A Peptide-Spectrum Alignment Model for Tandem Mass Spectrometry: Extended Version

John T. Halloran
Dept. of Electrical Engineering
University of Washington
Seattle, WA 98195, USA

Jeff A. Bilmes
Dept. of Electrical Engineering
University of Washington
Seattle, WA 98195, USA

William S. Noble
Department of Genome Sciences
University of Washington
Seattle, WA 98195

Abstract

We present a peptide-spectrum alignment strategy that employs a dynamic Bayesian network (DBN) for the identification of spectra produced by tandem mass spectrometry (MS/MS). Our method is fundamentally generative in that it models peptide fragmentation in MS/MS as a physical process. The model traverses an observed MS/MS spectrum and a peptide-based theoretical spectrum to calculate the best alignment between the two spectra. Unlike all existing state-of-the-art methods for spectrum identification that we are aware of, our method can learn alignment probabilities given a dataset of high-quality peptide-spectrum pairs. The method, moreover, accounts for noise peaks and absent theoretical peaks in the observed spectrum. We demonstrate that our method outperforms, on a majority of datasets, several widely used, state-of-the-art database search tools for spectrum identification. Furthermore, the proposed approach provides an extensible framework for MS/MS analysis and provides useful information that is not produced by other methods, thanks to its generative structure.

1 INTRODUCTION

A fundamental problem in biology and medicine is accurately identifying the proteins present in a complex sample, such as a drop of blood. The solution to this problem has many important applications, such as the early detection of diseases and congenital defects. For instance, sickle-cell disease is caused by a single substitution in the amino acid sequence of a protein found in humans [1], and the identification of this mutated protein can lead to preventive care [2]. The only high-throughput method for solving this problem is *tandem mass spectrometry* (*MS/MS*). Given a complex sample, an MS/MS experiment produces a collection of spectra typically on the order of thousands, each of which represents a single *peptide* (protein subsequence) that was present in the original sample. Fundamental to MS/MS is the ability to accurately identify the peptide responsible for generating a particular spectrum.

The most accurate methods for identifying MS/MS spectra make use of a peptide database typically derived from the mapped genome of the organism of interest. Given a peptide drawn from the database and an observed spectrum, these methods compare a *theoretical spectrum* of the peptide's idealized fragmentation events to a quantized or fixed-width thresholded observed spectrum. Such preprocessing necessarily discards potentially useful information. The spectrum identification problem is greatly complicated by experimental noise, corresponding both to the presence of unexpected peaks (insertions) and the absence of expected peaks (deletions) in the observed spectrum (Fig. 1). This paper describes a Dynamic Bayesian network for Rapid Identification of Peptides (DRIP), a database search method that serves as a generative model of the process by which peptides produce spectra in MS/MS. DRIP explicitly models insertions and deletions, without quantization or thresholding of the observed spectra.

We note that a DBN-based database search method, called Didea, was recently proposed [3], but this method does not model the underlying process by which peptides produce MS/MS spectra. Rather, in Didea both theoretical and observed spectra are observed, and the model contains only a single hidden variable, which is devoid of any physical meaning relative to the underlying MS/MS process. Consequently, Didea only provides a mechanism for scoring and ranking peptides, and no mechanism to meaningfully interpret the data. The theoretical spectrum in DRIP, by contrast, is hidden; insertions and deletions are explicitly modeled as latent variables (as in [4]), and the most probable alignment between the theoretical and observed spectra can be efficiently calculated (detailed in Section 4). Furthermore, Didea has a single hyperparameter that is optimized via grid search, making the model poorly adaptable to the wide range of machines with widely varying characteristics, a problem addressed by the highly trainable nature of DRIP.

We demonstrate, in fact, that against four state-of-the-art benchmarked competitors, DRIP is the most frequent top performer, dominating the others on four out of nine separate datasets. By contrast, other competitors, such as Didea, dominate on at most two datasets. Furthermore, DRIP, thanks to its generative approach, provides valuable auxiliary information, such as which observed peaks are most likely spurious, which theoretical peaks are most likely present, and the ability to calculate posteriors of interest via sum-product inference [5, 6]. Such posteriors include the probability of post-translational modifications given the observed spectrum, a task which previously required post-processing the results of a database search [7].

We first give a brief overview of a typical tandem mass spectrometry experiment and an overview of database search in Section 2. Readers are directed to [8] for further background in this area. Next, the four benchmarked competitors are described in Section 3. DRIP is described in detail in Section 4. Results are presented in Section 5, and we conclude and discuss future work in Section 6.

2 TANDEM MASS SPECTROMETRY AND DATABASE SEARCH

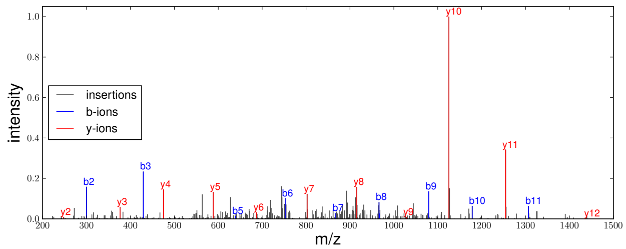


Figure 1: Sample tandem mass spectrum, where the peptide responsible for generating the spectrum is $x =$ LWEPLLDVLVQTK, the precursor charge c^s is 2, and the most probable alignment computed in DRIP is plotted. The b-ion peaks are colored blue, y-ion peaks are colored red, and insertions are colored gray. Note that fragment ions b_1, y_1, b_4, b_{12} correspond to deletions.

Although we are typically interested in the protein content of a complex mixture, the fundamental unit of observation in tandem mass spectrometry is the peptide, because peptides are more amenable to liquid chromatography and mass spectrometry analysis[8, 9]. Thus, a typical MS/MS experiment begins by digesting the proteins into peptides using a cleavage agent such as the enzyme trypsin. MS/MS then proceeds with two rounds of mass spectrometry. The first round measures the mass-to-charge ratio (*m/z*) of the intact peptide (called the *precursor m/z*), and the second round fragments the peptide and measures the m/z values of the resulting prefixes and suffixes. Each of these fragment m/z values is associated with an intensity value, which is roughly proportional to the number of copies of that peptide fragment. Figure 1 displays a sample tandem mass spectrum, along with the theoretical fragment ions (described below) of the generating peptide. A single unit along the m/z axis is called a *Thomson* (Th), and the intensity (y-axis) is unitless but can be seen as a measure of abundance or count.

Let $\mathbb{P}$ be the set of all possible peptides and S be the set of all tandem mass spectra. Given an observed spectrum $s \in S$ with observed precursor m/z m^s and precursor charge c^s, our task is to identify the peptide x from a given peptide database $\mathcal{D} \subseteq \mathbb{P}$ that is responsible for generating s. Any given mass spectrometry device is capable of isolating peptides with a specified precision on the precursor m/z; therefore, we may constrain the search to only consider peptides with precursor m/z $\pm w$ of m^s. The set of *candidate peptides* to be scored is then

$$D(m^s, c^s, \mathcal{D}, w) = \left\{ x : x \in \mathcal{D}, \left| \frac{m(x)}{c^s} - m^s \right| \leq w \right\}, \tag{1}$$

where $m(x)$ is the calculated mass of peptide x. The goal of database search, then, is to return the highest scoring candidate peptide

$$x^* = \underset{x \in D(m^s, c^s, \mathcal{D}, w)}{\operatorname{argmax}} \psi(x, s),$$

where $\psi : \mathbb{P} \times S \rightarrow \mathbb{R}$ is a function that assigns higher scores to higher quality matches and the pair (x, S) is referred to as a *peptide-spectrum match* (PSM). The primary distinguishing characteristic of any database search procedure is its choice of score function ψ.

2.1 THEORETICAL SPECTRA

Many score functions, including the one employed by the very first database search algorithm, SEQUEST [10], work by comparing the observed spectrum to a *theoretical spectrum* that is derived from a candidate peptide using basic rules of biochemistry. Let $x \in D(m^s, c^s, \mathcal{D}, w)$ be an arbitrary candidate peptide of length n. Note that $x = x_0 x_1 \ldots x_{n-1}$ is a string of *amino acids*, i.e. characters in a dictionary of size 20. For convenience, let $\tilde{n} = n - 1$. Our goal is to produce a theoretical spectrum v^x containing the fragment m/z values that we expect x to produce. In this work, we assume that the mass spectrometer employs collision-induced dissociation, which is the most widely employed method of peptide fragmentation. The model can be modified in a straightforward fashion to accommodate other fragmentation modes.

