Articles

Discovering Known and Unanticipated Protein Modifications Using MS/MS Database Searching

Wilfred H. Tang,* Benjamin R. Halpern, Ignat V. Shilov, Sean L. Seymour, Sean P. Keating, Alex Loboda, Alpesh A. Patel, Daniel A. Schaeffer, and Lydia M. Nuwaysir

Applied Biosystems, 850 Lincoln Centre Drive, Foster City, California 94404

We present an MS/MS database search algorithm with the following novel features: (1) a novel protein database structure containing extensive preindexing and (2) zone modification searching, which enables the rapid discovery of protein modifications of known (i.e., user-specified) and unanticipated delta masses. All of these features are implemented in Interrogator, the search engine that runs behind the Pro ID, Pro ICAT, and Pro QUANT software products. Speed benchmarks demonstrate that our modification-tolerant database search algorithm is 100-fold faster than traditional database search algorithms when used for comprehensive searches for a broad variety of modification species. The ability to rapidly search for a large variety of known as well as unanticipated modifications allows a significantly greater percentage of MS/MS scans to be identified. We demonstrate this with an example in which, out of a total of 473 identified MS/MS scans, 315 of these scans correspond to unmodified peptides, while 158 scans correspond to a wide variety of modified peptides. In addition, we provide specific examples where the ability to search for unanticipated modifications allows the scientist to discover: unexpected modifications that have biological significance; amino acid mutations; salt-adducted peptides in a sample that has nominally been desalted; peptides arising from nontryptic cleavage in a sample that has nominally been digested using trypsin; other unintended consequences of sample handling procedures.

The ultimate goal of proteomics research is to use the large-scale characterization of biological materials at the protein level to enable biological discovery. Tandem mass spectrometry, also known as MS/MS, has increasingly become the method of choice for identifying proteins in complex mixtures. In a typical tandem mass spectrometry experiment, the proteins in a sample are digested (usually with the enzyme trypsin), separated, ionized, and fragmented to produce MS/MS spectra. High-throughput analysis of MS/MS spectra requires automated software, and the most widely used type of software for automatically interpreting MS/MS spectra is database search software. Implementations of database search include Sequest,[1] ProteinProspector,[2] Mascot,[3] SCOPE,[4] Sonar MS/MS,[5] ProbID,[6] and OLAV[7] among others. In a database search, the computer performs a virtual (in silico) tandem mass spectrometry experiment on all the proteins in the selected protein database and systematically matches the resulting theoretical MS/MS spectra against the experimental MS/MS spectra. This strategy works well for identifying the peptide corresponding to an MS/MS spectrum if the exact amino acid sequence of the peptide is contained in the protein database searched and none of the amino acids of the peptide are modified in any manner. However, these conditions are often not satisfied, and moreover, peptides for which these conditions do not hold often reveal a great deal of interesting biology. For example, posttranslational modifications of proteins can have a significant effect on enzymatic activity, protein interactions, or protein turnover.

To identify modified peptides, a variety of strategies have been devised: (1) MS/MS database search engines such as those mentioned above generally have some capacity for dealing with modified peptides. However, in typical implementations of database search, looking for modifications is expensive in terms of computer resources, which means that, for all practical purposes, only a limited number of modification species can be looked for. Furthermore, any modification species searched for must be explicitly specified in advance by the user. (2) Error-tolerant sequence tag searching[8−10] has been demonstrated to be capable of identifying modified peptides. However, the sequence tags used

* To whom correspondence should be addressed. E-mail: tangwh@appliedbiosystems.com.

(1) Eng, J. K.; McCormack, A. L.; Yates, J. R. *J. Am. Soc. Mass Spectrom.* **1994**, *5*, 976−989.
(2) Clauser, K. R.; Baker, P.; Burlingame, A. L. *Anal. Chem.* **1999**, *71*, 2871−2882.
(3) Perkins, D. N.; Pappin, D. J. C.; Creasy, D. M.; Cottrell, J. S. *Electrophoresis* **1999**, *20*, 3551−3567.
(4) Bafna, V.; Edwards, N. *Bioinformatics* **2001**, *17* (Suppl. 1), S13−S21.
(5) Field, H. I.; Fenyö, D.; Beavis, R. C. *Proteomics* **2002**, *2*, 36−47.
(6) Zhang, N.; Aebersold, R.; Schwikowski, B. *Proteomics* **2002**, *2*, 1406−1412.
(7) Colinge, J.; Masselot, A.; Giron, M.; Dessingy, T.; Magnin, J. *Proteomics* **2003**, *3*, 1454−1463.
(8) Mann, M.; Wilm, M. *Anal. Chem.* **1994**, *66*, 4390−4399.
(9) Sunyaev, S.; Liska, A. J.; Golod, A.; Shevchenko, A.; Shevchenko, A. *Anal. Chem.* **2003**, *75*, 1307−1315.
(10) Tabb, D. L.; Saraf, A.; Yates, J. R. *Anal. Chem.* **2003**, *75*, 6415−6421.

10.1021/ac0481046 CCC: $30.25 © 2005 American Chemical Society
Published on Web 05/10/2005

for searching are often determined by manual interpretation of MS/MS spectra rather than automatically (although GutenTag[10] provides a fully automated solution). (3) The SALSA algorithm[11,12] seeks to identify modified peptides based on patterns explicitly specified by the user. These user-specified patterns can be diagnostic of either a particular modification species or a particular amino acid sequence. (4) MS-Shotgun,[13] CIDentify,[14] MS BLAST,[15] and FASTS[16] are adaptations of popular sequence homology analysis programs (BLAST[17,18] or FASTA[19]) in order to handle the peculiarities specific to mass spectrometry data. The quality of the results is highly dependent on the quality of the input (interpreted) peptides sequences, which can be obtained through either manual interpretation of MS/MS spectra or de novo sequencing software. Another approach for using de novo sequencing is implemented by OpenSea,[20,21] which performs mass-based alignment rather than sequence-based alignment. (5) Pevzner and co-workers[22,23] described a spectral alignment algorithm for performing modification-tolerant database searching. Below, we will provide a detailed comparison between the spectral alignment algorithm and our modification-tolerant algorithm.

While each of these algorithms has been demonstrated to be effective for identifying modified peptides under certain conditions, a general, reliable, automated, and rapid method for identifying modified peptides remains an elusive goal. This report describes an algorithm that we believe represents significant progress toward this goal. The MS/MS database search engine discussed here can perform high-throughput peptide identification and, furthermore, implements novel features that enable the rapid identification of peptides with a modification of arbitrary delta mass.[24] The search engine's ability to efficiently find modifications of arbitrary delta mass allows the rapid and automated discovery of peptides containing a known (i.e., user-specified) modification as well as peptides containing an unanticipated modification. We show specific examples where the ability to quickly identify peptides modified in an unanticipated manner leads to the following types of discoveries: unexpected modifications that have biological significance; amino acid mutations; salt-adducted peptides in a sample that has nominally been desalted; peptides arising from nontryptic cleavage in a sample that has nominally been digested using trypsin; other unintended consequences of sample handling procedures.

The next section (Algorithm) describes the Interrogator algorithm, including detailed descriptions of the protein database indexing, the signal processing, and our modification-tolerant search algorithm (called zone modification searching). The Experimental Methods section briefly describes the experimental protocols used in collecting the data. The Results section is split into several subsections. The first three subsections provide examples of interesting discoveries made by performing zone modification searches. The following subsection discusses peptide discrimination, the ability to distinguish correct peptide identifications from incorrect peptide identifications. The last subsection demonstrates the large efficiency advantage of Interrogator compared to other algorithms. The Discussion section examines the capabilities of zone modification searching and also compares the algorithm against other algorithms. Finally, the Conclusions section briefly summarizes our results.

ALGORITHM

We describe here the Interrogator database search algorithm, as implemented in Pro ID 1.1, Pro ICAT 1.1, Pro QUANT 1.1, Pro ID 1.4, Pro ICAT 1.4, Pro QUANT 1.4, and Interrogator Server (an experimental application which can run on a computer cluster). The algorithm has been tested on data from quadrupole/time-of-flight and triple quadrupole linear ion trap instruments, although only results based on quadrupole/time-of-flight data are presented in this paper.

The fundamental unit of computation performed in MS/MS database searching is matching the peaks from an experimental MS/MS spectrum against the theoretically expected fragments of a peptide from a protein database. In a typical database search, an enormous number of such computations are performed due to the large number of MS/MS spectra combined with the large size of the protein database. A basic LC/MS/MS experiment typically generates hundreds or thousands of MS/MS spectra, and a more sophisticated LC/MS/MS experiment can easily generate several hundred thousand MS/MS spectra. The July 11, 2003 version[25] of NR, the nonredundant protein database from NCBI,[26] contains 1 480 858 proteins, and digesting the proteins with the digest agent trypsin yields 50 904 596 peptides. Because of the enormous number of MS/MS spectrum versus peptide matches performed during a database search, it is important to minimize the computer time used per match performed. Interrogator's approach to making the matching efficient relies on precomputing indexes for a protein database.

Protein Database Indexing. Interrogator indexes a protein database in two directions: by the masses of the peptides obtained from digestion of the proteins using trypsin (or some other specified digest agent) and by the theoretically expected MS/MS fragment masses of each of the peptides. Indexing by tryptic

[11] Hansen, B. T.; Jones, J. A.; Mason, D. E.; Liebler, D. C. *Anal. Chem.* **2001**, *73*, 1676–1683.

[12] Liebler, D. C.; Hansen, B. T.; Davey, S. W.; Tiscareno, L.; Mason, D. E. *Anal. Chem.* **2002**, *74*, 203–210.

[13] Huang, L.; Jacob, R. J.; Pegg, S. C.; Baldwin, M. A.; Wang, C. C.; Burlingame, A. L.; Babbitt, P. C. *J. Biol. Chem.* **2001**, *276*, 28327–28339.

[14] Taylor, J. A.; Johnson, R. S. *Rapid Commun. Mass Spectrom.* **1997**, *11*, 1067–1075.

[15] Shevchenko, A.; Sunyaev, S.; Loboda, A.; Shevchenko, A.; Bork, P.; Ens, W.; Standing, K. G. *Anal. Chem.* **2001**, *73*, 1917.

[16] Mackey, A. J.; Haystead, T. A.; Pearson, W. R. *Mol. Cell. Proteomics* **2002**, *1*, 139–147.

[17] Altschul, S. F.; Gish, W.; Miller, W.; Myers, E. W.; Lipman, D. J. *J. Mol. Biol.* **1990**, *215*, 403–410.

[18] Altschul, S. F.; Madden, T. L.; Schäffer, A. A.; Zhang, J.; Zhang, Z.; Miller, W.; Lipman, D. J. *Nucleic Acids Res.* **1997**, *25*, 3389–3402.

[19] Pearson, W. R.; Lipman, D. J. *Proc. Natl. Acad. Sci. U.S.A.* **1988**, *85*, 2444–2448.

