

# Spectrum Fusion: Using Multiple Mass Spectra for De Novo Peptide Sequencing

Ritendra Datta<sup>1</sup> and Marshall Bern<sup>2</sup>

<sup>1</sup> Penn State University, University Park, PA 16801, USA  
datta@cse.psu.edu

<sup>2</sup> Palo Alto Research Center, Palo Alto, CA 94304, USA  
bern@parc.com

**Abstract.** We report on a new algorithm for combining the information from several mass spectra of the same peptide. The algorithm automatically learns peptide fragmentation patterns, so that it can handle spectra from any instrument and fragmentation technique. We demonstrate the utility of the algorithm, and the power of multiple spectra, by showing that combining pairs of spectra (one CID and one ETD) greatly improves *de novo* sequencing success rates.

## 1 Introduction

There are two basic approaches to peptide sequencing by tandem mass spectrometry (MS/MS): *database search* [11], which identifies the sequence by finding the closest match in a protein database, and *de novo sequencing* [3], which attempts to compute the sequence from the spectrum alone. *De novo* sequencing actually started first, but with the explosion of genomically derived protein sequences, database search quickly became the dominant approach, because it can make identifications from lower-quality spectra with less complete fragmentation. There are, however, good reasons to continue the pursuit of *de novo* sequencing. First, if databases are of low quality, as in the case of unsequenced organisms [19], then *de novo* analysis can outperform database search. Second, sometimes the unknown peptide—rather than a parent protein—is itself the object of interest, e.g., toxins [2], neurotransmitters, and hormones. Third, protein design techniques, such as directed mutation and recombination, often produce proteins without knowing exactly which gene produced them, and the protein sequence may have to be obtained directly through chemical and mass spectrometric sequencing.

Algorithm designers factor *de novo* sequencing into two subproblems: *candidate generation* and *scoring*. Candidate generation typically uses a graph algorithm, such as a longest- or best-path algorithm [6,8,17,22], to compute 1000s of possible sequences. The scoring phase then scores each of these candidates, using more detailed information such as the fragmentation propensities of residues [10,16], mass measurement recalibration [5], and so forth, that would be difficult to incorporate into the candidate generation phase. Because *de novo* sequencing requires essentially complete fragmentation, new fragmentation techniques (microwave assisted acid hydrolysis, IRMPD, ECD, and ETD) offer the best hope of performance improvement. The most interesting of the new techniques is electron-transfer dissociation (ETD) [20], because it is commercially

available, fast enough to be used with ion-trap instruments, and gives quite different, and hence complementary, information to the standard technique of collision-induced dissociation (CID). ETD reduces charge as it induces fragmentation, and hence it gives good fragmentation for highly charged (+3 and +4) parent ions, not-so-good fragmentation for +2 parents, and neutralizes and loses +1 parents.

In this paper, we give a generic algorithm for *spectrum fusion*, combining information from more than one spectrum of the same peptide. We apply the algorithm to *de novo* sequencing of peptides from pairs of spectra collected on a Thermo Electron LTQ instrument, run in a mode that alternates between CID and ETD fragmentation. We show significant improvements in *de novo* sequencing, approximately doubling the number of sequences that could be identified exactly.

Spectrum fusion has previously been done with special-case algorithms. Zhang and McElvain [24] used CID MS/MS and MS<sup>3</sup> pairs, and Bandeira et al. used overlapping [2] or differentially modified [1] MS/MS spectra, for *de novo* sequencing. Finally, and most relevant to the present work, Zubarev and collaborators pioneered the use of CID/ECD pairs for *de novo* sequencing [18]. ECD (electron-capture dissociation) gives similar spectra to ETD, but is less efficient, so it cannot generally be used with ion-trap instruments, only with expensive FTICR instruments. These instruments have about 100-fold better mass accuracy and resolution than ion-trap instruments, but take about 10 times longer to acquire each spectrum. Due to the high mass accuracy, *de novo* sequencing on FTICR is much easier than on ion-trap instruments. Indeed, Savitski et al. [18] achieve reasonable results from CID/ECD pairs using a simple greedy algorithm to compute a single approximate longest path.

## 2 Algorithms

We developed a generic algorithm for combining the information from multiple fragmentation spectra of the same peptide. The output is a *synthetic spectrum* with peaks at integer masses, representing the likelihood that the mass is equal to the sum of the (integer parts of) amino acid residue masses of a prefix of the peptide sequence. The synthetic spectrum is used as the input for candidate generation. By building a synthetic spectrum containing only prefixes, rather than prefixes and suffixes, we use spectrum fusion both to improve fragmentation completeness and to separate prefixes from suffixes. We previously applied graph partitioning [5] to peak separation; with complementary spectra (CID/ETD pairs) a global approach like graph partitioning is unnecessary. We have not yet applied spectrum fusion to scoring; for this phase we used ByOnic [4], our database-search tool, and scored the multiple spectra independently.

We demonstrate spectrum fusion on CID/ETD pairs, but the algorithm could be applied to other combinations of spectra and or other types of biomolecules (for example, glycans). The fusion algorithm is fully automated, so that the algorithm determines the information in the various spectra and peaks, with minimal dependence on prior knowledge. CID fragmentation patterns and peak intensities have been mapped [10,16,21], but no such statistical studies have been published for ETD.

At the core of our spectrum fusion algorithm lies a supervised learning phase that relieves dependence on prior knowledge. A sample consists of  $C$  MS/MS spectra and the corresponding peptide (reliably identified by database search and knowledge of the

biological material), where  $C$  is the number of spectra per peptide. Let us denote by  $\mathcal{S}_{i,c}$ ,  $c \in \{1, \dots, C\}$ , the spectrum of type  $c$  for sample  $i$ , consisting of mass over charge ( $m/z$ ) values and corresponding peak intensities. Let the measured parent mass be  $M_i$  and the “ground truth” peptide string be  $\mathcal{P}_i$ . (Following convention,  $M_i$  denotes M+H mass, the sum of residue masses + 19 for water + 1 for proton.) Thus our labeled data, which can be split into training and test sets, consists of tuples of the form  $(\mathcal{S}_{i,1}, \dots, \mathcal{S}_{i,C}, M_i, \mathcal{P}_i)$ ; the task is to predict prefix masses of  $\mathcal{P}_i$  from  $\mathcal{S}_{i,C}$  and  $M_i$ .