The first type of fragment m/z value corresponds to prefixes or suffixes of the candidate peptide, referred to respectively as *b-ions* and *y-ions*. In this work, we assume that the precursor charge c^S is 2, because this is the charge state of the high-quality set of PSMs used for training [11]. For $c^S = 2$, these b- and y-ions can be represented as functions $b(\cdot, \cdot)$ and $y(\cdot, \cdot)$, respectively, that take as input a peptide x

and integer $k < n$:

$$b(x,k) = \sum_{i=0}^{k-1} m(x_i) + 1, \quad y(x,k) = \sum_{i=\tilde{n}-k}^{\tilde{n}} m(x_i) + 19. \tag{2}$$

Note that the whole peptide mass is not considered and that, for $1 < k < n$, we have the recurrence relations $b(x,k) = b(x,k-1) + m(x_{k-1})$ and $y(x,k) = y(x,k-1) + m(x_{\tilde{n}-k})$. In Equation 2, the b-ion unit offset corresponds to the mass of a hydrogen atom while the y-ion offset corresponds to the masses of a water molecule as well as a hydrogen atom.

Thus, the b- and y-ions are simply the shifted prefix sums and suffix sums of x, respectively. When there is no ambiguity as to the peptide being described, it is typical to represent the b- and y-ion pairs as (b_k, y_{n-k}) for $k = 1, \dots, \tilde{n}$, where the subscript denotes the number of amino acids utilized in the ion computation. As an example, for peptide x = EALK, $(b_1, y_3) = (b(x,1), y(x,3)) = (130, 331)$. Denoting the number of unique b- and y-ions as n^x and, for convenience, letting $\tilde{n}^x = n^x - 1$, our theoretical spectrum is a sorted vector $v^x = (v_0, \dots, v_{\tilde{n}^x})$ consisting of the unique b- and y-ions of x. Figure 2 displays the theoretical spectrum for x = EALK, and Figure 1 displays an observed spectrum with annotated b- and y-ions.

The second type of theoretical fragment ion corresponds to the loss of a molecular group off of a b- or y-ion. Three types of losses are commonly encountered: loss of carbon monoxide, ammonia or water. The intensity of the neutral loss peaks are typically much lower than the corresponding b- and y-ion peaks. Thus, although many search algorithms explicitly model neutral losses [10, 12, 13], DRIP and Didea do not.

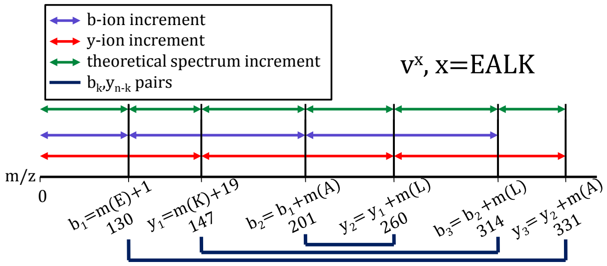


Figure 2: Theoretical spectrum of the peptide x = EALK. Note that the b- and y-ions correspond to prefix and suffix sums, respectively, of the peptide x.

3 PREVIOUS WORK

In this work, we compare DRIP's performance to that of four previously developed methods. All four methods begin by binning the observed spectrum, and all but Didea utilize both neutral losses in their theoretical spectra and scoring functions linear in a binned theoretical spectrum. Such linear scoring functions are equivalent to a weighted count of the fragment ions present in the observed spectrum, which we refer to as a *static alignment* between the observed and theoretical spectra. For the discussion that follows, elements of vectors are denoted using parenthesis, such that the jth element of v^x is $v^x(j)$.

All mass spectrometry devices have a finite observable Th range. Let the maximum observable Th range of the mass spectrometer be $\bar{o} \in \mathbb{N}$. For an arbitrary method, let z be the $(\bar{o}+1)$ length vector obtained by preprocessing s. Now, denote u as the $(\bar{o}+1)$ length vector based on the theoretical spectrum such that for integer $i \in [0, \bar{o}]$,

$$u(i) = \mathbf{1}\{i \in \{\bigcup_{j=0}^{n^x-1} v^x(j)\}\},$$

where $\mathbf{1}\{\cdot\}$ is the indicator function which returns 1 if its argument is true and 0 otherwise. Thus, u is simply a vector with ones at all indices corresponding to theoretical ions, and the static alignment between the observed and theoretical spectra is

$$\varphi(s,x) = \sum_{i=0}^{\bar{o}} u(i) z(i) = u^T z. \tag{3}$$

For example, when preprocessing consists of binning such that $z(i) \in \{0,1\}$ for all integers $i \in [0, \bar{o}]$, then $\varphi(s,x)$ is simply a count of the theoretical ions present in the binned observed spectrum. As we will see, many methods utilize Equation 3 in order to derive p-value based scoring functions(OMSSA, MS-GFDB) or a slightly augmented form of Equation 3 as a scoring function (SEQUEST's XCorr).

3.1 SEQUEST

The first database search algorithm for peptide identification, SEQUEST [10], weights each element of u by prior knowledge of the likelihood of fragment ions such that b- and y-ions have higher weight relative to neutral losses. Note that these weights are hardcoded into the algorithm. Let $\tilde{u}$ denote SEQUEST's weighted theoretical vector, and let z be the $(\bar{o}+1)$ length vector of SEQUEST's preprocessing(detailed in Section 4.6). SEQUEST ranks $x \in D(m^s, c^s, \mathcal{D}, w)$ using a weighted static alignment minus a background term meant to penalize overfitting. This background term is cross-correlation based, and the resulting scoring function is called XCorr. Using subscript τ to denote a vector whose elements are shifted by τ units, XCorr is defined as

$$\begin{aligned} \text{XCorr}(s,x) &= \tilde{u}^T z - \frac{1}{151} \sum_{\tau=-75}^{75} \tilde{u}^T z_\tau \qquad (4) \\ &= \tilde{u}^T (z - \frac{1}{151} \sum_{\tau=-75}^{75} z_\tau) \\ &= \tilde{u}^T z'. \end{aligned}$$

Note that the calculation $z' = (z - \frac{1}{151} \sum_{\tau=-75}^{75} z_\tau)$ may also be combined into the SEQUEST preprocessing. Thus, assuming that the output of the SEQUEST preprocessing is z', and using subscript w to signify the weighting of elements in the theoretical vector, we have that XCorr$(s, x) = \varphi_w(s, x)$, so that as a whole XCorr is simply a weighted static alignment.

3.2 OMSSA

The Open Mass Spectrometry Search Algorithm(OMSSA) [12] is an open source algorithm whose preprocessed observed spectrum, $z \in \mathcal{R}^{\bar{o}}$, consists of binning such that $z(i) \in \{0, 1\}$, for all integers $i \in [0, \bar{o}]$. As previously noted, under such preprocessing, $\varphi(s, x)$ is then simply a count of the theoretical ions present in the binned observed spectrum. OMSSA scores x using a p-value calculated by fitting $\varphi(s, x)$ to a Poisson distribution whose assumed mean parameter is derived from properties of the observed spectrum.

3.3 MS-GFDB

The MasS Generating Function DataBase (MS-GFDB) [13] algorithm calculates x's static alignment with a heavily preprocessed observed spectrum. Denoting the preprocessed observed spectrum as $z \in \mathcal{R}^{\bar{o}}$, MS-GFDB then ranks x by calculating the p-value of $\varphi(s, x)$ over a distribution of scores for all peptides with equal precursor m/z m^s,

$$\text{MS-GFDB}(s, x) = \frac{1}{|\mathbb{P}^{m^s}|} \sum_{\alpha \in \mathbb{P}^{m^s}} \text{p}_{\mathcal{D}}(\alpha) \mathbf{1}\{\varphi(s, \alpha) \geq \varphi(s, x)\}, \qquad (5)$$

where $\mathbb{P}^{m^s} = \{\alpha : m(\alpha) = m^s, \alpha \in \mathbb{P}\}$, $\text{p}_{\mathcal{D}}(\cdot)$ is a distribution over amino acids derived from the amino acid frequencies of the database $\mathcal{D}$, and amino acids are assumed i.i.d. such that $\text{p}_{\mathcal{D}}(x) = \prod_{i=0}^{n-1} p_{\mathcal{D}}(x_i)$. For arbitrary scoring functions, the exact p-value calculation in Equation 5 is difficult if not NP-hard. However, the linearity of Equation 3 allows Equation 5 to be computed efficiently using dynamic programming.

