DE NOVO PEPTIDE SEQUENCING FOR MASS SPECTRA BASED ON MULTI-CHARGE STRONG TAGS

KANG NING, KET FAH CHONG, HON WAI LEONG

Department of Computer Science,
National University of Singapore,
3 Science Drive 2, Singapore 117543

This paper presents an improved algorithm for de novo sequencing of multi-charge mass spectra. Recent work based on the analysis of multi-charge mass spectra showed that taking advantage of multi-charge information can lead to higher accuracy (sensitivity and specificity) in peptide sequencing. A simple de novo algorithm, called GBST (Greedy algorithm with Best Strong Tag) was proposed and was shown to produce good results for spectra with charge > 2. In this paper, we analyze some of the shortcomings of GBST. We then present a new algorithm GST-SPC, by extending the GBST algorithm in two directions. First, we use a larger set of multi-charge strong tags and show that this improves the theoretical upper bound on performance. Second, we give an algorithm that computes a peptide sequence that is optimal with respect to shared peaks count from among all sequences that are derived from multi-charge strong tags. Experimental results demonstrate the improvement of GST-SPC over GBST.

1. Introduction

De novo peptide sequencing of tandem mass (MS/MS) is a challenging problem in proteomics and high-throughput generation of MS/MS spectra with modern proteomics technology is compounding the problem. As the volume of MS/MS spectra grows, the accompanying algorithmic technology for automatically interpreting these spectra has to keep pace. An increasingly urgent problem is the interpretation of *multi-charge* spectra – MS/MS spectra with charge 3, 4, and 5 are available from the publicly accessible GPM (Global Proteome Machine) dataset [5]; and those with charge 3 are available from the ISB (Institute for Systems Biology) dataset [10]. It is foreseen that increasingly there will be more multi-charge spectra produced and so the problem of accurate interpretation of these spectra will become more important with time.

Many existing algorithms for peptide sequencing have focused largely on interpreting spectra of charge 1, even when dealing with multi-charge spectra, and only a few algorithms [4, 8, 11] take account for higher charge ions. Recent work by the authors [4] using this approach has shown that the sensitivity accuracy of Lutefisk [15] and PepNovo [8] (both of which consider only ion-types of charge 1 and 2) are very low (less that 25%) when applied to higher charge spectra from the GPM dataset. Their experimental study also showed that there is significant potential improvement in the performance if multiple charges are taken into consideration during the sequencing process. A simple de novo algorithm, called *GBST* (Greedy algorithm with Best Strong Tags) was also presented that uses multiple charges to achieve good results for multiple charge spectra. The GBST algorithm consists of two phases: in the first phase, a set of

"best" strong tags are computed based on strong evidence in the spectrum (charge 1, *b*-ions and *y*-ions, and no neutral loss, and direct connectivity); in the second phase, the GBST algorithm then link the set of "best" strong tags by taking into account more ion-types (charges) and greater connectivity. A standard algorithm was then used to generate the set of paths that corresponds to the top *k* predicted peptide sequences.

In this paper, we present an improved algorithm called *GST-SPC* that improves on GBST algorithm. In the first phase, the GST-SPC algorithm computes a *larger* set of strong tags – the set of all *maximal* multi-charge strong tags. We show that this improves the theoretical upper bound on the sensitivity. In the second phase, the GST-SPC algorithm computes a peptide sequence that is *optimal* with respect to *shared peaks count* (*SPC*) from among all sequences that are derived from strong tags. Our evaluation shows that the GST-SPC algorithm improves on GBST, especially on multi-charge spectra.

2. Review of Related Work and Problem Formulation

We first give a quick review of related work on de novo peptide sequencing for MS/MS. De novo algorithms [1, 3, 4, 6, 8, 11, 15] are used to predict sequences or partial sequences for novel peptides or for peptides that are not in the protein database. Most de novo sequencing algorithms [3, 6, 8, 15] uses a *spectrum graph* approach to reduce the search space of possible solutions. Given a mass spectrum, the spectrum graph [6] is a graph where each vertex corresponds to some ion type interpretation of a peak in the spectrum. Edges represent amino acids which can interpret the mass difference between two vertices. Each vertex in this spectrum graph is then scored using *Dancik scoring* based on its supporting peaks in the spectrum (see [6] for details). Given such a scoring the predicted peptide represents the optimal weighted path from the *sour*ce *vertex* (of mass 0) to the *end vertex* (of mass *M*).

