



A suffix tree approach to the interpretation of tandem mass spectra: applications to peptides of non-specific digestion and post-translational modifications

Bingwen Lu and Ting Chen*

Molecular and Computational Biology Program, Department of Biological Sciences,
University of Southern California, Los Angeles, CA 90089, USA

Received on March 17, 2003; accepted on June 9, 2003

ABSTRACT

Motivation: Tandem mass spectrometry combined with sequence database searching is one of the most powerful tools for protein identification. As thousands of spectra are generated by a mass spectrometer in one hour, the speed of database searching is critical, especially when searching against a large sequence database, or when the peptide is generated by some unknown or non-specific enzyme, even or when the target peptides have post-translational modifications (PTM). In practice, about 70–90% of the spectra have no match in the database. Many believe that a significant portion of them are due to peptides of non-specific digestions by unknown enzymes or amino acid modifications. In another case, scientists may choose to use some non-specific enzymes such as pepsin or thermolysin for proteolysis in proteomic study, in that not all proteins are amenable to be digested by some site-specific enzymes, and furthermore many digested peptides may not fall within the range of molecular weight suitable for mass spectrometry analysis. Interpreting mass spectra of these kinds will cost a lot of computational time of database search engines.

Overview: The present study was designed to speed up the database searching process for both cases. More specifically speaking, we employed an approach combining suffix tree data structure and spectrum graph. The suffix tree is used to preprocess the protein sequence database, while the spectrum graph is used to preprocess the tandem mass spectrum. We then search the suffix tree against the spectrum graph for candidate peptides. We design an efficient algorithm to compute a matching threshold with some statistical significance level, e.g. $p = 0.01$, for each spectrum, and use it to select candidate peptides. Then we rank these peptides using a SEQUEST-like scoring function. The algorithms were implemented and tested on experimental data. For

post-translational modifications, we allow arbitrary number of any modification to a protein.

Availability: The executable program and other supplementary materials are available online at: <http://hto-c.usc.edu:8000/msms/suffix/>.

Contact: tingchen@hto.usc.edu

INTRODUCTION

Protein sequence analysis is usually the first step to study a novel protein. With the growing of protein and nucleic acid sequence databases, partial sequence information can be useful for the identification of proteins by correlating experimental peptide analysis information with sequence databases. This new concept of protein identification was enhanced by the realization that mass spectrometers can be used to generate the wanted experimental data (Chait and Kent, 1992). The general approach called 'peptide mass finger-printing' employs site-specific proteolysis followed by measuring the mass-to-charge (m/z) ratios of the resulting peptides by mass spectrometry (MS). The set of observed m/z ratios is then used to search a protein database to find the corresponding peptide by applying the same proteolytic method to the database by *in silico* 'digestion'. One limitation of this peptide mapping approach is that it requires the sample to be fairly homogeneous, which can be overcome by tandem mass spectrometry.

Tandem mass spectrometry (MS/MS) plays a very important role in proteomics, e.g. for protein and peptide characterization (Eng *et al.*, 1994; Clauser *et al.*, 1999; Perkins *et al.*, 1999), and for *de novo* peptide sequencing (Taylor and Johnson, 1997; Dancik *et al.*, 1999; Chen *et al.*, 2001; Lu and Chen, 2003). One advantage of MS/MS over MS is that it is more discriminative, capable of analyzing complex mixture (Arnott *et al.*, 1993). In MS/MS, many peptides are ionized from the first MS,

*To whom correspondence should be addressed.

and then are further selected and fragmented, usually by collision-induced dissociation (CID). Mass-to-charge ratios of fragments after CID are measured. Usually a peptide bond is broken when the peptide is fragmented by CID, thus the resulting spectrum contains the information about the amino acid sequence of the peptide. The fragmentation of the peptide in CID is controlled by the physiochemical properties of the peptide and the energy of collision. Many fragments can be produced from a single peptide under CID. The charged fragment can be inferred by the position of the broken peptide-bond and the side retaining the charge. If the positive charge remains on the N-terminal side of the fragmented amide bond, this fragment ion is referred to as a b ion; the fragment ion is referred to as a y ion if the charge remains on the C-terminal side of the broken amide bond. Other types of ions such as $b-H_2O$, $b-NH_3$, $y-H_2O$ may also exist (Kinter and Sherman, 2000). The m/z ratios of the fragments after CID are quite informative. For example, the m/z ratio difference of two adjacent singly-charged y-ions is exactly the mass of the residue that differs between the two y-ions. However, in real MS/MS spectra, there is no information on the ion-type (b, y, \dots), charge of the ion (+1, +2, \dots), or position of the broken amide bond. Furthermore, it is usual that a complete ladder of an ion series is not present while some fake peaks exist. Additionally, the correctness of m/z ratios will depend on the accuracy of the instrument. As a result, the successful identification of peptide sequence using MS/MS is still a challenge.

