discovering protein-protein interactions and solving protein structures. We focus on new algorithms for interpretation of complex experimental data generated by this method in this paper.

Traditionally, three-dimensional structures of proteins are solved by x-ray crystallography and NMR. However, generating an accurate structure that satisfies constraints of experimental data can be extremely difficult. There has been some success in other computational methods to predict structures from energy functions, multiple alignments, and threading. However, the accuracy and general applicability of these methods lags far behind the rates at which new protein sequences are being identified. Classical methods of detecting protein-protein interactions involve complicated biochemical experiments, which are prohibitive to scale up to determine thousands to millions of interactions among thousands of proteins.

Cross-linking technology combined with mass spectrometry provides an alternative approach to detecting protein-protein interactions and adding reliable inter-amino acid constraints to protein structures. Previous studies of cross-linking have been able to produce low resolution interatomic distance constraints, which in conjunction with threading, has led to the determination of three-dimensional structure of a model protein [4]. Similar techniques can be applied to "dock" the structures of two interacting proteins.

Tandem mass spectrometry plays a powerful role in the identification of cross-linking



linker
p1 — · · · — pi — · · · — pm
(b)
p1 — p2 — . . . — pm
linker
q1 — · · · — qj — · · · — qm
(a)
linker
p1 — · · · — pi — · · · — pk — · · · — pm
(c)

Figure 1: Three cross-link structures: (a) two cross-linked peptides, (b) one decorated peptide, and (c) one self cross-linked peptide.

sites. Tandem mass spectrometry, combined with high performance liquid chromatography (HPLC), has been widely used to identify peptides and analyze protein sequences. Peptides, i.e. $\mathtt{NH_2CHR_1CO - NHCHR_2CO - \cdots - NHCHR_nCOOH}$, resulting from proteolytic digestion of proteins are separated by HPLC and then analyzed in a mass spectrometer. The latter is a two step process which consists of measuring the mass-to-charge (m/z) ratio of an ionized peptide (the "parent" ion) and then measuring the m/z of fragmentation products (the "daughter" ions) of the peptide after collision-induced dissociation (CID). CID produces a ladder of ions, typically N-terminal "b" ions ($\mathtt{NH_2CHR_1CO - \cdots - NHCHR_iCO^+}$) and C-terminal "y" ions ($\mathtt{NH_3^+CHR_{i+1}CO - \cdots - NHCHR_nCOOH}$), where two consecutive ions differ by one amino acid. These ions display a characteristic pattern for that peptide on the tandem mass spectrum. Computer programs such as SEQUEST [1] correlate peptide sequences in a protein database with the tandem mass spectrum, and report all peptides with significant correlation scores. An alternative approach [2, 3], called de novo peptide sequencing, extracts candidate peptide sequences from the spectral data before they are validated in a database.

We performed an experiment that first chemically cross-linked interacting proteins, then digested them proteolytically (using, for example, trypsin), and finally separated and identified cross-linking sites through HPLC-tandem mass spectrometry. This paper focuses on the algorithmic solutions to the identification of cross-linked peptides from the tandem mass spectral data. Cross-link structures are shown in Figure 1. Figure 1(a) shows an "H-structure" cross-link where two amino acids in two peptides are connected to a linker by covalent bonds, Figure 1(b) shows that the linker simply decorates some amino acid of a peptide, Figure 1(b) shows a self cross-linked peptide where two amino acids on this peptide are connected to a linker. Our approach to finding an H-structure cross-link (Figure 1(a)) for a tandem mass spectrum uses the following three steps:

- **Step 1**: Given the mass of the parent cross-link molecules measured in mass spectrometry, find every pair of peptides whose mass sum (plus the mass of the linker) equals this mass.
- **Step 2**: Given a pair of peptides, find the cross-linked amino acids that are optimally correlated to the tandem mass spectral data.
- **Step 3** Report the pair of peptides with the maximum correlation score.

