MS2DB: A Mass-Based Hashing Algorithm for the Identification of Disulfide Linkage Patterns in Protein Utilizing Mass Spectrometric Data

Timothy Lee, Rahul Singh
Department of Computer Science, San Francisco State University, 1600 Holloway Avenue, San Francisco, CA 94132-4025,U.S.A
timtlee@sfsu.edu, rsingh@cs.sfsu.edu

Ten-Yang Yen, Bruce Macher
Department of Chemistry and Biochemistry, San Francisco State University, 1600 Holloway Avenue, San Francisco, CA 94132-4025, U.S.A
ryen@sfsu.edu, macher@sfsu.edu

Abstract

The tertiary structure and biological function of a protein can be better understood given knowledge of the number and location of its disulfide bonds. By utilizing mass spectrometric (MS) experimental procedures that produce spectra of the protein's peptides joined by a disulfide bond, we can make initial identifications of these bonded cysteine pairings. The algorithmic problem then becomes how to match a theoretical mass space of all possible bonded peptides against the MS data. Our solution, MSHashID, utilizes the Expected Amino Acid Mass in combination with a hash structure to improve the time complexity of making an identification from worse than $O(n^2)$ to approximately $O(n)$, where n is the size of the mass space. We have developed a software package, MS2DB, which includes an implementation of this algorithm. Experiments using published data show that the MSHashID algorithm efficiently makes the correct initial identifications, which can then be confirmed using tandem mass spectrometry (MS/MS).

1. Introduction

Disulfide bonds occur in proteins when the sulfhydryl groups of cysteine residues become oxidized), forming a covalent bond derived from the coupling of thiol groups (S-H + S-H → S-S + 2H). As a result, residues that may be far removed from one another in the primary structure of the protein may in fact be physically cross-linked. Therefore, knowledge of the location of these bonds significantly contributes to our understanding of the protein's tertiary structure and function. For example, Angata *el al.* [1] has shown that the disulfide bond structures of ST8Sia IV are necessary for its polysialyation activity.

Various computational methods have also been developed to predict disulfide bonds based on the location of the cysteine residues in the protein's primary structure. The prediction accuracies of these methods are limited to ~60%. More fundamentally, Vullo *et al.* [2] showed that any prediction algorithm is computationally limited to proteins with only a few disulfide bonds. Ultimately, purely predictive methods must be complemented by an experimental methodology.

A promising approach is to apply mass spectrometric methodologies (MS) to determine disulfide bond structures [3]. Currently, MS-based approaches are widely used to solve protein identification problems, offering high throughput when used in conjunction with efficient data analysis algorithms. Existing web-based programs such as MS-Bridge in the ProteinProspector tools [4], the X! Protein Disulphide Linkage Modeler [5], and Peptidemap [6] are useful when analyzing MS data from MALDI-TOF (Matrix Assisted Laser Desorption Ionization- Time of Flight) experiments.

In the MS2DB project, we present algorithms for the mining and identification of disulfide bonds in proteins utilizing data obtained from liquid chromatography electrospray ionization tandem mass spectrometry (LC/ESI-MS/MS) experiments. As discussed in [7], this methodology yields highly specific structural fragments derived from the disulfide bonded pair of peptides. In [8], we describe how a tandem mass spectrum (MS/MS) of these fragments is utilized to confirm an initial identification of a disulfide bond. In this paper, our challenge is to develop algorithms that can accurately and efficiently process the MS data for an entire protein to make these initial identifications of its disulfide bond structures.

2. Computational Formulation

Let A denote the set of amino acid residues, each with mass m(a), a $\in$ A. A *peptide* $p = \{a_i\}$ is a string of amino acids with mass $m(p) = \sum m(a_i) + 18$ Daltons. The 18 Da. is included in this formula to account for the masses of H and OH of the the N- and C-termini of the peptide, respectively. A *disulfide bonded peptide* P_{12} is a pair of peptides p_1 and p_2, with mass $m(P_{12}) = m(p_1) + m(p_2) - 2m(H)$. For there to be a disulfide bond between p_1 and p_2, each peptide must contain at least one cysteine. Thus if $C = \{c_i\}$ denotes the set of all cysteine-containing peptides and $P = \{p_i\}$ denotes the set of all peptides, then $C \subseteq P$. In practice, it is very rare that C = P.

