A Neural Network Approach to the Identification of b-/y-ions in MS/MS Spectra

James P. Cleveland
Department of Computer Science and Engineering
University of South Carolina
Columbia, South Carolina
jimmycleveland@gmail.com

John R. Rose
Department of Computer Science and Engineering
University of South Carolina
Columbia, South Carolina
rose@cse.sc.edu

***Abstract*—The effectiveness of de novo peptide sequencing algorithms depends on the quality of MS/MS spectra. Since most of the peaks in a spectrum are uninterpretable 'noise' peaks it is necessary to carefully pre-filter the spectra to identify the 'signal' peaks that likely correspond to b-/y-ions. Selecting the optimal set of peaks for candidate peptide generation is essential for obtaining accurate results. A careful balance must be maintained between the precision and recall of peaks that are selected for further processing and candidate peptide generation. If too many peaks are selected the search space will be too large and the problem becomes intractable. If too few peaks are selected cleavage sites will be missed, the resulting candidate peptides will have large gaps, and sequencing results will be poor. For this reason pre-filtering of MS/MS spectra and accurate selection of peaks for peptide candidate generation is essential to any de novo peptide sequencing algorithm.**

We present a novel neural network approach for the selection of b-/y-ions using known fragmentation characteristics, and leveraging neural network probability estimates of flanking and complementary ions. We show a significant improvement in precision and recall of peaks corresponding to b-/y-ions and a reduction in search space over approaches used by other de novo peptide sequencing algorithms.

***Keywords*-Tandem mass spectrometry; proteomics; de novo; peptide sequencing; MS/MS preprocessing; b-/y-ion selection**

I. Introduction

Tandem mass spectrometry (MS/MS) is the most important tool in high-throughput proteomics. The primary application is peptide identification. For MS/MS spectra originating from proteins that are not present in a sequence database researchers must use de novo peptide sequencing algorithms to sequence the peptide. The goal of de novo peptide sequencing is to compute the peptide whose fragmentation produced the experimental spectrum. De novo peptide sequencing follows a general formulation: peak selection (often described as preprocessing), followed by peptide candidate generation, and then candidate scoring. In the peak selection step the objective is to identify a subset of peaks in the spectrum that likely correspond to b-/y-ion ladders. In the candidate generation step the selected peaks are used to generate a set of candidate peptides that could have produced the spectrum. It is during candidate generation that the effects of inadequate peak selection become problematic. In the candidate scoring step a scoring function is used to rank the candidates and choose the most probable peptide that produced the spectrum. Often candidate scoring is incorporated into candidate generation. The quality of the candidates that are generated and scored depends on the ability of the program to initially select peaks that will allow for the correct (or best) candidate peptide to exist in the search space, so it is important that peak selection be done in an optimal way.

The general approach used by other prominent de novo peptide sequencing algorithms depends primarily on relative peak intensity. PepNovo uses a sliding window of width 56 across the spectrum and keeps any peaks that are in the top 3 when ranked by intensity [1]. MSNovo selects peaks by using a sliding window of width 100 and selects the top 6 peaks from each window [2]. PILOT keeps only the top 125 peaks of highest intensity in the spectrum [3]. pNovo selects the top 100 peaks by intensity [4].

In our experiments we found that selecting peaks based on relative intensity alone could miss a nontrivial portion of b-/y-ions. If the complex dynamics of peptide fragmentation–including relative peak intensity–can be modeled and incorporated into a predictive ion-type classifier, then the accuracy of peak selection will be superior than the accuracy of a peak classifier that uses peak intensity alone. We demonstrate that this superior approach can be implemented via a neural network. A neural network approach was used because it allows us to construct a predictive model that does not require the complete understanding of the complex dynamics of peptide fragmentation. The Leveraged Neural Network (LNN) ion classifier described below selects peaks with higher precision and recall than other de novo peptides sequencing algorithms.

Increasing recall leads to better candidates in the candidate peptide search space. If recall is held fixed and the precision is increased the result will be a significantly smaller candidate peptide search space, without sacrificing the best candidate contained in the search space. Given the computational limits that all de novo agorithms face, low precision can render any de novo algorithm computationally impractical. Low recall will result in missing peaks, which in turn will result in large gaps in the spectrum graph. This in turn leads to an exploding combinatorial search as permutations of

978-1-4673-2560-8/12/$31.00

residues consistent with these large gaps must be considered. It is clear that a careful balance of improved precision and recall is important for peptide candidate generation.