3.4 DIDEA

The fourth algorithm that we consider, Didea [3], is most closely related to DRIP, in the sense that both methods employ a DBN. However, Didea differs from DRIP in four significant ways. First, the two methods use different notions of "time." In Didea, each frame of the DBN corresponds to one amino acid from the candidate peptide sequence. By contrast, successive frames in DRIP instead correspond to successive peaks in the observed spectrum. Accordingly, Didea must copy the entire pre-processed observed spectrum in every frame, whereas a frame in DRIP contains only a single observed peak's m/z and intensity values. Second, the theoretical spectrum in DRIP is hidden and inference is run to determine the best alignment between the observed and theoretical spectra while accounting for insertions and deletions. In Didea, the theoretical spectrum is not hidden as the amino acid variables in each frame are observed, so that performing inference only provides a score. Third, DRIP performs much less pre-processing on the observed spectrum than Didea. In particular, Didea must work with a version of the observed spectrum in which the m/z axis is discretized and the observed intensity values are reweighted using a complicated function of exponentials. DRIP instead scores m/z values in their natural resolution, without discarding information due to quantization. Fourth, whereas Didea is essentially a fixed model, DRIP offers the ability to learn its parameters using training data. The only learning available in Didea is the tuning of a single hyperparameter, via grid search, which controls the reweighting of peak intensities.

We now describe Didea's scoring function in detail. Let $z \in \mathcal{R}^{\bar{o}}$ be the output of Didea's preprocessing of s. Didea's scoring function is similar in form to XCorr. The b- and y-ions are calculated in successive frames and used as indices into shifted versions of z. Random access to the processed observed spectrum is granted by having all of z as observations in each frame, as earlier noted. A single hidden variable in the first frame, corresponding to the number of units to shift z, is then marginalized in order to compute a conditional log-likelihood probability consisting of a foreground score minus a background score (much like Equation 4), where the foreground score is the log-likelihood of the model and the background score is the log of the denominator of the aforementioned posterior.

3.4.1 Log-likelihood

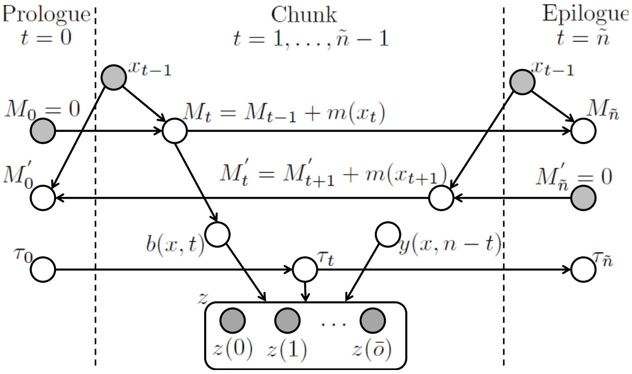


Figure 3: Graph of Didea, where edges denote deterministic functions. M_t and M_t' are the prefix and suffix masses, respectively, calculated recursively in successive frames and used to calculate b- and y-ions in the chunk frames. The entire preprocessed observed spectrum, z, is copied in each frame of the chunk. Note that, by the defined recursion, $M_{n^x} = m(x) = M_0'$.

The graph of Didea is displayed in Figure 3. Let $\tau_t \in [-L, L]$ be an integer random variable in frame t =

$0, \ldots, |x|-1$, such that τ_0 is uniform and $\mathrm{p}(\tau_t = \tau_0) = 1$ for $0 < t \leq \tilde{n}$, where $\tilde{n} = |x| - 1$. As with SEQUEST, denote z_{τ_t} as the vector resulting from shifting z by τ_t units.

The DBN is depicted as a *template* [6], where the first frame is called the *prologue*, the final frame is called the *epilogue*, and the *chunk* consists of multiple frames which dynamically expand based on the length of a peptide. Shaded variables are observed, i.e. take their values in each frame with probability 1, and unshaded variables are hidden(random). The amino acids of a peptide are observed in each frame after the prologue. The variable M_t successively accumulates the prefix masses of the peptide, while the variable $M_t^{'}$ successively accumulates the suffix masses of the peptide. M_t and $M_t^{'}$ are thus used to calculate the b- and y-ions of the peptide in the chunk. The variable τ_0 is hidden in the prologue and deterministically copied by its descendants in successive frames, such that $p(\tau_t = \tau | \tau_0 = \tau) = 1$ for $t > 0$. With τ_t and the b- and y-ions as parents to z, shifted versions of the preprocessed spectrum intensities are accessed.

For observed spectrum shift $\tau \in [-L, L]$, the log-likelihood of Didea is then

$$\begin{aligned}
&\log \mathrm{p}(\tau_0 = \tau, x, z) \\
&= \log \mathrm{p}(\tau_0 = \tau)\mathrm{p}(x, z | \tau_0 = \tau) \\
&= \log \mathrm{p}(\tau_0 = \tau) + \log \mathrm{p}(x, z | \tau_0 = \tau) \\
&= -\log |\tau_0| + \log \prod_{t=1}^{\tilde{n}} z_\tau(b(x,t)) z_\tau(y(x, n-t)) \\
&= -\log |\tau_0| + \sum_{t=1}^{\tilde{n}} (\log z_\tau(b(x,t)) + \log z_\tau(y(x, n-t)))
\end{aligned} \tag{6}$$

3.4.2 Conditional log-likelihood scoring function

Didea scores peptides by computing a conditional log-likelihood

$$\begin{aligned}
&\mathrm{Didea}(s, x) \\
&= \log \mathrm{p}(\tau_0 = 0 | x, z) \\
&= \log \mathrm{p}(\tau_0 = 0, x, z) - \log \mathrm{p}(x, z) \\
&= \log \mathrm{p}(\tau_0 = 0, x, z) - \log \sum_{\tau=-L}^{L} \mathrm{p}(\tau_0 = \tau, x, z) \\
&= \log \mathrm{p}(\tau_0 = 0, x, z) - \log \sum_{\tau=-L}^{L} \mathrm{p}(\tau_0 = \tau) \mathrm{p}(x, z_\tau | \tau_0 = \tau) \\
&= \log \mathrm{p}(\tau_0 = 0, x, z) - \log \frac{1}{|\tau_0|} \sum_{\tau=-L}^{L} \mathrm{p}(x, z_\tau | \tau_0 = \tau).
\end{aligned} \tag{7}$$

As previously mentioned, $\mathrm{Didea}(s, x)$ is a foreground score minus a background score, where the background score consists of averaging over $|\tau_0|$ shifted versions of the foreground score, much like SEQUEST's XCorr. Thus, Didea may be thought of as a probabilistic analogue of SEQUEST.

4 DRIP PEPTIDE SCORING

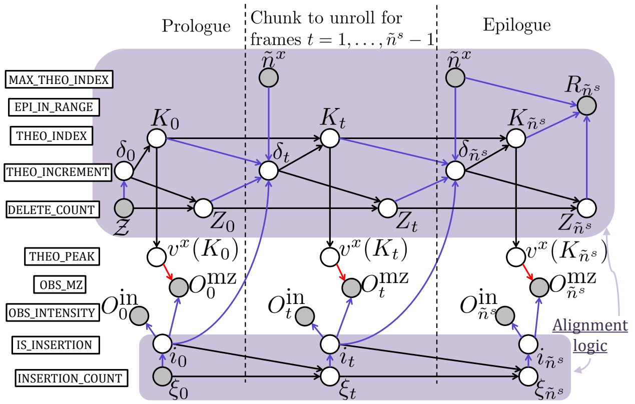


Figure 4: Graph of DRIP, where the boxed words on the far left summarize the role of the random variables on the same horizontal level to aid interpretation of the model. THEO and OBS are short for theoretical and observed, respectively. Shaded variables are observed and unshaded variables are hidden. Black edges denote deterministic functions, blue edges denote switching parents, and red edges denote continuous conditional distributions.