We make the simplifying assumption that the parent mass  $M_i$  is correct to  $\pm 0.5$  Daltons (Da). In reality, the estimated parent mass  $M'_i$  from ion-trap instruments may be off by 1 or 2 Da for peptides of charge +2, and by as much as  $\pm 7$  Da for peptides with greater charge. Accurate parent masses, however, can be obtained in various ways: by a high-resolution single-MS scan with another mass analyzer (e.g. Orbitrap), by a “zoom scan” with the same ion-trap mass analyzer, or by software parent mass correction. CID/ETD pairs offer improved software correction; see the Appendix. The training phase of spectrum fusion consists of two steps (Sections 2.1 and 2.2):

**Training:** (**Input:** Training data  $(\mathcal{S}_{i,1}, \dots, \mathcal{S}_{i,C}, M'_i, \mathcal{P}_i)$ , **Output:** Features, Model)

1. *Feature Selection:* Pick informative features across the different types of spectra.
2. *Statistical Learning:* Learn a probabilistic model on the selected features.

Then, an unknown sample  $(\mathcal{S}_{i,1}, \dots, \mathcal{S}_{i,C}, M'_i)$  is processed through a pipeline which includes spectrum fusion, candidate generation, and scoring (Sections 2.3 and 2.4):

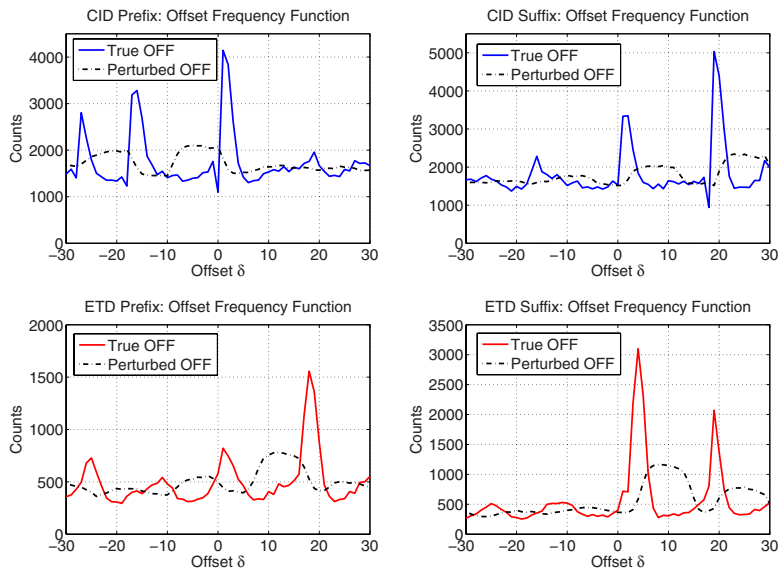
**De Novo Sequencing:** (**Input:** Test data  $(\mathcal{S}_{i,1}, \dots, \mathcal{S}_{i,C}, M'_i)$ , **Output:** Guess  $\mathcal{P}'_i$ )

1. *Parent Mass Correction:* Correct the reported parent mass  $M'_i$  to get  $M_i$ .
2. *Spectrum Fusion:* Generate one synthetic spectrum  $\widehat{\mathcal{S}}_i$  from spectra  $(\mathcal{S}_{i,1}, \dots, \mathcal{S}_{i,C})$ ; the synthetic spectrum should ideally contain most integer prefix masses, but few suffix or noise masses.
3. *Candidate Generation:* Compute the  $K$  best paths in  $\widehat{\mathcal{S}}_i$  to generate candidates.
4. *Scoring:* Score candidate peptides by summing ByOnic scores for each spectrum.

## 2.1 Feature Selection Using Offset Frequency Functions

The dominant peaks in CID spectra are well-understood and have known, fixed, relationships to prefix and suffix masses. For example, if a prefix mass (sum of residue masses of an initial subsequence of the peptide) is  $w$ , then there will usually be a spectral peak (called the “b-ion”) at  $w + 1$  and another peak at  $w + 2$  (the “isotope peak” containing one  $^{13}\text{C}$ ). Our goal was to introduce an algorithmic framework that would automatically “learn” these features. For this problem, machine learning offers many advantages: it can learn weights and dependencies among features; it can learn features for new techniques such as ETD; and it can make use of more, and more subtle, features.

Our features were selected from the *offset frequency function* (OFF), proposed by Dančik et al. [8]. The OFF is a histogram, giving the number of times a spectral peak is observed at a given integer mass offset from a known prefix or suffix mass. Suppose an ETD training-set spectrum  $\mathcal{S}$  consists of observed peaks at  $\{s_1, \dots, s_n\}$ , and the ground-truth peptide contains real-valued prefix masses at  $w_1, w_2, \dots, w_k$ . We compute the OFF as follows. If a prefix mass  $w_j$  is separated from  $s_i$  by an integer mass  $\delta$



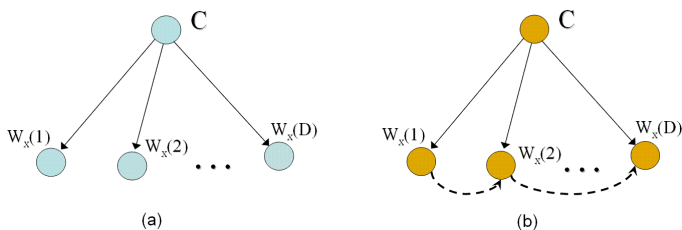
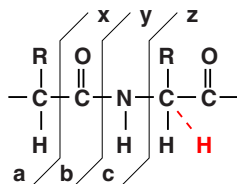
**Fig. 1.** Plots of Offset Frequency Functions for prefix/suffix and CID/ETD fragmentation types

