Bipartite matching generalizations for peptide identification in tandem mass spectrometry

Wenruo Bai
Department of Electrical Engineering
University of Washington
Seattle, WA 98195
wrbai@uw.edu

Jeffrey Bilmes
Department of Electrical Engineering
University of Washington
Seattle, WA 98195
bilmes@uw.edu

William S. Noble
Department of Genome Sciences, and Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195
william-noble@uw.edu

ABSTRACT

Motivation: Identification of spectra produced by a shotgun proteomics mass spectrometry experiment is commonly performed by searching the observed spectra against a peptide database. The heart of this search procedure is a score function that evaluates the quality of a hypothesized match between an observed spectrum and a theoretical spectrum corresponding to a particular peptide sequence. Accordingly, the success of a spectrum analysis pipeline depends critically upon this peptide-spectrum score function.

Results: We developed peptide-spectrum score functions that compute the maximum value of a submodular function under m matroid constraints. We call this procedure a *submodular generalized matching* (SGM) since it generalizes bipartite matching. We use a greedy algorithm to compute maximization, which can achieve a solution whose objective is guaranteed to be at least $\frac{1}{1+m}$ of the true optimum. The advantage of the SGM framework is that known long-range properties of experimental spectra can be modeled by designing suitable submodular functions and matroid constraints. Experiments on four data sets from various organisms and mass spectrometry platforms show that the SGM approach leads to significantly improved performance compared to several state-of-the-art methods. Supplementary information, C++ source code, and data sets can be found at https://melodi-lab.github.io/SGM.

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Biology and genetics; G.2.1 [**Discrete Mathematics**]: Combinatorics

General Terms

Algorithms, Experimentation

BCB'16, October 2–5, 2016, Seattle, WA, USA.

ACM 978-1-4503-4225-4/16/10 ...$15.00.
http://dx.doi.org/10.1145/2975167.2975201.

Keywords

proteomics, submodularity, mass spectrometry

1. INTRODUCTION

A shotgun proteomics experiment produces on the order of 10 mass spectra per second, each of which ideally is generated by a single peptide species. Hence, before the data can be used to answer high-level biological questions—like which functional classes of proteins are differentially expressed in one experimental condition versus another—we must first answer a simpler question, namely, "What peptide species was responsible for generating this observed spectrum?"

Over the past two decades, since the description in 1994 of the SEQUEST algorithm [1], by far the most common way to answer this question has been via database search. All such methods follow roughly the same form. The input is a set of observed spectra and a database of peptides, typically derived from the protein sequences of the organism under study. The database search algorithm is then deceptively simple: for each observed spectrum, we (1) extract from the database all peptides whose masses lies within a user-specified tolerance of the precursor mass associated with the spectrum, (2) compute a quality score for each peptide-spectrum match (PSM), and (3) assign to the spectrum the candidate peptide that received the best score.

Clearly, the success or failure of a database search method depends very strongly upon the quality of its score function. A good database search score function must exhibit at least three distinct properties. First, it must be quick to compute. At a production rate of 10 spectra per second, where each spectrum must be compared to hundreds or thousands of candidate peptides, an expensive score function will quickly become the bottleneck in any analysis pipeline. Second, the function must be accurate, in the sense that it usually succeeds in assigning the best score to the candidate peptide that actually was responsible for generating the observed spectrum. Third, the function must be well calibrated, so that the score assigned to the top peptide for one spectrum can be compared directly to the score assigned to the top peptide for a second spectrum. This third property is important because, in practice, the output of a database search algorithm is a ranked list of PSMs, one per observed spectrum. Because many observed spectra cannot be accurately identified, it is critical that the top of this ranked list of PSMs is highly enriched for correct identifications.

Dozens of database search score functions have been described in the literature (reviewed in [2]). Most rely on first transforming the peptide sequence into a theoretical spectrum and then computing some type of similarity score between the observed and theoretical spectra. Existing similarity functions rely on cross-correlation (SEQUEST) [1], dot product (X!Tandem) [3], hypergeometric scores (Myrimatch) [4], Poisson scoring (OMSSA) [5], probabilistic models (ProbID) [6] or simple counts of overlapping peaks (Morpheus) [7].

In this work, we propose to model the affinity between an observed and theoretical spectrum using a process we call a "submodular generalized matching" (SGM). This approach generalizes and provides greater modeling power than standard bipartite matching. In order to describe SGMs, we need first to describe bipartite matchings, submodular functions and their optimization, and matroids, all of which we briefly do in the next few paragraphs.

A maximum bipartite matching starts with a non-negative weighted bipartite graph (V, U, E, w), where V is a set of "left" vertices, U is a set of "right" vertices, $E \subseteq V \times U$ is a set of edges, and $w : E \to \mathbb{R}_+$ is a weight function on the edges, where $w(A) = \sum_{a \in A} w(a)$ for any edge set $A \subseteq E$. The goal of a maximum bipartite matching process is to find a set of edges $A \subseteq E$ that maximizes $w(A)$ but that is a matching, i.e., no vertex may be incident to more than one edge. Conceptually, one might treat computing a peptide-spectrum matching score as finding a maximum matching in a bipartite graph consisting of an observed spectrum (represented by the vertices V), a theoretical spectrum (the vertices U), and the edges E (feasible explanations of the observed by the theoretical spectra). In other words, given an edge $e \in E$ where $e = (v, u)$ with $v \in V$, $u \in U$, the weight $w(e)$ (which may be zero) indicates the degree to which theoretical peak u matches observed peak v.

For several reasons, however, maximum bipartite matching alone is inadequate to produce a good peptide-spectrum scoring function. First, only one edge in a traditional matching may be incident to a vertex, even though, as described below, several different theoretical peaks might potentially explain an observed peak. Conversely, a given theoretical fragmentation event might produce multiple effects in the observed spectrum. Second, the score function of a bipartite matching $w(A)$ is necessarily additive, meaning the weight of an edge does not change when considered in the context of other edges added to a matching. In practice, an optimal score function might need to combine matching scores in a non-additive fashion. To address the first problem, we use matroid constraints, and to address the second problem we use submodular functions. Together, these two approaches achieve our generalization.