For simplicity, assuming that we are given k protein sequences each with n proteolytic amino acids, we can

- Solve Step 1 in $O(kn^2 \log(kn))$ time and $O(kn)$ space.

Proteins with different numbers of proteolytic amino acids can be solved by this algorithm too. If we are given 2 peptide sequences of m-amino acid length and the tandem mass spectrum has h mass peaks, we can

- Solve Step 2 in $O(m \log h)$ time and $O(m+h)$ space.

Our paper is organized as follows. Section 2 describes the algorithm to find two protein subsequences whose mass sum equals a given mass. Section 3 further studies the algorithm of finding the best cross-links between two peptides. Section 4 reports the implementation of our algorithms and the test on a Hemoglobin cross-link experimental data.

2 Identifying two protein subsequences for a mass

There are $n(n+1)/2$ possible subsequences corresponding to a protein with n proteolytic amino acids. For example, trypsin cuts a protein sequence right after a lysine(K) or an arginine(R). The exact place of each cut depends on the distribution of Ks and Rs: if two Ks or Rs are very close, the first one may not be cut. Moreover, local sequence influences and modifications may protect some Ks and Rs from being cleaved by trypsin. Therefore, two protein sequences have $O(n^4)$ possible pairs of subsequences.

To speed up the computation, we translate a protein sequence P into an array A of $n+1$ masses, each of which corresponds to a unique subsequence between two adjacent proteolytic amino acids. For example, P=NRDNKT, when trypsin is used for digestion, is translated into an array $A = (A_1, A_2, A_3)$ where A_1 =the mass for NR, A_2 =the mass for DNK and A_3 =the mass for T. Thus any proteolytic subsequence of P has the mass sum equal to the sum of the elements in the corresponding interval of A. For example, the subsequence DNKT has the mass of $A_2 + A_3$.

Since the mass of the linker is fixed, we focus on finding two protein subsequences for a given mass M, which is equivalent to the Subsequence Sum Problem (SSP) defined below.

Definition 1 Subsequence Sum Problem (SSP): *Given two positive n-sequences $A = (a_1, ..., a_n)$ and $B = (b_1, ..., b_n)$, and a number M, find all possible pairs of subsequences, $(a_i, ..., a_j), 1 \leq i \leq j \leq n$, and and $(b_k, ..., b_l), 1 \leq k \leq l \leq n$, such that*

$$\sum_{s=i}^{j} a_s + \sum_{t=k}^{l} b_t = M$$

2.1 Algorithms for finding subsequences from one sequence

First, we consider a sequence A and a mass M, and we are asked to find a subsequence of A such that its sum equals M. The following Algorithm Find-A solves this problem.

Algorithm Find-A(A, M)

```
1. $i = 1$, $j = 1$;                                      # Subsequence: $(a_1)$
2. $sum = a_1$;                                           # $sum$ is the sum of the subsequence
3. While $j < n$ or $sum \geq M$                          # End if $j = n$ and $sum < M$
4.     If $sum = M$
5          then $output(i, j)$;                           # Solution: $sum = M$
6.     If $sum < M$
7.         then $j = \min(j+1, n)$, $sum = sum + a_j$;    # Add $a_{j+1}$ to $(a_i, \ldots, a_j)$
8.     Else                                               # $sum > M$ or $sum = M$
9.         $sum = sum - a_i$, $i = i + 1$.                # Delete $a_i$ from $(a_i, \ldots, a_j)$
```

Theorem 1 *Given a positive number sequence $A = (a_1, ..., a_n)$ and a number M,* Algorithm Find-A *finds all the subsequences $(a_i, ..., a_j), 1 \leq i \leq j \leq n$, satisfying*

$$\sum_{s=i}^{j} a_s = M \tag{1}$$

in $O(n)$ time and $O(n)$ space.

Proof. At Steps 3-9, either i or j increases by one at each iteration and the iteration will stop before $j > n$ and $i > n$. Therefore Algorithm Find-A runs in linear time. We will show that Algorithm Find-A finds all the solutions.