[20] Searle, B. C.; Dasari, S.; Turner, M.; Reddy, A. P.; Choi, D.; Wilmarth, P. A.; McCormack, A. L.; David, L. L.; Nagalla, S. R. *Anal. Chem.* **2004**, *76*, 2220–2230.

[21] Searle, B. C.; Dasari, S.; Wilmarth, P. A.; Turner, M.; Reddy, A. P.; David, L. L.; Nagalla, S. R. *J. Proteome Res.* **2005**, *4*, 546–554.

[22] Pevzner, P. A.; Dancik, V.; Tang, C. L. *J. Comput. Biol.* **2000**, *7*, 777–787.

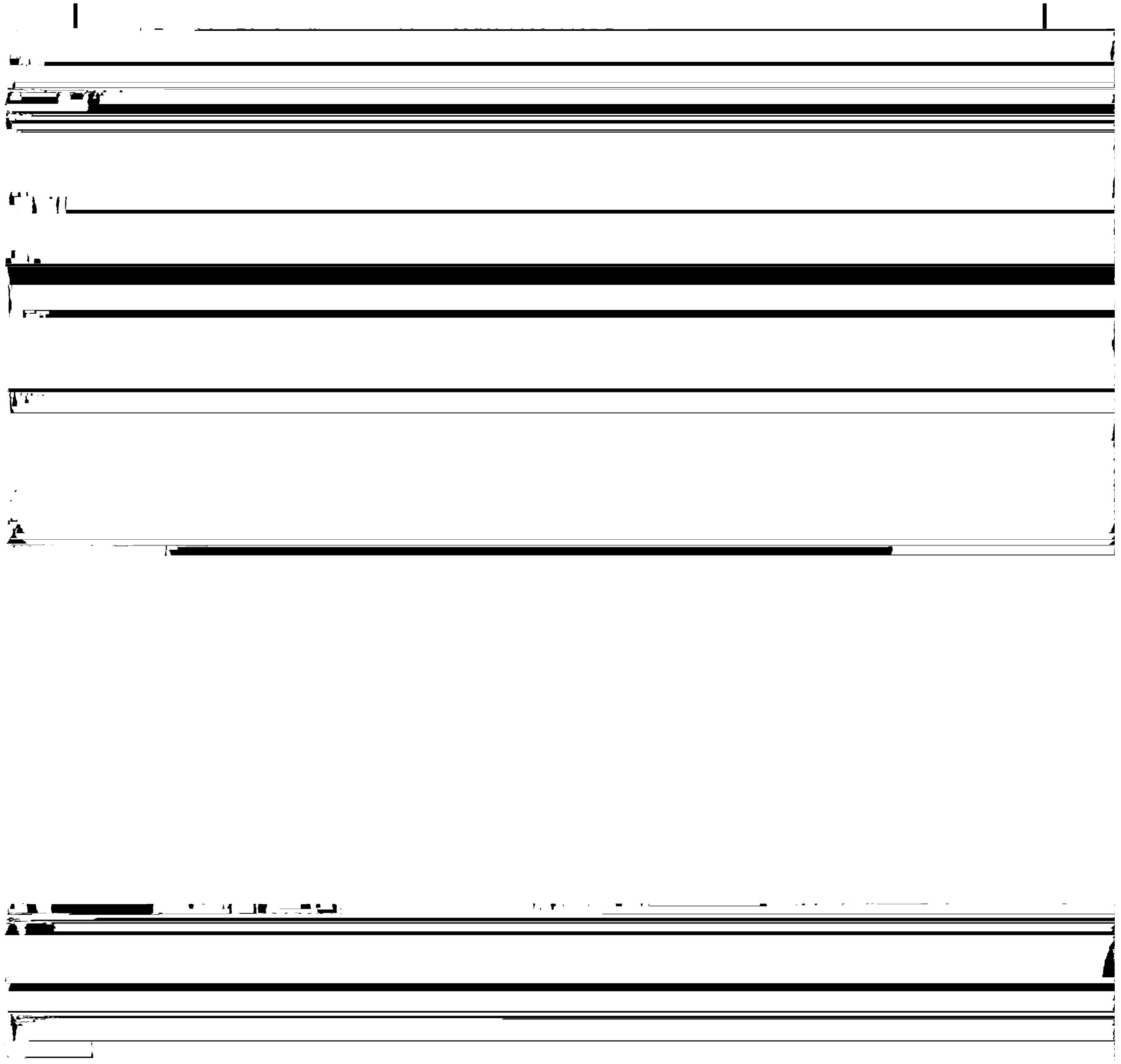
[23] Pevzner, P. A.; Mulyukov, Z.; Dancik, V.; Tang, C. L. *Genome Res.* **2001**, *11*, 290–299.

[24] In the context of mass spectrometry, delta mass is defined as the difference in mass between the experimentally measured mass and the theoretically expected mass. The theoretically expected mass is calculated using the amino acid sequence from the protein database.

[25] Hereafter, this version of NR will be referred to simply as NR.

[26] Databases can be downloaded from NCBI at ftp://ftp.ncbi.nlm.nih.gov/blast/db.

Chart 1. Schematic of the Interrogator Search Database's Index Structure[a]



[a] Interrogator indexes a protein database in two dimensions: by digest peptide mass and by theoretical MS/MS fragment ion mass. All the tables consist of lists of (key, value) pairs. The index structure enables the Interrogator scoring algorithm to directly and quickly access all peptides satisfying a given digest peptide mass range and MS/MS fragment ion mass. In this example, the Starts_B_Ions table and the Peptide_Indices_B_Ions table indicate that the peptides with mass 1160−1165 Da and containing a b ion fragment of mass 327.20 Da are as follows: LVNELTEFAK (molecular mass 1162.62 Da, b ion fragment LVN has mass 327.20 Da), LVNFVLESSR (molecular mass 1162.63 Da, b ion fragment LVN has mass 327.20 Da), and VNLLNAFASSK (molecular mass 1162.63 Da, b ion fragment VNL has mass 327.20 Da).

peptide mass is a common technique in database searching engines and generally results in significantly faster searches. However, indexing by both tryptic peptide mass and MS/MS fragment mass is, as far as we know, unique to Interrogator and results in even faster searches compared to indexing by tryptic peptide mass alone. To create the index in the first direction, every protein in a database is digested (in silico) according to the cleavage rules of trypsin (or the specified digest agent). The resulting peptides are sorted by mass and lumped into bins. Each bin contains at most 100 000 unique peptides. The mass range covered by each bin is set in order to ensure this condition holds true. For each bin, all the peptides within the bin are fragmented (in silico), and indexes in the second direction are created for the b and y ion[27] fragments separately. Chart 1 shows the details of the index structure. Finally, all the information is compressed, and the compressed indexes are stored on disk as an Interrogator search database. Once the protein database has been fully indexed, scoring an MS/MS spectrum is extremely rapid. However, before the indexed database is used, some signal processing on the MS/MS spectrum must be performed.

[27] If desired, other ion types can also be readily indexed and used for scoring.

Chart 2. Interrogator Pseudocode for Performing a No-Modification Search on an MS/MS Spectrum[a]

```
FOR each Peptide_Bin where there exists at least one Peptide such that "theoretical" ...
... Peptide.MW() - "experimental" Precursor_MW is between -MS_Tolerance and +MS_Tolerance
    INITIALIZE Score array to all 0's
    FOR each Query_Mass in the MS/MS spectrum
        Index_Low := Starts_B_Ions[(Query_Mass.Mass() - MS_MS_Tolerance) * 100]
        Index_High := Starts_B_Ions[(Query_Mass.Mass() + MS_MS_Tolerance) * 100]
        FOR each Index FROM Index_Low TO Index_High-1
            Peptide_Index := Peptide_Indices_B_Ions[Index]
            Score[Peptide_Index] := Score[Peptide_Index] + Query_Mass.Weight_For_Scoring()
        END FOR

        Index_Low := Starts_Y_Ions[(Query_Mass.Mass() - MS_MS_Tolerance) * 100]
        Index_High := Starts_Y_Ions[(Query_Mass.Mass() + MS_MS_Tolerance) * 100]
        FOR each Index FROM Index_Low TO Index_High-1
            Peptide_Index := Peptide_Indices_Y_Ions[Index]
            Score[Peptide_Index] := Score[Peptide_Index] + Query_Mass.Weight_For_Scoring()
        END FOR
    END FOR
    Keep only those elements of the Score array whose corresponding "theoretical" ...
    ... Peptide.MW() - "experimental" Precursor_MW is between -MS_Tolerance and +MS_Tolerance
END FOR

Merge results from different Peptide_Bins
High_Score := MAXIMUM of Score array
SAVE High_Score and corresponding Peptide
```

[a] The pseudocode refers to the index table structure described in detail in Chart 2. For each Query_Mass, if the renormalized intensity is among the largest 10 for each half of the MS/MS spectrum, the Weight_For_Scoring( ) method returns a weight value of 2; otherwise, the Weight_For_Scoring( ) method returns a weight value of 1. Note that in the future there may be an opportunity to improve how these weights are assigned.[61−63]

Query Mass List Construction. Signal processing begins with running a mass reconstruction algorithm, which consists of detecting peaks in the MS/MS spectrum followed by deisotoping. The (mass, intensity) pairs generated by the mass reconstruction algorithm are then subjected to an intensity renormalization relative to a crude intensity envelope, which is constructed as follows: the mass range of the MS/MS spectrum is divided into a number of intervals, and the envelope is constructed by drawing straight line segments between the (M_j, I_j) of successive intervals, where M_j is defined as the middle of the mass interval and I_j is defined as the maximum intensity in the interval. The (mass, renormalized intensity) pairs are then assembled into two lists: those with mass less than half the precursor mass and those with mass greater than half the precursor mass. From each of the two lists, the 20 (mass, renormalized intensity) pairs with the highest renormalized intensity are placed on a query mass list for matching against the indexed protein database. The goal of the intensity renormalization is to emphasize peaks that are locally intense—that is, intense compared to the peaks in the local neighborhood.

Scoring for No-Modification Searching. After the query mass list has been extracted from the MS/MS spectrum, the query masses are used for scoring against the protein database. For simplicity's sake, we first discuss searches that do not account for protein modifications; modification-tolerant searches will be discussed in detail below. From the protein database, every peptide from in silico digestion whose mass m_p matches the MS/MS spectrum's precursor mass m (to within a user-specified MS tolerance) is scored as follows. All the query masses from the MS/MS spectrum are (conceptually) "matched" (to within a user-specified MS/MS tolerance) against all the theoretically expected b and y ion fragment masses of the peptide from the database, and each match increments the score by a weight value.[28] The digest peptide that yields the highest score is deemed to be the most likely answer for that MS/MS spectrum, although the probability that this answer is actually the correct answer is a separate calculation and will be discussed later.