A set function is said to be *submodular* if it exhibits the quality of diminishing returns, i.e., the incremental "gain" associated with a given set v decreases as the context in which v is considered grows larger. More formally, a function f, which assigns a real value to any subset of a given finite set E, is submodular when for any $A \subseteq B \subset E$ and $v \in E \setminus B$, f satisfies $f(A \cup \{v\}) - f(A) \geq f(B \cup \{v\}) - f(B)$. Submodular functions naturally model notions of information, diversity, and coverage in many applications such as information gathering [8], image segmentation [9, 10, 11], document summarization [12, 13], and string alignment [14]. If f satisfies the above definition everywhere with equality, then the function is called *modular*, and if the inequality is reversed (i.e., $\leq$ rather than $\geq$), then the function is called *supermodular*.

A *matroid* $M = (E, \mathcal{I})$ is a pair consisting of a ground set E and a set of subsets $\mathcal{I} = (I_1, I_2, \dots)$ where $I_i \subseteq E$ for all i. The subsets are said to be "independent" and to be a matroid, the subsets must satisfy certain properties. Specifically, the pair $M = (E, \mathcal{I})$ is a matroid if it satisfies: (i) $\emptyset \in \mathcal{I}$; (ii) $A \subset B \in \mathcal{I}$ implies that $A \in \mathcal{I}$; and (iii) given $A, B \in \mathcal{I}$ with $|A| > |B|$ then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$. Matroids are extremely powerful combinatorial objects, despite their simple definition, and have undergone years of mathematical study [15]. It is often the case that the independent sets of matroids are used as constraints in discrete optimization. For example, we may wish to maximize a set function subject to the solution being independent with respect to one or more matroids.

In fact, bipartite matching can be described in exactly this way. Given a weighted bipartite graph (V, U, E, w), we can formulate maximum bipartite matching as maximizing $w(A)$ subject to A being independent in two matroids. Depending on the matroids (as described below) we may relax the constraint that an edge is incident only to one vertex. In fact, with this formulation, each vertex (within either V or U) may have its own limit on incident edges. This means that, for a vertex $x \in V \cup U$, we may define a limit k_x on how many edges in a generalized matching may be incident to x. Submodular matching generalizes this idea further as follows: rather than maximize an additive weight function $w(A)$, we instead maximize a submodular function f. That is, submodular matching finds an edge set $A \subset E$ that maximizes $f(A)$ subject to multiple matroid constraints, $A \in \mathcal{I}_1 \cap \mathcal{I}_2$. Submodular matching is NP-hard, but it can be well-approximated extremely efficiently using a greedy algorithm that has a mathematical quality guarantee, namely, that the solution provided by the greedy algorithm is no worse than 1/3 times the best possible solution—this approximation ratio is constant regardless of the problem size [16]. Submodularity can be further exploited to accelerate the greedy implementation, leading to an algorithm often called *accelerated* [17] or *lazy greedy* having almost linear time complexity in practice. Hence, computationally, the approach scales to very large data set sizes.

In this work, we demonstrate how submodular matching with matroid constraints can be used to design a natural mass spectrometry score function that incorporates two important pieces of prior knowledge about peptide fragmentation. First, the proposed score function keeps track of situations in which a single observed peak can be explained by more than one peak in the theoretical spectrum. Such a collision might occur, for example, in the fragmentation of the +2 charged peptide SSLEVHIR. One of the prefix ions (SS) has an m/z value nearly exactly equal to one of the suffix ions (R). If the observed spectrum has a peak at 175 Da/charge, then existing score functions must choose between scoring this peak as a single match or as two matches. The submodular approach, by contrast, allows us to assign a diminished score to the second match. Second, our proposed score function allows us to, in effect, assign "extra credit" to pairs of observed-theoretical matches that are mutually reinforcing. For example, when we evaluate the hypothesis that an observed spectrum was produced by the fragmentation of peptide QNSHLTIK, we expect a single cleavage event to produce a prefix ion (e.g.,

QNS with m/z=330 Da/charge) and its corresponding suffix ion (HLTIK with m/z=611 Da/charge). If the observed spectrum contains peaks at both 611 Da/charge and 330 Da/charge, then SGM offers full joint, or non-diminished, credit to these pair of peaks, to account for their complementary nature. The SGM approach also simultaneously discredits any other sets of peaks that should not be in a complementary relationship with each other, for the given peptide. As we see above, the edge interactions can be both local and global, and this is exactly the power of submodular function, which can model these properties easily while allowing fast approximate maximization.

We demonstrate that our proposed score function can be computed efficiently and that the resulting score function outperforms a variety of state-of-the-art methods across multiple data sets. Specifically, we compare SGMs with three existing methods, XCorr [18], MS-GF+ [19], and the XCorr p-value [20]. We compute the number of spectra identified at a 1% false discovery rate (FDR) threshold, observing statistically significant improvements relative to the second-best method ($p < 0.05$, Wilcoxon signed-rank test).

2. BACKGROUND: MASS SPECTROMETRY DATABASE SEARCH

The spectrum identification problem is described as follows. Given an observed spectrum s in a dataset S with precursor m/z value of m^s and precursor charge value c^s, and given a database $\mathcal{P}$, we wish to find the peptide $p \in \mathcal{P}$ responsible for generating s. Let P_s be a subset of $\mathcal{P}$ containing peptides with masses approximately equal to the precursor m^s, subject to a tolerance ω, i.e., $P_s = P(m^s, c^s, \mathcal{P}, \omega) = \{p : s \in \mathcal{P}, |m(y)/c^s - m^s| \leq \omega\}$ The value ω is determined by the instrument settings during the first round of mass spectrometry. In database search, only the peptides in P_s, the *candidate peptides*, are scored. A scored spectrum-peptide pair is called a peptide-spectrum match (PSM).