(within a tolerance  $\epsilon$ , set to 0.2 Da for our ion-trap data), then we increment the count for  $\delta$  for the combination (ETD, prefix). Similarly, if  $M - 19 - w_j$ , where  $M$  is the parent mass of the peptide, is separated from  $s_i$  by  $\delta$ , then we increment the count for  $\delta$  for (ETD, suffix). With an estimate of  $M$  accurate to the closest integer, the suffix OFF provides valuable corroborating evidence to the presence of a prefix at  $w$ . (Without an accurate  $M$ , the suffix OFF would still be useful but somewhat blurry.)

We determine which  $\delta$  offsets are most informative in an automated manner. For each combination of prefix/suffix and type of spectrum (CID, ETD, etc.), we generate a *perturbed OFF* in which the theoretical prefix and suffix masses are shifted by a uniform random sample in the range 1 – 10 Da; this OFF is used to estimate the statistics of insignificant OFF counts. We limit the computation of both OFFs, true and perturbed, to  $\pm 30$  Da offsets, since  $\delta$  values beyond this range are unlikely to be of any interest (Figure 1). Let the true OFF be  $\{v_{-30}, \dots, v_0, \dots, v_{+30}\}$ , and the deliberately perturbed OFF be  $\{v'_{-30}, \dots, v'_0, \dots, v'_{+30}\}$ , where each  $v$  or  $v'$  represents a count. We estimate the mean  $\mu$  and variance  $\sigma^2$  of insignificant OFF counts from the perturbed OFF, and compute a *z-score* for each count in the true OFF to determine informative peaks. More specifically, we say that  $\delta$  is an informative offset if its *z-score*  $(v_\delta - \mu) / \sigma$  exceeds 3, where  $\mu = \frac{1}{61} \sum_{j=-30}^{30} v'_j$ , and  $\sigma^2 = \frac{1}{60} \sum_{j=-30}^{30} (v'_j - \mu)^2$ . We denote the final set of informative offsets across prefixes/suffixes and all fragmentation methods by  $\hat{\Delta}$ , which is a set of  $D$  offsets,  $\{T_1, T_2, \dots, T_D\}$ . Each offset  $T_d$  is specified by a triple of fragmentation type (CID, ETD, etc.), orientation (prefix or suffix), and integer offset  $\delta$ . These triples comprise our set of selected *features*, for use in the model-based classification step that follows. As shown in Table 1, automatic feature selection discovered

**Table 1.** Set  $\hat{\Delta}$  of 21 features automatically selected by our algorithm for ETD and CID spectra. Mass offsets are relative to the sum of residue masses, so that the b-ion has an offset of +1 and the y-ion an offset of +19. In the interpretations, we use n to denote a neutron (that is,  $^{13}\text{C}$ ), H for hydrogen, and so forth. Ion naming follows the standard Biemann naming convention, but we use z-ion to denote the stable ion with one extra hydrogen (red), which is sometimes called the “z+1 ion”. Data used consists of 724 peptides with 12,574 prefixes and suffixes.

Frag'n	Pref/Suff	Offset	z-score	Interpretation
CID	Suffix	+19	11.68	y-ion
ETD	Suffix	+4	10.97	z-ion + n or H
CID	Prefix	+1	10.44	b-ion
CID	Suffix	+20	9.45	y-ion + n
ETD	Prefix	+18	9.39	c-ion
CID	Prefix	+2	9.15	b-ion + n
ETD	Prefix	+19	7.68	c-ion + n or H
ETD	Suffix	+5	7.62	z-ion + 2H, H + n, or 2n
ETD	Suffix	+3	7.04	z-ion
ETD	Suffix	+19	6.54	y-ion
CID	Prefix	-16	6.77	b-ion - H <sub>2</sub> O + n or NH <sub>3</sub>
CID	Prefix	-17	6.36	b-ion - H <sub>2</sub> O
CID	Suffix	+2	5.72	y-ion - H <sub>2</sub> O + n or NH <sub>3</sub>
ETD	Prefix	+17	5.71	c-ion - H ??
CID	Suffix	+1	5.68	y-ion - H <sub>2</sub> O
CID	Prefix	-27	4.75	a-ion
CID	Suffix	+21	4.43	y-ion + 2n
CID	Prefix	-15	4.24	b-ion - H <sub>2</sub> O + 2n or NH <sub>3</sub> + n
CID	Prefix	+3	3.93	b-ion + 2n
ETD	Suffix	+20	3.54	y-ion + n or H
ETD	Prefix	+20	3.45	c-ion + 2H, H + n, or 2n



**Fig. 2.** (a) In the naïve Bayes structure, each binary attribute variable  $W_x(d)$  (presence/absence of a peak at offset  $d$ ) depends only on the class variable  $C$ . (b) The TAN structure allows each attribute variable an in-degree of up to 2, thus generalizing (a) to allow more dependencies.

and ranked all the well-known ions such as b- and y-ions in CID spectra, and also discovered some less-known phenomena, such as the high intensity of isotope (or “neutral-gain”) peaks and the rarity of neutral losses in ETD spectra.