The *disulfide bond mass space* $D = \{d_i\}$ of a protein is the set of every possible pair of cysteine-containing peptides. A *mass spectrum* $S = \{s_j\}$ is the set of numbers obtained from a mass spectroscopy experiment that represent the masses of peptides, as well as chemical noise. The *match* $M = \{m_{ij}\}$ between D and S occurs between d_i and s_j when $|d_i - s_j| < m_t$, where m_t is defined as the *mass tolerance*, the ± amount of experimental uncertainty that s_j is allowed to have to determine the match. Each m_{ij} is then considered an *identified disulfide bond*. The **MS disulfide bond identification problem** can then be formulated as follows: given a mass space D and a spectrum S, find the set of matches $M = \{m_{ij}\}$.

3. Algorithmic solution

The construction of the mass space requires $O(k^2)$ time for one protein, where k is the number of sites at which the protein can be cleaved with a certain protease. This is because the k proteolytic amino acids divide the protein A into k+1 subsequences, leading to k(k+1)/2 unique pairs of subsequences that can be formed. For the case of disulfide bonds, we are concerned with forming unique pairs of subsequences from C as opposed to P. Because $C \subset P$ for almost all proteins and proteases, the disulfide bond mass space D is likely to be smaller than the mass space obtained from P. We generated D and P for the three proteins listed in Table 1, and found that the average reduction is mass space was 86%.

The quadratic time complexity can be further reduced if the data structure used to construct and search D did not require computing the mass of every member of D. Only in the final search phase would our algorithm compute the masses of the possible disulfide bonded peptides that are expected to be close in value S. This can be done by use of the *expected amino acid mass*, as defined below:

> DEFINITION 1. *Expected Amino Acid Mass* m_e. The weighted mean of A, i.e, $m_e = \sum w_i m(a_i)$, where $\{w_i\}$ denotes the relative abundance of each amino acid. Using the masses and relative abundances given in [9], we obtain $m_e = 111.17$ Da.

Using this definition, we can predict that the mass of a peptide $m(p) \approx ||p|| \, m_e + 18$, where ||p|| represents the number of amino acids contained in the peptide. The additional 18 Da. was explained in Section 2. Statistically, this is justified to a first approximation because the weighted standard deviation, again using the data in [9], is 28.86 Da. This observation leads to a second observation, that $||d_i||$ can be used to construct D in such a

way that it is approximately mass sorted. This is the motivation for exploring the use of a hash table to construct and search D.

3.1. Analysis of hashing based disulfide bond identification algorithm

The hash table is a well known data structure for efficient searching of a data space [10]. If the hash function employed satisfies the assumption of simple uniform hashing, then the expected time to search for an element is $O(1)$. Simple uniform hashing means that, given a hash table T, with ||T|| buckets, any data element d_i is equally likely to hash into any bucket, independently of where any other element has hashed to. Using the predicted mass of a peptide described in the previous section, we implement the simple hashing function $h(||d_i||) = ||d_i||$ as a first approximation.

To explore the efficacy of this choice of hashing function, we make the following definition:

> DEFINITION 2. *Expected Number of Amino Acids, e.* Given a mass spectrum value s_i, the expected number of amino acids $e_i = s_i / m_e$, rounded to the nearest integer.

For the Mouse Core 2 1,6-N-Acetyl glucosaminyltransferase I protein (hereafter referred to as "C2GnT-I") with 11 cysteines, the size of the disulfide bond mass space ||D|| = 55 (||c||(||c||-1)/2, where c is the number of cysteine residues). Each possible disulfide bonded peptide pair d_i is then placed into a bucket based on the simple hashing function described above. Then a simulated mass spectrum is generated by computing the mass of all 55 pairs, and dividing each by m_e to obtain 55 e_i to query the hash table.