The graph of DRIP is displayed in Figure 4, where each frame of the model corresponds to a single observed peak. Shaded nodes represent observed variables, and unshaded nodes represent hidden variables. Black edges correspond to deterministic functions of parent variables, and blue edges represent switching parent functionality (also known as Bayesian multi-nets [14]) where the parent nodes are known as *switching parents* and the parents (and hence conditional distributions) of their children may change given the values of their switching parents. Finally, red edges denote continuous conditional distributions. Random variables are grouped into frames, indexed by $t = 0, \ldots, n^s - 1$. Note that while elements of vectors are denoted using parentheses, a particular value of a sequence is denoted using subscripts, such that δ_t is a random variable in the tth frame. For convenience, let $\widetilde{n}^s = n^s - 1$, and recall that $\widetilde{n}^x$ denotes the number of peaks in the theoretical spectrum minus one. The first and last frames are known as the *prologue* and *epilogue*, respectively. The middle frame is called the *chunk* and is unrolled $n^s - 2$ times to frames $t = 1, \ldots, \widetilde{n}^s - 1$. Each frame of the graph contains observations O_t^{mz} and O_t^{in}, the tth m/z and normalized intensity values of s, respectively. Thus, we can view traversing the graph from left to right as moving across the observed spectrum.

The goal of DRIP is to calculate the most probable alignment between the observed and theoretical spectra, where an alignment is an instantiation of the random variables in

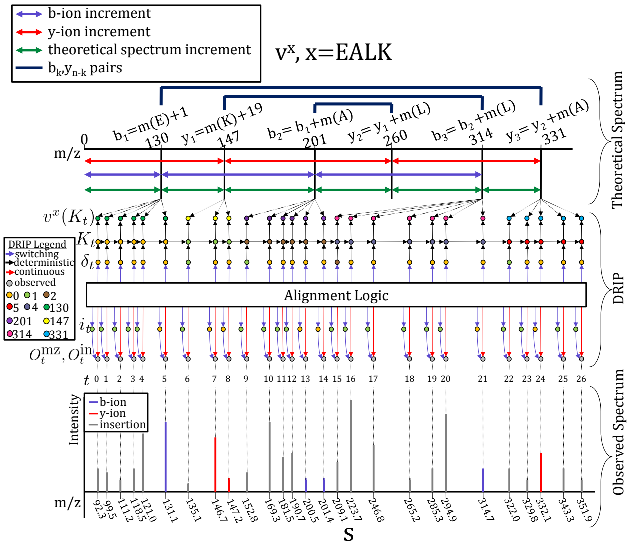


Figure 5: Illustration of a particular spectrum alignment (instantiation of random variables) in DRIP, where the node color denotes its instantiated value. The observed spectrum peaks serve as the observations in each frame while the theoretical spectrum is hidden. DRIP thus serves as a sequencer through all possible alignments between the theoretical and observed spectra, made efficient via dynamic programming.

the graph and the scoring of observed peaks as dictated by this instantiation. This concept is detailed in Figure 5 for a particular alignment, where random variable values are denoted by node colors, and the alignment corresponds to a traversal of both the theoretical (upper portion) and observed spectra (lower portion). In the center portion of Figure 5, the random variable K_t denotes the theoretical peak index and, given the increment random variable δ_t, moves us down the theoretical spectrum, while further alignment logic in DRIP constrains the manner in which the theoretical and observed spectra may be aligned. Figure 6 illustrates the scoring of observed peaks (discussed in Section 4.2) in this alignment, and the instantiation of random variables may be found in Table 1. We now discuss the details of how DRIP aligns the theoretical and observed spectra.

4.1 TRAVERSING THE THEORETICAL SPECTRUM

The variable K_t is the index of the theoretical peak used to score peaks in frame t, such that

$$\text{p}(K_0 = \delta_0 | \delta_0) = 1, \tag{8}$$
$$\text{p}(K_t = K_{t-1} + \delta_t | K_{t-1}, \delta_t) = 1,\ t > 0. \tag{9}$$

From (8) and (9), we see that δ_t is the number of theoretical peaks we traverse between frames t and $t + 1$. Note that a deletion thus occurs when $\delta_0 > 0$ and $\delta_t > 1$ for $t > 0$, i.e., the *hypotheses* such that one or more theoretical peaks are not accessed, where a hypothesis is an assignment of all random variables in the graph. The number of deletions occurring in a single frame is then δ_0 for the prologue and $(\delta_t - 1)\mathbf{1}\{\delta_t > 1\}$ for all subsequent frames. The total number of allowed deletions is $\mathcal{Z}$ and counts down in subsequent frames such that, denoting the number of deletions left in a frame as Z_t, we have

$$\text{p}(Z_0 = \mathcal{Z} - \delta_0 | \mathcal{Z}, \delta_0) = 1$$
$$\text{p}(Z_t = Z_{t-1} - (\delta_t - 1)\mathbf{1}\{\delta_t > 1\} | Z_{t-1}, \delta_t) = 1, t > 0.$$

The allowable number of insertions counts down in a similar manner to the deletions. ξ_0 is the maximum allowable insertions for all frames, i_t is a Bernoulli random variable which signifies whether the peak in frame t is an insertion, and $\text{p}(\xi_t = \xi_{t-1} - i_{t-1} | \xi_{t-1}, i_{t-1}) = 1$. Furthermore, the role of ξ_t as a switching parent of i_t is such that $\text{p}(i_t = 0 | \xi_t = 0) = 1$. Thus, when there are no insertions left, i_t is 0 for all remaining frames.

The hidden multinomial δ_t is such that $\text{p}(\delta_0 > \mathcal{Z}) = 0$, i.e. it respects the maximum deletion constraint of the first frame, and for $t > 0$, $\text{p}(\delta_t) = \sum_{i_{t-1}} \text{p}(\delta_t | \delta_{t-1}, Z_{t-1}, \tilde{n}^x, i_{t-1}) \text{p}(i_{t-1})$, where

$$\text{p}(\delta_t = 0 | \delta_{t-1}, Z_{t-1}, \tilde{n}^x, i_{t-1} = 1) = 1, \tag{10}$$
$$\text{p}(\delta_t > \tilde{n}^x - (K_t - Z_t) | \tilde{n}^x, K_t, Z_t, i_{t-1}) = 0. \tag{11}$$

Equation (10) prohibits DRIP from moving down the theoretical spectrum in a frame following an insertion. This constraint ensures that the theoretical spectrum may not be trivially traversed while observed peaks are scored as insertions, or equivalently that some observed peak must not be scored as an insertion in order to move down the theoretical spectrum for frames $t > 0$. Equation (11) constrains DRIP from incrementing past the range of valid theoretical peak indices. In order to discourage the use of deletions unless absolutely necessary, the distribution over δ_t is constrained to be monotone decreasing, such that for $i_{t-1} = 1$, $K_t - Z_t < \tilde{n}^x$, and $0 < h < \tilde{n}_t^x - (K_t - Z_t) - 1$, we have

$$\text{p}(\delta_t = h | \tilde{n}^x, K_t, Z_t, i_{t-1}) < \text{p}(\delta_t = h - 1 | \tilde{n}^x, K_t, Z_t, i_{t-1}).$$

The epilogue variable $R_{\tilde{n}^s}$, observed to 1, constrains which theoretical peaks may occur in the final frame. If the number of theoretical peaks left unexplored in the epilogue is greater than the number of remaining deletions, then such a hypothesis of random variables receives probability zero. Thus, we have

$$\text{p}(R_{\tilde{n}^s} = 1 | \tilde{n}^x, K_{\tilde{n}^s}, Z_{\tilde{n}^s}) = \mathbf{1}\{\tilde{n}^x - K_{\tilde{n}^s} \leq Z_{\tilde{n}^s}\}.$$

This boundary condition limits the number of valid hypotheses in DRIP which have non-zero probability, and by forcing the traversal of the theoretical spectrum from prologue to epilogue ensures that a peptide cannot align trivially well to the observed spectrum. Figure 5 and Table 1 detail the theoretical spectrum traversal for the alignment depicted in Figure 6.