Algorithm Find-A first considers subsequences starting with a_1 (Step 1) and computes their sums by adding $a_2, a_3, \ldots$ one by one into (a_1) (Steps 6 and 7), until the subsequence $(a_1, \ldots, a_j)$ satisfies

$$\sum_{s=1}^{j-1} a_s < M \leq \sum_{s=1}^{j} a_s. \tag{2}$$

Since all the numbers are positive, either $(a_1, \ldots, a_j)$ is a solution (Steps 4 and 5), or there is no solution for subsequences starting with a_1.

Then Algorithm Find-A looks at subsequences starting with a_2 (after Steps 8 and 9). By Equation 2, $\sum_{s=2}^{j-1} a_s < M$. Thus, the algorithm starts with the subsequence $(a_2, ..., a_j)$, adds $a_{j+1}, a_{j+2}, \ldots$ one by one into it, and calculates the sums (Steps 6 and 7). The addition at Step 7 stops when

$$\sum_{s=2}^{j+k-1} a_s < M \leq \sum_{s=2}^{j+k} a_s.$$

Then, the algorithm checks if $a_2, ..., a_{j+k}$ is a solution. As before, either $(a_2, \ldots, a_{j+k})$ is a solution (Steps 4 and 5), or there is no solution for subsequences starting with a_2.

The algorithm repeats this process for subsequences starting with $a_3, \ldots, a_n$, until every solution is found. At most one solution corresponds to subsequences starting with a_i for $i = 1, 2, \ldots, n$, so the total number of solutions is at most n. Algorithm Find-A requires $O(n)$ space. □

2.2 Algorithms for finding cross-linked subsequences

Following Algorithm Find-A and Theorem 1, we have

Lemma 2 *The SSP can be solved in $O(n^3)$ time and $O(n + p)$ space, where p is the number of solutions.*

Proof. For every subsequence $a_i, \ldots, a_j$ of A, Algorithm Find-A finds all subsequences of B that satisfy $\sum_{t=k}^{l} b_t = M - \sum_{s=i}^{j} a_s$ in $O(n)$ time. There are $O(n^2)$ subsequences of A, and thus SSP can be solved in $O(n^3)$ time. This algorithm stores all the solutions in $O(p)$ space and requires $O(n)$ space for A and B, a total of $O(n + p)$ space. □

From Theorem 1 and Lemma 2, there are at most $O(n)$ solutions (subsequences of B) for every subsequence of A. Thus,

Corollary 3 *The number of solutions for SSP is at most $O(n^3)$.*

If $O(n^2)$ space is allowed, then

Lemma 4 *The SSP can be solved in $O(n^2 \log n + p)$ time and $O(n^2 + p)$ space, where p is the number of solutions.*

Proof. We compute the sums of all subsequences of B in $O(n^2)$ time and store them into an array of $O(n^2)$ space. Then, the sums are sorted in $O(n^2 \log n)$ time. For every subsequence $a_i, \ldots, a_j$ of A, we can find all the subsequences of B that satisfy

$$\sum_{t=k}^{l} b_t = M - \sum_{s=i}^{j} a_s$$

in $O(logn + q)$ time using binary search, where q is the number of solutions. If p is the total number of solutions, finding all of them takes $O(n^2 \log n + p)$ times and $O(n^2 + p)$ space. □