PepNovo [8] uses a spectrum graph approach similar to [6], but uses an improved scoring function based on a probability network of different factors which affect the peptide fragmentation and how they conditionally affect each other (represented by edges from one vertex to another). The algorithm PEAKS [11] does not explicitly construct a spectrum graph but builds up an optimal solution by finding the best pair of prefix and suffix masses for peptides of small masses until the mass of the actual peptide is reached. A fast dynamic programming algorithm is used.

Problem Formulation of Multi-Charge Peptide Sequencing: Our formulation of multi-charge peptide sequencing follows that in [4]. We summarize it here to facilitate our discussion of the GBST algorithm (see [4] for detailed discussion).

Consider a multi-charge MS/MS spectrum S of charge α for a peptide $\rho = (a_1 a_2 \ldots a_n)$ where a_j is the j^{th} amino acid in the sequence. The *parent mass M*, of the peptide is given by $m(\rho) = M = \sum_{j=1}^{n} m(a_j)$. A peptide fragment $\rho_k = (a_1 a_2 \ldots a_k)$ $(k \leq n)$ has fragment mass $m(\rho_k) = \sum_{j=1}^{k} m(a_j)$. The peaks in the spectrum S come from peptide fragmentation. Each peak p can be characterized by its ion-type given by $(z, t, h) \in (\Delta_z \times \Delta_t \times \Delta_h)$, where z is the charge of the ion, t is the basic ion-type, and h is the neutral loss incurred by the

ion. In this paper, we use $\Delta = (\Delta_z \times \Delta_t \times \Delta_h)$, where $\Delta_z = \{1,2,\ldots,\alpha\}$, $\Delta_t = \{a,b,y\}$, and $\Delta_h = \{\phi, -H_2O, -NH_3\}$. The (z, t, h)-*ion* of the peptide fragment ρ_k will produced an observed peak p in the spectrum S, that has a mass-to-charge ratio of $mz(p)$, that can be computed by the following formula [4]:

$$m(\rho_k) = mz(\rho) \cdot z + (\delta(t) + \delta(h)) + (z - 1) \quad (1)$$

where $\delta(z)$ and $\delta(z)$ are the mass difference for the respective ion-type and neutral-loss.

The *theoretical spectrum of charge* α [4] for ρ is defined by $TS_\alpha^\alpha(\rho)$ = {p | p is observed peak for (z, t, h)-*ion* of peptide fragment ρ_k, for all $(z, t, h) \in \Delta$ and k=0,1,…,n}. It represents the set of *all possible* observed peaks that may be present in any experimental spectrum for the peptide ρ.

In peptide sequencing, we are given an experimental spectrum $S = \{p_1, p_2, \ldots, p_n\}$, where each peak p_k is described by two parameters: $mz(p_k)$, the observed mass-to-charge ratio and *intensity*(p_k), its intensity. The problem is to determine the peptide ρ that produced S. In practice, of course, only a small fraction of these peaks in $TS_\alpha^\alpha(\rho)$ are present in S and there are also noise peaks as well.

Extended Spectrum and Spectrum Graph: To account for the different ion-types considered by different algorithms, the notions of *extended spectrum* and *extended spectrum graph* were introduced S_β^α were introduced in [4], where α denotes the maximum charge for S, and β denotes the maximum charge considered by the algorithm (β = 2 for PepNovo [8] and Lutefisk [15]). In the *extended spectrum* S_β^α, for each peak $p_j \in S$ and ion-type $(z, t, h) \in (\{1,2,\ldots,\beta\} \times \Delta_t \times \Delta_h)$, we generate a pseudo-peak denoted by $(p_j, (z, t, h))$ with a corresponding *assumed* fragment mass given by (1). Then, the *extended spectrum graph* of connectivity d is a graph $G_d(S_\beta^\alpha)$ in which each vertex represents a pseudo-peak $(p_j, (z, t, h))$ in the extended spectrum S_β^α, namely to the (z, t, h)-*ions* for the peak p_j. Two special vertices are added – the *start vertex* v_0 to represent mass 0 and the *end vertex* v_M for the parent M. For each vertex v, we define *prefix residue mass* of v, denoted by $PRM(v)$, to be the *prefix mass* of the interpreted peptide fragment mass for vertex v. It is defined as $PRM(v) = m(v)$ if v is a prefix ion type, and $PRM(v) = M - m(v)$ if v is a suffix ion type, where M is the parent mass. There is a directed edge (u, v) from vertex u to vertex v if we can find a directed path of at most d amino acid with total mass equal to the $(PRM(v) - PRM(u))$. (The standard spectrum graph use d = 1.) Note that the number of possible paths to be searched is $O(20^d)$, which increases exponentially with d. In this paper, we use d=2, unless otherwise stated. The extended spectrum is a generalization because when β=1, all peaks are assumed to be of charge 1, and so $S_1^\alpha = S$ – namely, there is no extension. In the extended spectrum S_2^α, only ions of charge 1 or 2 are considered (even for spectra with charge α>2). Algorithms such as PepNovo [8] and Lutefisk [15] uses some subsets of S_2^α and $G_2(S_2^\alpha)$.