There are several algorithms developed by different groups regarding to this application of MS/MS. A popular program called SEQUEST correlates peptide sequences in a protein database with the empirical MS/MS spectrum (Eng *et al.*, 1994). Peptide sequences in a database that have the same mass as the parent ion mass of the spectrum are converted into hypothetical MS/MS spectra. The hypothetical spectra are then matched against the real spectrum using some scoring functions. The sequences with the top scores are reported. One limitation of this algorithm is that its scoring function lacks rigorous probability foundation. A similar program ProteinProspector (Clauser *et al.*, 1999) considered the impact of mass measurement accuracy on protein identification experiment, and tried *de novo* interpretation of MS/MS spectra. Another group developed a probability-based scoring program called Mascot (Perkins *et al.*, 1999) to search sequence databases using mass spectrometry data. Another probability-based scoring system is SCOPE developed by Bafna and Edwards (2001). All these programs are more or less successful at identifying proteins by database searching.

All the programs mentioned above have to index peptide sequences in the database by mass to speed up the search.

However, when the target peptides are generated from unknown enzymes (or deliberately by some non-specific enzymes such as pepsin or thermolysin) or have post-translational modifications, the indexing becomes very expensive and impractical. For example, if trypsin that cuts after K or R is specified as the enzyme, a protein sequence of 200 amino acids will generate about 20-40 peptide sequences, which can be easily indexed. If the enzyme is unknown or non-specific, the number of possible peptide sequences increases to $200 \times 20 = 4000$, roughly 100 times more. Indexing requires huge space and is really impractical. Even with indexing, the interpretation will be 100 times slower. Another problem of database searching is that the proteins might have gone under post-translational modifications. The post-translational modification is extremely important for the study of protein functions. Several algorithms have been designed to facilitate the identification of modified proteins (Yates *et al.*, 1995; Pevzner *et al.*, 2001; Hansen *et al.*, 2001). Protein modifications change the mass of a peptide. This again causes difficulty to indexing. The resulting database searching is very slow. Most programs can only handle at most 2 to 3 modifications. All the database search programs are facing the same challenge: a faster implementation of database searching for these two important applications.

Edwards and Lippert (2002) employed a suffix tree data structure for generating peptide candidates. Here we further explore the potential of employing suffix tree and present a method developed for fast database searching for tandem mass spectrum data generated by nonspecific enzyme digestion or of proteins that undergo post-translational modifications. The basic idea is to use suffix tree data structure (Gusfield, 1997) to capture the repeat information in the protein database, thus we can reduce the searching time. We gain another speedup by using spectrum graph and a matching threshold to eliminate peptide candidates so that the correct peptide can be selected more easily by a scoring function. For handling protein modifications, our algorithms allow arbitrary number of any modification to a protein.

This paper is organized as follows. We will describe the fundamental algorithms for the construction of suffix tree and spectrum graph, and then we provide two algorithms to find matches between these two data structures. Later, we will briefly describe how to find the matching threshold with statistical significance, and how to use it to speed up the database search. We will also mention the application for post-translational modifications. Finally, we implemented the algorithms and tested them on experimental data.

METHODS

Datasets

Mass spectra dataset: A collection of ten tandem mass spectra generated from trypsin digested bovine serum albumin (BSA) were used in this study. The spectra were generated by a Finnigan LCQ ESI-MS/MS mass spectrometer by the lab of Dr. George Church at Harvard Medical School. The spectra are treated as if they are non-specifically digested, i.e. we make no assumption about the cutting sites. **Protein sequence database:** The Baker's yeast (*Saccharomyces cerevisiae*) fasta protein database was used as our experimental database. The size of the database is about three megabytes. This database was downloaded from the website by European Bioinformatics Institute at the following URL: <http://www.ebi.ac.uk/proteome/>. The BSA protein sequence is also included in the protein database.