Algorithm Find-AB(A, B, M)

```
1. $S_A = \{(a_i, \ldots, a_j) \mid (\sum_{s=i}^{j} a_s < M \leq \sum_{s=i}^{j+1} a_s, j < n)$ or $(\sum_{s=i}^{j} a_s < M, j = n)\}$;
2. $S_B = \{(b_k), k = 1, \ldots, n\}$;
3. While $(S_A \neq \emptyset)$
4.     Find $(a_i, \ldots, a_j) \in S_A$, $v = \sum_{s=i}^{j} a_s$ is the maximum sum;
5.     For every $(b_k, \ldots, b_l) \in S_B$ and $\sum_{t=k}^{l} b_t < M - v$
6.         Delete $(b_k, \ldots, b_l)$ from $S_B$;
7.         If $(l < n)$, then add $(b_k, \ldots, b_{l+1})$ into $S_B$;
8.     For every $(b_k, \ldots, b_l) \in S_B$ and $\sum_{t=k}^{l} b_t = M - v$
9.         output $i, j, k, l$;
10.    Delete $(a_i, \ldots, a_j)$ from $S_A$;
11.    If $i < j$, then add $(a_i, \ldots, a_{j-1})$ into $S_A$.
```

For every a_i, Step 1 finds the longest subsequence $(a_i, \ldots, a_j)$ that satisfies $\sum_{s=i}^{j} a_s < M$ for every i. Any subsequence of A to be in a solution must have the sum less than M and be a prefix subsequence of some element in S_A. Then, Step 2 finds the shortest subsequences of

B. Any subsequence of B to be in a solution must be a prefix supersequence of some element in S_B.

Steps 3-11 check every subsequence of A with sum less than M in decreasing order, and search its corresponding solutions in B. Step 4 finds $(a_i, \ldots, a_j) \in S_A$ with the maximum sum v. Steps 5-7 delete every element $(b_k, \ldots, b_l) \in S_B$ that $\sum_{t=k}^{l} b_t < M - v$ and replace by its immediate prefix supersequence $(b_k, \ldots, b_{l+1})$. Steps 8-9 find every element $(b_k, \ldots, b_l) \in S_B$ such that $\sum_{t=k}^{l} b_t = M - v$ in S_B, and report the solution $(a_i, \ldots, a_j)$ and $(b_k, \ldots, b_l)$. After finding all solutions for $(a_i, \ldots, a_j)$, Step 10 deletes it, and Step 11 adds the immediate prefix subsequence $(a_i, \ldots, a_{j-1})$ into S_A.

Theorem 5 Algorithm Find-AB *solves SSP in* $O(n^2 logn + p \log n)$ *time and* $O(n + p)$ *space, where* p *is the number of solutions.*

Proof. Obviously, every output of Algorithm Find-AB is correct and unique. Assume that $(a_i, \ldots, a_j)$ and $(b_k, \ldots, b_l)$ satisfy $\sum_{s=i}^{j} a_s + \sum_{t=k}^{l} b_t = M$. We show that the algorithm finds this solution.

Let $(a_i, \ldots, a_{j'}) \in S_A$ in Step 1, which is the longest subsequence of A that starts with a_i and has a sum less than M. Because $\sum_{s=i}^{j} a_s < M$, then $j \leq j'$. In Step 4, the algorithm will check $(a_i, \ldots, a_{j'})$, and also all its prefix subsequences including $(a_i, \ldots, a_j)$: $\ldots$, $(a_i, \ldots, a_{j'})$, $\ldots$, $(a_i, \ldots, a_{j'-1})$, $\ldots$, $(a_i, \ldots, a_j)$, $\ldots$. On the other hand, every subsequence of B in the order of $\ldots$, (b_k), $\ldots$, (b_k, b_{k+1}), $\ldots$ are checked in Steps 5, 6 and 7.

Let Step 4 generate a series of v-values: $v_1, v_2, \ldots,$. This series is in decreasing order, because Step 4 always finds an element in S_A with the maximum sum and Steps 8 and 9 replace it by a subsequence with a smaller sum. Step 5 deletes subsequences of B that has sum less than $M - v_1$, $M - v_2$, $\ldots$ at each iteration respectively. Let $w = \sum_{s=i}^{j} a_s$. When the algorithm is deleting $(a_i, \ldots, a_j)$ in Step 10, all the subsequences of B with a sum less than $M - w$ would have been deleted. Thus, $(b_k, \ldots, b_l)$ is present in S_B because $\sum_{t=k}^{l} b_t = M - w$.