Upper Bound on Sensitivity: Given any spectrum graph G defined on an experimental spectrum S from a *known* peptide ρ, the notion of *theoretical upper bound on sensitivity* was defined in [4] as follows: Given G, we can compute the path in G that *maximizes* the number, p^*, of amino acids from the (*known*) peptide ρ. Then, $U(G) = p^*/|\rho|$ is an upper bound on the *sensitivity* for *any sequencing algorithm* based on the spectrum graph

approach using the graph G. Then $U(G_d(S_\beta^\alpha))$ is the *theoretical upper bound on sensitivity* for the extended spectrum graph $G_d(S_\beta^\alpha)$, namely using the extended spectrum S_β^α with all ion types in Δ and a connectivity of d. PepNovo and Lutefisk which considers charge of up to 2 (and connectivity of up to 2) are bounded by $U(G_2(S_2^\alpha))$ and there is a sizeable gap between $U(G_2(S_2^5))$ and $U(G_2(S_5^5))$.

3. Evaluation of Greedy Strong Tag Algorithm for Multi-Charge Spectra

The GBST algorithm [4] was a simple algorithm that takes into account higher charge ions. It performs well on multi-charge spectra compared to other de novo algorithms. However, we show in this section that there is still a big gap in performance with respect to the theoretical upper bound $U(G_2(S_\alpha^\alpha))$.

The GBST Algorithm: The GBST algorithm first computes a set, *BST*, of "best" (or reliable) strong tags. To find strong tags, they use ion-types that appear most frequently, namely, charge 1, *b-ions* and *y-ions* with no neutral loss. The restricted set is given by $\Delta^R = (\Delta_z^R \times \Delta_t^R \times \Delta_h^R)$, where $\Delta_z^R = \{1\}$, $\Delta_t^R = \{b, y\}$, and $\Delta_h^R = \{\phi\}$. They also define $G_1(S_1^\alpha, \Delta^R)$, the spectrum graph $G_1(S_1^\alpha)$ where the ion types considered are restricted to those in Δ^R. Then, a *strong tag* T of ion-type $(z, t, h) \in \Delta^R$ is a maximal path $\langle v_0, v_1, v_2, ..., v_r \rangle$ in the graph $G_1(S_1^\alpha, \Delta^R)$, where every vertex $v_i \in T$ is of a (z, t, h)-*ion*. In each "component" of this graph, GBST compute a "best" strong tag with respect to some scoring function [4]. Then, the set *BST* is the set comprising the best strong tag for each component in the spectrum graph $G_1(S_1^\alpha, \Delta^R)$.

After the set of best tag, *BST*, is computed, the GBST algorithm then proceeds to find the best sequence that result from paths obtained by "extending" the tags from *BST* using all possible ion-types. It search for paths in the graph $G_2(BST)$ defined as follows: the vertices are the strong tags in *BST*, and we have a directed edge from the tail vertex u of a strong tag T_1 to the head vertex v of another strong tag T_2 if there is a directed edge (u,v) in the graph $G_2(S_\alpha^\alpha)$. We note two major difference between $G_2(BST)$ and the extended spectrum graph $G_2(S_\alpha^\alpha)$ – firstly, the number of vertices in $G_2(BST)$ is smaller; and secondly, the number of edges is also much smaller since only strong tags are linked in a head-to-tail manner. However, all ion types are considered in the graph $G_2(BST)$.