A detailed description of the scoring algorithm is provided in Chart 2. A close examination of the pseudocode in Chart 2 reveals that the "matching" of query masses is actually done via direct table lookups only. The direct table lookups are enabled by Interrogator's unique index structure (see Chart 1) and, moreover, result in a much faster scoring algorithm compared to ordinary database search algorithms which do not use this index structure. To understand this large difference in speed, let us consider what the pseudocode for exactly the same scoring functionality looks like in the absence of Interrogator's index structure (see Chart 3). The sparseness of theoretical fragment masses[29] means that the answer to the IF statement's query mass comparison is only occasionally TRUE. The great majority of the time the answer to the IF statement's query mass comparison is FALSE (leading to no increment of the score). Thus, the number of IF statement comparisons performed is much larger than the number of score increments performed, and the algorithm's running time is essentially determined by the number of query mass comparisons performed. By contrast, the index structure of Interrogator enables the scoring to be done with no query mass comparisons—scores are implemented using index table lookups only. Interrogator's running time is thus determined by the number of score increments, which is a much smaller number than the number of query mass comparisons. Thus, Interrogator runs considerably faster

[28] The weight depends on the renormalized intensity, with larger values of the renormalized intensity corresponding to larger weights.

[29] Averagine, a computationally constructed "average" amino acid, has a mass of 111 Da (Senko, M. W.; Beu, S. C.; McLafferty, F. W. *J. Am. Soc. Mass Spectrom.* **1995**, *6*, 229−233), so there should be, on average, 111 Da between consecutive theoretical y ion fragment masses (similarly for b ions).

Chart 3. Pseudocode for Performing an Unindexed No-Modification Search on an MS/MS Spectrum

```
FOR each Theoretical_Peptide whose MW equals Precursor_MW (to within ± MS_Tolerance)
    Score := 0
    FOR each Query_Mass in the MS/MS spectrum
        IF Query_Mass.Mass() IS EQUAL TO Theoretical_Fragment_Mass (to within ± MS_MS_Tolerance) THEN
            Score := Score + Query_Mass.Weight_For_Scoring()
        ELSE
            [DO NOTHING]
    END FOR
END FOR
```

than ordinary database search algorithms because Interrogator does not waste time performing unproductive query mass comparisons.

We note a source of inefficiency in the Interrogator scoring algorithm for simple no-modification searches. In general, the molecular weight range covered by a peptide bin in the index structure is wider than the mass range delimited by (experimental precursor mass) ± (MS tolerance). For any given peptide bin, all the peptides contained in the bin are scored before the irrelevant database peptides (i.e., those peptides whose theoretical molecular weights do not agree with the experimental peptide molecular weight to within the MS tolerance) are eliminated from consideration. In other words, the algorithm spends time scoring database peptides that are subsequently deemed to be irrelevant. This inefficiency can be reduced by making the peptide bins smaller, but making the peptide bins too small is also undesirable due to the overhead associated with each bin. The size of the peptide bin is optimized in Interrogator to balance these two competing factors. Below, in the Algorithm Efficiency subsection, we provide evidence that, despite the inefficiency discussed here, the no-modification search runs faster in Interrogator compared to ordinary database search engines. In addition, as discussed below in the description of Interrogator's modification-tolerant search algorithm, this inefficiency is eliminated or greatly mitigated for zone modification searches looking for a broad range of possible delta masses.

Memory management is also crucial to database search efficiency. Because loading data from disk into memory is a much slower operation than CPU computation, it is important to minimize disk-to-memory retrievals. Interrogator loads an index table for a particular molecular weight range into memory once and makes sure that all computations that refer to the molecular weight range are complete before allowing the block to be discarded from memory. The importance of carefully controlled memory management is particularly evident when Interrogator is run on a cluster of machines with lots of memory. If the cluster has enough memory to load the entire contents of the indexed protein database into memory all at once, the database search speeds up significantly. More precisely, on a cluster with sufficient memory,[30] the first time a particular indexed protein database is searched, the search runs at normal speed. However, on all subsequent searches against the same indexed protein database, the search runs faster, as long as the indexed database has not been swapped out of memory.

Scoring for Modification-Tolerant Searching. There are two methods of searching for protein modifications in Interrogator. In the first method, a possible modification is accounted for during the creation of the Interrogator search database (the preindexed protein database). Hence, when the Interrogator search database is subsequently used during a database search, the modification is automatically searched for. This first method of modification searching is equivalent to modification-tolerant searches in other database search algorithms and is sometimes dubbed "variable modification searching".

Interrogator's second method of searching for modifications is called zone modification searching. This novel feature enables the rapid discovery of modifications of both known and unknown delta mass. We define m as the precursor mass of the MS/MS spectrum being searched and define Δm as a candidate delta mass. Note that for a given MS/MS spectrum, m is uniquely determined, but Δm can be varied through a wide range of different values. The values of Δm can be positive, which corresponds to modifications leading to an increase in the peptide mass relative to the database sequence, or negative, which corresponds to modifications leading to a decrease in the peptide mass. For each candidate Δm, every digest peptide from the protein database whose mass m_p satisfies the relation $m_p + \Delta m = m$ (to within a user-specified MS tolerance) is scored under six different assumptions. The mass range $[0, m]$ is divided into six equally sized zones, and a modification of delta mass Δm is assumed to occur in each of the six zones in turn.

To calculate the score, each query mass from the query mass list is (conceptually) "matched" against the digest peptide under the hypothesis that the query mass arises from a b ion as well as under the hypothesis that the query mass arises from a y ion. We note that this scoring is exactly the same in nature as the scoring for the no-modification searching discussed earlier. In the case of no-modification searching, matching of a query mass under the hypothesis that the query mass arises from a b ion is accomplished (conceptually) simply by checking to see if the query mass equals (to within a user-specified MS/MS tolerance) any of the theoretically expected b ion fragment masses of the digest peptide; similarly, matching of a query mass under the hypothesis that the query mass arises from a y ion is accomplished by checking to see if the query mass equals (to within a user-specified MS/MS tolerance) any of the theoretically expected y ion fragment masses of the digest peptide. For the zone modification searching discussed here, the scoring is somewhat more complex, which we illustrate with the help of an example.

Consider the hypothetical "experimental" MS/MS spectrum that would result from perfect fragmentation of the peptide AHSSVM*QK with the methionine residue M oxidized (modification delta mass Δm of +16). "Perfect" fragmentation means that the hypothetical "experimental" MS/MS spectrum contains all of

[30] For example, sufficient memory for NR is ~20 GB, and sufficient memory for Swiss-Prot is ~2 GB. The disk space occupied by the indexed protein database is smaller by ~10-fold due to compression. Nonetheless, the relatively large space requirements make the Interrogator algorithm unwieldy for some types of searches, even with compression. For example, using a relatively nonspecific digest agent (or no digest agent at all) would lead to an unmanageable explosion in the size of the indexed database.

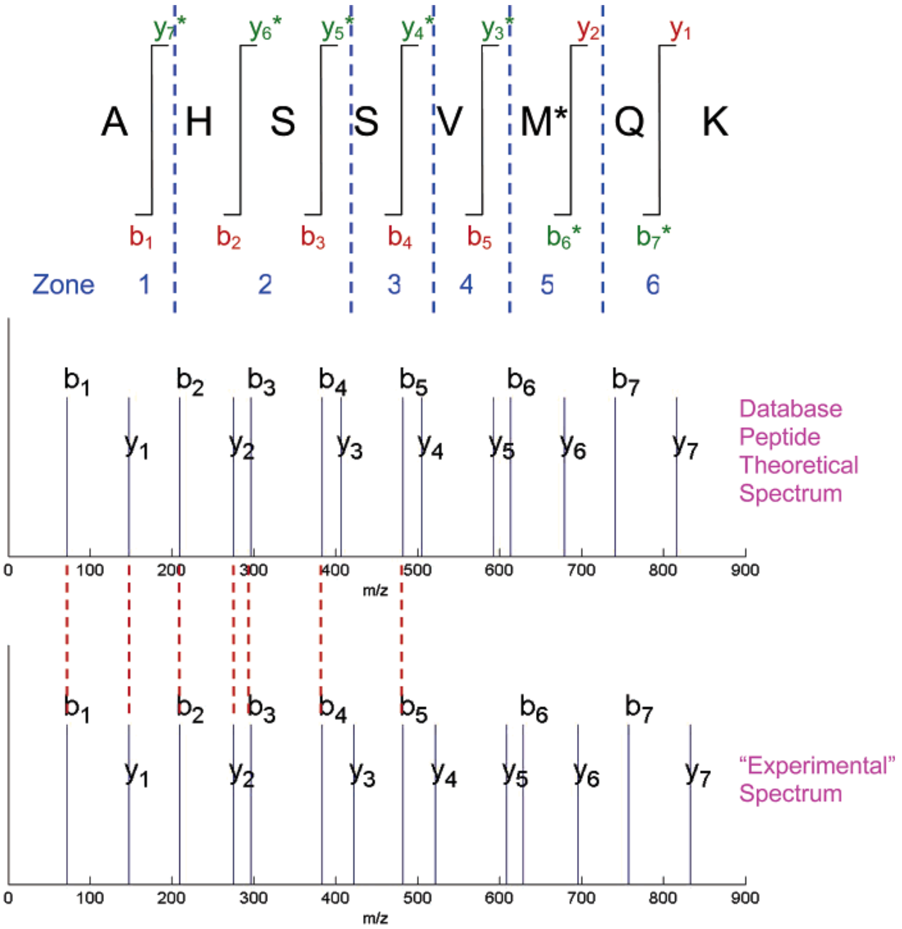


Figure 1. Example for illustrating zone modification searching. Note that, from a purely computational viewpoint, the zone numbers are a property of the experimental spectrum, but it is helpful conceptually to visualize where the zones would fall on the (modified) peptide sequence. The zone numbers are based on b ion mass, so zone numbers for y ion hypotheses are actually assigned based on the complement mass. See text for further details.

the theoretically expected b and y ions (and no other ions). We study the matching of the spectrum of the methionine-modified peptide AHSSVM*QK against the database digest peptide AHSSVMQK (unmodified). From Figure 1, we see that the spectral ion peaks b_1, b_2, b_3, b_4, b_5, y_1, and y_2 do not contain the modified methionine and thus match the database peptide fragment ions. The other spectral ion peaks b_6, b_7, y_3, y_4, y_5, y_6, and y_7 (marked by asterisks) contain the modified methionine and thus do not directly match the database peptide fragment ions; however, these spectral ion peaks can be matched indirectly by introducing the concept of ion complements.[31] For any query mass m_Q, define the complementary query mass $m_Q' = m + 2\, m_{\mathrm{H+}} - m_Q$, where $m_{\mathrm{H+}}$ is the mass of a proton and m is the precursor mass as defined earlier. The complement of the spectral (methionine-modified) b_6 ion matches the database y_2 ion, and the complement of the spectral (methionine-modified) b_7 ion matches the database y_1 ion. Similarly, for the spectral y ions, the complement of the spectral y_3 ion matches the database b_5 ion, the complement of the spectral y_4 ion matches the database b_4 ion, and so on. Thus, all spectral ion peaks match, either directly or indirectly (via complements), to the database peptide fragment ions.