Our results show that 23 of the 55 e_i index immediately into the correct bucket. The other 32 queries require a search of neighboring buckets. The farthest search to find its match is V = 3 buckets. We believe that this justifies the use of the simple hashing function $h(||d_i||) = ||d_i||$, to a first approximation.

3.2. Consideration of missed cleavages and intramolecular bonded cysteines

In the laboratory, a protease used to digest a protein will sometimes miss a cleavage point. For example, a protein with sequence NRDKTA should be digested by trypsin into three peptides: NR, DK, and TA. However, if one cleavage point is missed, two peptides are created: either NRDK and TA, or NR and DKTA. Our software models this behavior by including the parameter m_{max}, the maximum number of missed cleavages allowed.

It can be inferred by induction that a protein with k cleavage sites and a $m_{max} = m$ will digest into (m + 1)k unique peptides, assuming $k >> m$. Note that m_{max} includes all smaller values of missed cleavage levels, e.g., $m_{max} = 2$ includes m = 1 and m = 0 as well. If m_{max} is small (e.g., three or smaller), missed cleavages can be considered to be a constant multiplicative factor in our time complexity analysis as described earlier.

Since the proteolytic digestion process produces peptides that contain two or more cysteine residues, there is the possibility that intramolecular bonds may occur, i.e. disulfide bonds exist within a single peptide. These peptides must be included into D, with mass $m(p) = \sum m(a_i) - 2$, if at most one disulfide bond per peptide is considered. As discussed in [8], if a peptide contains more than two cysteine residues, the precise determination of the linkage pattern is not possible, requiring further laboratory analysis to correct this condition. The impact on time complexity is simply the larger mass space D, which can be modeled as an additive factor, $f(||P||, ||C||, m_{max})$.

3.3. The algorithms

The complete algorithms to create the mass space, called Generate-D, as well as the hashing based algorithm MSHashID to search this space, is listed in this section. We also perform a time complexity analysis for each algorithm. Our overall result indicates a

marked improvement in performance in comparison with an O(||C||!) time complexity as reported for the X! Disulfide Protein Linkage Modeler [5, 11].

Algorithm Generate-D (protein sequence, protease, m_{max})
1 From the protein sequence P, obtain $\{c_i\}$, the set of all cysteine locations
2 For each c_i,
3 Using the protease, obtain the m_{max} cleavage site locations s_i closest to c_i such that each $s_i<c_i$.
4 Using the protease, obtain the m_{max} cleavage site locations b_i closest to c_i such that each $b_i>c_i$.
5 $j \leftarrow m_{max}$.
6 While $j \geq 0$
7 k = 0.
8 While $k \leq j$
9 Cleave P at locations $s[m_{max} - j]$ and b[j].
10 Add subsequence to set of peptides, $\{p_i\}$.
11 If subsequence contains > 1 ci, add to $\{d_i\}$.
12 $k \leftarrow k + 1$.
13 $j \leftarrow j - 1$.
14 For each p_i, $d_i = m(Comb(p_i, 2)) - 2$, where Comb is the combinatorial function.

Algorithm MSHashID (D, S)
1 Construct and initialize hash table T. The size of T depends on h(), the hashing function used.
2 For each d_i in D
add d_i to $T[h(||d_i||)]$.
3 For each s_i in S
compute e_i, the expected number of amino acids.
4 For each e_i,
5 Let j = 0.
6 While j < V,
7 Obtain t_i, defined to be every d_i from the hash bucket $T[h(e_i) \pm j]$.
8 For each t_i,
9 compute $m(t_i)$.
10 If $s_i = m(t_i)$,
output t_i as an identified disulfide bond.
11 Else, $j \leftarrow j + 1$.
12 If j = V, output "no match".

THEOREM 1. Generate-D executes in $O(||C||^2)$ time.