Suffix tree and suffix tree construction

A brief introduction to suffix tree is given here. The audience are referred to Gusfield (1997) for extensive suffix tree treatments. A suffix tree T is a rooted tree data structure for a string S where each suffix of the string S is represented by a path running from the root to a leaf. A single suffix tree can also be built from multiple strings, for example, by concatenating the strings into a single string with some letters that are not presented in the strings (Gusfield, 1997). Suffix trees built on multiple strings are called generalized suffix trees. This feature of suffix tree is very useful when we want to build a suffix tree out of a biological sequence database, since sequence databases usually contain multiple sequences. A generalized suffix tree example for the two sequences 'RQPKL' and 'RQPKG' is given in Figure 1. In this example, the two sequences are concatenated by the letter '@', which is not presented in either of the sequences.

For database searching programs, a mass spectrum is used to search against a sequence database. Since the database that we search against is invariant, we can preprocess it to simplify the search. Here a suffix tree is created online using Ukkonen's linear time algorithm (Ukkonen, 1995) based on the sequence database.

Construction of spectrum graph

An NC-spectrum graph ('NC' is brief for 'N-terminal and C-terminal', since the graph is built on the N-terminal b-ions and the C-terminal y-ions) is constructed according to the input mass spectrum using the algorithm by Chen *et al.* (2001). Here we briefly summarize the ideas of the construction.

Tandem mass spectrometry measures mass/charge ratios of selected peptides and then measures their fragmented ions. Assume that the charges are known and the masses

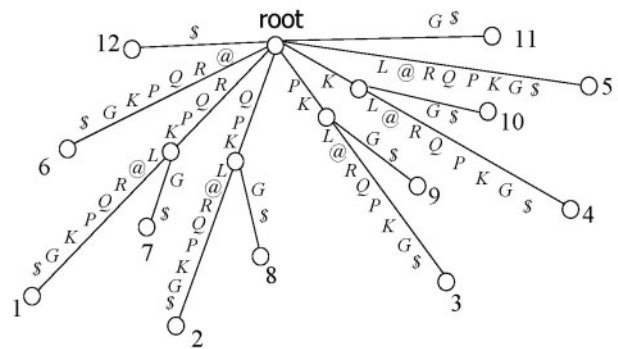


Fig. 1. Generalized suffix tree for the two sequences 'RQPKL' and 'RQPKG'. The two strings are concatenated by '@' to form the total string 'RQPKL@RQPKG\$'. The number at a leaf indicates the starting position of the suffix in the total string.

can be derived. Assume that an unknown peptide q has molecular weight W (uncharged) and k fragmented ions I_1, \dots, I_k with masses w_1, \dots, w_k , respectively. A spectrum graph $G = (V, E)$ is created as follows.

Let $m = 2k + 1$. We first create two nodes, z_0 and z_m , on a line to represent the zero mass and the total residue mass, $W - 18$, of q , respectively. The 18 daltons are for the two extra hydrogens and one extra oxygen in q , besides the residues. All other nodes are created on the line between z_0 and z_m such that their distances to z_0 correspond to the associated masses. For each I_j , because it is unknown whether it is a b-ion or a y-ion, we create a pair of nodes, z_j and z_{m-j} , placed at the mass of $w_j - 1$ and $W - (w_j - 2)$, respectively, to represent two mutually exclusive assumptions: (1) I_j is a b-ion, and z_j represents the node with the residue mass of this b-ion; and (2) I_j is a y-ion, and z_{m-j} represents the node with the residue mass of its complementary b-ion. If this ion is real, either z_j or z_{m-j} , but not both, represents the real b-ion.

The edges of the spectrum graph G always point from the lower mass nodes to the higher mass nodes. If the mass difference between two nodes z_i and z_j equals the total mass of some amino acid residues, we draw a directed edge between z_i and z_j , pointing from the low-mass node to the high-mass node. Thus, the spectrum graph G is a directed acyclic graph along a line, and all edges point to the right on the real line (so it is also in topologically sorted order). See Figure 2 for an example of a spectrum graph. In the following sections, the starting node z_0 and ending node z_m will also be referred to as N_0 (N-terminal) node and C_0 (C-terminal) node, respectively.