II. Methods

The dataset used in this study is composed of doubly charged tryptic peptides produced by LC/MS/MS. We limited our dataset to doubly charged peptides since this charge state is most common in MS/MS experiments. Of the original dataset containing 8610 mass spectra we kept 3373 spectra of unique peptides which had an Xcorr score greater than 2.5 (providing high confidence peptide spectrum matches), and a mass between 600 and 3000 Da. Our data came from the PNNL Salmonella Typhimurium datasetwhich is publicly available on the web.[1]

The dataset (D) was randomly divided in half for 2-fold cross validation (D_1 and D_2). For each fold 890 spectra (324 684 peaks for D_1 and 329 382 peaks D_2) were used for the training and classification. First, D_1 was used for training ($D_T \leftarrow D_1$) and D_2 for classification/evaluation ($D_E \leftarrow D_2$), and then the reverse ($D_T \leftarrow D_2$ and $D_E \leftarrow D_1$). The results from each fold were averaged and are presented in Figure 2. For each spectrum in the training dataset we first removed peaks with intensity below an experimentally derived threshold, in this case 15, which dramatically sped up the training of the neural network without sacrificing performance. Before the neural network can be trained D_T must be transformed. Each peak in D_T is assigned its correct class label (target vector), either b-ion, y-ion, or u-ion (unknown ion), each of which is a binary vector of length three. For each peak in D a feature vector (pattern) is generated that will later be presented to the input layer of the neural network for training and classification. The features used are described in Table I. D_T is randomly divided again such 95% of the spectra (13 884 and 13 836 peaks for D_1 and D_2 respectively) were used for backpropagation (D_{T_B}) and 5% of the spectra (13 270 and 14 934 peaks for D_1 and D_2 respectively) for validation (stopping criteria) (D_{T_V}). The number of peaks in D_{T_B} and D_{T_V} are roughly equivalent since the backpropagation dataset is filtered so that there are an equal number of b, y, and u ions.

The training process of the neural network requires the use of an objective error function. In our implementation the output (**o**) of the neural network represents an estimate of the posterior probability that the input pattern belongs to the respective class in the target vector (**t**). When interpreting the outputs as probabilities it is appropriate to use the cross entropy error function [6].

$$\text{network error} = -\sum_{i=0}^{2} [\mathbf{t}_i log(\mathbf{o}_i) + (1 - \mathbf{t}_i) log(1 - \mathbf{o}_i)]$$

[1] http://omics.pnl.gov/view/dataset_80292.html

Algorithm 1 Leveraged Neural Network Training and Classification

$net_1 \leftarrow train(D_{T_B}, D_{T_V})$
$D_T \leftarrow classify(D_T, net_1)$ {peaks in D_T now have b-/y-/u-ion probability estimates}
$net_2 \leftarrow train(D_{T_B}, D_{T_V})$
$D_E \leftarrow classify(D_E, net_1)$
$D_E \leftarrow classify(D_E, net_2)$

The neural network will train on all of the patterns in D_{T_B} numerous times (epochs) until the network performance no longer improves. This is determined by classifying the patterns in D_{T_V} after each epoch until the error on D_{T_V} begins to increase, at which point the training terminates.

In our classifier two neural networks are used in succession for peak classification, which we refer to as a leveraged neural network. Each is trained in the manner described above except for differences in the feature vector used. The structure of each neural network consists of an input layer with as many nodes as features in the pattern, a single hidden layer with twice as many nodes as the input layer, and an output layer with three nodes corresponding to the three possible classes. A general formulation for training the neural networks is given in algorithm 1. In the first neural network (net_1) the peak features are computed from data in the spectrum alone as described below and in Table I. In the second neural network (net_2) the outputs from net_1 are leveraged as additional features in net_2, described in Table I. In the net_2 input pattern the complement ion feature is modified by replacing the normalized relative intensity of the complementary peak with the maximum of the b-/y-ion probability estimates in the output from net_1 for the complementary peak. In the net_2 input pattern there are two additional features corresponding to flanking residues on the N and C terminal sides of the current peak (peak for which the feature vector is being computed). The N terminal flanking residue feature is computed by taking the maximum b-/y-ion probability (as estimated by net_1) of any peak with a mass offset from the current peak equivalent to the mass of an amino acid. The C terminal flanking residue feature is computed similarly. The reasoning for these 'leveraged' features is that if the current peak has a complement or flanking peak with a high probability of being a $b-$/$y-$ion, then the current peak has increased probability of being a $b-$/$y-$ion itself. Our experiments show that leveraging the output from net_1 to train a second neural network in this way yields a higher recall than does classification with net_1 alone.