PROOF. Step 1 takes $O(||P||) = O(||C||)$ time. Because m_{max} is a constant, Steps 3-13 are each constant, so Step 2 takes $O(||C||)$ time. The combinatorial function $Comb(p_i, 2)$ in Step 14 takes $O(||P||^2) = O(||C||^2)$ time. Thus, Generate-D executes in $O(||C|| + ||C|| + ||C||^2) = O(||C||^2)$ time. □

THEOREM 2. MSHashID can be solved in $O(||D|| + ||S||)$ time if $||S|| << ||D||$ and if the hashing function h() executes in constant time and is a good fit to the data.

PROOF. The mass space D is constructed in steps 1 and 2 of MSHashID, which take $O(||D||)$ time if h() executes in constant time. The last two steps search D for matches. Steps 3 and 4 take $O(||S||)$ time. The first inner loop, Step 6, can be approximated by a constant if V is small, which is the case when h() is a good fit to the data. The innermost loop, Step 8, is computed for only the values t_i in the bucket. If h() is a good fit to the data, this can also be approximated by a constant. Thus, MSHashID executes in $O(||D|| + ||S||)$ time. □

4. Implementation of the Generate-D and Hash-D algorithms

Our implementation of MSHashID accepts the input of the protein's sequence information in FASTA format, followed by the protease used to digest the protein. Our software models the action of the proteases trypsin (T) and chymotrypsin (C), endoproteinase Glu-C (G), as well as both trypsin and chymotrypsin (TC) together. This is followed by the maximum number of missed cleavages m_{max} allowed. The next number is the value for the mass tolerance, m_t. Finally, the number of spectrum masses followed by the spectrum mass values is input. All mass values are entered in units of Daltons. The program displays a list of cysteine pairs (each labeled by its position in the primary structure, as in [12]) that are identified as being in a disulfide bond.

5. Experimental results

Table 1 summarizes the results of validating MSHashID program against data from three proteins, with varying numbers of cysteines and disulfide bonds. We obtained the primary sequences from the Swiss-Prot database [13], and the MS masses are computed from the spectra that are shown in the referenced figures. We first note the absence of *false negatives* (i.e. no missed bond identifications) in these results. A *false positive* is generated when MSHashID outputs a cysteine pair that doesn't match the identification made in the reference. Two false positives were generated for the C2GnT-I Fig. 9 data because the combination of both proteases and $c_{max} = 3$ created a relatively large, fine-grained mass space. For each experiment, the size of each disulfide bond mass space, ||D||, is also listed.

Table 1. MSHashID validation testing results for laboratory data.

Experiment	Protein	Reference	Protease used	m_t	c_{max}	‖D‖	Disulfide bond to verify	Bond identified?	Number of false positives
1	C2GnT-I	[12], Fig. 7	T	5.0	0	55	Cys^{59}—Cys^{413}	Yes	0
2	C2GnT-I	[12], Fig. 8	T	5.0	0	55	Cys^{372}—Cys^{381}	Yes	0
3	C2GnT-I	[12], Fig. 9	TC	1.0	3	5356	Cys^{100}—Cys^{172}	Yes	2
4	C2GnT-I	[12], Fig.10	C	2.0	2	1653	Cys^{151}—Cys^{199}	Yes	0
5	ST8sia IV	[1], Fig. 5	C	2.0	1	120	Cys^{142}—Cys^{292}	Yes	0
6	ST8sia IV	[1], Fig. 6	C	2.0	1	120	Cys^{156}—Cys^{356}	Yes	0
7	FT VII	[14], Fig. 5	T	1.0	1	28	Cys^{318}—Cys^{321}	Yes	0
8	FT VII	[14], Fig. 7	C	1.0	1	28	Cys^{211}—Cys^{214}	Yes	0
9	FT VII	[14], Fig. 9	C	1.0	1	28	Cys^{68}—Cys^{76}	Yes	0