Table 1: Random variable hypothesis for alignment displayed in Figure 6. Recall the deterministic relationships $K_0 = \delta_0$, $K_t = K_{t-1} + \delta_t$ for $t > 0$, and given the theoretical peak index K_t we have the theoretical peak $v^x(K_t)$. For instance, from the theoretical spectrum of $x =$ EALK in Figure 5, we have $K_6 = K_5 + \delta_6$ and $v^x(K_6) = v^x(1) = 147$.

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
δ_t	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	2	0	0	0	0	0	1	0	0	0	0	0
K_t	0	0	0	0	0	0	1	1	1	2	2	2	2	2	2	4	4	4	4	4	4	4	5	5	5	5	5
i_t	1	1	1	1	1	0	1	0	0	1	1	1	1	0	0	1	1	1	1	1	1	0	1	1	0	1	1

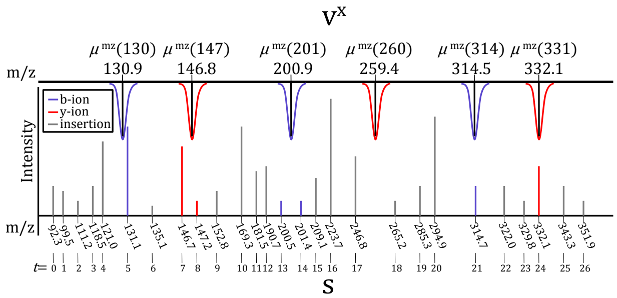


Figure 6: Spectra alignment in DRIP given observed spectrum s, $x =$ EALK, the theoretical Gaussian peaks of v^x, and hypothesis from Figure 5 (the hypothesis is also listed in Table 1). Note that t is the frame number, all Gaussians have equal variance, and $v^x(3)$ is a deletion.

4.2 SCORING OBSERVED PEAKS

The m/z observation is scored by a Gaussian centered near the theoretical peak accessed in a frame, and the intensity observation is scored using a Gaussian with mean greater than or equal to the most intense peak value. In this manner, DRIP aligns the theoretical and observed spectra by penalizing observed peaks far from theoretical peaks and penalizing peaks with intensity less than unity. The conditional distributions of the observations are then

$$\mathrm{p}(O_t^{\mathrm{mz}} | v^x(K_t)) = \sum_{i_t} \mathrm{p}(O_t^{\mathrm{mz}} | v^x(K_t), i_t)\mathrm{p}(i_t),$$
$$\mathrm{p}(O_t^{\mathrm{in}}) = \sum_{i_t} \mathrm{p}(O_t^{\mathrm{in}} | i_t)\mathrm{p}(i_t).$$

When $i_t = 0$, i.e., the tth observed peak is not an insertion, then the observations are scored as

$$\mathrm{p}(O_t^{\mathrm{mz}} | v^x(K_t), i_t = 0) \sim \mathcal{N}(\mu^{\mathrm{mz}}(v^x(K_t)), \sigma^2),$$
$$\mathrm{p}(O_t^{\mathrm{in}} | \mu^{\mathrm{in}}, i_t = 1) \sim \mathcal{N}(\mu^{\mathrm{in}}, \bar{\sigma}^2),$$

where μ^{in} and $\bar{\sigma}^2$ are the mean and variance of the Gaussian used to score peak intensities, σ^2 is the variance of the Gaussian used to score m/z observations, and μ^{mz} is a vector of means such that an arbitrary theoretical peak j scores m/z observations using $\mathcal{N}(\mu^{\mathrm{mz}}(j), \sigma^2)$. The theoretical peaks serve as indices into a vector of Gaussians, each such Gaussian having equal variance and centered at a unique m/z position (illustrated in Figure 6). Thus, when describing a theoretical peak as scoring an observed peak, we are referring to scoring using the Gaussian accessed by the theoretical peak. To avoid confusion, we refer to the Gaussian accessed by a theoretical peak as the *theoretical Gaussian peak*.

All DRIP Gaussian means and variances are learned using expectation-maximization(EM) [15] and a high-confidence set of PSMs used in [11]. Note that this adds a great deal of modeling power to DRIP, allowing for adaptability of the expected m/z location of theoretical peaks. In MS/MS data, the m/z measurements will be stochastically offset from the true m/z values in a manner dependent upon machine precision, and may also be systematically offset in a nonlinear fashion due to machine miscalibration [16]. Learning the means of the Gaussians allows DRIP to account for and adapt to these trends. Furthermore, the learned means themselves may be of interest to researchers in the field as a method of studying the nonlinear warping of the m/z axis encountered in specific experiments.

4.2.1 Insertion penalties

Due to the noisy nature of MS/MS data, only a small minority of observed peaks actually correspond to fragment ions. As an example, Figure 1 displays excellent fragmentation of the peptide present in the observed spectrum. However, only 22 of 324 peaks correspond to fragment ions. Furthermore, these fragment ions only occur within a sufficiently small Th window of their theoretical values. Indeed, the learned variance σ^2 dictates that 99.9937% of the mass for a theoretical Gaussian peak lies within an approximately $1\mathrm{Th}$ range, so that attempting to score all m/z observations with theoretical Gaussian peaks would see a majority of peptides score poorly for almost all observed spectra (save sufficiently sparse spectra, which are not often encountered in practice). Such an approach would also make the comparison of scores across different spectra in the same dataset difficult, because the variance among the PSMs would be incredibly high. Thus, when $i_t = 1$, the observations are scored as

$$\mathrm{p}(O_t^{\mathrm{mz}} | v^x(K_t), i_t = 1) = \mathrm{p}(O_t^{\mathrm{mz}} | i_t = 1) = a$$
$$\mathrm{p}(O_t^{\mathrm{in}} | i_t = 1) = b,$$

where a and b are constants.

The insertion constant a imposes a tradeoff between, on the one hand, receiving exponentially bad scores from scoring observed peaks far from theoretical Gaussian peaks and, on the other hand, simply receiving an arbitrarily large constant penalty. To balance this tradeoff, a is set to the score received evaluating an m/z observation 4 standard deviations from the theoretical Gaussian peak mean, $\mu^{\text{mz}}(v^x(K_t))$, i.e., $a = f(4\sigma - \mu^{\text{mz}}(v^x(K_t))|\mu^{\text{mz}}(v^x(K_t)), \sigma^2)$, where $f(z|\mu, \bar{\sigma}^2)$ is the scalar Gaussian with mean μ and variance $\bar{\sigma}^2$, evaluated at $z \in \mathbb{R}$. Thus, for arbitrary m/z observation z,

$$f(z|\mu^{\text{mz}}(v^x(K_t)), \sigma^2) > a, \text{ if } |z \pm \mu^{\text{mz}}(v^x(K_t))| < 4\sigma,$$
$$f(z|\mu^{\text{mz}}(v^x(K_t)), \sigma^2) \leq a, \text{ else,}$$

and we have that an m/z observation score is greater than a so long as it remains within 99.9937% of the centered mass of $\mathcal{N}(\mu^{\text{mz}}(v^x(K_t)), \sigma^2)$. Similarly, the penalty b is set such that an intensity observation score is greater than b so long as it is within a specified percentage of the centered mass of $\mathcal{N}(\mu^{\text{in}}, \bar{\sigma}^2)$. The percentage used for the results in Section 5 was 20%, prioritizing aligning the observed and theoretical peaks over simply scoring high intensity peaks. Furthermore, the number of allowable insertions is limited per peptide (described in Section 4.4), restricting the ability of arbitrary peptides to score observed peaks well.