The algorithm should find it and report the solution $(a_i, \ldots, a_j)$ and $(b_k, \ldots, b_l)$.

Both S_A and S_B maintain $O(n)$ elements throughout the algorithm. We store them in a data structure called a Red-Black Tree [5]. It is a binary tree that allows retrieval, adding and deletion in $O(logn)$ time while keeps the tree height $O(logn)$ through efficient tree rotations. Thus the algorithm requires only $O(n+p)$ space.

In Steps 3-11, every subsequence of A is checked and deleted at most once. Every subsequence of B is checked and deleted at most once in Steps 5-9 if it is not in any solution. If a subsequence is in $r \geq 1$ solutions, it will be checked exactly $r+1$ times. It takes $O(logn)$ time for every retrieval in Steps 4, 5 and 8, every deletion in Steps 6 and 10, and every adding in Steps 7 and 11. The total running time is $O(n^2 \log n + p \log n)$. □

Algorithm Find-AB requires basically only $O(n)$ space if the solutions are not stored. It has an advantage in applications where n is huge, because an $O(n^2)$-space algorithm could easily blow up the whole virtual memory. Similar algorithms can be applied to solve the SSP problem with k sequences:

Theorem 6 *Given M and k positive n-sequences $A_t = (a_{t1}, ..., a_{tn})$, $t = 1 \ldots k$, finding all possible pairs of subsequences of A_1, A_2, ..., or A_k such that their mass sum equals M takes $O(kn^2 \log(kn) + p)$ time and $O(kn+p)$ space, where p is the number of solutions.*

In a protein database of $k \cdot n$ size where $k \gg n$, Theorem 6 implies that the solutions can be found in the scale of $O(k \log k)$ time.

3 Identifying cross-linked amino acids

3.1 Cross-Linking Structure

Two peptide sequences $P = (p_1, ..., p_m)$ and $Q = (q_1, ..., q_m)$ can be cross-linked into an H-structure shown in Figure 1(a). In Figure 1(a), amino acids p_2 and q_j are connected by a

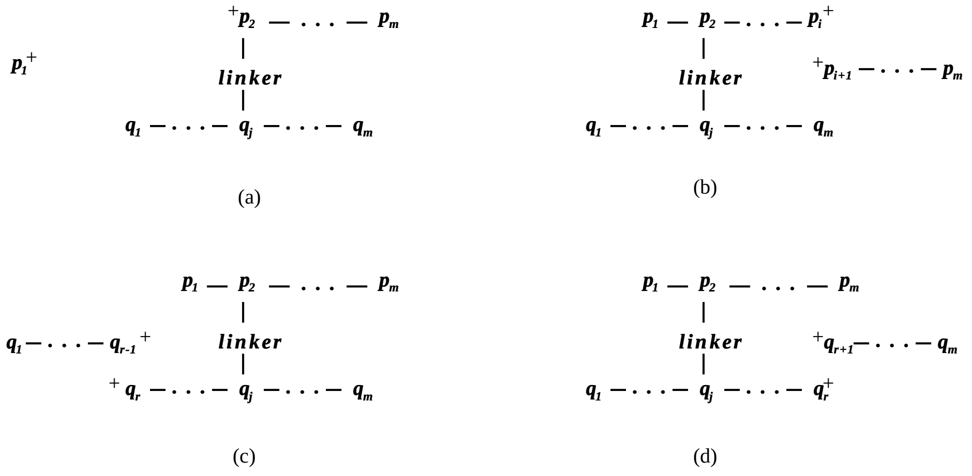


Figure 2: Four fragmentation patterns for a cross-linked molecule in tandem mass spectrometry.

linker introduced by a cross-linking agent. A typical agent, disuccinimidyl glutarate (DSG), cross-links two lysines to form a Lys-linker-Lys structure. If each peptide sequence has r ($r \leq m$) such amino acids to be cross-linked, there are r^2 possible cross-links.