## 2.2 Combining Features with a Tree-Augmented Naïve Bayes Network

With the set  $\hat{\Delta} = \{T_1, \dots, T_D\}$  of informative offsets automatically determined, the next step is to use them to learn a statistical model to perform a binary classification: For each integer  $x$  within the range of 1 to parent mass  $M$ , decide if  $x$  is a prefix mass or not. The feature vector for this binary classifier consists of a length- $D$  binary vector  $W_x$ , with each element (attribute variable) indicating the absence or presence (0 or 1) of a peak (within a mass tolerance  $\epsilon$ ) at the corresponding offset. More specifically, a CID prefix entry of  $W_x$  is set to 1 if there is a peak in the CID spectrum at position  $x + \delta$ , and a suffix entry is set to 1 if there is a peak at  $M - 19 - x + \delta$ , where  $\delta$  are the informative integer offsets. ETD entries of  $W_x$  are treated analogously.

There is one subtlety in converting real-valued-masses to integer-masses in the spectra. A real-valued mass  $M$  should be rounded to the closest integer of  $0.9995M$  in order to remove the characteristic mass defect (fractional part) of a peptide. Thus 1814.1 Da rounds to 1813. We used this correction wherever required, so that integer amino acid residue masses sum to M+H peptide masses (minus 19).

One straightforward approach to the classification problem would be to use a naïve Bayes classifier [9], meaning that, given the class variable, each feature is assumed conditionally independent of every other feature (Figure 2). Denoting by indicator variable  $prf_x$  whether  $x$  is a prefix mass or not, the class conditional is written as follows:

$$\text{Prob}(prf_x = 1 | W_x) \propto \text{Prob}(prf_x = 1) \cdot \prod_{d=1}^D \text{Pr}(W_x(d) | prf_x = 1).$$

To do away with the proportionality constant, classification can be done based on some thresholding of the odds  $\text{Prob}(prf_x = 1 | W_x) / \text{Prob}(prf_x = 0 | W_x)$ . This model is computationally efficient and easy to estimate, but the assumption of conditional independence of the attributes is too strong. There are obvious dependencies among the attribute variables, as isotope peaks are almost sure to co-occur with peptide peaks. This motivated us to explore a generalization of the naïve Bayes classifier to a richer network structure that allows dependencies. One such generalization is the *tree-augmented* naïve Bayes classifier (TAN) [14], shown in Figure 2(b), which allows each attribute variable to depend on the class variable along with at most one other variable. The TAN model has many attractive properties: (1) An in-degree of two should capture the most important interactions among spectral peaks; an isotope or neutral-loss peak depends on the monoisotopic peak, but has little dependence upon peaks at other cleavages. (2) Unlike general Bayes nets, TAN is efficiently estimated in polynomial time, and inferencing is fast once the structure is known. (3) TAN is simple enough to be visually interpretable.

We estimate the TAN structure based on the polynomial-time *Construct-TAN* algorithm from [14], which in turn is based on earlier work on second-order product approximation of discrete joint distributions [7]. Learning the TAN structure for prefix/non-prefix classification is presented in Algorithm 1. The learning time is  $O(n^2 D)$ , where  $n$  is the total number of positive and negative examples (binary vectors) used for training. Note that the TAN so constructed is optimal, in the sense that of all network structures possible given the TAN restrictions, the one obtained maximizes the likelihood given

**Algorithm 1.** Learning a TAN Structure for Prefix/Non-Prefix Classification

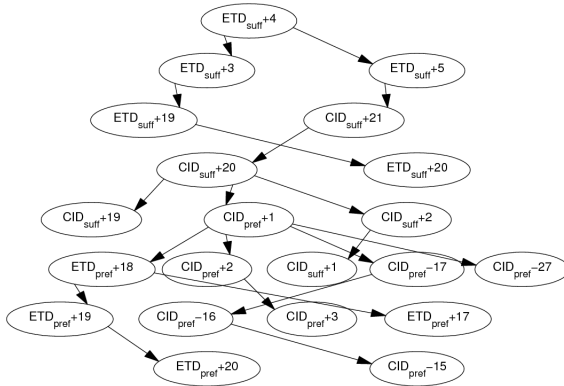
- Require:** Selected features  $\widehat{\Delta} = \{T_1, \dots, T_D\}$ , spectra of multiple types  $(S_{i,1}, \dots, S_{i,C}, M'_i)$  for known peptides.
- 1: To create positive examples, for each peptide  $P_i$ , and each prefix position  $x$  for it, compute length  $D$  binary vector  $W_x$  based on presence/absence of the selected features  $\widehat{\Delta} \rightarrow$  *True vectors*.
  - 2: To create negative examples, perturb each such true prefix position by a uniformly sampled integer between (1-10) to simulate non-prefix positions, and compute binary vector  $W'_x$  as before  $\rightarrow$  *Perturbed vectors*.
  - 3: Empirically compute *conditional mutual information* between attribute pairs,  $j, k \in \{1, \dots, D\}$ ,  $j \neq k$ :

$$I_P(T_j; T_k | prf) = \sum_{\substack{w(j) \in \{0,1\} \\ w(k) \in \{0,1\} \\ p \in \{0,1\}}} \text{Prb}(w(j), w(k), prf=p) \cdot \log \frac{\text{Prb}(w(j), w(k) | prf=p)}{\text{Prb}(w(j) | prf=p)} \cdot \text{Prb}(w(k) | prf=p)$$

where  $\text{Prb}(- | prf = 1)$  are estimated from *true vectors*, and  $\text{Prb}(- | prf = 0)$  from *perturbed vectors*.

- 4: Build full undirected graph  $\mathcal{G}$  with nodes  $\{T_1, \dots, T_D\}$ , edge weight  $-I_P(T_j; T_k | prf)$  between  $T_j$  and  $T_k$ .
- 5: Apply Prim's Algorithm to find the minimum spanning tree in  $\mathcal{G}$  (with the -ve edge weights  $\Rightarrow$  a *max.* spanning tree).
- 6: Select a node in  $\mathcal{G}$  arbitrarily and set all edges outward from it, to get directed graph  $\mathcal{G}'$ .
- 7: Add class variable  $prf_x$  to  $\mathcal{G}'$  as a node, and direct edges from it to each  $T_j \rightarrow$  desired TAN structure (e.g., Fig. 2 (b))

the training data. The structure can be described by the set of parents of each attribute  $T_d$ , which includes the class variable  $prf$ , and at most one other attribute, which we refer to as  $T_{d'}$ . An example TAN structure estimated over CID/ ETD pairs is shown in Figure 3. It is worth noting that PepNovo [13] also uses a tree of dependencies for scoring CID spectra; however, PepNovo's dependencies were determined manually and only the weights were learned automatically.