Denoting an arbitrary scoring function as $\mathfrak{s}(p, s)$, the spectrum identification problem, for a given s, computes:

$$p^* \in \underset{p \in P(m^s, c^s, \mathcal{P}, \omega)}{\operatorname{argmax}} \mathfrak{s}(p, s) \quad (1)$$

The key difference between various search methods is the scoring function $\mathfrak{s}(p, s)$, which strongly determines a method's success. We next introduce the widely used SEQUEST method and then show how we generalize it using submodular generalized matching.

2.1 The SEQUEST algorithm

SEQUEST [1] is the very first database search engine and is still in widespread use. We begin by describing SEQUEST because it is relatively simple and runs very fast while achieving acceptable performance. SEQUEST also provides a good starting point for SGM. Two other score functions, MS-GF+ and the XCorr p-value, are described in Section 4.2. The SEQUEST search procedure proceeds as follows.

Prior to analysis, each observed spectrum is pre-processed and each candidate peptide is used to generate a theoretical spectrum [21]. The spectrum pre-processing discretizes the m/z axis and reduces the amount of intensity variation along the m/z axis. The theoretical spectrum generation includes prefix (b-ion) and suffix (y-ion) peaks, as well as several neighboring peaks that represent secondary losses of ammonia

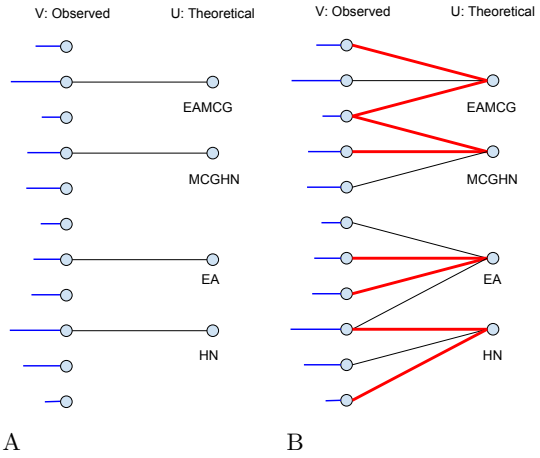


Figure 1: PSM bipartite graphs. (A) V is the set of all peaks in an observed spectrum. The horizontal lines attached on the left represent the peak intensities in the observed spectrum. U is the set of all fragment ions for a given theoretical spectrum derived from a given peptide. An edge (v, u) connects $v \in V$ with $u \in U$ if v might possibly be explained by u with an associated non-negative weight. (b) Red lines are selected edges for a particular match. Non-horizontal edges might correspond to neutral losses.

(NH_3), water (H_2O) and carbon monoxide (CO) molecular groups.

The traditional SEQUEST score function, called XCorr, is a dot product between one theoretical and one observed spectrum, and can be calculated as follows:

$$\text{SEQUEST}(p, \tilde{s}) = \langle p, \tilde{s} \rangle - \frac{1}{151} \sum_{\tau=-75}^{75} \sum_{i=1}^{N} p(i)\tilde{s}(i-\tau) \quad (2)$$

$$= \langle p, \tilde{s} - \frac{1}{151} \sum_{\tau=-75}^{75} \tilde{s}_\tau \rangle \quad (3)$$

where x is the observed spectrum, y is the theoretical spectrum and $\tilde{s}' = \tilde{s} - \frac{1}{151}\sum_{\tau=-75}^{75} \tilde{s}_\tau$ is called the background spectrum with $\tilde{s}_\tau(i) = \tilde{s}(i-\tau)$.

3. SUBMODULAR GENERALIZED MATCHINGS

Producing a score function S via the submodular matching method requires four steps. First, for each PSM to be scored, we create a distinct bipartite graph where the left vertices V correspond to the observed peaks and the right vertices U to theoretical peaks. Second, we produce a submodular evaluation function $f(A)$ defined over edges of that bipartite graph. Third, we define a set of matroids $\mathcal{M}_v = (E, \mathcal{I}_v)$ and $\mathcal{M}_u = (E, \mathcal{I}_u)$ whose independent sets are to be used as constraints. Fourth, we compute the score itself, $\mathfrak{s} = \max_{A \in \mathcal{I}_v \cap \mathcal{I}_u} f(A)$. We discuss each of these steps in detail below.

3.1 Bipartite graph production

All of our analyses depend upon a bipartite graph representation of pairs of observed and theoretical spectra (Fig-

ure 1A). A bipartite graph is one whose vertices can be divided into two disjoint sets U and V such that every graph edge $e = (v, u)$ connects a vertex $v \in V$ to a vertex $u \in U$. We create a bipartite graph $G = (V, U, E)$ for each PSM, where V is the set of peaks in the observed spectrum, U is the set of peaks in the theoretical spectrum, and for an edge $e = (v, u) \in E$, theoretical peak u explains the existence of observed peak v to a degree with a corresponding weight $w(e)$.

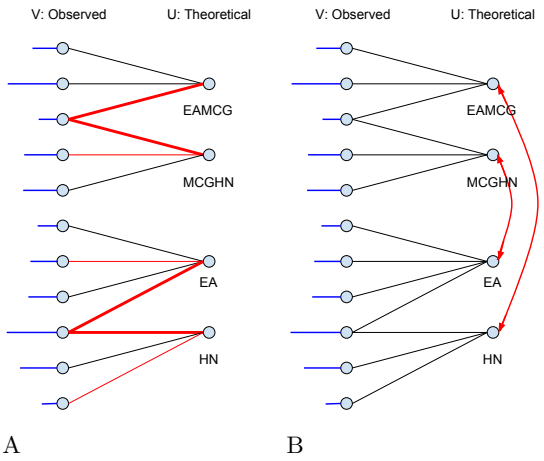
The weight assigned to a given edge $e = \{v, u\}$ is determined by the m/z and intensity of the observed and theoretical peaks. Specifically, $w(\{v, u\})$ is defined as $w'(\{v, u\})x_v y_u$, where $w'(\{v, u\})$ is a weight matrix. $w'(\{v, u\})$ describes the general biological relationship between the observed and theoretical peaks given their mass to charge ratio, m_v and m_u. For example, if $m_u - m_v$ is close to 0 or 18 (water loss), $w'(\{v, u\})$ is high. Note that the matrix w' is sparse since we do not expect relationships to exist between two peaks at an arbitrary relative m/z offset. Also, the values of w' are a function only of the m/z difference, i.e., $w'(\{v, u\}) = h(m_v - m_u)$ where $h(\cdot)$ is a function. We learn $h(\cdot)$ empirically using the average intensity near high-confidence b-ions and y-ions (Supplementary Section B). Note that our empirical weights may also be applied to other scoring methods. We show in Section 5.3 that using our empirical weighting scheme does indeed yield improvements when used with a method like SEQUEST; however, the submodular function introduced in the next section makes better use of this weighting information.