Description of features

The features described capture known fragmentation characteristics and correlations between b-/y-ion peaks and other mass peaks produced by CID peptide fragmentation. The

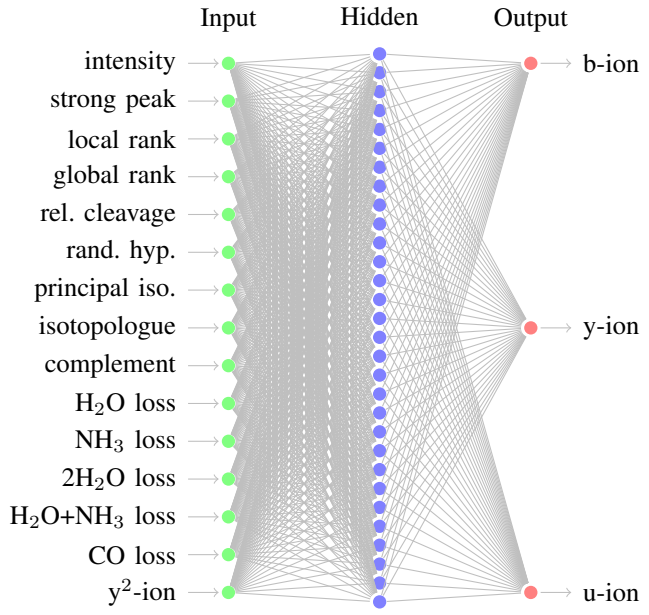

Figure 1. Topology of net_1

net_1 pattern features		net_2 pattern features	
feature	value	feature	value
intensity	N, D	intensity	N, D
strong peak	B	strong peak	B
local intensity rank	N	local intensity rank	N
global intensity rank	N	global intensity rank	N
relative cleavage position	N, D	relative cleavage position	N, D
random peak hypothesis	P	random peak hypothesis	P
principal isotope	B	principal isotope	B
isotopologue	B	isotopologue	B
complement	N, D	complement	P_{net_1}
H_2O neutral loss	N, D	H_2O neutral loss	N, D
NH_3 neutral loss	N, D	NH_3 neutral loss	N, D
H_2O-H_2O neutral loss	N, D	H_2O-H_2O neutral loss	N, D
H_2O-NH_3 neutral loss	N, D	H_2O-NH_3 neutral loss	N, D
CO neutral loss (a-ion)	N, D	CO neutral loss (a-ion)	N, D
y^2-ion	N, D	y^2-ion	N, D
		N-term flanking ion	P_{net_1}
		C-term flanking ion	P_{net_1}

Table I
PATTERN FEATURES FOR net_1 AND net_2: N DENOTES A NORMALIZED VALUE, D DENOTES A DISCRETIZED VALUE, B DENOTES A BINARY VALUE, AND P DENOTES A PROBABILITY ESTIMATE. EACH PEAK IN THE SPECTRUM IS CLASSIFIED BY BOTH NEURAL NETWORKS IN SUCCESSIVE PASSES OVER THE SPECTRUM. net_2 FEATURES DEPEND ON THE CLASSIFICATION RESULTS OF net_1.