**Fig. 3.** The TAN structure for the CID/ETD spectra pairs, estimated over 724 peptides using Algorithm 1. Dependencies are interpretable, e.g., the top of the TAN shows that the monoisotopic z-ion and the +2 isotope both depend upon the +1 isotope. The connection from  $CID_{pref}+1$  to  $CID_{pref}-27$  implies that a-ion depends upon b-ion. Class variable  $prf$  is omitted to avoid clutter.

The training process is completed by empirical estimation of  $\text{Prob}(W(d) | prf=1)$  and  $\text{Prob}(W(d) | prf=0)$  if  $T_d$  has only one parent, or  $\text{Prob}(W(d) | prf=1, W(d'))$

and  $\text{Prob}(W(d) \mid \text{prf}=0, W(d'))$  if  $T_d$  has two parents. As suggested in [14], the empirical estimates are smoothed to avoid poor estimates from a limited sample size  $N$ :

$$\text{Prob}(W(d) \mid \text{prf}_x = 1) \leftarrow \frac{N}{N+5} \text{Prob}(W(d) \mid \text{prf}_x = 1) + \frac{5}{N+5} \text{Prob}(W(d)).$$

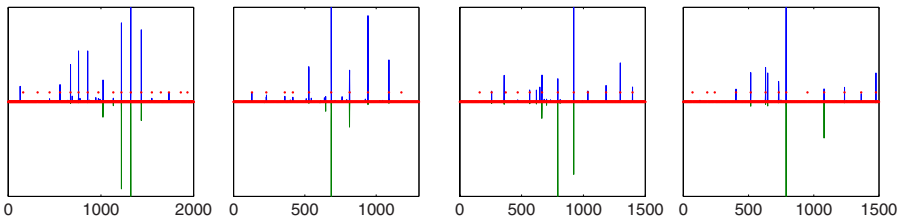
Probability estimates for attributes with two parents are smoothed similarly. Here we have made a reasonable choice of prior for the conditional, the marginal distribution. The classification of a mass  $x$  as prefix or not is ultimately based on the odds  $\phi_x$ ,

$$\phi_x = \frac{\text{Prob}(\text{prf}_x=1 \mid W_x)}{\text{Prob}(\text{prf}_x=0 \mid W_x)} = \frac{\text{Prob}(\text{prf}_x=1)}{\text{Prob}(\text{prf}_x=0)} \cdot \prod_{d=1}^D \frac{\text{Prob}(W_x(d) \mid \text{prf}_x=1[, W_x(d')])}{\text{Prob}(W_x(d) \mid \text{prf}_x=0[, W_x(d')])}.$$

### 2.3 Producing a Fusion Spectrum

The trained classifier can be used to combine multiple spectra,  $S_{i,c}$ ,  $c \in \{1, \dots, C\}$ , with integer parent mass  $M_i$ , into a single synthetic spectrum  $\hat{S}_i$  (Figure 4), which contains most prefix masses but as few other types of peaks as possible. The steps:

1. Initialize spectrum  $\hat{S}_i$  to be an empty (no-peak) spectrum with mass range 1 to  $M_i$ .
2. In each  $S_{i,c}$ ,  $c \in \{1, \dots, C\}$ , the intensities at  $x$  are replaced by rank-based intensities, namely  $\max\{0, 200 - rk(x)\}$ , where  $rk(x)$  is descending order rank.
3. For each integer  $x$  in the range 1 to  $M_i$ , a binary vector  $W_x$  is created based on the feature set  $\hat{\Delta}$  using spectra  $S_{i,c}$ ,  $c \in \{1, \dots, C\}$ , indicating presence/absence of peaks at the  $D$  offsets. The vector  $W_x$  is used to compute the odds  $\phi_x$  as above.
4. The  $M_i/10$  positions with greatest  $\phi_x > 1$  (odds in favor of position being a prefix mass) are picked to be synthetic peaks. Peak intensities are set to the sum of intensities corresponding to the  $D$  offsets in the various spectra (with nothing added if no peak is present at a given offset).
5. Some peaks are removed since they cannot be prefix masses: (a) those from 1 to 56 Da, (b) those from  $M_i - 19 - 56$  to  $M_i$ , and (c) masses within 250 Da of either 0 or  $M_i - 19$  that cannot be completed with a sum of amino acid residue masses.
6. Each peak  $x$  is then compared with its complementary peak  $M_i - 19 - x$ , and the peak with lower intensity is removed. We found this step to be very effective in eliminating suffixes while retaining prefixes.



**Fig. 4.** Sample synthetic spectra generated by Naïve Bayes (below, inverted) vs. TAN (above, upright). *Note:* TAN spectra contain many more true prefix masses (red dots) at high intensity.

## 2.4 De Novo Sequencing Steps

The final steps in our *de novo* sequencing algorithm are standard steps with some minor modifications. We first construct a *spectrum graph* [8] on the synthetic integer-mass spectrum  $\widehat{S}_i$ . Denoting the peak masses in  $\widehat{S}_i$  by  $V_i$ , we make a node for each integer in  $\mathcal{V}_i = V_i \cup \{0\} \cup \{M_i - 19\}$ . For each  $a, b \in \mathcal{V}_i$ , we add a directed edge  $a \rightarrow b$  if  $b - a$  equals the mass of one of the amino acids, and label it accordingly.