Via tandem mass spectrometry, each cross-linked molecule in Figure 1(a) will result in one of the four fragmentation patterns shown in Figure 2. In Figure 2(a), the peptide bond $p_1 - p_2$ is broken, and the molecule is fragmented into an N-terminal ion, p_1^+, and an C-terminal ion, the H-structure ion on the right. In Figure 2(b), the peptide bond $p_i - p_{i+1}$ is broken, the molecule is fragmented into an N-terminal ion, the H-structure ion on the left, and a C-terminal ion, $p_{j+1}^+ - \cdots - p_n$. Similarly, if a peptide bond of Q is broken, the molecule has two fragmentation patterns shown in Figure 2(c) and 2(d). Depending on the cross-linking agent used in the experiment, a linker may contain a peptide bound. If the linker is broken, the H-structure molecule is cleaved into two peptide ions, each of which has a decoration similar to the molecule shown in Figure 1(c). Theoretically, there are $O(n)$ ions for cross-linked P and Q. A tandem mass spectrum is a collection of mass peaks, ideally, each of which corresponds to some ion in Figure 2.

3.2 Algorithms for identifying cross-linked amino acids

Definition 2 Cross-Linked Amino Acids Problem (CLAA): *Given an h-mass peak tandem mass spectrum $T = (t_1, ..., t_h)$ of two cross-linked m-amino acid peptides $P = (p_1, ..., p_m)$ and $Q = (q_1, ..., q_m)$, each having r ($r \leq m$) amino acids that can be cross-linked, find the cross-linked amino acids that maximize the value of a given scoring function $\mathcal{F}(P, Q, T)$.*

There are r^2 cross-linked amino acid pairs corresponding to r^2 possible cross-link structures. For every such structure, we can generate a hypothetical tandem mass spectrum, $S = (s_1, ..., s_{4m})$ from the masses of all the possible fragmented ions in Figure 2. A scoring function $\mathcal{F}$ compares the mass peaks of T with the mass peaks of S, and gives a score based on how well they are correlated. Generally, $\mathcal{F}$ considers a *match* between two mass peaks t_i of T and s_j of S, if $|t_i - s_j| \leq \varepsilon$, where ε is the maximum measurement error allowed in an experiment. The higher the number of matches is, the higher the $\mathcal{F}$-score is.

Now we consider the function $\mathcal{F}$ to be the counts of the number of matches. For every mass peak of S, finding its match in T takes $O(logh)$ time by binary search. The total number of matches can be determined in $O(m \log h)$ time. Thus, finding the most-likely cross-linked amino acids takes a total of $O(r^2 m \log h)$ time. However, there is better way to do it.

Theorem 7 *CLAA problem can be solved in $O(m \log h)$ time and $O(m + h)$ space if the scoring function $\mathcal{F}$ counts the number of matches.*

Proof. Figure 2(a)-(d) shows $4(m-1)$ ions for one cross-link, and thus r^2 cross-links have a total of $4(m-1)r^2$ ions. However, there are only $8(m-1)$ different masses for these ions. Figure 3 shows two types of N and C-terminal ions after breaking the peptide bond $p_i - p_{i+1}$. In Figure 3(a), the cross-linked amino acid of P locates after p_i, and in Figure 3(b), the cross-linked amino acid of P locates before p_i. The C-terminal ion in Figure 3(a) has a fixed mass no matter where the cross-link may be located, and so does the N-terminal ion in

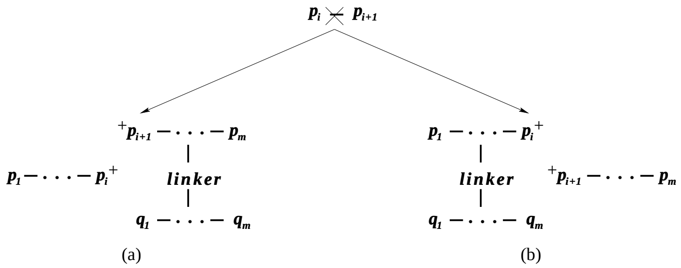


Figure 3: Two types of N and C-terminal ions after breaking the $p_i - p_{i+1}$ peptide bound.