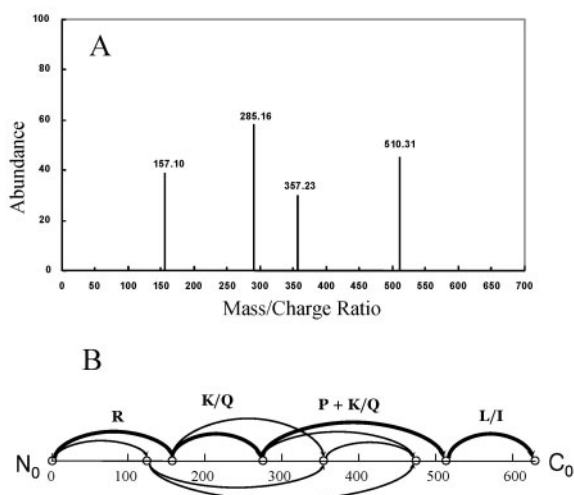


Fig. 2. An NC-spectrum graph example. A: a hypothetical tandem mass spectrum of the peptide RQPKL(622.39 Daltons.) B: a sparse NC-spectrum graph constructed from the spectrum shown in A. A path running from N_0 to C_0 is bolded and labelled with possible corresponding amino acids. The symbol '+' means 'and', while '/' means 'or'. For example, 'P+K/Q' means 'P and K' or 'P and Q'.

Searching the suffix tree against the spectrum graph

We have preprocessed the tandem mass spectrum using an NC-spectrum graph. In the following two sections we show two algorithms to search the suffix tree against the spectrum graph. In Algorithm 1 we perform a suffix tree-based searching, while in Algorithm 2 the searching is NC-spectrum graph-based. Before we introduce the algorithms, we will first introduce the basic notations to make the algorithms easier to understand.

We will use 'ST' to denote 'suffix tree' and 'NC' for 'NC-spectrum graph'. We will use r , u and v to denote the nodes in the suffix tree, where r is referring to the root. We will use N_0 , C_0 , w and z to denote the nodes in the spectrum graph, where N_0 is the starting node and C_0 is the ending node. Sometimes the subscripts 'ST' and 'NC' are also used to make the notations clearer.

Algorithm 1: ST-NC searching—ST based

We will traverse the suffix tree by a depth-first-search (DFS) algorithm (Cormen *et al.*, 2001). The spectrum graph will be used as a checking table. We will start the traversal beginning from the root r of the suffix tree. Suppose the current path is (u, v) , where u denotes the last position in the tree where the path (r, u) has been mapped onto the spectrum graph. u is not necessarily an internal node or a leaf. At the beginning of the traversal, u and r

are the same, or say u is the root of the suffix tree. We will check whether there is an edge in the spectrum graph corresponding to the current path (u, v) . Let w_{NC} be the last node in the NC-spectrum graph where some incoming path from the root to w_{NC} is mapped by the (r, u) . At the beginning of the traversal, w_{NC} is N_0 of the spectrum graph. We have two cases here:

1. The mass of the current path (u, v) can be mapped to some outgoing edge (w_{NC}, z_{NC}) . We have two possibilities in this situation:
 - (i) z_{NC} is the ending node in the spectrum graph. In this case, the path from r to v has been mapped to some path from N_0 to C_0 . Thus we will report the path (r, v) and continue the suffix tree traversal.
 - (ii) z_{NC} is not the ending node in the spectrum graph. We will then let $u \leftarrow v$, and $w_{NC} \leftarrow z_{NC}$ and continue the traversal.
2. If the mass of the current path (u, v) can not be mapped to any outgoing edge of w_{NC} , we will extend the path (u, v) one more letter:
 - If v is neither a leaf nor an internode, we will let v be the next letter in the edge-label of current edge.
 - If v is an internal node, then we can extend the path (u, v) one more letter to one of v 's children.
 - If v is a leaf, we will stop extension but continue suffix tree traversal. In this situation, we will need to backtrack on the suffix tree. When we backtrack on the suffix tree, we will also backtrack the corresponding edges in the spectrum graph.

After the extension, we will check again whether the path (u, v) can be mapped to some outgoing edge of w_{NC} . We will also add one more restriction that there can be at most three extensions in a row. The number 'three' is somewhat arbitrary, by assuming that at most three consecutive b-ions or y-ions are allowed to be missed from the tandem mass spectrum. If the quality of the spectrum is good enough, we can tighten the number of extensions to be 'two'; conversely, if the quality is not that good, one might want to relax the restriction by setting this number to be larger than 'three'. There are two cases that can happen within this scenario:

- (i) If the path (u, v) can be mapped to some outgoing edge of w_{NC} within 'three extensions', we will follow step (1) mentioned above.
- (ii) if the path (u, v) cannot be mapped to some outgoing edge of w_{NC} within 'three extensions', we will conclude that the path from root r to the current position v can not be mapped to any path from N_0 to

C_0 in the spectrum graph. In the latter case, we will stop further extension but continue the suffix tree traversal (again we need to backtrack on the suffix tree, and also backtrack the corresponding edge in the spectrum graph).