3.2 Submodular evaluation function

In this section, we define the set function $f(A)$ which produces a score that corresponds to how well a set of theoretical peaks (the vertex subset of U incident to edges A) explains a set of observed peaks (the vertex subset of V incident to edges A). While the function f may evaluate a set of edges that are not matchings, when we produce PSM scores, we only consider sets A that constitute matchings due to the matroid constraints. Therefore, we describe this function on the assumption that A is a matching.

In this section, we describe a set of properties of the scoring function f that naturally lead us to the class of monotone non-decreasing submodular functions. Then we discuss the particular submodular function we have designed for PSM scoring. First, in general, we want the theoretical spectrum to explain as much of the observed spectrum as possible. This means that if $A \subseteq B$ are two matches, one a subset of the other, then we expect B to score no worse than A. In other words, f should be monotone non-decreasing, or $f(A) \leq f(B)$ whenever $A \subseteq B$. Second, in many cases, we would not wish to over-credit a given set of theoretical-observed matches. For example, if an observed peak is well-explained by a given theoretical peak, then any other theoretical peak that also explains the same observed peak should be discounted or, in some sense, "explained away." Without some kind of discounting procedure, we would be overconfident about the observed peak. Moreover, a non-discounted score function might discourage an optimization procedure from finding alternative explanations of the theoretical peak. This is a natural diminishing returns property, and can be described as $f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B)$, $\forall A \subseteq B \subset E$ and $e \notin B$. This is in fact a defining property of submodular functions.

In general, even approximately maximizing an arbitrary set function costs $O(2^m)$, where $m = |E|$, exponential in the set of edges, and hence is hopelessly intractable. When the function is submodular, however, a simple and efficient greedy algorithm can maximize a monotonically non-decreasing submodular function subject to cardinality constraints can achieve a $1 - \frac{1}{e}$ guarantee. The same greedy algorithm maximizes said function subject to two matroid constraints (described below) with a 1/2 guarantee. Hence, submodular functions are both natural for the problem of generalized matching for PSM scores, but also allow efficient algorithms to be used to obtain high quality approximate optima.



A B

Figure 2: Illustration of submodular functions. V is the set of all peaks in an observed spectrum, with blue horizontal lines representing the observed spectrum intensities. U is the set of all fragment ions for a given theoretical spectrum. Edge (v, u) with $v \in V$ and $u \in U$ if v might possibly be explained by u with an associated non-negative weight. E is set of all edges. (A) Two theoretical peaks match the same observed peak. The submodular function assigns a score intermediate between the max and the sum of the two edge scores. (B) Two complementary theoretical peaks match to different observed peaks. The submodular function assigns a "bonus" for this complementarity.

There are many possible submodular set functions that one might choose from. We have developed a family of such functions, as described below, that are uniquely suited to PSM scoring and that allow for a rich and powerful relationship to exist between a peptide and its observed spectrum. Accordingly, we describe two submodular functions, f_1 and f_2, that each capture an important part of PSM scoring. Taking positive weighted sums of such functions preserves submodularity (e.g., $f = f_1 + f_2$ is submodular when the f_i's are submodular), so we combine the two functions for our final scoring method. This is discussed next.

3.2.1 *Matching one observed peak to multiple theoretical peaks*

In general, an observed peak should not be over-explained and hence over-valued by multiple ions. When an observed peak is accounted for on the left side, if it is matched again, then the second match should not be given as much credit as

when the second match is considered alone. This diminishing returns property is perfectly modeled by a submodular function since if e_i is the i^{th} match, then $f(e_i|e_1,\ldots,e_{i-1}) \leq f(e_i)$ is a consequence of submodularity. This relationship is depicted in Figure 2A and can be represented using the following function:

$$f_1(A) = \sum_{v \in V} g_1\left(\sum_{e \in A \cap \delta v} w(e)\right), \tag{4}$$

where $g_1(x)$ is a monotonically non-decreasing concave function mapping from $\mathbb{R}^+$ to $\mathbb{R}$, and $\delta v \subseteq E$ are the edges incident to node $v \in V$. The function $f_1(A)$ therefore provides submodularity on the observed side thanks to the use of δv. $f_1(A)$ is a submodular function since it is a sum of monotonically non-decreasing concave functions composed with non-negative modular set functions [22]. Note that the choice of $g_1(x)$ is flexible, and different $g_1(x)$ functions allow us to model this interaction in diverse ways. In our experiments we use $g_1(x) = \beta \log(1 + \beta^{-1}x)$, where β is a parameter. We also experimented with $g_1 = x^\alpha$ for $\alpha = 1/2$, $1/3$ and $1/4$, but this function yielded worse empirical results (data not shown).