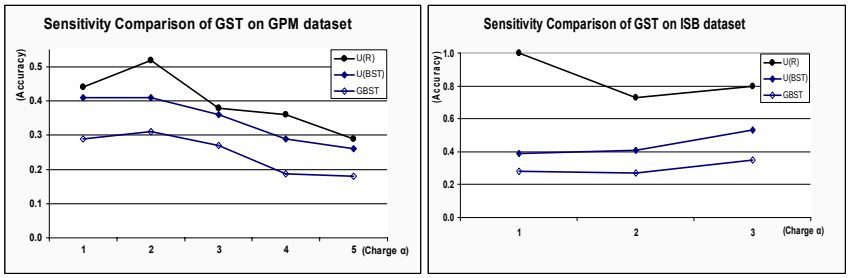
Upper Bounds on Sensitivity for GBST: Since the GBST algorithm uses a restricted set of ion-types Δ^R in its search for best strong tags, we let $U(R) = U(G_1(S_1^\alpha, \Delta^R))$ be the upper bound on sensitivity *with ion-type restriction*. For the second phase, we define $U(BST) = U(G_2(BST))$, the upper bound on sensitivity *with best strong tag restriction*.

Datasets Used: To evaluate the performance of GBST vis-à-vis the upper bounds, we used spectra that are annotated with their corresponding peptides – the GPM-Amethyst dataset [5] (Q-star data with good resolution[1]) and the ISB dataset [10] (Ion-Trap data with low resolution). For each dataset, we selected subsets of spectra with annotated

[1] Though these GPM spectra are high resolution spectra, they have been pre-processed using deconvolution [16], and so charge state determination using mono-isotopes is not possible.

peptides validated with an X-correlation score (Xcorr) greater than 2.0. The selected GPM dataset we use contains 2328 spectra, with 756, 874, 454, 205, and 37 each of charge 1,2,3,4, and 5, respectively, with an average of 46.5 peaks per spectrum.
The selected ISB dataset contains 995 spectra, with 16, 489, and 490 of charge 1,2, and 3, respectively, with an average of 144.9 peaks per spectrum.

The Evaluation Results: We have computed these upper bounds on sensitivity for both the GPM and the ISB datasets and the results are shown in Figure 1, together with the actual sensitivity obtained by the GBST algorithm. The results in Figure 1 show that for GPM datasets, $U(BST)$ is near to $U(R)$, but the GBST results have sensitivities about 10% less than $U(BST)$. This indicates that the GBST has not been able to fully utilize the power of *BST*. For the ISB datasets, even $U(BST)$ is far from $U(R)$. Therefore, it is natural the GBST algorithm can not perform well on ISB datasets.



(a) (b)

Figure 1: The comparison of sensitivity results of *GBST* with theoretical upper bounds. $U(R)$ and $U(BST)$ on (a) GPM dataset, and (b) ISB datasets.

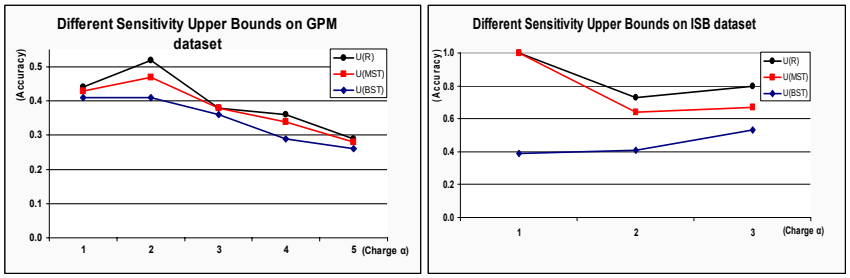
4. An Improved Algorithm – GST-SPC

In this paper, we present an improved algorithm, called GST-SPC, for de novo sequencing of multi-charge spectra, which improves the GBST algorithm in two ways: (a) by selecting a larger set of multi-charge strong tags, and second, and (b) by improving the sequencing algorithms for a given set of multi-charge strong tags.

(a) Using a Larger Set of Strong Tags: A straight-forward improvement of GBST [4] is to expand the set of strong tag under consideration. We do this as follows: (i) when searching for strong tags, we include multi-charge ions (using S_α^α instead of just S_1^α), and (ii) instead of choosing *only one* "best" strong tag from each component of the graph $G_1(S_1^\alpha, \Delta^R)$, we allow a set of all *multi-charge strong tags* in each component of the graph $G_1(S_\alpha^\alpha, \Delta^R)$ to be chosen. Namely, a *multi-charge strong tags* of ion-type (z^*, t, h) $\in \Delta^R$ is a maximal path $\langle v_0, v_1, v_2, \ldots, v_r \rangle$ in $G_1(S_\alpha^\alpha, \{\Delta^R\})$, where every vertex v_i is of a (z^*, t, h)-ion, in which t and h should be the same for all vertices, but z^* can be different number from $\{1,\ldots\alpha\}$. We let *MST* denote this set. The algorithm for computing the *MST* is the almost identical to that for tag generation (a depth-first search) with slight modification to store the *MST*. Running the GBST algorithm with the MST (instead of the BST) improves the results slightly (the details not shown here).