In order to use a standard  $K$ -shortest-path algorithm [12] (with implementation [15]) to generate candidates, rather than a special-purpose algorithm [6,17], we devised a special edge-weighting scheme. Denoting by  $Int(x)$  the intensity of a peak in  $\widehat{S}_i$  at position  $x$  (trivially 0, if a peak is absent), we have

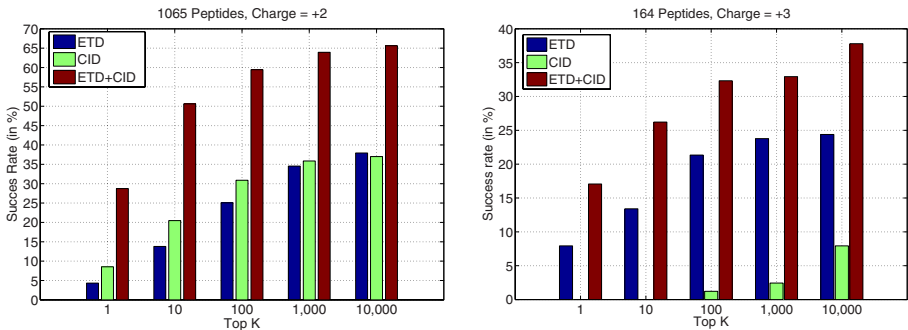
$$Wt.(a \rightarrow b) = \underbrace{\frac{50}{(b-a)}}_{\text{Path hop control}} - \underbrace{(Int(a) + Int(b))}_{\text{Prefix intensity sum}} - \underbrace{\frac{1}{2}(Int(M_i - 19 - a) + Int(M_i - 19 - b))}_{\text{Suffix intensity sum}}$$

The intensity terms in the equation above are negated to convert a longest path problem into one of shortest path. The suffix intensities are added (with lower weight 1/2), because we observed that some suffix peaks remained in the artificial spectrum  $\widehat{S}_i$ , and we wanted to take advantage of their presence by treating them as corroborative evidence. Suppose a peptide is AGPTRK, and let  $a$  and  $b$  correspond to the mass of prefixes AGP and AGPT respectively. While the high intensities at  $a$  and  $b$  do suggest amino acid T, peaks at their complementary positions (suffixes TRK and PTRK) also support the occurrence of T. The first term “path hop control” introduces a small bias toward paths with fewer hops. This bias helps avoid generating peptide candidates containing long sequences of low mass amino acids, so that, e.g., N is generally favored over GG (having same mass) and K over AG and GA. The numerator arbitrarily controls this bias, and a choice of 50 worked well. The bias toward fewer hops can be explained by the fact that  $\frac{1}{x} + \frac{1}{y} > \frac{1}{x+y}$ , when  $x, y > 0$ . This means that, everything else remaining same, an edge of length  $(x + y)$  is favored over two edges of lengths  $x$  and  $y$ .

Each of the  $K$  candidate peptides is then scored using ByOnic [4]. ByOnic scores each candidate against each of the  $C$  different original spectra, and we simply sum these scores and pick the candidate  $\mathcal{P}'_i$  with the highest total. While more complex functions of the  $C$  scores can be used, we have not experimented with any variants.

## 3 Experiments

For a test, we used well-identified CID/ETD pairs of spectra collected by Christopher Becker and Shanhua Lin (PPD, Inc., Menlo Park) on a Thermo Electron LTQ equipped with ETD source. The sample material was human blood plasma, digested with either Lys-C or trypsin, alkylated, reduced, and run through multiple affinity removal system (MARS) and reverse-phase columns. We trained 4 different TANs, one for each choice of digestion (Lys-C or trypsin) and parent charge (+2 or +3/+4). The trypsin data set included 1543 +2 peptides (724 training and 719 test), and 317 +3/+4 peptides (155 training and 162 test). The Lys-C data set included 1025 +2 peptides (520 training and 505 test), 539 +3 peptides (274 training and 264 test), and 178 +4 peptides (50 training

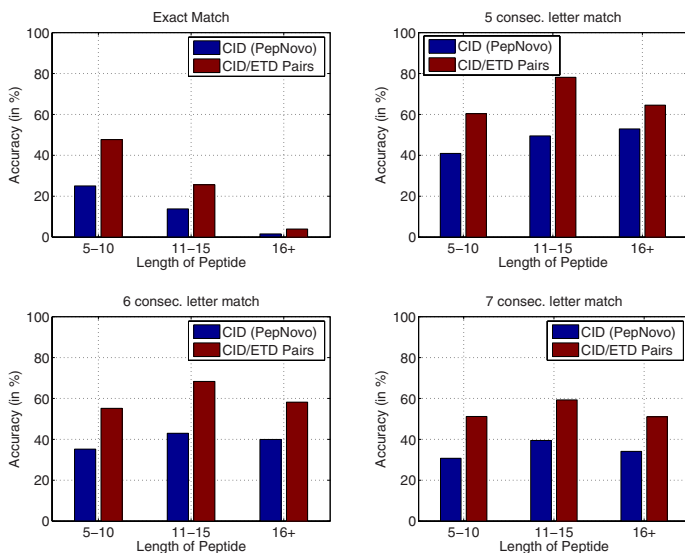


**Fig. 5.** Comparison of our candidate generation algorithm using only one type of spectrum versus the same algorithm using CID/ETD pairs. A successful trial was one in which the correct peptide was in the list of the top  $K$  generated candidates. The average length of the +3 charged peptides is 17, making this a challenging data set.

and 128 test). In all our tests, we considered L and I interchangeable, since they have exactly the same mass and similar chemistry, but we considered K and Q different since their masses differ slightly and chemistries differ considerably.