3.2.2 *Scoring complementary pairs of matched observed peaks*

Relative to the precursor mass, a b-ion and corresponding y-ion always appear in pairs in a proper spectrum. Therefore, if a b-ion v_b is found, then a score function should ideally provide a boost if the corresponding y-ion $v_{\bar{b}}$ is also present (Figure 2B). This leads to a relationship of the form $f(a_b \cup a_{\bar{b}}) > f(a_b) + f(a_{\bar{b}})$, where $a_b \in \delta v_b$ and $a_{\bar{b}} \in \delta v_{\bar{b}}$. Unfortunately, this relationship is supermodular (essentially a negative submodular), which is a much more difficult objective to optimize with mathematical quality guarantees. Therefore, in order to express the inherent complementarity between b- and corresponding y-ions, we use a modular function. Such functions are as close to supermodular as possible while still being submodular. Hence, relationships among edges that are naturally supermodular use a modular function; relationships that are naturally modular use a weakly submodular function, and relationships that are naturally submodular use a strong submodular function. The "strength" of the submodularity, in this context, corresponds precisely to the curvature (magnitude of the second derivative) of the concave functions that are employed (e.g., a linear function is concave, but leads to a modular function). For PSM scoring, we therefore use the following function, which acts as our submodular surrogate for a supermodular relationship between ions and co-ions:

$$f_2(A) = \gamma_2 \sum_{i \in U_{\text{b-ion}}} \left[\sqrt{m\left(A \cap \left(E_i \cup \sum_{j \neq i} E_{\bar{j}}\right)\right)} + \sqrt{m\left(A \cap E_{\bar{i}}\right)}\right] \tag{5}$$

where $U_{\text{b-ion}}$ is the set of b-ions for a given theoretical spectrum, and if i is an b-ion then $\bar{i}$ is the corresponding y-ion, and vice verse. (We say that $\bar{i}$ is the co-ion of i, and any $j \neq i$, $\bar{j}$ is a "non-co" ion.) The function $m(A) = \sum_{e \in A} w_e$ is a modular weight function. The coefficient $\gamma_2 = \sqrt{\sum_{e \in E} w(e)}$ scales the function to be combined with the other f_i's. Hence, we see that f_2 maximally credits any edge sets in A that correspond to an ion and its co-ion (since they are in different components in each term of the sum), whereas any edges that do not have this complementary (i.e., an ion and its non-co ions) are discounted if they are jointly selected within A.

3.2.3 *Combination of submodular functions*

The above submodular functions can be used together by using a weighted sum, but the values of the weights are important. It is particularly important, when summing submodular functions, that one does not dominate the other. For example, if f_1 is naturally larger than f_2, then when we maximize the sum $f_1 + f_2$, the solution will focus primarily on f_1 without bothering to score f_2 well. To avoid this problem, we us the following combination:

$$f(A) = [\lambda_1 f_1(E) + (1-\lambda_1) f_2(E)] \left[\lambda_2 \frac{f_1(A)}{f_1(E)} + (1-\lambda_2)\frac{f_2(A)}{f_2(E)}\right]$$

Inside the brackets, we normalize each function so that $0 \leq f_1(A)/f_1(E) \leq 1$ are combinable in a convex mixture controlled by parameter λ_2, where $0 \leq \lambda_2 \leq 1$. The mixture, however, is still normalized to be in the range $[0, 1]$ so we then calibrate this result by multiplying by the constant $\lambda_1 f_1(E) + (1-\lambda_1) f_2(E)$. Intuitively, this calibration ensures that PSM scores are comparable across different observed spectra (Figure 7 and Supplementary Section A).

3.3 Matroid constraints

A simple way to produce a PSM score is to use $\mathfrak{s} = f(E)$, thereby allowing *all* possible edges to comprise a score. This would be a poor choice, however, since most observed spectra contain many noise peaks, and there is no oracle to decide whether a given observed peak is real signal or not. Although preprocessing steps can be used to limit the influence of noise, it is impossible to fully eliminate it. Another reason is that the observed peaks of one fragment ion may accidentally appear in the position of other ions. In this case, the latter ion will have an unexpected co-ion, and we do not want to reward this. Thus, using all edges E will produce a final score that suffers from many false interactions. Fortunately, when we use a submodular function as described above, the edges that do accurately explain the observed spectrum will, in general, be much more highly weighted than the noise edges. Moreover, limiting the set of edges being scored to satisfy certain constraints (which we call a "generalized matching") forces the edges comprising a score to compete with each other. This competition limits the ability of incorrect edges to artificially boost the score. Moreover, for PSMs that are not true peptide-spectrum matches, even the best set of edges will not lead to a high score.

The next question is how to produce the feasible subsets of edges that may be scored. One possible approach would use standard bipartite matching constraints. This, as mentioned in Section 1, is problematic since every edge in such a matching maybe incident to no more than one vertex. In practice, one fragment ion might truly explain multiple observed peaks, and one observed peak might be explained by multiple fragment ions. What we would like is a "generalized matching" where every vertex may be incident to more than one but still a limited number of edges.

This generalized matching property can easily be handled using two matroid constraints. As mentioned in Section 1, a matroid is a pair $(E, \mathcal{I})$, where E is a finite set and $\mathcal{I}$ is a family of what are called "independent" subsets of E. A matroid constraint means that any edge set solution A would have $A \in \mathcal{I}$. A particular kind of matroid that is useful for our purposes is a *partition matroid*. A partition matroid is based on a partition of E into ℓ disjoint subsets sets $\{E_i\}_{i=1}^{\ell}$, and ℓ non-negative integers $\{k_i\}_{i=1}^{\ell}$, where $\cup_{i=1}^{\ell} E_i = E$ and $E_i \cap E_j = \emptyset$ for $i \neq j$. The independent sets of a partition

matroid are defined as $\mathcal{I} = \{A|A \subseteq E, |A \cap E_i| \leq k_i, \forall i\}$. Two natural partition matroids over the edges E may be defined based on the edges incident either to vertices $v \in V$ or $u \in U$. That is, we define two partition matroids $\mathcal{M}_v = (E, \mathcal{I}_v)$ and $\mathcal{M}_u = (E, \mathcal{I}_u)$, where $\mathcal{I}_v = \{A|A \subseteq E, |A \cap \delta(v)| \leq k_v \, \forall v \in V\}$ and $\mathcal{I}_u = \{A|A \subset E, |A \cap \delta(u)| \leq k_u \, \forall u \in U\}$ and where, again, $\delta(v)$ is the set of edges incident to v and likewise for $\delta(u)$. We immediately see that a matching A corresponds to a set with $A \in \mathcal{I}_v \cap \mathcal{I}_u$, where $k_v = k_u = 1$ for all $v \in V$ and $u \in u$. A natural and immediate generalization of bipartite matching, moreover, is to set $k_v \geq 1$ and/or $k_u \geq 1$, where k_v (resp. k_u) corresponds to the limit of allowable incident edges to vertex v (resp. u) in any generalized matching. The values k_v and k_u are seen as parameters of the constraint and correspond, e.g., to allowing an observed peak to be explained by multiple fragment ions and one ion to be explained by multiple observed peaks in the spectrum. In practice, we set $k_v = \infty$, because in the four datasets we studied, the maximum number of edges connected to v is only 2. Hence, there is no need for constraints on observed side.