We now discuss how the zone modification scoring procedure is applied to our hypothetical “experimental” MS/MS spectrum under the assumption that a modification occurs in zone 5 (as is the case for our “experiment”). We first match each “experimental” query mass under the hypothesis that the query mass arises from a b ion. Each query mass is assigned a zone number based on mass, and the type of matching performed depends on the zone number. In zones 1, 2, 3, and 4, b ions are not modified, so if a query mass falls in zone 1, 2, 3, or 4, it is matched against the database peptide b ions. In zone 6, b ions are modified, so if a query mass falls in zone 6, its complement is matched against the database peptide y ions. In zone 5, b ions may or may not be modified, so if a query mass falls in zone 5, both types of matching are performed: the query mass is matched against the database peptide b ions, and the query mass’s complement is matched against the database peptide y ions. We must also match each query mass under the hypothesis that the query mass arises from a y ion. In contrast to the b ion hypothesis, for the y ion hypothesis, the zone number of each query mass is assigned based on the complement mass (see Figure 1). The type of matching performed again depends on the zone number. In zone 6, y ions are not modified, so if a query mass falls in zone 6, it is matched against the database peptide y ions. In zones 1, 2, 3, and 4, y ions are modified, so if a query mass falls in zone 1, 2, 3, or

[31] The use of complements in database search can be traced to: Clauser, K. R.; Baker, P.; Burlingame, A. L. *Proceedings of the 44th ASMS Conference on Mass Spectrometry and Allied Topics*, Portland, OR, May 12−16, 1996.

4, its complement is matched against the database peptide b ions. In zone 5, y ions may or may not be modified, so if a query mass falls in zone 5, both types of matching are performed—the query mass is matched against the database peptide y ions, and the query mass's complement is matched against the database peptide b ions. In summary, for the scoring performed in this example, all the query masses from the modified peptide match, directly or indirectly, to the database peptide (unmodified) b and y ions. The scoring procedure is performed similarly for all the other zones. Note, however, that if we assume the modification occurs in any zone other than the correct zone (5, in this example), some of the query masses would not match, resulting in a lower score.

In a zone modification database search, for each MS/MS spectrum, the scoring procedure described in the example is applied to all of the database digest peptides whose mass m_p satisfies the relation $m_p + \Delta m = m$ (to within a user-specified MS tolerance), where Δm can assume any value within a range specified by the user. Similarly to the no-modification search, the digest peptide that yields the highest score is deemed to be the most likely answer for that MS/MS spectrum, though in the case of the zone modification search discussed here, there are two additional free parameters—namely, the delta mass Δm and the zone number. The delta mass Δm defines the modification to the peptide, and the zone defines the approximate location of the modification. While the zone modification search is more complex than the no-modification search, it is still very efficient since the six different zone searches can actually be done simultaneously with very little additional computation.[32]

A detailed description of the zone modification scoring algorithm is provided in Chart 4. Similar to the above discussion in the context of the no-modification search, all the "matching" of query masses is actually done via direct table lookups only, resulting in rapid scoring. In addition, for zone modification searches looking for a broad (continuous) range of delta masses, the inefficiency discussed above in the context of no-modification searches is eliminated or greatly mitigated. For no-modification searches, more peptides are scored than necessary because each peptide bin's molecular weight range is generally wider than the mass range delimited by (experimental precursor mass) ± (MS tolerance). On the other hand, for broad zone modification searches, the molecular weight range searched (as dictated by the experimental precursor mass together with the delta mass range the user wishes to investigate) spans several complete peptide bins, resulting in a possibility of wasted work only at the ends of the molecular weight range; thus, a lot less unnecessary computation is performed, and the algorithm is much more efficient. Below, in the Algorithm Efficiency subsection, we provide speed benchmarks demonstrating the tremendous efficiency of broad zone modification searches.

We note that the zone modification scoring technique is absolutely critical for effectively making use of a fully preindexed protein database in the case of modified peptides. To understand this, let us compare the zone modification search algorithm against other simpler but less effective algorithms. A naïve approach for using a preindexed database is to simply match all the query masses against the preindexed database, as is done in a no-modification search. (Call this algorithm I.) This method is ineffective for modified peptides because any fragment ion query mass containing a modified amino acid would not match—the preindexed protein database is not aware of modifications. A somewhat less naïve approach is to match all the query masses as well as all the complementary query masses against the protein database. (Call this algorithm II.) Under this approach, fragment ions containing a modified amino acid match via complementary query masses, while fragment ions containing no modified amino acids match query masses directly. However, half of the query masses/complementary query masses is expected to not match, as determined by the location of the modification in the amino acid sequence. Moreover, these extraneous query masses/complementary query masses raise the background "noise" level for the database search, where the "noise" comes from random matches that occur by chance. The zone modification search algorithm improves on algorithm II by intelligent selection of a subset of query masses/complementary query masses. Only those query masses/complementary query masses that are consistent with the location of the modification are selected for matching; there are no extraneous query masses/complementary query masses to raise the background noise level. In a comparison of the three algorithms on actual data, the zone modification search clearly outperforms algorithm I for modified peptides and also significantly outperforms algorithm II (more true positive peptide identifications, fewer false positive peptide identifications) due to the lower background noise level inherent to the zone modification search algorithm.

Peptide Confidences. We now discuss the probability estimate that the highest scoring peptide for any given MS/MS spectrum is actually a legitimate answer. For each MS/MS spectrum, some peptide will have the highest score, as long as at least one peptide is scored during the database search. However, to have any degree of confidence that this highest scoring peptide is indeed a legitimate answer, we require additional evidence.

We approached this problem empirically, by making use of an annotated data set in which the correct answer (or lack thereof) for each MS/MS spectrum is known with a reasonable degree of certainty. The data set is divided into two groups—those MS/MS spectra in which the highest scoring peptide is correct, and those MS/MS spectra in which the highest scoring peptide is not a legitimate answer. We screened a variety of metrics for effectiveness at discriminating between the two groups. Metrics investigated included (highest) score, various distance-based metrics, and delta mass, as well as linear combinations of metrics optimized via linear discriminant analysis. The metric that provides the best discrimination is a distance-based metric alone. We call this optimal metric the distance score, and it is defined as the difference between the highest score and the seventh highest score for each MS/MS spectrum. The distance score provides a measure of the separation between the highest scoring peptide and the "pack" of wrong peptides. The larger the distance score is, the larger the probability that the highest scoring peptide is indeed a legitimate answer. To actually compute probabilities, we

[32] For the no-modification search, for each peptide-spectrum scoring, a single number is accumulated as the score. For the zone modification search, 12 partial scores are accumulated (6 zones × 2 ion types (b, y)) during the course of the search, and at the very end, these partial scores are summed in various combinations to yield the 6 different zone search scores. This step at the very end takes a small amount of computation, and consequently, the zone modification search is only slightly slower than a no-modification search on the same search space.

Chart 4. Interrogator Pseudocode for Performing a Zone Modification Search on an MS/MS Spectrum[a]

```
Mass_H := 1.0078 Daltons
Num_Zones := 6
Zone_MW_Width := Precursor_MW / Num_Zones

FOR each Peptide_Bin where there exists at least one peptide and there exists Delta_Mass ...
... such that "theoretical" Peptide.MW() + Delta_Mass - "experimental" Precursor_MW is ...
... between -MS_Tolerance and +MS_Tolerance and Delta_Mass is in the range specified by the user
    INITIALIZE (2-dimensional) Partial_Zone_Score_B array to all 0's
    INITIALIZE (2-dimensional) Partial_Zone_Score_Y array to all 0's
    FOR each Query_Mass in the MS/MS spectrum
        B_Zone_Index := Query_Mass.Mass() / Zone_MW_Width
        Y_Zone_Index := Num_Zones - 1 - Query_Mass.Mass() / Zone_MW_Width
        Complementary_Query_Mass := Precursor_MW + 2*Mass_H - Query_Mass.Mass()

        Index_Low := Starts_B_Ions[(Query_Mass.Mass() - MS_MS_Tolerance) * 100]
        Index_High := Starts_B_Ions[(Query_Mass.Mass() + MS_MS_Tolerance) * 100]
        FOR each Index FROM Index_Low TO Index_High-1
            Peptide_Index := Peptide_Indices_B_Ions[Index]
            Partial_Zone_Score_B[B_Zone_Index][Peptide_Index] :=
                Partial_Zone_Score_B[B_Zone_Index][Peptide_Index] +
                Query_Mass.Weight_For_Scoring()
        END FOR
        Index_Low := Starts_Y_Ions[(Query_Mass.Mass() - MS_MS_Tolerance) * 100]
        Index_High := Starts_Y_Ions[(Query_Mass.Mass() + MS_MS_Tolerance) * 100]
        FOR each Index FROM Index_Low TO Index_High-1
            Peptide_Index := Peptide_Indices_Y_Ions[Index]
            Partial_Zone_Score_Y[Y_Zone_Index][Peptide_Index] :=
                Partial_Zone_Score_Y[Y_Zone_Index][Peptide_Index] +
                Query_Mass.Weight_For_Scoring()
        END FOR
        Tolerance := MS_Tolerance + MS_MS_Tolerance
        Index_Low := Starts_B_Ions[(Complementary_Query_Mass - Tolerance) * 100]
        Index_High := Starts_B_Ions[(Complementary_Query_Mass + Tolerance) * 100]
        FOR each Index FROM Index_Low TO Index_High-1
            Peptide_Index := Peptide_Indices_B_Ions[Index]
            Partial_Zone_Score_B[B_Zone_Index][Peptide_Index] :=
                Partial_Zone_Score_B[B_Zone_Index][Peptide_Index] +
                Query_Mass.Weight_For_Scoring()
        END FOR
        Tolerance := MS_Tolerance + MS_MS_Tolerance
        Index_Low := Starts_Y_Ions[(Complementary_Query_Mass - Tolerance) * 100]
        Index_High := Starts_Y_Ions[(Complementary_Query_Mass + Tolerance) * 100]
        FOR each Index FROM Index_Low TO Index_High-1
            Peptide_Index := Peptide_Indices_Y_Ions[Index]
            Partial_Zone_Score_Y[Y_Zone_Index][Peptide_Index] :=
                Partial_Zone_Score_Y[Y_Zone_Index][Peptide_Index] +
                Query_Mass.Weight_For_Scoring()
        END FOR
    END FOR
    Keep only those elements of the Score array where, for the corresponding Peptide, there ...
    ... exists Delta_Mass such that "theoretical" Peptide.MW() + Delta_Mass - ...
    ... "experimental" Precursor_MW is between -MS_Tolerance and +MS_Tolerance and Delta_Mass ...
    ... is in the range specified by the user

    FOR each Peptide_Index FROM 0 TO Num_Peptides-1
        FOR each Zone_Index FROM 1 TO Num_Zones-1
            Partial_Zone_Score_B[Zone_Index][Peptide_Index] :=
                Partial_Zone_Score_B[Zone_Index][Peptide_Index] +
                Partial_Zone_Score_B[Zone_Index-1][Peptide_Index]
        END FOR

        FOR each Zone_Index FROM Num_Zones-2 DOWNTO 0
            Partial_Zone_Score_Y[Zone_Index][Peptide_Index] :=
                Partial_Zone_Score_Y[Zone_Index][Peptide_Index] +
                Partial_Zone_Score_Y[Zone_Index+1][Peptide_Index]
        END FOR

        FOR each Zone_Index FROM 0 TO Num_Zones-1
            Zone_Score[Zone_Index][Peptide_Index] :=
                Partial_Zone_Score_B[Zone_Index][Peptide_Index] +
                Partial_Zone_Score_Y[Zone_Index][Peptide_Index]
        END FOR
    END FOR
END FOR
Merge results from different Peptide_Bins
High_Score := MAXIMUM of Zone_Score array
SAVE High_Score and corresponding Peptide, Zone_Index
```