Theoretically, the size of the *MST* may be exponential. However, in practice, our experiments show that the MST does not exhibit exponential growth from BST. For GPM datasets (average of about 46 peaks) the increase in the average number of strong tags is from 10 to about 50. For ISB datasets (average of 145 peaks) the increase is from 15 to about 90. The average length of strong tags in MST is 4.65 amino acids for GPM datasets, and 2.26 for ISB datasets.

We define $U(MST) = U(G_2(MST))$ the *theoretical upper bound on sensitivity with respect to the set MST* of multi-charge strong tags. The increase from $U(BST)$ to $U(MST)$ is shown in Figure 2. From Figure 2, it is easy to see that the introduction of MST has pushed up the theoretical upper bounds for both datasets. For GPM dataset, the best sequencing results obtainable from MST is about 5% higher in accuracies than BST. We also note that $U(MST)$ is very close to the $U(R)$, the theoretical upper bounds with Δ^R. For ISB datasets, the increase is more pronounced – partly because the ISB datasets have more peaks. The best sequencing results obtainable from *MST* is about 10%~60% higher in accuracies than *BST*, and with in 20% to the theoretical upper bounds. This shows a great potential for sequencing algorithms based on *MST*.



(a) (b)

Figure 2**.** Comparing the theoretical upper bounds on sensitivity for MST and BST. Results are based on (a) GPM dataset, and (b) ISB datasets.

(b) Optimal Shared Peaks Count: While the GBST algorithm modified to use MST (in place of BST) is slightly better, there is still a gap in performance. This motivates us to formulate the problem of *maximizing the shared peaks counts* with respect to the computed set of multi-charge strong tags. The shared peak count (SPC) is a commonly used and fairly objective criterion for determining the "quality" of de novo peptide sequencing. We also show that we can solve this problem *optimally in polynomial time*.

Suppose that we are given the set, say MST, of strong tags. Define a *multi-charge strong tag path* Q to be a path from v_0 to v_M given by $Q = (q_0\ T_1\ q_1\ T_2\ q_2\ T_3\ q_3\ \ldots\ q_{k-1}\ T_k\ q_k)$ where each T_j is a strong tag in MST and each q_j is a path of at most *two* amino acids, or mass difference that "links" the preceding tag to the succeeding tag in the usual head-to-tail fashion. A strong tag path Q gives rise to a peptide sequence $P(Q)$ obtained by interpreting the "gaps" in the path Q. A example of P(Q) is "[50]CGV[100]PK". Given the peptide sequence $P(Q)$, we can compute the shared peaks count of $P(Q)$. Then our problem can be stated as the following: Among all the possible strong tag paths, we want

to find an *optimal multi-charge strong tag path Q** that *maximize the shared peak count* in the peptide sequence $P(Q^*)$.

Our solution to this problem is to form the graph $G_2(MST)$ defined in the same ways as the graph $G_2(BST)$. We first pre-compute the shared peaks count for each tag in *MST*. For each edge (u, v) connecting two tags T_u and T_v, we compute the path Q of length with at most two amino acids that locally maximizes that shared peak count of Q against experimental spectrum. Then we can compute the path with maximum shared peaks count in the graph $G_2(MST)$, which is a DAG. Additional processing has to be done if either end vertex is not connected to the first (or last) vertex in the path, or the sparse areas are not connectable – we connect this via mass difference. It is easy to see that this algorithm optimizes the shared peaks count among all peptide sequences obtained by extending the multi-charge strong tags in MST via connectivity 2. Next, we present an algorithm that produces provably better result.

Improving the Spared Peaks Counts using *H*(*MST*): We can further improve the shared peaks count if we increase the maximum connectivity *d*. However, this will cause the running time to grow exponentially due to the number of paths to be searched. We propose a graph $H(MST)$, a superset of $G_2(MST)$ which is simple to define, and yet not too computationally expensive. In $H(MST)$, we have an edge from the tail vertex u of T_u to the head vertex v of T_v if the mass difference $(PRM(v)–PRM(u))$ is in the range [57.02, 186.08] Da, where 57.02 and 186.08 are the minimum and maximum mass of an amino acid, respectively. In addition, we can also pre-compute the path from u to v that locally maximizes the shared peak count. For this sub-problem, we have fast procedure that does this efficiently. The length of the computed path from u to v varies depending on the mass difference. The rest of the algorithm is to interpret edges in $H(MST)$.