intensity feature is the normalized and discretized relative peak intensity of the current peak (the peak for which a feature vector is being created). Normalized intensities are computed by dividing each peak intensity by the maximum peak intensity in the spectrum. Normalized and discretized intensities are then rounded up to either 0.05, 0.10, 0.20, 0.40, 0.80, or 1.00. The strong peak feature is a binary value that indicates whether or not the current peak was selected as a 'strong peak' using a sliding window method; in this case the top three peaks were selected in a sliding window of width 56 Da. The local and global intensity ranks give the normalized rank by intensity of the current peak within a 'local' window, or globally. These first four peak intensity based features are informative due to the fact that b-/y-ions tend to be of higher abundance than other ion types in CID spectra. The relative cleavage position gives the normalized and discretized position of the current peak relative to parent ion mass. This feature captures the variation in peak intensity across the mass range of the instrument. Typically, peaks tend to be more intense near the center of the peptide and less intense or missing near the terminal ends. The random peak hypothesis estimates the probability that the current peak is a random peak rather than an ion of interest. This feature is modeled after the random peak hypothesis described in Frank and Pevzner [1]. The principal isotope feature is a binary value that indicates whether or not the current peak appears to be a principal isotope, and the isotopologue feature indicates the converse. The complement feature in the net_1 pattern is the normalized and discretized intensity of any peak found at the expected complement mass position. If the current peak is a b-ion then we expect to see the complimentary y-ion peak, and likewise for the reverse. In the case of the net_2 pattern the complement feature gives the maximum b-/y-ion probability estimate using net_1 of any peak found at the expected complement mass position. The H_2O, NH_3, H_2O-H_2O, H_2O-NH_3, and CO neutral loss features all give the normalized and discretized intensity of peaks found in their respective offsets from the current peak. The y^2-ion feature vector gives the normalized and discretized intensity of any peak existing in the offset from the current peak where a doubly charged y-ion is expected. The N-term and C-term flanking ion features are the maximum b-/y-ion probability estimates using net_1 for any peaks that are found at a mass offset from the current peak corresponding to the mass of a single amino acid. If the current peak is indeed a b-/y-ion then we expect it to be part of an ion ladder, and thus we expect to find other peaks that are likely b-/y-ions at mass offsets equivalent to the mass of an amino acid. This feature value is the flanking peak intensity and not the mass difference, and therefore does not capture any sequence information.

III. RESULTS

Results comparing precision and recall are shown in Figure 2. We compared the performance of the leveraged neural network (LNN) peak selection with two other de novo peptide sequencing algorithms. The window method selects peaks by choosing the 3 most intense peaks in a window of width 56 Da. This is the method Frank described in the original PepNovo publication [1]. Peak selection in PepNovo was subsequently improved. As shown in this figure, the actual performance of PepNovo is substantially better with respect

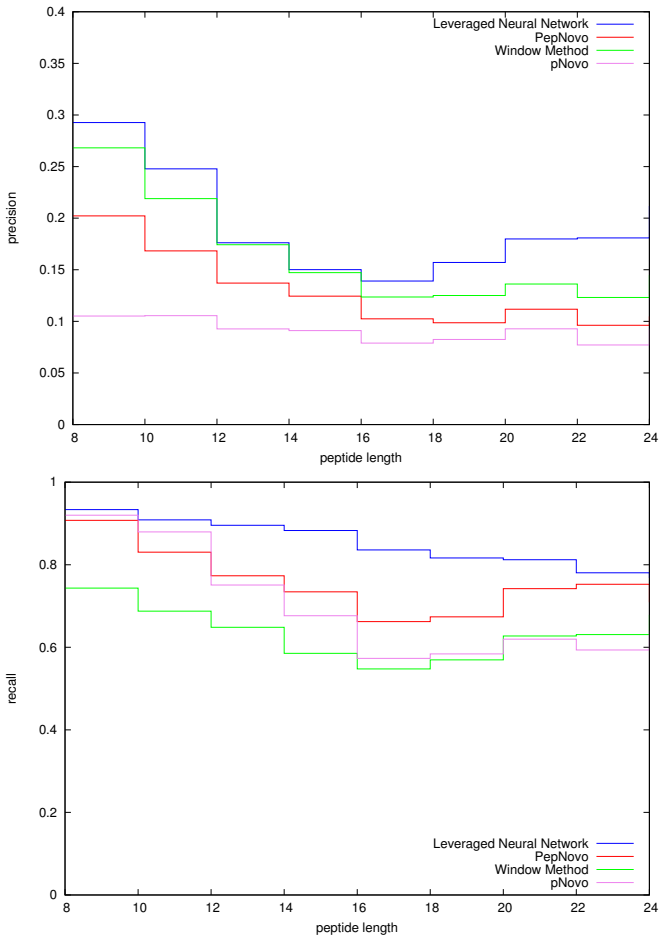

Figure 2. Results comparing the precision and recall for b-/y-ion selection across varying peptide length. Average of D_1 and D_2 cross validation results for PNNL spectra. Our neural network approach is compared to PepNovo and the ms2preproc window method ($X = 3, Y = 0, Z = 56$)

peptide length	LNN	PepNovo
8	11	43
9	20	46
10	47	51
11	46	52
12	42	58
13	62	55
14	71	77
15	64	87
16	97	72
17	89	107
18	84	89
19	50	93
20	76	93
21	38	102