[a] The pseudocode refers to the index table structure described in detail in Chart 1.

again turn to our annotated data set to provide an empirical model for estimating the probability. The distributions of distance scores for correct peptides and incorrect peptides were studied and found to be approximately Poisson. We also found that the distributions vary as a function of the number of peptides scored for a spectrum—for larger numbers of peptide scored, the distributions shift toward lower distance scores. Thus, we derived approximate Poisson distributions for correct and incorrect peptides as a function of distance score and number of peptides scored. These empirically derived distributions are used to estimate probability. A separate annotated data set has been used to verify that the probabilities computed by the model accurately reflect the observed probability of a correct result. In reporting, probabilities are multiplied by 100 to yield percent confidences.

EXPERIMENTAL METHODS

Simple Protein Mixture. The simple protein mixture was created by mixing together 24 proteins purchased from Sigma. The resulting sample was denatured, reduced with dithiothreitol, alkylated with iodoacetic acid, digested with trypsin, separated by reversed-phase liquid chromatography, and analyzed via electrospray ionization with an Applied Biosystems/MDS Sciex API QSTAR Pulsar Hybrid LC/MS/MS system using information-dependent acquisition.

Transthyretin from Patient Samples. Immunoprecipitation was used to isolate transthyretin from each patient's serum.[33−35] The samples were purified, digested with trypsin, and injected via electrospray ionization into an Applied Biosystems/MDS Sciex API QSTAR Pulsar Hybrid LC/MS/MS system. These experiments were performed by the Amyloid Treatment and Research Program and the Mass Spectrometry Resource at Boston University School of Medicine, and we subsequently analyzed a portion of their data using the Interrogator algorithm. Further experimental details are described in refs 34−36.

Human Heart Mitochondrial Protein Sample. A purified human heart mitochondrial protein sample was fractionated by sucrose gradient density centrifugation. Each fraction was reduced with TCEP, alkylated with iodoacetamide, digested with trypsin, separated by reversed-phase liquid chromatography, and injected via electrospray ionization into an Applied Biosystems/MDS Sciex API QSTAR Pulsar Hybrid LC/MS/MS system. This experiment was performed by Gaucher and co-workers, and we subsequently analyzed a portion of their data using the Interrogator software. Further experimental details are described in refs 37 and 38.

Table 1. Delta Masses Found the Most Number of Times in the Simple Protein Mixture Using the Zone Modification Algorithm[a]

delta mass	no. of times found	modification
+58	22	carboxymethylation of N-terminus
+44	18	carboxymethylation of methionine
+1	13	deamidation
−18	13	loss of water, or pyroglutamic acid conversion of N-terminal glutamic acid
−17	9	loss of ammonia, or pyroglutamic acid conversion of N-terminal glutamine
−48	8	decomposed carboxymethylated methionine

[a] The algorithm enables the discovery of common modifications without any prior knowledge.

RESULTS

Simple Protein Mixture. The collection of 1292 MS/MS spectra for the simple protein mixture has undergone extensive manual curation. Consequently, it is known with a reasonable degree of certainty which peptide identifications are accurate. For database searching, we specify trypsin as the digest agent, build carboxymethylation of cysteine into the Interrogator search database, and perform a zone modification search looking for modifications with delta mass in the range from −400 to +400 Da. We find that doing this modification-tolerant search results in a significantly higher number of MS/MS spectra being identified. Out of 473 correctly identified MS/MS scans, 158 scans correspond to modified peptides (i.e., have nonzero delta masses).

Table 1 lists the most frequently found modifications in the sample. We note that three of these frequently found modifications arise from unintentional side reactions with iodoacetic acid. During sample preparation, iodoacetic acid is mixed into the sample in order to carboxymethylate the cysteine residues. While the intended reaction between the cysteine residues and iodoacetic acid is nearly quantitative, the methionine residues and the N-terminus of each peptide have sufficient nucleophilic character to react with iodoacetic acid (albeit to a lesser degree), resulting in carboxymethylated methionines and carboxymethylated N-termini. These types of side reactions have been documented in the literature, both for the N-terminus[39] and for methionine.[40,41]

All of the frequently found modifications listed in Table 1 are readily explainable in retrospect. However, it may not be obvious a priori which modifications are expected to be the most abundant in the sample. It is interesting to note some modifications that are not on the list of frequently found modifications in the sample. Phosphorylation is not found at all in the sample, and oxidized methionine is only found rarely in the sample (once, to be precise), even though phosphorylation and oxidation of methionine are common modifications in general in many other samples. Thus, if the objective is to maximize the number of modified peptides identified, performing a zone modification search removes from the user the burden of carefully selecting (or guessing) which modifications are expected to be the most abundant in the sample,

[33] Lim, A.; Prokaeva, T.; McComb, M. E.; Connors, L. H.; Skinner, M.; Costello, C. E. *Protein Sci.* **2003**, *12*, 1775−1785.

[34] Lim, A.; Prokaeva, T.; McComb, M. E.; O'Connor, P. B.; Théberge, R.; Connors, L. H.; Skinner, M.; Costello, C. E. *Anal. Chem.* **2002**, *74*, 741−751.

[35] McComb, M. E.; Lim, A.; Prokaeva, T.; Connors, L. H.; Skinner, M.; Costello, C. E. *Proceedings of the 50th ASMS Conference on Mass Spectrometry and Allied Topics*, Orlando, FL, June 2−6, 2002.

[36] McComb, M. E.; et al., manuscript in preparation.

[37] Gaucher, S. P.; Taylor, S. W.; Fahy, E.; Zhang, B.; Warnock, D. E.; Ghosh, S. S.; Gibson, B. W. *J. Proteome Res.* **2004**, *3*, 495−505.

[38] Gaucher, S. P.; Nimkar, S.; Fahy, E.; Taylor, S. W.; Gibson, B. W.; Ghosh, S. S. *Proceedings of the 51st ASMS Conference on Mass Spectrometry and Allied Topics*, Montreal, PQ, Canada, June 8−12, 2003.

[39] Boja, E. S.; Fales, H. M. *Anal. Chem.* **2001**, *73*, 3576−3582.

[40] Gundlach, H. G.; Moore, S.; Stein, W. H. *J. Biol. Chem.* **1959**, *234*, 1761−1764.

[41] Lapko, V. N.; Smith, D. L.; Smith, J. B. *J. Mass Spectrom.* **2000**, *35*, 572−575.

Table 2. Semitryptic Peptides Discovered in the Simple Protein Mixture Using the Zone Modification Search[a]

delta mass	peptide
−381	FFS***ASCVPGATIEQK***
−380	***SELQDAIGSL***HSR
−362	***CACSNHEPYFGY***SGAF
−333	LGY***VLTCPSNLGTGLR***
−330	***GGDDLDPHYVL***SSR
−243	***SELQDAIGSLH***SR
−228	MP***CTEDYLSLILNR***
−228	VE***DIWSFLSK***
−227	AVG***KVIPELDGK***
−214	TL***GLYGKDQR***
−198	VV***SVLPIQHQDWLK***
−171	GN***PTVEVDLHTAK***
−101	T***GPNLHGLFGR***
−71	A***DDGRPFPQVIK***

[a] For each row in the table, the entire sequence represents the fully tryptic peptide reported by Interrogator (accompanied by a reported large, negative delta mass), while the bold portion of the sequence represents the semi-tryptic peptide which is actually in the sample. Note that the nonbold portion of the sequence has a mass equal in magnitude to the reported delta mass.

as would be the case if an ordinary modification-tolerant search were performed. Rather, a zone modification search over a broad delta mass range automatically accomplishes the goal without any need for a priori knowledge or guesswork.

Many of the modifications reported with large, negative delta masses correspond to semitryptic peptides (i.e., peptides in which one end follows trypsin's cleavage rules but the other end does not). Because we specify trypsin as the digest agent, all of the peptides in the preindexed protein database are fully tryptic (i.e., both ends of the peptide follow trypsin's cleavage rules). Thus, all of the reported peptides are fully tryptic. Converting a fully tryptic peptide into a semitryptic peptide involves clipping off a few amino acids from one end of the peptide. Hence, if the magnitude of the reported negative delta mass equals (to within the user-specified MS tolerance) the combined mass of amino acids from one end of the reported fully tryptic peptide and if the reported modification zone is at the same end, then the peptide is in reality a semitryptic peptide. The semitryptic peptides discovered are listed in Table 2. For each table entry, the entire sequence represents the reported fully tryptic peptide (accompanied by a reported large, negative delta mass), while the bold portion of the sequence represents the semitryptic peptide that is actually in the sample. We note that the nontrivial number of semitryptic peptides identified (a phenomenon we have observed in other samples as well) is inconsistent with a recent publication stating that trypsin digestion results in virtually no nonspecific cleavage.[42]

An interesting and initially puzzling peptide identified has the sequence MAALKDQLIHNLLK and a delta mass of −89 Da. None of the popular databases of delta masses[43] contain an entry for a delta mass of −89 Da. Noting that the peptide comes from the N-terminus of the protein L-lactate dehydrogenase A chain [rabbit] [gi:126050] provides a clue to the nature of the modified peptide. It turns out that there are actually two modifications on the peptide. The first modification, which has an associated delta mass of −131 Da, is loss of methionine (M) from the N-terminus. This cleavage of methionine off of a protein's N-terminus is a well-known and commonly occurring process, although the physiological importance is still not completely understood.[44] The second modification, which has an associated delta mass of +42 Da, is acetylation of the alanine (A) on the new N-terminus. N-terminal acetylation is also a well-known and commonly occurring process, although the physiological importance is also not well-understood.[44,45] Combining the delta mass of the two modifications (−131 and +42 Da) yields the observed delta mass of −89 Da. Zone modification searching is able to identify this doubly modified peptide because in this case, the two modifications are located close to each other on the peptide.