Figure 3(b). Since P and Q together have $2(m-1)$ peptide bonds, there are only $8(m-1)$ different masses.

Let $\mathcal{M}(\alpha, T)$ count the match of a mass α with a spectrum T. $\mathcal{M}(\alpha, T) = 1$ if there is a match, and 0 otherwise. In the following steps, we can find the best cross-link:

1. Calculate ion masses P: $(L^N_{p_1}, L^C_{p_1}), \ldots, (L^N_{p_{m-1}}, L^C_{p_{m-1}})$ for the N and C-terminal ions in Figure 3(a) corresponding to the breaking of peptide bounds $(p_1 - p_2)$, ..., $(p_{m-1} - p_m)$, and $(R^N_{p_1}, R^C_{p_1}), \ldots, (R^N_{p_{m-1}}, R^C_{p_{m-1}})$ for the N and C-terminal ions in Figure 3(b).

2. Look up these masses in T for matches: $\mathcal{M}(L^N_{p_i}, T)$, $\mathcal{M}(L^C_{p_i}, T)$, $\mathcal{M}(R^N_{p_i}, T)$, and $\mathcal{M}(R^C_{p_i}, T)$, for $i = 1, ..., m-1$.

3. For every possible cross-linked amino acid p_i, calculate $score_i = \sum_{j=1}^{i-1} (\mathcal{M}(L^N_{p_j}, T) + \mathcal{M}(L^C_{p_j}, T)) + \sum_{j=i}^{m} (\mathcal{M}(R^N_{p_j}, T) + \mathcal{M}(R^C_{p_j}, T))$. Find the maximum score, $score_a$, which indicates the most likely cross-linked amino acid p_a.

4. Repeat Steps 1, 2 and 3 and find the most likely cross-linked amino acid q_b for Q.

5. Report cross-link $p_a - q_b$.

In Step 1, it takes $O(m)$ time to calculate $(L^N_{p_1}, L^C_{p_1})$ and from $(L^N_{p_1}, L^C_{p_1})$ to $(L^N_{p_2}, L^C_{p_2})$ takes only $O(1)$ time for adding and subtracting p_2. Thus Step 1 takes $O(m)$ time. The matching in Step 2 takes $O(m \log h)$ time by using binary search. Similarly, it takes $O(m)$ time to calculate $score_1$, and only additional $O(1)$ time for $score_2$. Thus, Step 3 takes $O(m)$ time. Step 4 has similar time complexity. Therefore, finding the cross-linked amino acids takes a total of $O(m \log h)$ time. □

Although Theorem 7 is for the particular function $\mathcal{F}$ that counts the number of matches, this theorem holds for almost every function we generally use, such as the correlation function which we use in the next section, the convolution function, and so on.

4 Experiments and data analysis

Chemical modifications and crosslinks were introduced into the human blood protein hemoglobin (Hb; Sigma, St. Louis, MO) by reaction with disuccinimidyl glutarate (DSG; Pierce, Rockford, IL). Hemoglobin is a four subunit protein consisting of two subunits each of two unique polypeptide chains, α and β. The approximate molecular weight of each chain is 15,000 daltons. DSG is a homobifunctional crosslinker which creates a link between two separate lysine residues by inserting a 5-carbon chain between them.