We will carry out the suffix tree traversal according to steps 1 and 2 above until the whole suffix tree is traversed.

LEMMA 1. *Algorithm ‘ST-NC searching—ST Based’ has space complexity $O(n + |V|)$ and time complexity $O(n)$, where n is the total length of sequences in the database and $|V|$ is the number of vertices in the spectrum graph.*

PROOF. Using Ukkonen’s algorithm, the suffix tree will take space $O(n)$ (Gusfield, 1997). The NC-spectrum graph will take space $O(|V|)$ (Chen *et al.*, 2001). Thus, the space complexity for this algorithm will be $O(n + |V|)$.

The algorithm will take time $O(n)$ to traverse the tree. Since we are using the spectrum graph as a checking table, each checking up will take constant time. The time spent on checkup is constant because the number of outgoing edges for each node is limited and for each checking we are only checking the outgoing edges for the node w_{NC} . Thus the time complexity for this algorithm will be $O(n)$.

THEOREM 2. *Algorithm ‘ST-NC searching—ST Based’ correctly finds all the candidate peptide sequences that have corresponding paths in the spectrum graph.*

PROOF. The statement is proved inductively as follows.

At the beginning of the algorithm, u and r are the same in the suffix tree, and w_{NC} is N_0 of the spectrum graph, or say, we have mapped r to N_0 .

Assume that the path (r, u) has been mapped to some path (N_0, w_{NC}) , the algorithm will extend the path (r, u) to v if and only if there is a corresponding outgoing edge (w_{NC}, z_{NC}) in the spectrum graph. And when we backtrack on the suffix tree, we are also backtracking the corresponding edge in the spectrum graph, which again makes sure that (r, u) is corresponding to some path (N_0, w_{NC}) .

Since we are traversing the whole tree, all candidate peptide sequence will be visited exactly once. Thus the algorithm will correctly find all the candidate peptides that have corresponding paths in the spectrum graph.

One straightforward implementation of algorithm ‘ST-NC searching—ST Based’ is to use a binary array to represent the coordinates (masses) of the NC-spectrum graph, by rounding the coordinates to the nearest integers. The values for the binary array will be 1’s for the indices that are equal to the rounded integers of the coordinates of the NC-spectrum graph. The values for the binary array

will be 0’s otherwise. Using this binary array, we can keep track of the number of matches of the path from root to the current position when we do the DFS. We can then set a minimal number of matches as a selection criterion for our candidate peptides. In Section ‘Choosing a matching threshold’ the minimal number of matches will be treated more rigorously.

Algorithm 2: ST-NC searching – NC based

In this algorithm, we will do an exhaustive search on the spectrum graph, meanwhile we will keep track of the corresponding path in the suffix tree. We will start from the leftmost node of the spectrum graph N_0 , which will be our source of exhaustive search. The spectrum graph is a directed acyclic graph (DAG) with all edges pointing from left to right (Chen *et al.*, 2001), thus it is already in topologically sorted order. We will do the exhaustive search according to the topological order of the spectrum graph.

Let (w, z) be the current edge in the spectrum graph, where w is the last node in the spectrum graph with some incoming edge(s) mapped to some path(s) in the suffix tree. At the beginning of the algorithm, w is the same as N_0 . When we say an edge in the spectrum graph is mapped to a path in the suffix tree, we mean the corresponding mass for the edge in the spectrum graph is the same as the mass for the path in the suffix tree. Let $W_{(w,z)}$ be the mass for the edge (w, z) . We will check whether we have the corresponding paths in the suffix tree for $W_{(w,z)}$. This can be done by augmenting the node data structure for the NC-spectrum graph with a field called ‘ST-pointers’.

DEFINITION In an NC-spectrum graph, the field ‘ST-pointers’ for a node w is a collection of pointers, with each pointer pointing to a position in the suffix tree where the path from the root (of the suffix tree) to that position corresponds to a path from N_0 to w in the spectrum graph.

With the help of ST-pointers, we will update our spectrum graph in the following way. For each pointer in the ST-pointers for w , let u be the position that the pointer is pointing to and we search from u downward the suffix tree to see whether we can find a corresponding path (u, v) for edge (w, z) . If such a path is found, then a pointer to v will be included in the ST-pointers of node z ; if no such path is found, we will do nothing.