Transthyretin from Patient Samples. Transthyretin (TTR) is synthesized predominantly in the liver and secreted into plasma where it functions as a transport protein for the hormone thyroxine and for retinol (in association with retinol binding protein). Mutations in TTR may lead to familial transthyretin amyloidosis, characterized by deposition of abnormal, aggregated protein in tissues and organs. The main clinical symptoms are neuropathy, cardiomyopathy, and vitreous opacities, usually leading to death within 7−15 years following the onset of symptoms. Since the only effective treatment known to date is liver transplantation, accurate diagnosis is critical.[34,46,47] We discuss here two data sets obtained from the Amyloid Treatment and Research Program and the Mass Spectrometry Resource at Boston University School of Medicine as part of a collaborative project pertaining to the development of automated LC/MS/MS methods for the characterization of TTR mutations. Data were obtained from two patient samples, which have been previously characterized by MS via differential peptide mapping and by ESI-MS/MS[33,34,48] and more recently characterized by LC/MS/MS via the Pro ID software platform.[35,36] For database searching, we specify trypsin as the digest agent and perform a zone modification search looking for modifications with delta mass in the range from −400 to +400 Da. All of the modified peptides discussed in this section have also been verified by manual inspection of the corresponding MS/MS spectra. We also note that the majority of the peptides identified by the database search are unmodified, but the discussion here focuses on the modified peptides since the novel feature enabled by performing a zone modification search is modification discovery.

In the first patient sample, the zone modification search identifies the peptide AADDTWEPFASGK both in its unmodified form (delta mass of 0 Da) and in a modified form with a delta mass of −73 Da. This delta mass of −73 Da, with the modification being located in a zone near the TW residues, can be explained by a tryptophan (W) → leucine/isoleucine (L/I) mutation located at position 41 on human transthyretin. This particular mutation

[42] Olsen, J. V.; Ong, S.-E.; Mann, M. *Mol. Cell. Proteomics* **2004** *3*, 608−614.

[43] For example, see http://www.abrf.org/index.cfm/dm.home or http://www.unimod.org.

[44] Bradshaw, R. A.; Brickey, W. W.; Walker, K. W. *Trends Biochem. Sci.* **1998**, *23*, 263−267.

[45] Polevoda, B.; Sherman, F. *J. Biol. Chem.* **2000**, *275*, 36479−36482.

[46] Bergethon, P. R.; Sabin, T. D.; Lewis, D.; Simms, R. W.; Cohen, A. S.; Skinner, M. *Neurology* **1996**, *47*, 944−951.

[47] Pomfret, E. A.; Lewis, W. D.; Jenkins, R. L.; Bergethon, P.; Dubrey, S. W.; Reisinger, J.; Falk, R. H.; Skinner, M. *Transplantation* **1998**, *65*, 918−925.

[48] Théberge, R.; Connors, L.; Skinner, M.; Skare, J.; Costello, C. E. *Anal. Chem.* **1999**, *71*, 452−459.

(a) GPTGTGESKCPLMVKVLDAVRGSPAINVAVHVFRKAADDTWEPFASGKTSESG
ELHGLTTEEEFVEGIYKVEIDTKSYWK**ALGISPFHEHAEVVFTANDSGPRR**YTIA
ALLSPYSYSTTAVVTNPKE

(b) ALGISPFHEHAEVVFTANDSGPR**R**

⇩

RALGISPFHEHAEVVFTANDSGPR

Figure 2. (a) Amino acid sequence of transthyretin (TTR). The underlined tryptic peptide (with one missed cleavage) undergoes a rearrangement reaction. (b) Peptide rearrangement reaction, catalyzed by trypsin.

has been identified previously in the literature as causing amyloidosis.[34,49] Since the disorder is autosomal dominant (i.e., the disorder is expressed in those individuals who inherit only one copy of the mutation), it is not surprising that both the mutated and nonmutated forms of the proteins are present in the patient sample.

In the second patient sample, the peptide GSPAINVAVHVFR is identified in its unmodified form (delta mass of 0 Da) and in two modified forms with delta masses of +32 and of +48 Da, with the modifications being located in a zone near the AV residues. The delta mass of +32 Da corresponds to a valine (V) → methionine (M) mutation on the second of the three valines in the peptide (position 30 on human transthyretin).[50] This mutation has also been identified previously in the literature as being amyloidogenic.[49] The delta mass of +48 Da corresponds to the valine (V) → methionine (M) mutation (delta mass of +32 Da) accompanied by oxidation of the methionine residue (delta mass of +16 Da).

In addition to finding amino acid mutations, the zone modification search also discovers several other modifications in the second patient sample. The peptide CPLMVK is identified with a delta mass of −46 Da, which is consistent with loss of the side chain of the cysteine (C) residue. The peptide CPLMVKVLDAVR is identified with a delta mass of +80 Da, which can be explained by sulfonation of the cysteine (C) residue. Both of these cysteine modifications have been observed previously.[33,34,48,51]

The peptide ALGISPFHEHAEVVFTANDSGPR is identified with a delta mass of +156 Da, with the modification being located in a zone at the N-terminal end of the peptide. The delta mass of +156 Da matches the residue mass of arginine, suggesting that the actual peptide in the sample is RALGISPFHEHAEVVFTANDSGPR. However, starting from the intact protein, it is impossible to obtain this peptide by cleavage, even allowing for missed tryptic cleavages or nontryptic cleavages. (See Figure 2a.) We believe that RALGISPFHEHAEVVFTANDSGPR is actually present in the sample and arises from a trypsin-catalyzed rearrangement reaction, perhaps via a cyclic intermediate. The starting material for this rearrangement reaction is the peptide with one missed tryptic cleavage: ALGISPFHEHAEVVFTANDSGPRR. In the rearrangement reaction, an arginine (R) is clipped off of the C-terminus and attached to the N-terminus, yielding the rearranged peptide RALGISPFHEHAEVVFTANDSGPR. (See Figure 2b.) The clipping of arginine occurs due to regular tryptic cleavage, while the attachment of arginine at the other end can be understood as arising from the usual trypsin cleavage reaction proceeding in the reverse direction. Several examples of this type of rearrangement reaction have been reported previously,[52−54] and the MS/MS data here support this finding.

Table 3. Delta Masses Found the Most Number of Times for the Protein NADH Dehydrogenase (Ubiquinone) Fe−S Protein 1, 75 kDa [gi: 31377539] from the Human Heart Mitochondrial Sample Using the Zone Modification Algorithm[a]

delta mass	no. of times found	modification
+16	19	oxidation of methionine
+38	15	potassium ion adduct
+1	7	deamidation
+22	6	sodium ion adduct

[a] In addition to finding common modifications such as oxidation of methionine and deamidation, the algorithm enables the discovery that salt-adducted peptides are present in the sample.

Human Heart Mitochondrial Protein Sample. The data discussed here come from an extensive effort to create a catalog of the mitochondrial proteome. For database searching, we specify trypsin as the digest agent, build carboxamidomethylation of cysteine into the Interrogator search database, and perform a zone modification search looking for modifications with delta mass in the range from −400 to +400 Da.

In the pooled data resulting from combining together 12 LC/MS/MS runs, the mitochondrial protein with the largest number of peptides identified (213 total) is NADH dehydrogenase (ubiquinone) Fe−S protein 1, 75 kDa [gi: 31377539]. Table 3 lists the most common modifications identified for this protein. We note that two out of the four most commonly found modifications arise from salt adducts (sodium or potassium adducts). This trend of finding relatively high numbers of sodiated and potassiated peptides holds true for other proteins in the sample as well. This finding is also consistent with ref 38, in which a small subset of the MS/MS spectra from this mitochondrial protein sample was examined manually.

We can speculate as to why such a high number of sodiated and potassiated peptides are found. One hypothesis is that the desalting column used during sample preparation is not sufficiently efficient. Another hypothesis is that certain peptides bind salt so tightly that no amount of desalting will eliminate the salt adducts. Overall, the finding that such a large number of salt adducts are present in this sample has initiated further attention to sample preparation conditions.[55]

Peptide Discrimination. Peptide discrimination, the ability to distinguish correct peptide identifications from incorrect peptide identifications, is of vital importance in any automated or high-

[49] Connors, L. H.; Lim, A.; Prokaeva, T.; Roskens, V. A.; Costello, C. E. *Amyloid* **2003**, 160−184.

[50] The localization of the mutation to the specific valine residue is done by manual inspection of the data since the zone modification search only identifies a zone, or range of amino acids, in the vicinity of the modification (mutation) rather than a specific amino acid residue.

[51] Kishikawa, M.; Nakanishi, T.; Miyazaki, A.; Shimizu, A. *Amyloid* **1999**, *6*, 48−53.

[52] Lippincott, J.; Hess, E.; Apostol, I. *Anal. Biochem.* **1997**, *252*, 314−325.

[53] Witkowska, H. E.; Shaw, B.; Nimkar, S.; Hall, S. C.; Zittin-Potter, S.; Nerissian, A.; Faull, K.; Valentine, J. *Proceedings of the 51st ASMS Conference on Mass Spectrometry and Allied Topics*, Montreal, PQ, Canada, June 8−12, 2003.

[54] Fodor, S.; Zhang, Z. *Proceedings of the 51st ASMS Conference on Mass Spectrometry and Allied Topics*, Montreal, PQ, Canada, June 8−12, 2003.

[55] Gaucher, S. P., personal communication.