Table II
AVERAGE NUMBER OF PEAKS PER SPECTRUM CLASSIFIED AS b-/y-IONS BY LNN AND PEPNOVO.

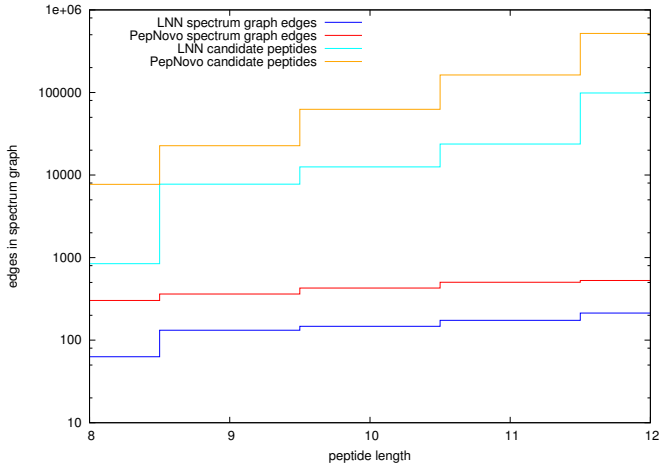

Figure 3. Comparison of the median number of edges in the spectrum graph (bottom pair) and the median number of candidate peptides generated from the spectrum graph (top pair). Note that the y scale is logarithmic and the relationship between the number of edges in the spectrum graph and the number of candidate peptides is exponential.

to recall than the window method. The actual performance of PepNovo was determined by modifying the source code to output the peaks from the raw spectrum that are used to construct the spectrum graph.

Note that the LNN precision is consistently greater than that of PepNovo. A comparison of the number of peaks selected by LNN and PepNovo are shown in Table II. The number of peaks selected has a direct impact on the size of the search space. This effect can be seen when the search space of candidate peptide sequences is generated using the peaks selected by the two algorithms (Figure 3). We implemented a basic dynamic programming approach as described in Lu and Chen [7] to generate candidate peptides using the peaks selected by PepNovo and the LNN. Keep in mind that programs such as PepNovo use much more sophisticated approaches to generate candidates from the spectrum graph. This allows them to avoid an exhaustive search of the implicit search space. The size of the candidate peptide search space generated by the de novo algorithm using PepNovo's peak selection is larger than the size of the candidate peptide search space generated by the de novo algorithm using LNN peak selection. The difference in the size of the candidate peptide search space is shown in Figure 3 for peptides of length 8 to 12.

The size of the candidate peptide search space is exponentially proportional to the number of edges in the spectrum graph. Consequently we use this as a measure of search space to extend our results to longer peptides by comparing the number of edges in the spectrum graphs produced by each peak selection algorithm without having to exhaustively enumerate the candidate peptides.

It should be noted that on balance the b-/y-ion recall of the LNN peak selection is greater for all peptide lengths included in our experiments. Thus on balance we can expect that the top scoring candidate peptides that would be

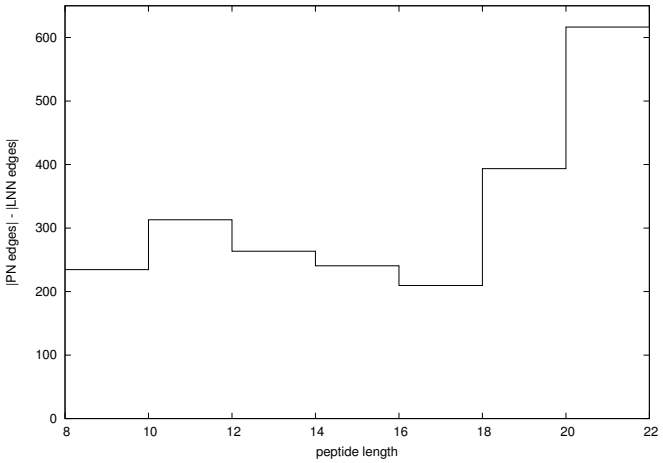

Figure 4. LNN net reduction in spectrum graph edges compared to PepNovo. These data points were produced by subtracting the median number of edges in the LNN spectrum graph from the median number of edges in the PepNovo spectrum graph for peptide length bins of width 2.