We will carry out the search in the manner described above until we come to C_0 . The ‘ST-pointers’ in C_0 will automatically give us the collection of pointers to the positions in the suffix tree where each path from the root to that position corresponds to a path from N_0 to C_0 .

Similar to the algorithm ‘ST-NC searching – ST Based’, we have the following theorems. The proofs are similar

and thus omitted here but provided at the supplementary website.

LEMMA 3. Algorithm ‘ST-NC searching – NC Based’ has space complexity $O(n + |V|)$ and time complexity $O(n + |E|)$, where $|E|$ is the number of edges in the spectrum graph.

THEOREM 4. Algorithm ‘ST-NC searching – NC Based’ is correct.

Choosing a matching threshold

In this section we study how to choose a matching threshold to screen for good candidate peptides.

Given a spectrum s with parent ion mass m , what is the chance that a random peptide q (with mass m) has at least t matches with s ? If we find a peptide having t matches, how significant is this match? Let a function $f(q, s)$ be the number of matches between q and s . Then we need to compute $\Pr(f(q, s) \geq t \mid s, m)$. Actually, in our database search, the goal is to find the threshold t such that

$$\Pr(f(q, s) \geq t \mid s, m) = p, \quad (1)$$

where p indicates the statistical significance. For example, we can set $p = 0.01$. We briefly describe two algorithms to compute t such that (1) is satisfied.

Random peptide sampling. For the simplicity of description, we assume that every peptide sequence has the same probability to exist, and that the mass for each amino acid is an integer. We want to randomly generate r peptides q_1, \dots, q_r , all with the same mass m , and compute $f(q_i, s)$ for $i = 1, \dots, r$. Let N_t be the number of q_i where q_i satisfies $f(q_i, s) \geq t, i = 1, \dots, r$. We can then choose a threshold t such that

$$p \times r = N_t. \quad (2)$$

That is, the product of p and r equals N_t .

How do we efficiently sample r random peptides with mass m ? We use an array C : $C[i]$ indicates the total number of peptides having mass i . Then we have the following recursion:

$$C[i] = \sum_{j=1}^{20} C[i - \text{mass}(\alpha_j)], \quad (3)$$

where $\alpha_1, \dots, \alpha_{20}$ represent 20 amino acids and $\text{mass}()$ returns the mass of an amino acid. With this recursion, C can be computed in linear time. With C , we can generate peptides in reversed order. Starting from $C[m]$, we generate the last amino acid α_j of the peptide using the following probability:

$$\Pr(\alpha_j) = C[m - \text{mass}(\alpha_j)] / \sum_{i=1}^{20} C[m - \text{mass}(\alpha_i)], \quad (4)$$

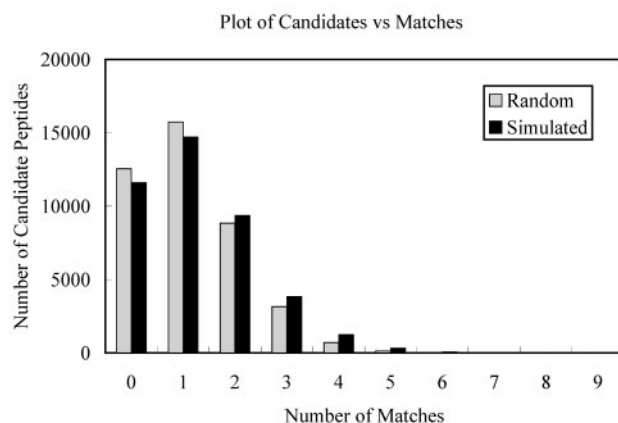


Fig. 3. Plot showing the number of candidate peptides vs the number of matches for an example spectrum (Spectrum 10 in Table 1). The distribution of the matches is approximately Poisson. The light grey histogram is the plot of the matches of 40 thousand random candidate peptides. The dark histogram is a simulated Poisson distribution, using the mean computed from the matches of random peptides and using the same number of events (matches).

and we repeat on $C[m - \text{mass}(\alpha_j)]$ until the first amino acid is generated. We can repeat this process to generate r peptides. It can be proved that this process generates each peptide with the same probability. The total time is $O(m + l)$, where l is the total number of amino acids for r peptide sequences.