Briefly, 50 mM Hb was reacted with 5 mM DSG for 1 hour at room temperature in the presence of 90 mM sodium phosphate pH 7.5 and 10% DMSO. The reaction was stopped by addition of an equal volume of 1.0 M Tris pH 7.5. An aliquot of this reaction was mixed with 2 volumes of 6M urea and TPCK-treated trypsin (Sigma, St. Louis, MO) was added at a ratio of 1:125 w/w trypsin to Hb. This mixture was allowed to react at 37 C overnight. 100 ml of the digestion mixture was subject to LC/MS/MS analysis. The sample was loaded onto a YMC ODS-AQ S3m 120 Å1.0 x 150 mm column (Waters, Milford, MA) and the peptides were separated by a gradient of 5-80% acetonitrile (0.03% trifluoroacetic acid and 0.10% acetic acid as ion pairing reagents). The eluent was directed to a Finnigan LCQ (Thermoquest, San Jose, CA) ion trap mass spectrometer fitted with an ESI source and operated in positive ion mode. Significantly ionized peptides were automatically measured for m/z and subjected to

CID. Data were automatically collected.

The data were analyzed visually and with several in-house algorithms, as well as the commercial package SEQUEST [1]. For this experiment, we implement the algorithm in Lemma 4, and calculate the correlation between a pair of sequences S and a tandem mass spectrum T using the following function:

$$cor(T,S) = \frac{\sum_{s\in S, t\in T, \mathcal{M}(s,t)=1} h(s)\cdot h(t)}{\sqrt{\sum_{s\in S} h^2(s)\cdot \sum_{t\in T} h^2(t)}} \cdot \frac{\#matches}{|S|+|T|-\#matches}$$

where $h(t)$ gives the abundance for the mass peak t, and $h(s)$ is the hypothetical abundance for the theoretical mass peak s, corresponding to some ion of S.

Several types of modification events were recognized: inter-peptide cross-links (Figure 1(a)), decoration (Figure 1(b)), and internal cross-links (Figure 1(c)). Of specific interest to this work, a crosslink between lysine 82 of both β subunits of Hb was detected. Structurally, this would correspond to bridging the central channel of the Hb protein. The inter-residue distance between the two lysines was measured to be 8.18 Å (derived from the PDB file 1A3N), which is comparable with the 7.7 Å spacer length of DSG [6]. This crosslink was found in an ion m/z of 1174.35 at z=3, corresponding to a molecular ion of 3521.08. This mass is equivalent to two molecules of the tryptic peptide 67-VLGAFSDGLAHLDNLK-82 of Hb after being carbamylated (a N-terminal modification of 43 daltons due to urea) and cross-linked by the glutarate moiety of the DSG. Partial sequence data from this peptide was observed in the daughter ion spectra which gives credence to the result.

A more detail report will be shown in the full version of this paper.

5 Discussion

The algorithms suggested here do not consider the case of amino acid modifications. However, if a modification such as phosphorylation is known, our algorithms work too. In the real experimental data, there are many complications because of noise, multiple charged ions, mass measurement errors, ions losing a water molecule, internal ions from double fragmentation, isotopic ions, and so on. All these can affect the interpretation and some of them are machine dependent and some others are experiment dependent. Also, a good scoring function is critical to judge what is the best interpretation for a tandem mass spectrum. What is the best scoring function and how to use the abundance information remain unsolved.

References

[1] 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.

[2] 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.*

[3] Chen, T. & Kao, M. & Tepel, M. & Rush, J. & Church, G.M. (2000) A Dynamic Programming Approach to De Novo Peptide Sequencing via Tandem Mass Spectrometry *The 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, Page 389-398, 2000

[4] Young, M.M. & Tang, N. & Hempel, J.C. & Oshiro, C.M. & Taylor, E.W. & Kuntz, I.D. & Gibson, B.W. & Dollinger, G. (2000) High Throughput protein fold identification by using experimental constraints derived from intramolecular cross-links and mass spectrometry. *Proceedings of National Academy of Sciences, USA*, Page 5802-5806, Vol 97, No.11, 2000.

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

[6] Tame, J.R. & Vallone, B. (2000) The structures of deoxy human haemoglobin and the mutant Hb Tyralpha42His at 120 K. *Acta Crystallogr D Biol Crystallogr*, 2000 Jul;56 (Pt 7):805-11.