generated by these spectrum graphs will be of higher quality. When we compare the number of edges in the spectrum graphs we find that the number of edges generated by PepNovo's peak selection contain on average approximately 300 more edges than the corresponding LNN spectrum graph as shown in Figure 4.

IV. Discussion and Conclusion

Peak selection is an important preprocessing step in de novo sequencing. As a practical matter, it is important that the number of peaks be reduced so that the candidate peptide search space is constrained. A reduction in the number of peaks used to create the spectrum graph makes it possible to process spectra faster. It also makes it possible to process longer peptides than would otherwise be impractical. The results in the preceding section demonstrate that the LNN approach results in fewer peaks being selected. The resulting search space is smaller. As noted earlier, the LNN spectrum graphs contains on average 300 fewer edges than do the spectrum graphs resulting from PepNovo's peak selection. If one uses spectrum graph size as a measure of the search space then the median LNN search space for peptides of length 20 is smaller than the median PepNovo search space for peptides of length 15.

Quality of search space is as important as reduction of search space. It is critical that those peaks corresponding to b-/y-ions be identified so that the resulting candidate search space contains the correct peptide. As demonstrated in the preceding section, LNN peak selection precision and recall are superior to PepNovo's precision and recall for all peptide lengths from 8 residues to 24 residues. The LNN approach selects fewer peaks than PepNovo. It also selects more peaks corresponding to b-/y-ions than does PepNovo.

We have presented a peak filtering approach that demonstrates an improvement in precision and recall over current approaches. The result is a smaller, more targetted search space. This promises to support the faster processing of spectra and the ability to handle longer peptides.

References

[1] A. Frank and P. Pevzner, "PepNovo: de novo peptide sequencing via probabilistic network modeling." *Analytical chemistry*, vol. 77, no. 4, pp. 964–73, Feb. 2005. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/15858974

[2] L. Mo, D. Dutta, Y. Wan, and T. Chen, "MSNovo: a dynamic programming algorithm for de novo peptide sequencing via tandem mass spectrometry." *Analytical Chemistry*, vol. 79, no. 13, pp. 4870–4878, 2007. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/17550227

[3] P. a. DiMaggio and C. a. Floudas, "De Novo Peptide Identification via Tandem Mass Spectrometry and Integer Linear Optimization," *Analytical chemistry*, vol. 79, no. 4, pp. 1433–46, Feb. 2007. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2730153\&tool=pmcentrez\&rendertype=abstracthttp://pubs.acs.org/doi/abs/10.1021/ac0618425

[4] H. Chi, R. Sun, B. Yang, C. Song, and LH, "pNovo: De novo Peptide Sequencing and Identification Using HCD Spectra," *Journal of Proteome*, pp. 2713–2724, 2010. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/pr100182k

[5] C. Ansong, N. Tolić, S. O. Purvine, S. Porwollik, M. Jones, H. Yoon, S. H. Payne, J. L. Martin, M. C. Burnet, M. E. Monroe, P. Venepally, R. D. Smith, S. N. Peterson, F. Heffron, M. McClelland, and J. N. Adkins, "Experimental annotation of post-translational features and translated coding regions in the pathogen Salmonella Typhimurium." *BMC genomics*, vol. 12, no. 1, p. 433, Jan. 2011. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3174948\&tool=pmcentrez\&rendertype=abstract

[6] L. Silva and J. M. de Sá, "Data classification with multilayer perceptrons using a generalized error function," *Neural Networks*, vol. 21, no. 9, pp. 1302–10, Nov. 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608008000749

[7] B. Lu and T. Chen, "A suboptimal algorithm for de novo peptide sequencing via tandem mass spectrometry." *Journal of computational biology : a journal of computational molecular cell biology*, vol. 10, no. 1, pp. 1–12, Jan. 2003. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/12676047