3.4 The final score function

Our final PSM score, for a given peptide and spectrum, is computed as $\max_{A \in \mathcal{I}_v \cap \mathcal{I}_u} f(A)$, where $\mathcal{I}_v$ and $\mathcal{I}_u$ are the independent sets of the corresponding partition matroids with appropriate values of k_v and k_u. Fortunately, as mentioned previously, this optimization problem can be easily and scalably calculated using a simple greedy algorithm which has a guarantee of 1/3 [23].

Our score function can be regarded both as a generalization of XCorr and of maximum bipartite matching. XCorr uses a dot product operation to calculate the score which is, in fact, equivalent to maximizing a modular set function subject to a particular bipartite matching constraint, namely one where $|\delta v| = |\delta u| = 1$, i.e., where edges exist only between a theoretical peak and its corresponding observed peak. One difference between such a bipartite score and XCorr is that XCorr uses a normalized spectrum $\tilde{s}' = (\tilde{s} - \frac{1}{151}\sum_{\tau=-75}^{75} \tilde{s}_\tau)$ (the difference between a foreground and average background spectra, which can be negative) rather than just a non-negative foreground spectrum. In a submodular matching, we cannot use the background spectrum directly since it is not always positive, something that would violate both the monotonicity and submodularity of our objective $f(A)$ and render the efficient greedy algorithm mathematically vacuous. An alternative strategy to use background information without resulting in negative values is to subtract a similar background factor after finding the max matching, as in: $\mathfrak{s} = \max_{A \in \mathcal{I}_v \cup \mathcal{I}_u} f(A) - \alpha\tau$ where $\tau = \sum_{i \in U} \frac{1}{151}\sum_{j=-75}^{75} \tilde{s}(m/z_i + j)$, and where α is a parameter and τ is a background factor. Subtracting τ from the score is not precisely the same as using the background spectrum in XCorr but has the same intended purpose and, as we show below, works well while preserving submodularity, monotonicity, and hence the mathematical guarantees and applicability of the greedy algorithm.

3.5 Score calibration

As described in Section 1, a good score function must be well calibrated. We say that a PSM score function is well calibrated if a score of x assigned to spectrum σ_i has the same meaning or significance as a score of x assigned to spectrum σ_j. During a database search, the top-scoring PSMs from many different are combined into a final, ranked list. If the scores of different observed spectra are not comparable, then the ranking will not be reflective of the true qualities of the PSMs. We therefore calibrate each PSM's score by subtracting the average score of all of the PSMs involving that spectrum, as follows

$$\mathfrak{s}^*(p, s) = \mathfrak{s}(p, s) - \frac{\sum_{p' \in P_s} \mathfrak{s}(p', s)}{|P_s|}, \quad (6)$$

where $\mathfrak{s}(p, s)$ is the non-calibrated score, and P_s is the set of candidate peptides associated with spectrum s. The subtraction term is a constant with respect to each spectrum and does not affect which peptide is chosen. Rather, it only helps to produce a good overall of ranking of top-scoring PSMs. In other search methods, such as the P-value [20] approach, scores are calibrated using dynamic programming. Using such a technique for SGM scoring is left to future research, since SGM scoring is inherently non-linear, unlike SEQUEST which is a simple linear dot-product-based score.

4. METHODS

We use four different data sets to benchmark the performance of SGM relative to three state-of-the-art methods, and we employ several different quality measures to compare the results.

4.1 Data sets

The yeast (*S. cerevisiae*) and worm (*C. elegans*) data sets were collected using tryptic digestion followed by acquisition using low-resolution precursor scans and low-resolution fragment ions. A total of 108,291 yeast and 68,252 worm spectra with charges ranging from 1+ to 3+ were collected. Each search was performed using a ±3.0 Da tolerance for selecting candidate peptides. Peptides were derived from proteins using tryptic cleavage rules without proline suppression and allowing no missed cleavages. A single fixed carbamidomethyl modification was included. Further details about these data sets, along with the corresponding protein databases, may be found in [24].

The malaria parasite *Plasmodium falciparum* was digested using Lys-C, labeled with an isobaric tandem mass tag (TMT) relabeling agent, and collected using high-resolution precursor scans and high-resolution fragment ions. The data set consists of 240,762 spectra with charges ranging from 2+ through 6+. Searches were run using a 50 ppm tolerance for selecting candidate peptides, a 0.03 Da fragment mass tolerance, a fixed carbamidomethyl modification, a fixed TMT labeling modification of lysine and N-terminal amino acids. Further details may be found in [25].

The human dataset was digested using trypsin, labeled with an isobaric tandem mass tag (TMT) relabeling agent, and collected using high-resolution precursor scans and high-resolution fragment ions. The data set consists of 1,133,534 spectra with charges ranging from 2+ through 6+. Searches were run using a 10 ppm tolerance for selecting candidate peptides, a 0.02 Da fragment mass tolerance, a fixed carbamidomethyl modification, a fixed TMT labeling modification of lysine and N-terminal amino acids. Further details may be found in [26].

4.2 Database search methods

We compare our method with three baseline methods: SEQUEST, MS-GF+ [19], and the XCorr p-value [20]. We have

already described SEQUEST and its XCorr score function in Section 2. However, in practice, as demonstrated below, the raw XCorr score is poorly calibrated. The MS-GF+ [19] search engine uses an alternative score function and employs dynamic programming to exactly compute the score distribution over the universe of candidate peptides for a linear scoring function. This leads to far better calibration. For the evaluations, we use MS-GF+ version 9980, and PSMs are ranked by the "Evalue" score. The XCorr p-value [20] uses a similar dynamic programming approach to calibration, applied to the SEQUEST XCorr score. Our experiments use a re-implementation of SEQUEST called "Tide" [18], available in Crux version 2.1.16790 [27]. For clarity, we refer to the two variants of SEQUEST as "XCorr" (for results based on ranking with the raw XCorr score) and "p-value" (for results based on ranking by the XCorr p-value).