4.3 DRIP SCORING FUNCTION

Peptides are scored by their optimal alignment using their per-frame log-Viterbi Score,

$$\psi(s, x) = \frac{1}{n^s} \max_{\delta_t, i_t, \forall t} \log \text{p}(s|x) = \frac{1}{n^s} \log \text{p}^*(s|x). \quad (12)$$

Dividing by the number of frames allows comparability of PSMs from different spectra. In order to further analyze the scoring function, assume inference has been completed and we have computed the Viterbi path, using * to denote a variable's Viterbi value. Let $\lambda = \sum_{t=0}^{\tilde{n}^s} i_t^*$ denote the number of used insertions and note that $\text{p}(i_t = 0) = \text{p}(i_0 = 0)$. DRIP's score is then

$$\begin{aligned} &\log \text{p}^*(s|x) = \quad (13)\\ &\lambda[\log(ab) + 3\log \text{p}(i_0 = 1)] + 3(n^s - \lambda)\log \text{p}(i_0 = 0) + \\ &\sum_{t=0}^{\tilde{n}^s} \Big[\log \text{p}(\delta_t^*) + \log f(O_t^{\text{in}}|\mu^{\text{in}}, \sigma^2)) + \\ &\qquad \mathbf{1}\{i_t^* = 0\}(\log f(O_t^{\text{mz}}|\mu^{\text{mz}}(v^x(K_t^*)), \sigma^2)\Big] \end{aligned}$$

where, as before, $f(z|\mu, \bar{\sigma}^2)$ is the scalar Gaussian with mean μ and variance $\bar{\sigma}^2$, evaluated at $z \in \mathbb{R}$. The learned model variances are such that $\sigma^2 < \bar{\sigma}^2$, i.e., there is more uncertainty in intensity measurements than m/z measurements. Thus, it is easy to see that when a peptide does not align well with the observed spectrum (i.e., many observed peaks are far from the closest theoretical peak), then the $\log f(O_t^{\text{mz}}|\mu^{\text{mz}}(v^x(K_t^*)), \sigma^2)$ term severely penalizes the score. Furthermore, this score decreases quickly as the distance between the observed peak and theoretical peak mean increases. This also implies that peptides which arbitrarily match intense peaks will still receive poor scores if they do not align well.

4.4 CALCULATION OF ALLOWABLE DELETIONS AND INSERTIONS

The maximum number of deletions, $\mathcal{Z}$, and insertions, ξ_0, are tightly controlled in order to prevent peptides with poor maximal alignments to score the observed spectrum arbitrarily well. While $\mathcal{Z}$ is upper bounded by the number of theoretical peaks n^x, this value is extremely lax since, for any peptide, it does not restrict hypotheses where observed peaks are never scored by theoretical Gaussian peaks. Similarly, ξ_0 is upper bounded by n^s, however this lax value allows all peptides to score all observed peaks as insertions. We utilize the design of the insertion penalties(Section 4.2.1) in bounding $\mathcal{Z}$ and ξ_0.

For a candidate peptide x with theoretical spectrum v^x, begin by constructing a vector of counts, s', for which $s'(v^x(j))$ denotes the number of observed peaks in s within $4\sigma\mathsf{Th}$ of a theoretical peak $v^x(j)$, where $j = 0, 1, \ldots, \tilde{n}^x$. By the design of the insertion penalties, $\sum_{j=0}^{\tilde{n}^x} 1\{s'(v^x(j)) > 0\}$ theoretical Gaussian peaks will be used to score observed peaks. Consequently, $\mathcal{Z}$ is observed to be $\mathcal{Z} = n^x - \sum_{j=0}^{\tilde{n}^x} \mathbf{1}\{s'(v^x(j)) > 0\}$. Under similar arguments, an upper bound on the true number of insertions would be $n^s - \sum_{j=0}^{\tilde{n}^x} s'(v^x(j))$, the number of observed peaks lying $4\sigma\mathsf{Th}$ from any theoretical peaks. Thus, ξ_0 is observed to be $\xi_0 = n^s - \sum_{j=0}^{\tilde{n}^x} s'(v^x(j))$.

4.5 APPROXIMATE INFERENCE

Table 2: DRIP per-peptide run-times (in seconds) for 3 yeast spectra, 1000 scored candidate peptides each.

Spec.	Exact Inf.	$k = 1500$	$k = 1000$	$k = 500$
s_1	0.07816	0.01056	0.00779	0.00436
s_2	0.30003	0.02070	0.01496	0.00820
s_3	3.61777	0.04105	0.02861	0.01586

The state space of the random variables in DRIP grows rapidly as the number of observed and theoretical peaks increases. Although the observed variables $\tilde{n}^x, R_{\tilde{n}^s}, \mathcal{Z}, \xi_0$ greatly decrease the number of states necessary to explore by limiting the hypotheses in DRIP which receive non-zero probability, there are still an exponentially large number of states to score in order to find the Viterbi path. However, the problem of interest is ideally suited for approximate inference techniques, specifically beam pruning [17]. In beam pruning, assuming a beam width of $k \in \mathbb{N}$, only the top k most probable states in a frame are allowed to persist. Although under this methodology we are no longer theoretically guaranteed to find the Viterbi path, the structure of the problem and the value of the learned theoretical Gaussian variances ensures that, per frame, many of the hypotheses will be of extremely low probability.

For instance, the hypothesis that the first theoretical peak matches the last observed peak is highly improbable. In general, the hypothesis that a theoretical peak centered many Thomsons away from an observed peak is also highly improbable. Thus, we can retain the k most probable states in a frame without deleteriously affecting the Viterbi score. However, care must be taken such that k is not too small or else globally good alignments where a frame must be explained by a low probability event may not be allowed to persist. Table 2 displays the per-peptide run times for 3 randomly chosen yeast spectra, scoring 1000 candidate peptides per spectrum using exact inference and various k values. Each test was performed on the same machine with an Intel Core 2 Quad Q9550 and 8GB RAM. For $k \in \{1500, 1000\}$, all peptide scores were equal to their exact scores. For $k = 500$, 0.1% of the peptide scores differed from their exact scores. The top ranking PSM scores did not change for all beam widths. The results found in Section 5 were generated using $k = 1500$.

4.6 DRIP OBSERVED SPECTRUM PREPROCESSING

As with all other search algorithms, we score database peptides of at most a fixed maximum length, specified prior to run time. Practical values of the maximum peptide length (the maximum peptide length considered for all results in Section 5 is 50) mean that the number of observed peaks is typically an order of magnitude larger than the number of theoretical peaks for any scored peptide. Furthermore, most of these peaks are noise peaks [18], and as such we filter all but the most l intense peaks, where in practice, $l = 300$.

After filtering peaks, the observed spectrum is renormalized as in SEQUEST [10], the steps of which are as follows. Firstly, all observed peak intensity values are replaced with their square root. Secondly, the observed spectrum is partitioned into 10 equally spaced regions along the m/z axis and all peaks in a region are normalized to some globally maximum intensity, which in our case is 1. Steps 1 and 2 greatly decrease the high variance of observed intensities, and step 2 helps ensure that scoring is not severely biased by many large peaks lying arbitrarily close to one another. Lastly, any peak with intensity less than 1/20 of the most intense peak is filtered. Note that through all of these preprocessing steps, the m/z values for all remaining observed peaks remain unaltered and, as discussed in Section 4.3, the scoring of these unaltered values dominates the returned DRIP score.