Algorithm GST-SPC: Finally, our GST-SPC algorithm uses the multi-charge strong tag set *MST* and the graph $H(MST)$ to compute a peptide with optimal *shared peaks count*.

5. Performance Evaluation of Algorithm GST-SPC

We have compared the performance of our algorithms with two other algorithms with freely available implementation, Lutefisk [15] and PepNovo [8]. For specific spectrum and algorithm, the sequencing results with best scores are compared. To compare performance of GST-SPC with the GBST [4], Lutefisk [15], and PepNovo [8], we use the following accuracy measures:

Sensitivity = # correct / $|\rho|$ Tag-Sensitivity = # tag-correct / $|\rho|$

Specificity = # correct / $|P|$ Tag-Specificity = # tag-correct / $|P|$

where #correct is the "*number of correctly sequenced amino acids*" and #tag-correct is "*the sum of lengths of correctly sequenced tags* (*of length > 1*)". The number of correctly sequence amino acids is computed (approximated) as the longest common subsequence (lcs) of the correct peptide sequence ρ and the sequencing result P. The *sensitivity* indicates the quality of the sequence with respect to the correct peptide sequence and a high sensitivity means that the algorithm recovers a large portion of the correct peptide.

The *tag-sensitivity* accuracy takes into consideration of the continuity of the correctly sequenced amino acids. For a fairer comparison with algorithms like PepNovo that only outputs the highest scoring tags (subsequences) we also use specificity and tag-specificity measures, which measures how much of the results are correct.

The comparison of the different algorithms based on the four accuracy measures is summarized in Figure 3 (for the GPM datasets) and Figure 4 (for the ISB datasets). Overall, the results obtained by our GST-SPC algorithm using the shared peaks count scoring functions are promising. On the GPM datasets, the GST-SPC outperforms the other algorithms. For example, it has higher sensitivity than Lutefisk (by 10% for charge ≥ 2) and PepNovo (by about 10%) in sensitivity and tag-sensitivity. It has comparable specificity and tag-specificity PepNovo for charge 1 and 2. It is constantly better than GBST and Lutefisk (for charge > 1) on all accuracy measures.

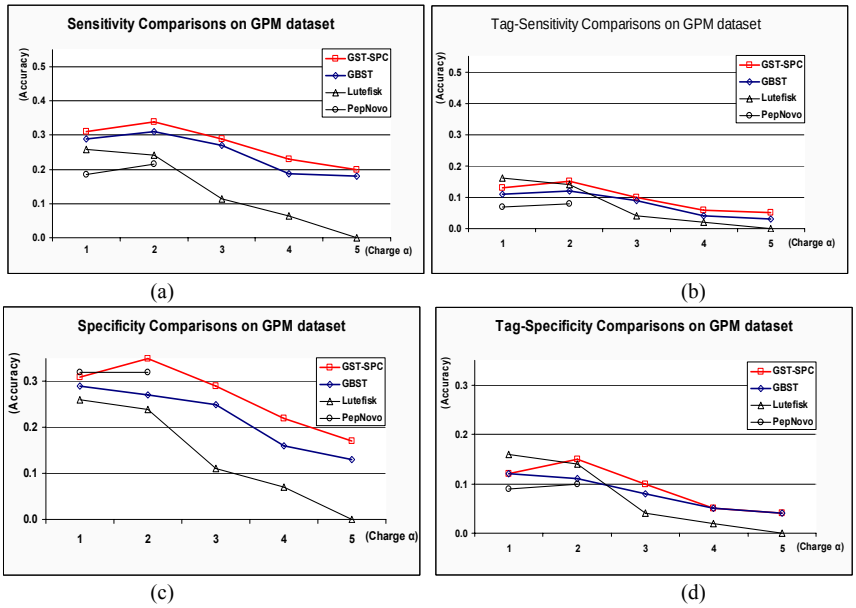


Figure 3. Comparison of different algorithms on GPM dataset – based on (a) sensitivity, (b) tag-sensitivity, (c) specificity and (d) tag-specificity. PepNovo only has results for charge 1 and 2.

For the ISB dataset, the results shows that the ranking as follows: (PepNovo, GST-SPC, GBST, Lutefisk) for all the accuracy measures. The ISB datasets contains many noises and PepNovo has a sophisticated scoring function that may account for its best performance, especially on datasets with charge 1. For spectra with charge 2, the difference in performance is not as high. However, since PepNovo do not (as yet) handle spectra with charge greater than 2, there was no way to compare results for charge 3. That comparison would be interesting given the apparent trend exhibited in the results.