To ensure a fair comparison, search settings for all algorithms were set to equivalent values whenever possible. In general, we set the search engine parameters so that the discretization of the fragment m/z axis is appropriate for the given data set. The only exception is that, for technical reasons related to the dynamic programming procedure, the XCorr p-value can only be calculated using an m/z resolution of 1.0005079 Da. For both Tide and MS-GF+, default search parameters are used, except that, to make a fair comparison, isotope peak errors are turned off in MS-GF+. Furthermore, to avoid variability in how proteins are digested to peptides or how "decoy" peptides (see Section 4.3) are generated, we use the same digested peptide database as input to all search algorithms. These databases are created by using the "tide-index" command in Crux, with "clip-nterm-methionine" set to "True."

4.3 Evaluation of methods

The major difficulty in comparing the performance of different search engines is that we lack a ground truth against which to judge accuracy. A widely used approach is target/decoy search [28]. The "target" set is the real candidate peptide set. The "decoy" set is created by randomly permuting the (non-terminal) amino acids of each target peptide to create a corresponding "decoy" peptide. For each spectrum, we perform the search on a database comprised of targets plus decoys, and the single peptide with the highest score is selected [29]. If more than one peptide has the highest value, then such ties are broken randomly.

We consider two complementary metrics for comparing two algorithms using target/decoy search. The first, simpler approach is the "target match percentage" (TMP), defined as the fraction of observed spectra for which the top-scoring match involves a target peptide. For a perfectly random score function, we expect the TMP to be ∼50%. The best possible TMP is 100%; however, in practice any real data set will contain spectra that cannot be identified, either because the corresponding generating peptide is not in the given peptide database or because the spectrum was generated by a non-peptide contaminant and these spectra are expected to match targets and decoys with equal frequency. TMP is not a widely used performance measure; however, we employ it here because TMP provides a measure of the quality of a score function that is independent of a score function's calibration. This is because the TMP never involves comparing scores for PSMs involving different spectra. Hence, the distribution of PSM scores for spectrum A can be dramatically different from the distribution of PSM scores for spectrum B, but the TMP achieved by the score function can still be high.

Table 1: Target match percentage achieved by the four score functions on four data sets. In each row, the maximal value is shaded red.

Dataset	SGM	MS-GF+	p-value	XCorr
yeast	70.59	66.19	65.47	64.91
worm	82.83	77.59	77.39	76.43
Plasmodium	69.91	66.85	65.53	69.39
human	74.40	60.40	73.26	74.12

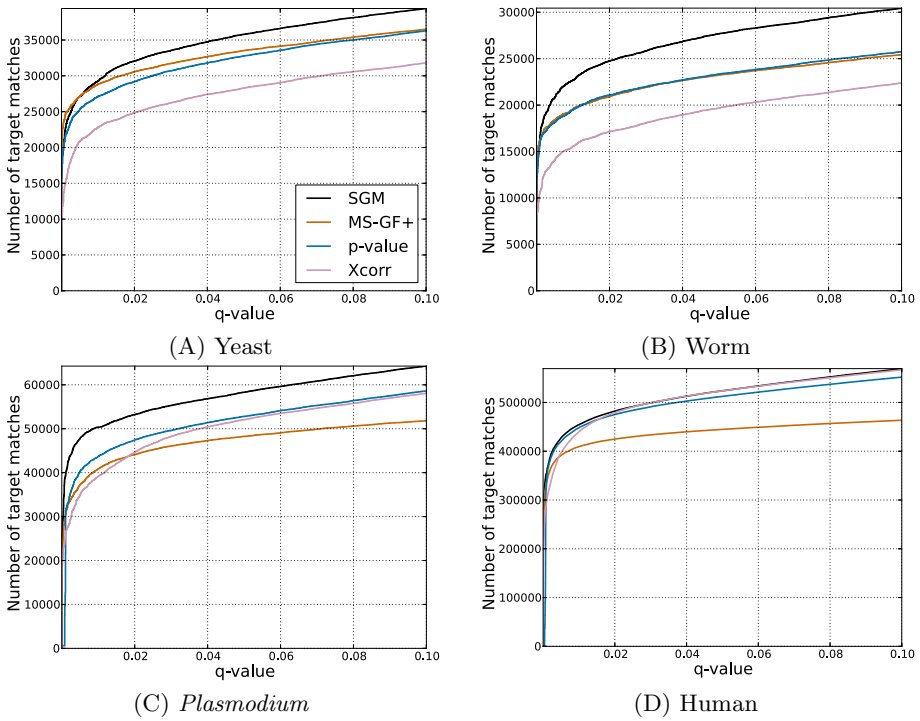
Evaluating the calibration of a score function requires additional steps. After finding the top-scoring PSM for each spectrum, the next step is to set a score threshold and to label every PSM scoring better than the threshold as "accepted." Then we can calculate the false discovery rate at a given threshold as $\text{FDR} = \frac{\text{number of accepted decoy PSMs}}{\text{total number of accepted PSMs}}$. In practice, we compute for each PSM its corresponding q-value, defined as the minimum FDR at which a PSM with that score is accepted [30]. Because many mass spectrometry studies report results using an FDR threshold of 1%, we sometimes report the number of target PSMs accepted at $q \leq 0.01$. To evaluate the performance of a search engine over a variety of q-value thresholds, we also plot the number of accepted target PSMs as a function of q-value threshold and compute the area under the plot from $0 \leq q \leq 0.1$. Because the FDR-based evaluation involves creating a ranked list of top-scoring PSMs from many different spectra, this metric requires good cross-spectrum calibration.