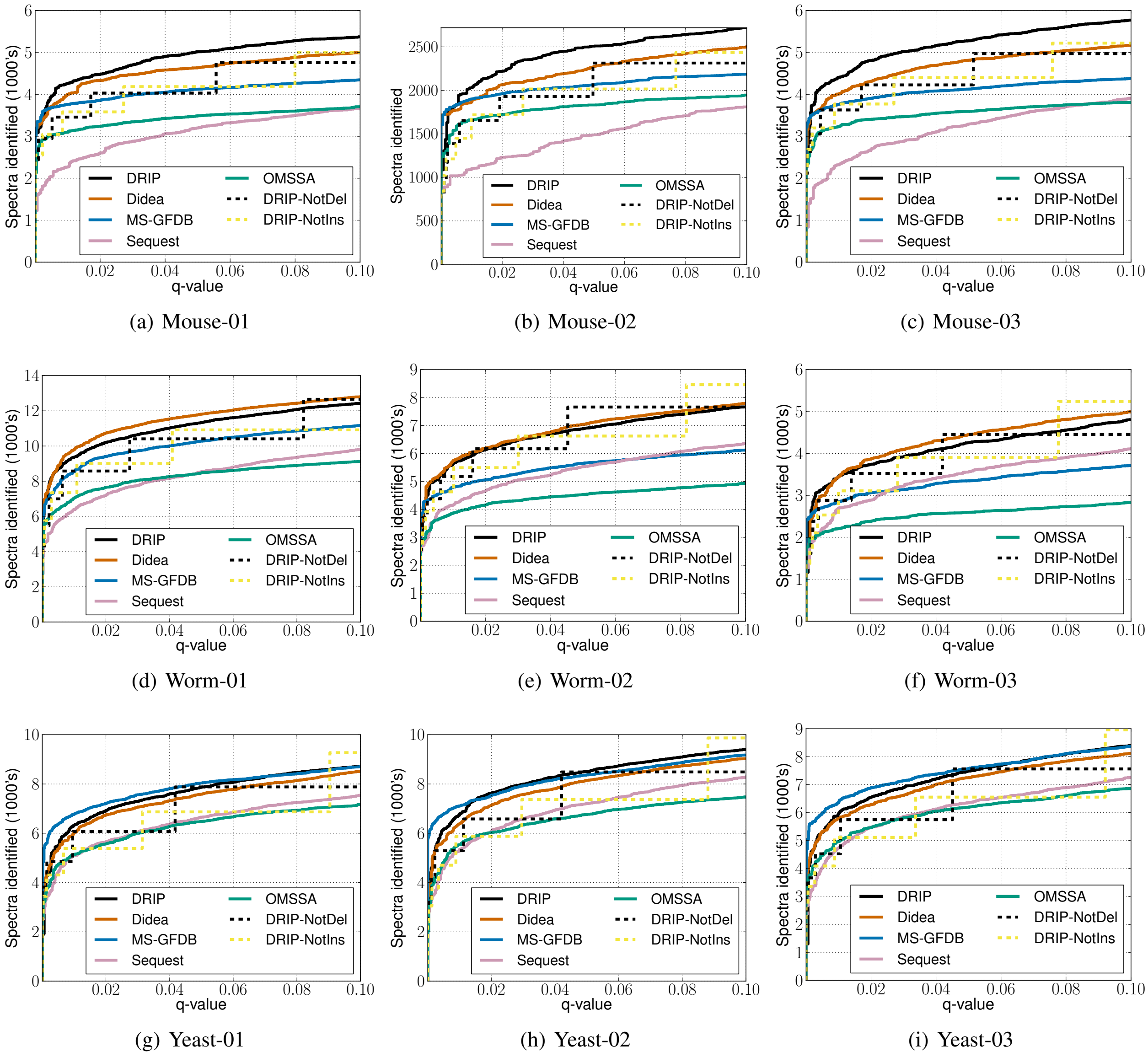
5 RESULTS

We compared the performance of DRIP to four competing database search methods (Section 3). In these evaluations, we do not have an independently labeled gold standard set of identified spectra. Although it is possible to send a purified sample of known peptides through the MS/MS pipeline to obtain high confidence identifications, the low complexity of the input sample yields spectra that are less noisy than real spectra. Therefore, as is common in this field, we estimate for each search procedure the *false discovery rate* (i.e., the proportion of spectra above a given threshold that are incorrectly identified, or 1 – precision) by searching a decoy database of peptides [19]. These decoys are generated by shuffling the peptides in the target database. Because FDR is not monotonically related to the underlying score, we compute a q-value for each scored spectrum, defined as the minimum FDR threshold at which that score is deemed significant. Once the target and decoy PSMs are calculated, we plot the number of identified targets as a function of q-value threshold. In practice, search results with an FDR $>10\%$ are not practically useful, so we only plot $q \in [0, 0.1]$.

We use eight previously described datasets [3] as well as another yeast dataset (yeast-03) taken from the same repository, considering only spectra with charge 2+. All benchmark methods were searched using the same target and decoy databases, and all parameters across search algorithms were set as equivalently as possible. All datasets and reported PSMs per benchmark method may be found at http://noble.gs.washington.edu/proj/drip. As seen in the results panel in Figure 5, DRIP outperforms SEQUEST and OMSSA at all q-value thresholds on all datasets. DRIP is the most frequent top performer, beating all other methods on four datasets, compared to MS-GFDB and Didea, each of which is top performer on only two datasets. Furthermore, DRIP individually outperforms MS-GFDB and Didea on six of the nine datasets. DRIP also offers the most consistent performance compared to MS-GFDB and Didea across the different organisms: DRIP is always ranked first or second, whereas MS-GFDB and Didea rank third on many datasets. Both DRIP and Didea only model b- and y-ions, whereas the other algorithms [10, 12, 13] use more complicated models of peptide fragmentation, including neutral losses.

5.1 INSERTION, DELETION COUNT-BASED SCORES

Once a peptide's Viterbi path has been decoded, the total number of insertions and deletions used by a peptide to score an observed spectrum may be calculated. These two quantities may be used as quality measures for a PSM, as well as to exactly compute the signal-to-noise ratio per observed spectrum. To illustrate the utility of these two quantities, we show that using both as scoring functions allows some discriminative power to differentiate between target and decoy peptides, outperforming OMSSA and SEQUEST as well as MS-GFDB over some datasets. Plotted in Figure 7, DRIP-NotDel and DRIP-NotIns correspond to scoring functions utilizing the number of a peptide's theoretical peaks not deleted and the number of observed peaks a peptide did not consider an insertion, respectively. Note the piece-wise



(a) Mouse-01

(b) Mouse-02

(c) Mouse-03

(d) Worm-01

(e) Worm-02

(f) Worm-03

(g) Yeast-01

(h) Yeast-02

(i) Yeast-03

Figure 7: Performance curves for DRIP. The x-axis may be thought of as the significance threshold, and the y-axis the number of correctly identified spectra at a threshold. Thus, higher on the y-axis denotes better performance. DRIP-NotDel and DRIP-NotIns utilize the decoded DRIP Viterbi path to calculate the number of theoretical peaks not deleted and number of observed peaks not inserted, respectively, as scoring functions.

linear behavior, which is caused by scoring ties due to the scoring functions being integer based.

It is worth noting that scoring methods, such as SEQUEST and Didea, which perform binning typically do so by taking the maximum observed peak intensity falling within a bin. Under such binning schemes, the number of theoretical peaks not deleted is equal to the number of observed peaks which are not insertions. In DRIP, where quantization is not performed, these two quantities are not equal (Figure 7). Such quantities are typically used by post-processors as features for the task of reranking target and decoy scores for improved accuracy [20, 21], and these quantities (and potentially others) calculated from DRIP's Viterbi path may similarly be used as features.

5.2 IMPACT OF LEARNED PARAMETERS ON PERFORMANCE

The use of Gaussians allows DRIP to avoid quantization of m/z measurements, unlike all existing competitors. Learning the Gaussian means and variances in DRIP provides both a tool to study the nonlinear m/z offsets caused by machine error [16] as well as a significant increase in performance. As previously mentioned, the Gaussian parameters are learned using EM and a high-confidence set of charge 2 PSMs used in [11]. Figure 8 displays the performance benefits of jointly learning these parameters.

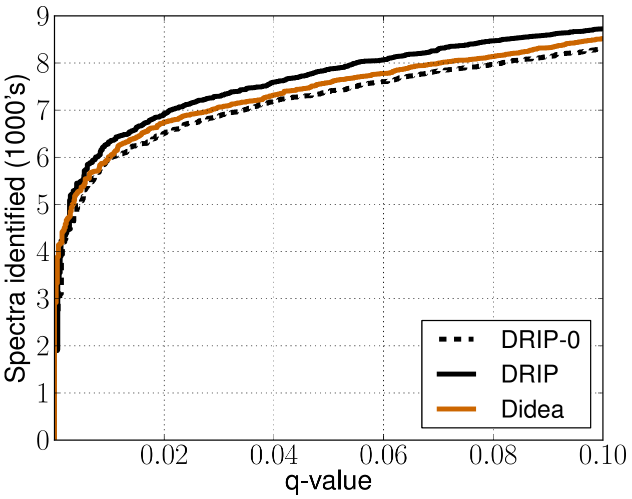


Figure 8: DRIP run with different model parameters over Yeast-01, where DRIP-0 consists of hand-tuned Gaussian parameters and DRIP consists of EM jointly learned Gaussian means and variances.