Approximation by Poisson distribution. We assume that the distribution of $f(q, s)$ is Poisson, which is approximately correct shown by Figure 3. Let λ be the mean, and let $k = f(q, s)$, the probability mass function of k is

$$p(k) = e^{-\lambda} \frac{\lambda^k}{k!}, \quad k = 0, 1, 2, \dots \quad (5)$$

Then, we choose the largest t that satisfies

$$\sum_{k=t}^{\infty} p(k) \leq (1 - p). \quad (6)$$

How to calculate λ for s and m ? λ can be calculated by random peptide sampling. We first construct an array C using the algorithms shown above. Let S be the sum of matches between s and every peptide having mass m . Then $\lambda = S/C[m]$. Define an array D : $D[i]$ equals the number of peptides (having mass m) that have i as the mass of a b-ion. With D , we can calculate S using the following rule: if $D[i] > 0$ and i (b-ion) matches some mass peak in s , we add $D[i]$ to S ; if $m - i$ (y-ion) matches some mass peak in s , we also add $D[i]$ to S . Similar to the algorithm for calculating C , D can be done in linear time. Thus λ can be

Table 1. Test results of the algorithm ‘ST-NC searching – ST Based’

Spectrum	Real Sequence	Top Candidate	ST Building Time ^a	Searching Time ^b	Minimal Matches ^c	Number of Candidates ^d
1	DLGEEHFK	DLGEEHFK	25	8	3	1293
2	AEFVEVTK	AEFVEVTK ^e	26	8	3	1349
3	LYEYIAR	LYEYIAR ^e	26	7	3	913
4	LVNELTEFAK	LVNELTEFAK	23	10	5	1401
5	KQTALVELLK	KQTALVELLK	23	9	4	1319
6	LSQKFPK	ENFYLYL ^f	26	6	5	824
7	LGEYGFQNALIVR	LGEYGFQNALIVR	25	13	4	823
8	DAFLGSFLYEYSR	DAFLGSFLYEYSR	25	13	3	526
9	KVPQVSTPTLVEVSR	KVPQVSTPTLVEVSR	23	14	4	2468
10	RHPEYAVSVLLR	RHPEYAVSVLLR	29	13	4	1151

Notes: *a*) Time used to build the suffix tree in seconds; *b*) Time used to search the suffix tree for the candidate peptides in seconds; *c*) The minimal number of matches required to be as a candidate peptide; *d*) Total number of candidate peptides obtained using the Minimal Matches(*c*); *e*) More than one candidate peptides scored top one; *f*) The real peptide scored top 7.

Table 2. The number of candidate peptides decreases as the minimal number of matches to the NC-spectrum graph increases

Spectrum	Minimal Number of Matches										Matches of Real Peptide ^a	Final Ranking of Real peptide ^b
	0	1	2	3	4	5	6	7	8	9		
1	53611	35424	9742	1293	82	1	0	0	0	0	4	1
2	45732	28836	8252	1349	134	6	0	0	0	0	3	1
3	44579	27120	6540	913	68	4	0	0	0	0	4	1
4	48880	44080	30309	15144	5446	1401	279	43	8	I	9	1
5	44404	35318	17634	5815	1319	205	18	0	0	0	5	1
6	54314	47178	30859	14026	4274	824	93	10	0	0	5	7
7	47021	35219	14680	4065	823	117	9	3	0	0	7	1
8	43821	28321	5629	526	29	1	0	0	0	0	4	1
9	46606	39004	22734	9008	2468	460	67	13	3	I	11	1
10	41147	30716	14856	4855	1151	190	28	3	1	I	9	1

The 10 spectra are in the same order as in Table 1. For each spectrum, the collection containing the real peptide with the smallest number of candidate peptides is indicated by an italic font. Note: *a*) The number of matches of the real peptide to the NC-spectrum graph. *b*) The final ranking of the real peptide using a SEQUEST-like scoring function.

SEQUEST-like scoring function; for the rest spectrum (#6 in Table 1), the program ranked the real peptide top 7 among all the candidate peptides.

Using NC-spectrum graph can greatly reduce the number of candidate peptides, thus saving our time on scoring the candidate peptides against the real spectrum. Table 2 showed clearly how the minimal number of matches to the NC-spectrum graph can reduce the number of candidate peptides. Table 2 also shows that for three out of the ten spectra (#4, 9, 10), the real peptides have the largest number of matches. This reinforces that it is a good practice to use the number of matches as a screening criterion for candidate peptides.