Figure 5 compares the performance of our candidate generation algorithm run on single spectra, either CID or ETD, with the performance of the same algorithm using CID/ETD pairs. Even with a single input spectrum, the TAN output is enriched in prefix peaks and depleted in suffix and noise peaks, yet candidate generation is much worse than with CID/ETD pairs. We found that for +2 peptides, CID spectra have better fragmentation than ETD spectra, with a median of 75% of the possible b- and y-ions present among the top 200 peaks compared to 44% of the c- and z-ions for ETD. A median of 90% of the cleavages were represented by either a b- or y-ion for CID compared to 81% represented by either a c- or z-ion for ETD. Separating prefix from suffix peaks is somewhat easier for ETD spectra than for CID spectra, due to fewer noise peaks and the frequent co-occurrence of z- and y-ions for ETD suffixes, so that the ETD candidate generation catches up with CID when the number of paths  $K$  is large (Figure 5, left). For +3 parents, CID spectra had worse fragmentation, with medians of 44% of possible b- and y-ions and 71% of cleavages, compared to 64% and 86% for ETD. In a CID/ETD pair, a median of 93% (respectively, 88%) of cleavages are represented by at least one of the four possibilities (b-, y-, c-, z-ion) for +2 (respectively +3) parents. Better fragmentation, along with easier prefix and suffix separation, gave quite dramatic improvement in candidate generation performance for both +2 and +3 peptides.

Figure 6 compares the results of our complete *de novo* sequencing pipeline versus PepNovo [13], which we believe to be the best available *de novo* sequencing program for ion-trap spectra. This experiment is not meant to be a fair comparison of algorithms or software, but rather an assessment of the “bottom-line” advantage of CID/ETD pairs over single CID spectra. Savitski et al. [18] reported that CID/ECD pairs were advantageous, but did not attempt to quantify the performance improvement offered by multiple spectra. In Figure 6, we see that the number of exactly correct peptides increases by about a factor of 2, yet remains at a modest level, and that partially correct peptides



**Fig. 6.** Comparison of the results of our algorithm, which makes use of ETD/CID spectra together, with the results of PepNovo [13] run on the CID spectra only. We counted exact matches (every letter correct except for L and I swaps), and longest consecutive sequences of correct letters.

(with 5, 6, or 7 correct consecutive letters) increases by more than 50%. About 70% of the peptides of lengths 11–15 have 6 or more consecutive letters correct.

## 4 Discussion

Although *de novo* sequencing has been used for various experiments, it is unlikely to become the technique of choice for well-studied complex samples, such as human plasma or tissue. For these samples, it will continue to play an important niche role by identifying polymorphisms and alternate splices.

For complex samples from highly variable organisms such as pathogens, *de novo* sequencing will likely play a more central role. Savitski et al. [18] argue that ECD/CID pairs provide the first “proteomics-grade” *de novo* sequencing, meaning that their processing pipeline makes approximately as many peptide identifications as would a CID-only, database-search strategy employing the same FTICR instrument. The slow duty cycle of CID/ECD-FTICR, however, limits the number of MS/MS spectra that can be acquired on one run. We expect *de novo* sequencing using CID/ETD on LTQ ion-trap instrument to actually outperform CID/ECD-FTICR, if the measure of success is the number of distinct peptides with useful sequence tags (say 6+ letters) per unit time.

Finally, *de novo* sequencing probably is indeed the technique of choice for studies of relatively simple mixtures of peptides or proteins with high biological activity, such as toxins and neurotransmitters. Such peptides are often from unsequenced organisms, have been heavily processed post-translationally, and may vary from strain to strain or even from individual to individual. Such samples are also important enough to warrant

extra work, such as the acquisition of multiple fragmentation spectra per peptide, perhaps many more than two spectra per peptide. The proposed spectrum fusion algorithm would then be useful in automatically extracting the information in the suite of spectra.

## References

1. Bandeira, N., Tsur, D., Frank, A., Pevzner, P.A.: Protein identification by spectral networks analysis. *Proc. Natl. Acad. Sci. USA* 104, 6140–6145 (2007)
2. Bandeira, N., Clauser, K.R., Pevzner, P.A.: Assembly of peptide tandem mass spectra from mixtures of modified proteins. *Molecular Cell. Proteomics* 6, 1123–1134 (2007)
3. Bartels, C.: Fast algorithm for peptide sequencing by mass spectrometry. *Biomedical and Environmental Mass Spectrometry* 19, 363–368 (1990)
4. Bern, M., Cai, Y., Goldberg, D.: Lookup peaks: a hybrid of de novo sequencing and database search for protein identification by tandem mass spectrometry. *Anal. Chem.* 79, 1393–1400 (2007)
5. Bern, M., Goldberg, D.: De novo analysis of peptide tandem mass spectra by spectral graph partitioning. *J. Computational Biology* 13, 364–378 (2006)
6. Chen, T., Kao, M.-Y., Tepel, M., Rush, J., Church, G.M.: A dynamic programming approach to de novo peptide sequencing by mass spectrometry. *J. Computational Biology* 8, 325–337 (2001)
7. Chow, C.K., Liu, C.N.: Approximating discrete probability distributions with dependence trees. *IEEE Trans. on Information Theory* 14, 462–467 (1968)
8. Dančik, V., Addona, T.A., Clauser, K.R., Vath, J.E., Pevzner, P.A.: De novo peptide sequencing via tandem mass spectrometry. *J. Computational Biology* 6, 327–342 (1999)
9. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*. Wiley-Interscience, Chichester (2000)
10. Elias, J.E., Gibbons, F.D., King, O.D., Roth, F.P., Gygi, S.P.: Intensity-based protein identification by machine learning from a library of tandem mass spectra. *Nature Biotechnology* 22, 214–219 (2004)
11. Eng, J.K., McCormack, A.L., Yates III., J.R.: An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *J. Am. Soc. Mass Spectrom.* 5, 976–989 (1994)
12. Eppstein, D.: Finding the  $k$  shortest paths. *SIAM J. Computing* 28, 652–673 (1998)
13. Frank, A., Pevzner, P.: PepNovo: De Novo Peptide Sequencing via Probabilistic Network Modeling. *Anal. Chem.* 77, 964–973 (2005)
14. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian Network Classifiers. *Machine Learning* 29, 131–163 (1997)
15. Graehl, J.: Implementation of David Eppstein's  $k$  Shortest Paths Algorithm., <http://www.ics.uci.edu/~eppstein/>
16. Havilio, M., Haddad, Y., Smilansky, Z.: Intensity-based statistical scorer for tandem mass spectrometry. *Anal. Chem.* 75, 435–444 (2003)
17. Ma, B., Zhang, K., Hendrie, C., Liang, C., Li, M., Doherty-Kirby, A., Lajoie, G.: PEAKS: powerful software for peptide de novo sequencing by tandem mass spectrometry. *Rapid Comm. in Mass Spectrometry* 17, 2337–2342 (2003), <http://www.bioinformaticsolutions.com>
18. Savitski, M.M., Nielsen, M.L., Kjeldsen, F., Zubarev, R.A.: Proteomics-Grade de Novo Sequencing Approach. *J. Proteome Research*, 2348–2354 (2005)
19. Shevchenko, A., et al.: Charting the proteomes of organisms with unsequenced genomes by MALDI-quadrupole time-of-flight mass spectrometry and BLAST homology searching. *Anal. Chem.* 73, 1917–1926 (2001)