DRIP-0 consists of setting the Gaussian means to values halfway between integer units along the m/z axis and setting the intensity variance to an order of magnitude larger than the m/z variance, so as to penalize misalignment between the theoretical and observed spectra greater than small intensities. DRIP consists of jointly learning both the Gaussian means and variances, the parameters used for testing over all datasets in Figure 7. Interestingly, the learned intensity variance is larger than the learned m/z variance, so that the learned parameters dictate m/z measurements have the largest impact when scoring. Jointly learning both the means and variances improves performance compared to hand-tuned parameters, leading to improved performance relative to Didea. This trend of improved performance via learning the Gaussian parameters is observed on the other datasets, as well.

6 CONCLUSIONS AND FUTURE WORK

We have presented DRIP, a generative model of peptide fragmentation in MS/MS. Through DRIP, a database peptide is scored by maximally aligning the peptide's theoretical spectrum to the observed MS/MS spectrum via Viterbi decoding. Unlike previous database search methods, the observed spectrum is not quantized; instead, the m/z measurements are scored in their natural resolution. Considering the recent push in the field toward high-resolution data [22], for which other search methods must reevaluate their quantization schemes, DRIP's handling of m/z values at full resolution is particularly important.

DRIP's scoring function outperforms state-of-the-art algorithms on many of the presented datasets, and is far superior to the popular search algorithms SEQUEST and OMSSA. Furthermore, unlike a recent DBN-based database search method [3], DRIP is a highly trainable model which allows it a great deal of adaptability to the wide variety of machines and experimental conditions. The Viterbi path calculated in DRIP also provides a large amount of information, which otherwise typically requires post-processing after database search. Finally, via sum-product inference, DRIP may be used to calculate posteriors of particular interest to end users, a task which has previously required complicated post-processing [7].

We plan to pursue several avenues for future work. Initially, we will collect high-quality training sets of PSMs charge states other than 2 and for high-resolution spectra. Perhaps the most exciting avenue for future work is that a minor change to the DRIP model will allow it to align an observed spectrum to not just one but many different peptides simultaneously. We plan to investigate sequential variants of algebraic decision diagrams [23] to represent (potentially exponentially) large collections of peptides in polynomial space and to exploit the dynamic programming nature of DBNs to be able to score such peptide collections efficiently. Such a framework will also generalize to de novo sequencing, in which we search over the set of all possible peptides as opposed to simply a database. Finally, we plan to investigate generalizing the use of algebraic decision diagrams to allow DRIP to calibrate its scores relative to the entire peptide set. This would be similar in spirit to the dynamic programming calibration employed by methods like MS-GFDB.

Acknowledgements: This work was supported by the National Institutes of Health (NIH) under awards R01 GM096306, P41 GM103533, and T32 HG00035, and by the National Science Foundation (NSF) under grant CNS-0855230.

References

[1] L. Pauling, H. A. Itano, S. J. Singer, and I. C. Wells, "Sickle Cell Anemia, a Molecular Disease," *Science*, vol. 110, pp. 543–548, Nov. 1949.

[2] M. C. Walters, M. Patience, W. Leisenring, J. R. Eckman, J. P. Scott, W. C. Mentzer, S. C. Davies, K. Ohene-Frempong, F. Bernaudin, D. C. Matthews, *et al.*, "Bone marrow transplantation for sickle cell disease," *New England Journal of Medicine*, vol. 335, no. 6, pp. 369–376, 1996.

[3] A. P. Singh, J. Halloran, J. A. Bilmes, K. Kirchoff, and W. S. Noble, "Spectrum identification using a dynamic bayesian network model of tandem mass spectra," in *Uncertainty in Artificial Intelligence (UAI)*, (Catalina Island, USA), AUAI, July 2012.

[4] K. Filali and J. Bilmes, "A Dynamic Bayesian Framework to Model Context and Memory in Edit Distance Learning: An Application to Pronunciation Classification," in *Proceedings of the Association for Computational Linguistics (ACL)*, 43, (University of Michigan, Ann Arbor), 2005.

[5] J. Pearl, *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[6] J. Bilmes, "Dynamic graphical models," *IEEE Signal Processing Magazine*, vol. 27, pp. 29–42, Nov 2010.

[7] S. A. Beausoleil, J. Villen, S. A. Gerber, J. Rush, and S. P. Gygi, "A probability-based approach for high-throughput protein phosphorylation analysis and site localization," *Nature Biotechnology*, vol. 24, no. 10, pp. 1285–1292, 2006.

[8] H. Steen and M. Mann, "The ABC's (and XYZ's) of peptide sequencing," *Nature Reviews Molecular Cell Biology*, vol. 5, pp. 699–711, 2004.

[9] M. Kinter and N. E. Sherman, *Protein sequencing and identification using tandem mass spectrometry*. Wiley-Interscience, 2000.

[10] J. K. Eng, A. L. McCormack, and J. R. Yates, III, "An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database," *Journal of the American Society for Mass Spectrometry*, vol. 5, pp. 976–989, 1994.

[11] A. A. Klammer, S. Reynolds, M. J. MacCoss, J. Bilmes, and W. S. Noble, "Improved peptide identification and spectrum prediction using a probabilistic model of peptide fragmentation.," in *ASMS*, 2006.

[12] L. Y. Geer, S. P. Markey, J. A. Kowalak, L. Wagner, M. Xu, D. M. Maynard, X. Yang, W. Shi, and S. H. Bryant, "Open mass spectrometry search algorithm," *Journal of Proteome Research*, vol. 3, pp. 958–964, 2004. OMSSA.

[13] S. Kim, N. Mischerikow, N. Bandeira, J. D. Navarro, L. Wich, S. Mohammed, A. J. R. Heck, and P. A. Pevzner, "The generating function of cid, etd, and cid/etd pairs of tandem mass spectra: Applications to database search," *Molecular and Cellular Proteomics*, vol. 9, no. 12, pp. 2840–2852, 2010.

[14] J. Bilmes, "Dynamic Bayesian Multinets," in *UAI '00: Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence* (C. Boutilier and M. Goldszmidt, eds.), (San Francisco, CA, USA), Morgan Kaufmann Publishers, 2000.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, pp. 1–22, 1977.

[16] J. D. Egertson, J. K. Eng, M. S. Bereman, E. J. Hsieh, G. E. Merrihew, and M. J. MacCoss, "De novo correction of mass measurement error in low resolution tandem ms spectra for shotgun proteomics," *Journal of The American Society for Mass Spectrometry*, pp. 1–8, 2012.

[17] H. Ney and S. Ortmanns, "Progress in dynamic programming search for lvcsr," *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1224–1240, 2000.

[18] B. Y. Renard, M. Kirchner, F. Monigatti, A. R. Ivanov, J. Rappsilber, D. Winterc, J. A. Steen, F. A. Hamprecht, and H. Steen, "When less can yield more–computational preprocessing of ms/ms spectra for peptide identification," *Proteomics*, vol. 9, no. 21, pp. 4978–4984, 2009.

[19] L. Käll, J. D. Storey, M. J. MacCoss, and W. S. Noble, "Assigning significance to peptides identified by tandem mass spectrometry using decoy databases," *Journal of Proteome Research*, vol. 7, no. 1, pp. 29–34, 2008.

[20] L. Käll, J. Canterbury, J. Weston, W. S. Noble, and M. J. MacCoss, "A semi-supervised machine learning technique for peptide identification from shotgun proteomics datasets," *Nature Methods*, vol. 4, pp. 923–25, 2007.

[21] A. Keller, A. I. Nesvizhskii, E. Kolker, and R. Aebersold, "Empirical statistical model to estimate the accuracy of peptide identification made by MS/MS and database search," *Analytical Chemistry*, vol. 74, pp. 5383–5392, 2002.

[22] C. D. Wenger and J. J. Coon, "A proteomics search algorithm specifically designed for high-resolution tandem mass spectra," *Journal of proteome research*, 2013.

[23] R. I. Bahar, E. A. Frohm, C. M. Gaona, G. D. Hachtel, E. Macii, A. Pardo, and F. Somenzi, “Algebraic decision diagrams and their applications,” in *Computer-Aided Design, 1993. ICCAD-93. Digest of Technical Papers., 1993 IEEE/ACM International Conference on*, pp. 188–191, IEEE, 1993.