6. Conclusion

While many computational treatments of the problem of identifying and characterizing protein have utilized the fact that each amino acid has a known mass, this paper is to our knowledge the first to utilize the fact that the average value of an amino acid mass can be used to mine for information in a proteomics mass space. Encapsulating this into our definition of the Expected Amino Acid Mass, and employing a hash table data structure to store and retrieve matches from the space of possible mass combinations, we arrive at an algorithm, MSHashID, which efficiently solves the MS disulfide bond identification problem. Together with the MS/MS data analysis program IndexID, the MS2DB package enables highly accurate, high throughput identification of disulfide linkage patterns in proteins. The implementation of MSHashID will soon be included in publicly available version of MS2DB at http://tintin.sfsu.edu:33191/ms2db/.

Acknowledgments

This work is supported in part by funding from the NSF (IIS-0644418 and CHE-0619163), the Center for Computing for Life Sciences at San Francisco State University, and a grant (P20 MD000262) from the Research Infrastructure in Minority Institutions Program, National Center on Minority Health and Health Disparities (NCMHD), NIH.

7. References

[1] K. Angata, T. Yen, A. El-Battari, B.A. Macher, and M. Fukuda, "Unique disulfide bond structures found in ST8Sia IV polysialyltransferase are required for its activity", *J Biol Chem.*, 2001, May 4;276(18):15369-7.

[2] Vullo, A. and P. Frasconi, “Disulfide connectivity prediction using recursive neural networks and evolutionary information”, *Bioinformatics*, 2004. 20(5): p. 653-9.

[3] Smith, D.L. and Z. Zhou, “Strategies for Locating Disulfide Bonds in Proteins”, in *Methods in Enzymology: Mass Spectrometry*, (J. A. McCloskey, ed.), Vol. 193, Academic Press, Inc., 374-389, 1990.

[4] http://prospector.ucsf.edu/prospector/4.0.8/html/msbridge.htm.

[5] http://www.systemsbiology.ca/x-bang/DisulphideModeler/DisulphideModeler.html.

[6] http://prowl.rockefeller.edu/prowl/peptidemap.html.

[7] Yen, T.Y. and Macher, B.A. Determination of glycosylation sites and disulfide bond structures using LC/ESI-MS/MS analysis. *Methods in enzymology*, **415**, (2006), 103-113.

[8] T. Lee, R. Singh, R.Yen and B. Macher "MS2DB: An Algorithmic Approach to Determine Disulfide Linkage Patterns in Proteins by Utilizing Tandem Mass Spectrometric Data", *IEEE International Symposium on Computer-Based Medical Systems* (CBMS), 2006, pp. 947-952.

[9] http://prowl.rockefeller.edu/aainfo/struct.htm.

[10] T.H. Cormen , C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, 2001.

[11] R. Craig, O. Krokhin, J. Wilkins, and R.C. Beavis, “Implementation of an algorithm for modeling disulfide bond patterns using mass spectrometry”, *Journal of proteome research*, **2**, (2003), 657-661.

[12] T.Y. Yen, B.A. Macher, S. Bryson, X. Chang, I. Tvaroska, R. Tse, S. Takeshita, A.M. Lew, and A. Datti, “Highly Conserved Cysteines of Mouse Core 2 1,6-N-Acetyl glucosaminyltransferase I Form a Network of Disulfide Bonds and Include a Thiol That Affects Enzyme Activity,” *J Biol Chem.*, (2003), Nov 14;278(46):45864-81.

[13] Swiss-Prot database: http://cs.expasy.org/.

[14] T. De Vries, T. Yen, R.K. Joshi, J. Storm, D.H. van den Eijnden, R.M.A. Knegtel, H. Bunschoten, D.H. Joziasse, and B.A. Macher, “Neighboring cysteine residues in human fucosyltransferase VII are engaged in disulfide bridges, forming small loop structures: a proposed 3D model based on location of cysteines, and threading and homology modeling”, *Glycobiology*, 11, 423-432.