We also compare the algorithm with respected to the number of *completely correct* identified peptide sequences. Our results (not shown here due to space limitations) show

that the GST-SPC algorithm out-performs Lutefisk, but is slightly worse than PepNovo. We have also listed (in Table 1) a few sample "good" interpretations of the GST-SPC algorithm, on which Lutefisk does not provide good results. It is interesting to note that GST-SPC algorithm can identify more correct amino acids – illustrating the power of using multi-charge strong tags.

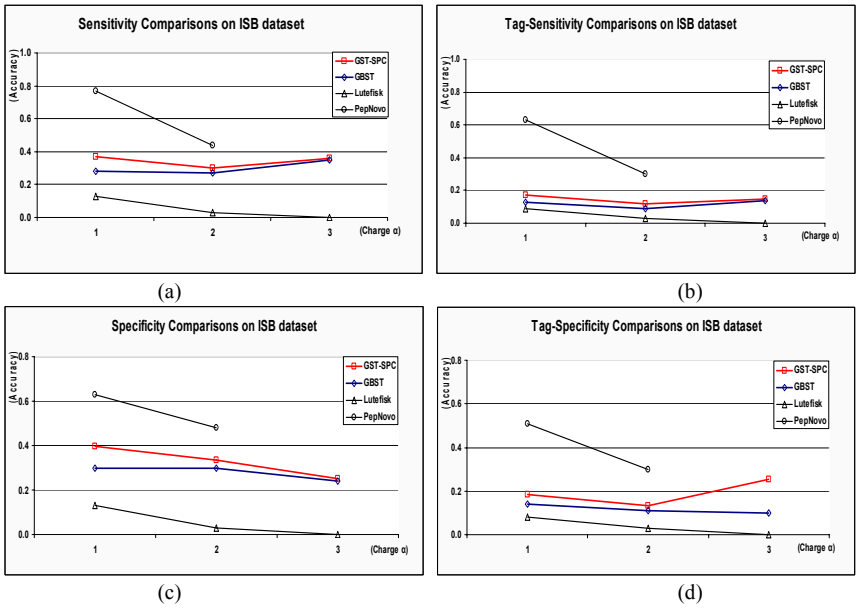


Figure 4. Comparison of different algorithms on ISB dataset – based on (a) sensitivity, (b) tag-sensitivity, (c) specificity and (d) tag-specificity. PepNovo only has results for charge 1 and 2.

Table 1: The sequencing results of Lutefisk, PepNovo and GST-SPC algorithm on some spectra. The accurate subsequences are labeled in bold and red. "-" means there is no result.

M/Z	Z	Real	Lutefisk	PepNovo	GST-SPC
1219.8	2	VAQLEQVYIR	[170.1]E***LEKVYLR***	GL***QLEQVYLR***	AVE***IEQVYIR***
1397.9	2	ELEEIVQPIISK	[242.1]***EEL***AVG[LP]***LSK***	***EELVKPLLSK***	***EIEEI***A[101.0]QH***ISK***
1644.9	2	PAAPAAPAPAEKTPVKK	[AP]***AAPA***[HS]AP[198.1]***PA***AA[CS]	***AAPA***DFEAMTNLPK	***APAAPAPA***[56.1]APAMTKVPK
1838.8	3	SSYSLSGWYENIYIR	[172.1]L[303.2][243.1][NP][MT]***LYLR***	-	***SS***IYI[27.3]IIEPCE***IYIR***
2000.2	4	PAAPAAPAPAEKTPVKKKAR	[323.1]R***PA***[AP]EKTN[LP]K[199.1]R	-	***APAAPA***MWNYNHKPYIR
1936.1	4	SIRVTQKSYKVSTSGPR	[199.1][PW][259.1]L[250.1]***KVSTSGPR***	-	VVIS***VTQK***[63.8]W***KVSTSGPR***
2101.1	4	KIETRDGKLVSESSDVLPK	[243.1]***LV***R[TY]YT***SESS***AE[PV]R	-	IKQHTHECY***SESSDVIPK***
2359.0	5	CDKDLDTLSGYAMCLPNLTR	-	-	AF***CDYA***[417.2]RNQKIRCP***TR***

6. Conclusion

In this paper, we propose a novel algorithm, GST-SPC for de novo sequencing of multi-charge MS/MS spectra. Our algorithm is based on the idea of using multi-charge strong tags to assist in reducing the size of the problem space to be searched. For a fixed set of strong tags, the GST-SPC algorithm optimizes the shared peaks count among all possible augmentations of the tags to form peptide sequences. The experimental results on ISB and GPM datasets show that GST-SPC is better than the GBST algorithm and Lutefisk. Against PepNovo, it performs better on GPM datasets and is worse on the ISB datasets. We have also showed theoretical upper bound results for our algorithms.