20. Syka, J.E., Coon, J.J., Schroeder, M.J., Shabanowitz, J., Hunt, D.F.: Peptide and protein sequence analysis by electron transfer dissociation mass spectrometry. *Proc. Natl. Acad. Sci. USA* 101, 9528–9533 (2004)
21. Tabb, D.L., Smith, L.L., Brechi, L.A., Wysocki, V.H., Lin, D., Yates III, J.R.: Statistical characterization of ion trap tandem mass spectra from doubly charged tryptic digests. *Anal. Chem.* 75, 1155–1163 (2003)
22. Taylor, J.A., Johnson, R.S.: Implementation and uses of automated de novo peptide sequencing by tandem mass spectrometry. *Anal. Chem.* 73, 2594–2604 (2001)
23. Venable, J.D., Xu, T., Cociorva, D., Yates III, J.R.: Cross-correlation algorithm for calculation of peptide molecular weight from tandem mass spectra. *Anal. Chem.* 78, 1921–1929 (2006)
24. Zhang, Z., McElvain, J.S.: De novo peptide sequencing by two-dimensional fragment correlation mass spectrometry. *Anal. Chem.* 72, 2337–2350 (2000)

## Appendix: Parent Mass Correction

In the context of spectrum fusion, the problem of parent mass correction is as follows. The MS instrument outputs a real-valued, nominal parent mass  $M'_i$  for peptide  $\mathcal{P}_i$ , and our goal is to correct it to  $\pm 0.5$  Da. (A precise estimate of the parent mass is not required by our algorithm.) Once again, we have multiple spectra per peptide available for this classification problem, and once again, we use the M+H convention, meaning that we would like the sum of the residue masses plus 19 Daltons, to account for water and one proton. Empirically, we find that the true parent mass  $M_i$  is never beyond  $\pm 7$  Da of  $M'_i$ . Therefore, in each case, there are only 15 candidate integers in the neighborhood of  $M'_i$  that could be the correct answer.

1. **Complementary Peak Sums:** We take pairwise sums of the 30 tallest peaks in the original spectra. (We make adjustments for the M+H convention, and the fact that the sum of an ETD complementary pair is one Dalton greater than the sum of a CID pair.) For those sums that fall within  $M'_i \pm 7$ , we add the sum of intensities for that pair to the “intensity” of the parent mass candidate.
2. **Known Peak Positions:** We observed that spectra often contain peaks at positions that directly reflect the true mass  $M_i$ . For example, CID spectra of peptides with parent charge +2 often contain a peak at  $(M_i + 1 - 18)/2$ , representing the entire peptide minus water, doubly charged. ETD spectra of peptides with parent charge +3 often contain peaks at  $M_i + 2$  and  $M_i - 15$ , for the entire peptide with two protons reduced to hydrogens, and the entire peptide with two protons reduced, minus ammonia. We developed a simple method to automatically deduce such mass cues from different types of spectra, given some training data. Here, we hypothesize in turn each of the 15 candidates as the actual mass, and seek these known cues in the spectra. If a cue is found, a score equal to the intensity of that peak is added to the parent mass candidate.
3. **Suffix/Prefix Alignment:** As in [23], we complement all the peaks in a spectrum relative to the candidate parent mass, and thereby create a mirror spectrum. We expect this spectrum to align with the original spectrum best when the correct mass candidate was used for complementing. Therefore, we take dot products of the

intensities of the original and mirror spectra (separately, for each type of spectrum) and add the dot products as scores for the 15 mass candidates.

Finally, we normalize each of the three scores by the respective maximum values among the candidates, and sum the scores to determine the winning candidate.

Parent mass correction performed in this manner is fairly effective. For the +2 charged peptides, the closest integer to the nominal M+H parent mass (after correction for mass defect) matched the correct mass only 28% of the time. In contrast, our correction leads to about 90% accuracy. For the +3 charged peptides, the nominal mass is correct only about 4% of the time. (Due to isotopes shifting the center of the single-MS peak, the nominal mass is most commonly one or two Daltons too high.) In contrast, our mass correction leads to about 75% accurate parent mass estimation. We also found out that a majority of the times that our mass correction made wrong estimates, even a correct mass estimate would not lead to successful *de novo* sequencing (5 or more consecutive letters correct), suggesting poor quality of such spectra.