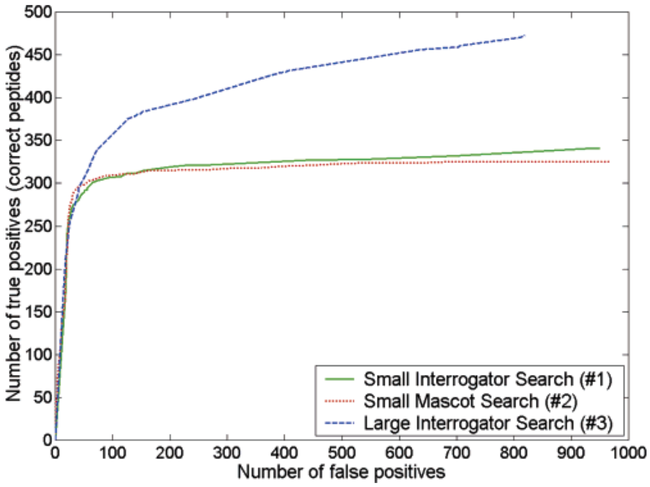


Figure 3. Counting ROC plots comparing three different database searches, all of which are performed on the same simple protein mixture data set. The small Interrogator search (1) is an Interrogator search with no zone modification searching (details: protein database is NR, digest agent is trypsin, allow one missed cleavage, carboxymethylation of cysteine is built into the Interrogator search database, MS tolerance 0.09 Da, MS/MS tolerance 0.04 Da). The small Mascot search (2) is the Mascot equivalent of the small Interrogator search (details: protein database is NR, digest agent is trypsin, allow one missed cleavage, carboxymethylation of cysteine is a variable modification, MS tolerance 0.09 Da, MS/MS tolerance 0.04 Da). The large Interrogator search (3) is an Interrogator search with zone modification searching looking for modifications in the range from −400 to +400 Da (details: protein database is NR, digest agent is trypsin, allow one missed cleavage, carboxymethylation of cysteine is built into the Interrogator search database, MS tolerance 0.09 Da, MS/MS tolerance 0.04 Da).

throughput protein identification experiment. To evaluate peptide discrimination, we make use of the data from the simple protein mixture. This data set has been subjected to extensive manual curation, meaning that we know which peptide identifications are correct and which are incorrect. We use this data set to evaluate Interrogator's discrimination ability as well as to compare the discrimination ability of Interrogator versus another commonly used database search engine, Mascot.[3]

A common method for visualizing and comparing discrimination ability is the receiver operating characteristic plot (ROC plot),[56] in which one can read off the false positive level that must be tolerated in order to obtain any given true positive level. In our case, false positives are those peptides deemed to be correct by virtue of surpassing a threshold but are in fact incorrect, as determined by manual curation; similarly, true positives are those peptides deemed to be correct by virtue of surpassing a threshold and are indeed correct, as determined by manual curation. In a standard ROC plot, the true positive fraction is plotted versus the false positive fraction. The more closely a plot approaches the upper left corner, the better the discrimination ability of the corresponding database search. We modify the ROC plot slightly to a counting ROC plot, in which the number of true positives is graphed versus the number of false positives. Figure 3 shows counting ROC plots comparing three different database searches, all of which are performed on the same simple protein mixture data set.

Small Interrogator search (1): Interrogator search with no zone modification searching.

Small Mascot search (2): Mascot equivalent of small Interrogator search (1).

Large Interrogator search (3): Interrogator search with zone modification searching looking for modifications in the range from −400 to +400 Da.

The small Interrogator search (1) and the small Mascot search (2) represent equivalent searches using Interrogator and Mascot. The plots suggest that the peptide discrimination ability of the two search engines is very similar—in comparing these two searches, we see that the Mascot search is slightly more discriminant (the plot for Mascot is slightly closer to the upper left corner than the corresponding plot for Interrogator), while the Interrogator search correctly identifies a few more peptides (the plot for Interrogator extends slightly higher than the plot for Mascot). More detailed studies comparing the two search engines on a variety of pairs of equivalent searches also suggest that the peptide discrimination ability of the two search engines is very similar.[57,58]

The large Interrogator search (3) represents an Interrogator zone modification search covering a broad range of candidate modifications (−400 to +400 Da). Figure 3 shows that the zone modification search identifies significantly more peptides than either the small Interrogator search (1) or the small Mascot search (2). Furthermore, the number of additional peptides identified by making use of the zone modification feature is much greater than the relatively minor differences between the two small searches.

Algorithm Efficiency. Table 4 shows the computer time required to complete each of the database searches shown in Figure 3, and Table 5 shows the speed of each of these database searches in units of number of spectrum-sequence comparisons performed per second. In comparing equivalent small searches (1 versus 2), we see that the extensive preindexing done by Interrogator enables it to perform the search several times faster than Mascot. More detailed studies show that this trend holds for a variety of pairs of equivalent searches.[57,58]

Table 5 also shows that the Interrogator zone modification search looking for a broad range of candidate modifications runs much faster (by 100-fold, or 2 orders of magnitude) than Mascot. This reflects the synergy between two of the novel features of the Interrogator algorithm: extensive preindexing of the protein database and zone modification searching. While extensive preindexing alone results in efficiency gains, as can be seen by comparing the two equivalent small searches (1 versus 2), the combination of extensive preindexing and zone modification searching results in even greater efficiency gains and enables vast search space coverage in a reasonable amount of computer time. The computer time required to complete the large Interrogator search (3) is comparable to the computer time required to perform a Mascot modification-tolerant search looking for nine variable modifications (the maximum number allowed by the software) on tryptic peptides, but the large Interrogator search covers a

[56] Swets, J. A. *Science* **1988**, *240*, 1285−1293.

[57] Seymour, S. L.; Loboda, A.; Rajagopalan, B.; Tang, W. H.; Shilov, I. V.; Patel, A. A.; Phu, L. M.; Hunter, C. L.; Nuwaysir, L. M.; Schaeffer, D. A. *ABRF 2004: Integrating Technologies in Proteomics and Genomics*; Portland, OR, February 28−March 2, 2004; Poster P201-T.

[58] Seymour, S. L.; et al., unpublished results.

Table 4. Computer Time Required for Three Different Database Search Types on Four Different Computer/Cluster Configurations[a]

database search	search time			
	on 10-blade server[b]	on 1-blade server[c]	on desktop cluster[d]	on laptop[e]
small Interrogator search (1)[f]	6 s (0.1 min)	274 s (4.6 min)	272 s (4.5 min)	639 s (10.7 min)
small Mascot search (2)[f]	84 s (1.4 min)	671 s (11.2 min)	921 s (15.4 min)	1217 s (20.3 min)
large Interrogator search (3)[f]	617 s (10.3 min)	6570 s (109.5 min)	6761 s (112.7 min)	9686 s (161.4 min)

[a] All searches are performed on the same simple protein mixture data set against the same protein database, NR. (Note: When the same search is run repeatedly, the second search runs more quickly than the first search due to computer memory caching effects. This holds true for all three database searches on all four machine configurations. The search times reported in the table are for the second search; all subsequent searches have run times approximately equal to the run time of the second search.) [b] IBM eServer BladeCenter with 10 HS20 blades (each blade has two Intel Xeon 2.4 GHz processors and 2 GB RAM) [c] IBM eServer BladeCenter with 1 HS20 blade (each blade has two Intel Xeon 2.4 GHz processors and 2 GB RAM) [d] Cluster made by linking together 5 older-model Dell desktop machines (average: Intel Pentium II 410 MHz processor, 640 MB RAM) [e] Dell Latitude C840 laptop with a single Intel Mobile Pentium 4 2.4 GHz processor and 1 GB RAM [f] See Figure 3 caption for details about each of the searches. Briefly, small Interrogator search (1) is an Interrogator search with no zone modification searching, small Mascot search (2) is the Mascot equivalent of the small Interrogator search, and large Interrogator search (3) is an Interrogator search with zone modification searching in the range −400 to +400 Da.

Table 5. Speed of Three Different Database Search Types on Four Different Computer/Cluster Configurations in Units of Number of Spectrum-Sequence Comparisons Performed Per Second[a]

database search	search speed (no. of spectrum-sequence comparisons/s)			
	on 10-blade server[b]	on 1-blade server[b]	on desktop cluster[b]	on laptop[b]
small Interrogator search (1)[c]	2.8×10^6	6.1×10^4	6.1×10^4	2.6×10^4
small Mascot search (2)[c]	2.0×10^5	2.5×10^4	1.8×10^4	1.4×10^4
large Interrogator search (3)[c]	4.2×10^7	4.0×10^6	3.9×10^6	2.7×10^6

[a] All searches are performed on the same simple protein mixture data set against the same protein database, NR. [b] See Table 4 for details about each of the computer configurations. [c] See Figure 3 caption for details about each of the searches. Briefly, small Interrogator search (1) is an Interrogator search with no zone modification searching, small Mascot search (2) is the Mascot equivalent of the small Interrogator search, and large Interrogator search (3) is an Interrogator search with zone modification searching in the range −400 to +400 Da.

much larger search space and enables identification of many more modification species.

Table 6 shows search times for broad zone modification searches on a single-CPU laptop for all of the samples discussed in this paper. With modest computational resources, a broad Interrogator zone modification search can be run in a time comparable to the instrument time required to collect the data. We believe that this capability has exciting implications for high-throughput proteomics. Historically, scientists have not routinely looked for a broad range of peptide modifications in their MS/MS data. We speculate that, using traditional database search engines or other protein identification software, the cost in computer time to perform such comprehensive searches has been prohibitive (usually several days, if at all possible). The efficiency of the zone modification search algorithm overcomes this barrier and enables scientists to systematically mine MS/MS data for a broad range of modifications on a routine basis.

DISCUSSION

The most striking feature of zone modification searching is the ability to rapidly find a peptide with a modification of arbitrary delta mass (provided that the delta mass is somewhere within the range specified by the user). This is in contrast with ordinary modification-tolerant searching, where a modified peptide can be found only if the modification is explicitly specified in advance by the user. Furthermore, in ordinary modification-tolerant searching, each prespecified modification leads to a significant increase in search time; hence, for all practical purposes, only a limited number of modifications can be tried. On the other hand, the high efficiency of zone modification searching allows the user to look for a very large variety of different modifications all at once. Thus, the zone modification search allows the scientist to explore a much broader landscape compared to ordinary database searching.

A limitation of zone modification searching is that the algorithm is designed to only look for a single modification on a peptide. However, under certain conditions, two or more modifications can be found by zone modification searching. If all the modifications occur in the same zone (or in nearby zones), a zone modification search can in fact identify the multiply modified peptide. An example was discussed above for the simple protein mixture. By contrast, ordinary modification-tolerant searching can readily find multiple modifications regardless of location on the peptide.

There is a slight difference in sensitivity between zone modification searching and ordinary modification-tolerant searching. It has been observed empirically that ordinary modification-tolerant searching is somewhat more sensitive than zone modification searching—that is, ordinary modification-tolerant searching is capable of detecting a few more modified peptides than zone modification searching.

Table 6. Search Times on a Laptop (Dell Latitude C840 with a Single Intel Mobile Pentium 4 2.4 GHz Processor and 1 GB RAM) for Broad Zone Modification Searches Looking for Modifications with Delta Mass in the Range from −400 to +400 Da

data set	no. of spectra in data set	protein database searched	no. of proteins in protein database	no. of amino acid residues in protein database	search time	search time/ spectrum
simple protein mixture	1292	NR[a]	1 480 858	476 119 222	9686 s (161.4 min)	7.5 s
TTR patient 1	560	NR[a]	1 480 858	476 119 222	4648 s (77.5 min)	8.3 s
TTR patient 2	449	NR[a]	1 480 858	476 119 222	3816 s (63.6 min)	8.5 s
human heart mitochondrial sample[e]	8539	NR[a]	1 480 858	476 119 222	66672 s (1111.2 min)	7.8 s
simple protein mixture	1292	Swiss-Prot[b]	127 873	46 856 345	1333 s (22.2 min)	1.0 s
TTR patient 1	560	Swiss-Prot[b]	127 873	46 856 345	642 s (10.7 min)	1.1 s
TTR patient 2	449	Swiss-Prot[b]	127 873	46 856 345	534 s (8.9 min)	1.2 s
human heart mitochondria sample[e]	8539	Swiss-Prot[b]	127 873	46 856 345	8554 s (142.6 min)	1.0 s
simple protein mixture	1292	yeast[c,d]	6 298	2 974 038	133 s (2.2 min)	0.10 s
TTR patient 1	560	yeast[c,d]	6 298	2 974 038	58 s (0.97 min)	0.10 s
TTR patient 2	449	yeast[c,d]	6 298	2 974 038	47 s (0.78 min)	0.10 s
human heart mitochondria sample[e]	8539	yeast[c,d]	6 298	2 974 038	825 s (13.8 min)	0.097 s

[a] Downloaded from NCBI (ftp://ftp.ncbi.nlm.nih.gov/blast/db), dated July 11, 2003. [b] Downloaded from NCBI (ftp://ftp.ncbi.nlm.nih.gov/blast/db), dated June 4, 2003. [c] Downloaded from NCBI (ftp://ftp.ncbi.nlm.nih.gov/blast/db), dated November 26, 2003. [d] This search against the yeast database does not make sense biologically because the proteins in the samples are not from yeast, but it is instructive to have search times against a small database such as yeast. [e] Combination of 12 LC/MS/MS runs.