However, it is interesting to note that none of these algorithms is close to the theoretical upper bound of the sensitivity (based on Δ^R restriction) shown in Figure 2. This indicates that there is hope that there can be an algorithm based on *MST* that outperforms all of these algorithms. Since there is still room for algorithms based such ideas to improve, the idea itself is very promising. Other research directions are also possible – we are currently looking at more flexible method to connect strong tags rather than the head-to-tail manner, for example; and statistical significance (rather than SPC) of the strong tags and peptide sequencing results are also important for us to investigate.

Acknowledgments

The authors would like to thank Pavel Pavzner and Ari Frank of UCSD for insightful discussion and help with the PepNovo program. The work was partially supported by the National University of Singapore under grant R252-000-199-112.

References

1. Bandeira, N., Tang, H., Bafna, V. and Pevzner, P. Shotgun Sequencing by tandem mass spectra assembly. *Analytical Chemistry*, 76:7221-7233, 2004.
2. Birkinshaw, K. Deconvolution of mass spectra measured with a non-uniform detector array to give accurate ion abundances. *J Mass Spectrom.*, 38:206-210, 2001.
3. Chen, T., Kao, M.-Y., Tepel, M., Rush, J. and Church, G. M. A Dynamic Programming Approach to De Novo Peptide Sequencing via Tandem Mass Spectrometry. *Journal of Computational Biology*, 8:325-337, 2001.
4. Chong, K. F., Ning, K. and Leong, H. W. De Novo Peptide Sequencing For Multiply Charged Mass Spectra. *To appear APBC2006*, 2006.
5. Craig, R., Cortens, J. P. and Beavis, R. C. Open source system for analyzing, validating, and storing protein identification data. *J Proteome Res.*, 3:1234-1242, 2004.
6. Dancik, V., Addona, T., Clauser, K., Vath, J. and Pevzner, P. De novo protein sequencing via tandem mass-spectrometry. *J. Comp. Biol.*, 6:327-341, 1999.
7. Eng, J. K., McCormack, A. L. and John R. Yates, I. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *JASMS*, 5:976-989, 1994.
8. Frank, A. and Pevzner, P. PepNovo: De Novo Peptide Sequencing via Probabilistic Network Modeling. *Anal. Chem.*, 77:964 -973, 2005.
9. Han, Y., Ma, B. and Zhang, K. SPIDER: Software for Protein Identification from Sequence Tags with De Novo Sequencing Error. *2004 IEEE Computational Systems Bioinformatics Conference (CSB'04)*, 2004.
10. Keller, A., Purvine, S., Nesvizhskii, A. I., Stolyar, S., Goodlett, D. R. and Kolker, E. Experimental protein mixture for validating tandem mass spectral analysis. *OMICS*, 6:207-212, 2002.
11. Ma, B., Zhang, K., Hendrie, C., Liang, C., Li, M., Doherty-Kirby, A. and Lajoie, G. PEAKS: Powerful Software for Peptide De Novo Sequencing by MS/MS. *Rapid Communications in Mass Spectrometry*, 17:2337-2342, 2003.
12. Mann, M. and Wilm, M. Error-tolerant identification of peptides in sequence databases by peptide sequence tags. *Analytical Chemistry*, 66:4390-4399, 1994.
13. Perkins, D. N., Pappin, D. J. C., Creasy, D. M. and Cottrell, J. S. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, 20:3551-3567, 1999.
14. Tabb, D., Saraf, A. and Yates, J. r. GutenTag: high-throughput sequence tagging via an empirically derived fragmentation model. *Anal Chem.*, 75:6415-21, 2003.
15. Taylor, J. A. and Johnson, R. S. Implementation and uses of automated de novo peptide sequencing by tandem mass spectrometry. *Anal Chem.*, 73:2594-2604, 2001.
16. Zheng, H., Ojha, P. C., McClean, S., Black, N. D., Hughes, J. G. and Shaw, C. Heuristic charge assignment for deconvolution of electrospray ionization mass spectra. *Rapid Commun Mass Spectrom.*, 17:429-436, 2003.