Our program is flexible to incorporate other kinds of ions such as b–H₂O, b–NH₃, y–H₂O and a-ions, and assign different weights to different kinds of ions during the search between the suffix tree and the spectrum graph.

The idea of using the NC-spectrum graph as a screening criterion for candidate peptides can be further extended to the case where we do not know the parent ion mass. Sometimes due to the reason that the charge status of the parent ion is unknown, we are unable to know the parent ion mass beforehand. In the case of SEQUEST, the program will make several tries, by assuming that the parent ion is charged +1, +2, +3, and then calculate the respective parent ion masses. There is also one technique to predict the parent ion mass by using a zoom scan. In the scenario of the NC-spectrum graph, if we do not know the charge of the parent ion, we can simply regard all the ions as b-ions and give an upper bound of the parent ion mass. We can then use the b-ion series and the upper bound of the parent ion mass to screen for the wanted candidate peptides during our suffix tree searching, again by setting a minimal number of matches as a selection criterion.

ACKNOWLEDGMENT

This research was partially supported by grants NSF-ITR EIA - 0112934, NIH NIGMS 1-R01-RR16522-01 and University of Southern California.

REFERENCES

- Arnott,D., Shabanowitz,J. and Hunt,D.F. (1993) Mass spectrometry of proteins and peptides: sensitive and accurate mass measurement and sequence analysis. *Clin. Chem.*, **39**, 2005–2010.
- Bafna,V. and Edwards,N. (2001) SCOPE: a probabilistic model for scoring tandem mass spectra against a peptide database. *Bioinformatics*, **17 (Suppl.)**, S13–S21.
- Chait,B.T. and Kent,S.B.H. (1992) Weighing naked proteins: practical, high-accuracy mass measurement of peptides and proteins. *Science*, **257**, 1885–1890.
- Chen,T., Kao,M.Y., Tepel,M., Rush,J. and Church,G.M. (2001) A dynamic programming approach for *de novo* peptide sequencing via tandem mass spectrometry. *J. Comput. Biol.*, **8**, 325–337.
- Clauser,K.R., Baker,P.R. and Burlingame,A.L. (1999) Role of accurate mass measurement (± 10 ppm) in protein identification strategies employing MS or MS/MS. *Analytical Chem.*, **71**, 2871–2882.
- Cormen,T.H., Leiserson,C.E. and Rivest,R.L. (2001) *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. *J. Comput. Biol.*, **6**, 327–342.
- Edwards,N. and Lippert,R. (2002) Generating peptide candidates from amino-acid sequence databases for protein identification via mass spectrometry. *2nd Workshop on Algorithms in Bioinformatics (WABI 2002)*. Rome, Italy.
- 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. *J. Amer. Soc. Mass Spectrometry*, **5**, 976–989.
- Gusfield,D. (1997) *Algorithms on strings, trees and sequences: computer science and computational biology*. Cambridge University Press.
- Hansen,B.T., Jones,J.A., Mason,D.E. and Lieber,D.C. (2001) SALSA: a pattern recognition algorithm to detect electrophile-adducted peptides by automated evaluation of CID spectra in LC-MS-MS analysis. *Analytical Chem.*, **73**, 1676–1683.
- Hewick,R.M., Hunkapiller,M.W., Hood,L.E. and Dreyer,W.J. (1981) A gas-liquid solid phase peptide and protein sequenator. *J. Biol. Chem.*, **256**, 7990–7997.
- Kinter,M. and Sherman,N.E. (2000) *Protein sequencing and identification using tandem mass spectrometry*. John, Wiley & Sons, Inc..
- Kurtz,S. (1999) Reducing the space requirement of suffix tree. *Software-Practice and Experience*, **29**, 1149–1171.
- Lu,B. and Chen,T. (2003) A suboptimal algorithm for *de novo* peptide sequencing via tandem mass spectrometry. *J. Comput. Biol.*, **10**, 1–12.
- Pevzner,P.A., Mulyukov,Z., Dancik,V. and Tang,C.L. (2001) Efficiency of database search for identification of mutated and modified proteins via mass spectrometry. *Genome Res.*, **11**, 290–299.
- 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.
- Ukkonen,E. (1995) On-line construction of suffix-tree. *Algorithmica*, **14**, 249–260.
- Yates,III,J.R., Eng,J.K., McCormack,A.L. and Schieltz,D. (1995) Method to correlate tandem mass spectra of modified peptides to amino acid sequences in the protein database. *Analytical Chem.*, **67**, 1426–1436.