Table 7. Comparison of Ordinary Modification-Tolerant Database Searching versus Zone Modification Searching

ordinary modification-tolerant database searching[a]	zone modification searching
modifications must be explicitly specified by user in advance	can find unanticipated modification (no need to specify particular modification in advance); the modification can be of arbitrary delta mass
can find multiple modifications on a peptide	designed to find one modification on a peptide (but can find multiple modifications if all the modified residues are close together in the sequence)
expensive in computer time → can only look for a limited number of modifications	efficient → can look for a very large variety of different modifications
slightly more sensitive	slightly less sensitive

[a] For Interrogator, this is accomplished by building prespecified modifications into the Interrogator search database

The advantages and disadvantages of zone modification searching versus ordinary modification-tolerant searching are summarized in Table 7. The Interrogator software is capable of performing both types of searches, individually or in conjunction. (Zone modification searching is a novel feature unique to the Interrogator software, while ordinary modification-tolerant searching is accomplished in the Interrogator software by building prespecified modifications into the Interrogator search database; see the algorithm description above for further details.)

It is instructive to also compare the zone modification search algorithm versus some other algorithms for identifying modified peptides. The error-tolerant sequence tag searching method of Mann and Wilm[8] relies on sequence tags, which consist of three regions. The central region consists of a partial peptide sequence derived from a run of consecutive MS/MS fragment ions, and the two end regions consist of the N-terminal and C-terminal masses flanking the partial peptide sequence. In matching a sequence tag against a protein database, the error-tolerant search allows for one mismatch in one region. Since the sequence tag is derived from consecutive fragment ions, the error-tolerant sequence tag search can only make use of information from consecutive fragment ions with the possible exception of a one amino acid long gap. By contrast, the zone modification search can make use of information from all fragment ions regardless of their proximity to one another. Thus, we expect the zone modification search to give improved peptide identification results (more true positives, less false positives) compared to the error-tolerant sequence tag search. We also compare zone modification search efficiency versus sequence tag search efficiency. GutenTag is the only fully automated sequence tag search software we are aware of[59] and requires ~1 s/spectrum on a single-CPU computer (AMD Athlon XP 1700+) for searches against the yeast database (~6500 proteins).[10] Table 6 shows that a broad zone modification search is ~1 order of magnitude faster, requiring ~0.1 s/spectrum for searches against the yeast database.

We also compare the zone modification search algorithm versus the spectral alignment algorithm of Pevzner et al.[22,23] The spectral alignment algorithm is capable of solving a more general problem than the zone modification search algorithm—the spectral alignment algorithm can identify peptides containing an arbitrary

(59) For most sequence tag search software, the process of calling sequence tags is done by manual inspection.

number of modifications (of arbitrary delta mass), while the zone modification search algorithm is designed for finding singly modified peptides (of arbitrary delta mass). However, we expect the zone modification search algorithm to be much faster than the spectral alignment algorithm due to the extensive preindexing used in the zone modification search. Consider the amount of computation required to score a spectrum versus a peptide sequence where no fragment ions match. During a database search, this scenario should be one of the most common scenarios since an incorrect spectrum-sequence pair only has ion matches that arise by chance and thus should be low in number (and far more incorrect spectrum-sequence pairs than correct spectrum-sequence pairs are encountered during the course of a database search). For a zone modification search, the scoring for this type of spectrum-sequence pair takes no time except for a small amount of bookkeeping at the end of the search. During the lookup process using the preindexed protein database, there are no references pointing to the spectrum-sequence pair, and thus, no computation takes place. More generally, the time spent scoring a spectrum-sequence pair is proportional to the number of actual fragment ion matches, not the number of potential ion matches checked. We analyzed an Interrogator search to explore the impact of this efficiency and found that 60% of spectrum-sequence comparisons produced no fragment ion matches. Another 39% of spectrum-sequence comparisons produced between one and five fragment ion matches. By contrast, the spectral alignment algorithm must perform a full alignment on every spectrum-sequence pair. This alignment requires computational time proportional to the square of the number of theoretical or observed ions, regardless of how many fragment ion matches are eventually found. Another advantage of the zone modification search algorithm compared to the spectral alignment algorithm is that the score computed by the zone modification search algorithm is more representative of the complexities of MS/MS fragmentation. The spectral alignment algorithm sorts MS/MS fragment ions in a single list and completely disregards the distinction between b and y ions. In particular, the complement relation between b and y ions is lost. By contrast, the zone modification search algorithm carefully accounts for the relationship between b and y ions and, as discussed above in the Algorithm section, accounting for this complexity improves the performance of the algorithm. Pevzner et al. did recognize this deficiency and suggested a possible approach for addressing this deficiency (by doing something similar to the antisymmetric longest path approach for solving the de novo sequencing problem[60]) but, as far as we know, accounting for b/y complementarity in the spectral alignment algorithm is still an unsolved problem.

Next, we provide some guidelines for using the zone modification search. We recommend that for modifications which are prevalent in the sample, the modifications should be built into the Interrogator search database (that is, ordinary modification-tolerant searching should be used in looking for those modifications). For example, if the sample preparation procedure includes a step in which cysteine residues are carboxymethylated via reaction with iodoacetic acid, we recommend that the carboxymethylation modification of cysteine be built into the Interrogator search database. While this results in larger Interrogator search databases and longer search times, there are a number of advantages to doing an ordinary modification-tolerant search rather than a zone modification search, and some of these advantages are particularly pronounced when the modification is prevalent in the sample. To continue with the example of carboxymethylated cysteines, we note that the relatively high occurrence of cysteine implies that a significant fraction of peptides contain at least one carboxymethylated cysteine residue, and a nontrivial number of peptides contain two or more carboxymethylated cysteine residues. The only search that can reliably identify peptides containing two or more carboxymethylated cysteines is ordinary modification-tolerant searching (i.e., build the carboxymethylation modification of cysteine into the Interrogator search database), as the zone modification search algorithm is not designed to detect multiply modified peptides. Furthermore, in the event that the user chooses to perform a zone modification search, the zone modification search is much more effective if the two types of modification-tolerant searching are used in conjunction—account for the carboxymethylation modification of cysteine by building the modification into the Interrogator search database and look for other modifications using the zone modification search. If the zone modification search is performed without building the carboxymethylation modification of cysteine into the Interrogator search database, in general it would not be possible to find peptides with carboxymethylated cysteine along with another modification because the zone modification search algorithm is not designed to find doubly modified peptides, resulting in missed identifications. On the other hand, if the carboxymethylation modification of cysteine is built into the Interrogator search database, carboxymethylated cysteine is accounted for automatically and thus does not count as a modification from the perspective of the zone modification search, making it possible for the zone modification search to find another modification on a peptide known to contain carboxymethylated cysteine.

We recommend that zone modification searching be used as a tool for modification "fishing." That is, we recommend that zone modification searching be used for the discovery of unknown modifications of unknown delta mass or for trying out a large variety of possible modifications, each of which the user expects to be relatively rare individually (though collectively the modifications could be large in number). This approach contrasts sharply with ordinary modification-tolerant database searching, in which the user is forced to select a limited number of candidate modifications based on prior knowledge or guesswork. The tremendous efficiency of zone modification searching allows the user to test out all possible modifications within a delta mass range (subject to the limitations discussed above and summarized in Table 7).

Once a modified peptide (i.e., a peptide with nonzero delta mass) has been identified, understanding the nature of the modification requires additional detective work. This delta mass rationalization is a challenge faced by many modification-tolerant peptide identification software packages[8−10,12,20−23] and is a task

[60] Dancik, V. International Patent Application WO 99/62930, December 9, 1999.

[61] Zhang, Z. *Anal. Chem.* **2004**, *76*, 3908−3922.

[62] Kapp, E. A.; Schutz, F.; Reid, G. E.; Eddes, J. S.; Moritz, R. L.; O'Hair, R. A. J.; Speed, T. P.; Simpson, R. *J. Anal. Chem.* **2003**, *75*, 6251−6264.

[63] Huang, Y.; Triscari, J. M.; Pasa-Tolic, L.; Anderson, G. A.; Lipton, M. S.; Smith, R. D., Wysocki, V. H. *J. Am. Chem. Soc.* **2004**, *126*, 3034−3035.

often left to the scientist to perform by hand. Tables listing modifications together with their associated delta masses are useful resources.[43] Interrogator offers some support for delta mass rationalization by offering an option in which the software suggests possible modifications that are consistent with the delta mass and the amino acid sequence. OpenSea[21] improves on this by also rescoring accounting for the modification, which often allows OpenSea to localize the modification to a specific amino acid.

CONCLUSIONS

We have presented here a novel algorithm for the rapid and automated identification of peptides with known as well as unanticipated modifications. The algorithm facilitates discovery of the unexpected, and we have illustrated this with a series of examples.

A survey of the mass spectrometry proteomics literature quickly reveals that performing a database search to identify proteins is a routine practice but that performing a database search that comprehensively looks for a broad variety of modifications is quite rare. We hypothesize that the reason for this observation is that such comprehensive searches, while undoubtedly informative, require more computing power than most scientists have at their disposal. The efficiency of our modification-tolerant search algorithm offers a possible solution to this dilemma by enabling the scientist to perform a comprehensive modification-tolerant search in a time comparable to the instrument time required for collecting the data.

ACKNOWLEDGMENT

We thank the Amyloid Treatment and Research Program and the Mass Spectrometry Resource at Boston University School of Medicine for providing the TTR data and helpful discussion. We thank Dr. Sara Gaucher and co-workers at the Buck Institute for Age Research and at MitoKor for providing the human heart mitochondrial data and helpful discussion. We are indebted to our former and current colleagues in the mass spectrometry software, applications, and marketing groups for their invaluable advice, assistance, and support.

Received for review December 22, 2004. Accepted March 25, 2005.

AC0481046