5. RESULTS

5.1 Comparison of four search methods

We begin by computing the target match percentage of the four search methods—SGM, MS-GF+, p-value and XCorr—on the four data sets described in Section 4.1. In all four cases, SGM achieves the greatest TMP (Table 1). Each of these data sets consists of multiple mass spectrometry runs: 3, 3, 20, and 100, for the yeast, worm, *Plasmodium* and human data sets, respectively. Consequently, for the latter two data sets we were able to compute the TMP separately for each run and then use a Wilcoxon signed-rank test to identify statistically significant differences. This analysis (Supplementary Figure 6) indicates that, for the malaria data set, SGM performs significantly better than XCorr ($p = 0.11$), and for the human data set, SGM performs significantly better than all three competing methods ($p = 1.6 \times 10^{-5}$). The consistently good TMP performance of the SGM method on these diverse data sets indicates that, for each observed spectrum, this score function does a very good job of ranking the generating peptide above all other candidate peptides.

Next we evaluate the methods using false discovery rate estimation, thereby additionally taking into account the calibration of the scores. In practice, this evaluation is the most important, since it directly reflects how the end user will interpret the results of the search. The results (Figure 3) suggest that, once again, SGM performs better than MS-GF+, XCorr and the XCorr p-value for the yeast, worm and *Plasmodium* data sets. We quantified the performance of each method by counting the number of accepted PSMs at a q-value threshold of 0.01, and we again used a Wilcoxon



(A) Yeast (B) Worm

(C) *Plasmodium* (D) Human

Figure 3: FDR-based comparison of search methods. Each panel plots, for a single data set and a variety of score functions, the number of spectra identified as a function of FDR threshold.

signed-rank test to estimate significant differences for the data sets comprised of many runs. This analysis shows that SGM significantly outperforms all three competing data sets on both the *Plasmodium* and human data sets. The p-values relative to the second-ranked method, XCorr, are 0.0015 for *Plasmodium* and 0.0014 for the human data set (Figure 6).

5.2 Verifying the Utility of Submodularity

The SGM approach employs a combination of two submodular functions, each of which is designed to capture a particular property of high quality matches between spectra and peptides. To verify that the good results in Section 5.1 are indeed a reflection of these properties, we examined more closely the high-confidence identifications produced by SGM. For this analysis, we focus on a single, randomly selected run ("TMT10") from the *Plasmodium* data set.

First, we show that f_1 discourages choosing multiple edges incident to one observed peak. To do so, we compare the number of multiply matched observed peaks in high confidence PSMs ($q \leq 0.01$) generated using two methods: a simple, modular approach versus using f_1. The distribution of the number of multiply matched peaks decreases when we use f_1 (Figure 4A), which implies that our submodular function discourages multiply matched observed peaks.

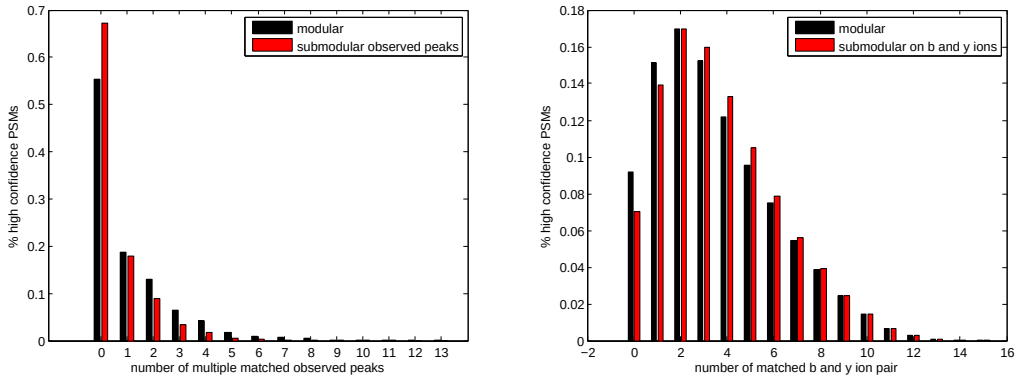
Second, we show that f_2 encourages choosing a b-ion if its corresponding y-ion is already chosen, and vice verse. As before, we compute the number of jointly matched b- and y-ion pairs among the high confidence PSMs, with and without inclusion of f_2. As expected, the number of matched b- and y-ion pairs increases when we use f_2 (Figure 4B), implying that our submodular function indeed encourages such matching.

5.3 Investigation of empirical weights

Our method is different from XCorr in two respects. First, we use empirically derived edge weights based on an analysis of the data (Section 3.1). Second, we generalize the dot product score to one based on a submodular generalized matching. We performed further experiments to ascertain which of these two changes are primarily responsible for the good results reported above.

Traditional methods, such as XCorr and MS-GF+, use simple distinct weights such as 10, 50, or even binary values. We say they are using "classical" weights, where $w'(\{v,u\}) = \delta_{m_v,m_u} + 0.2\sum_{l\in\{-17,-18,-28\}} \delta_{m_v+l,m_u}$, $\delta_{i,j} = 1$ if $i = j$ and 0 otherwise. So the classical weight is 1 if $m_v - m_u = 0$; 0.2 if $m_v - m_u \in \{-17,-18,-28\}$ and 0 otherwise. Switching to our empirical weights also helps performance in their case as well as ours.

To do this, we analyze the contributions of these two components (weights vs. SGM score) separately. We do the test on a single run ("TMT10") from the *Plasmodium* data set, evaluating performance using target-decoy q-values. We compare four different search methods, SGM with and without empirical weights, and XCorr with and without empirical weights. In this experiment (Figure. 5), the combination of SGM with empirical weights achieves by far the best performance. While the empirical weights help both XCorr and SGM, the SGM with the standard XCorr weights is still better than XCorr with our empirical weights. Hence, it is fair to say that the SGM process is the more important of the two. This is not surprising because the power of SGM, and the use of submodular functions, is that we can allow global long-range interaction amongst edge scores, something that is not at all possible with XCorr.



A B

Figure 4: PSM properties captured by the submodular function. (A) The number of multiple matched observed peaks decreases when we use the submodular function f_1. (B) The number of multiple matched b- and y-ion pairs increases when we use the submodular function f_2.

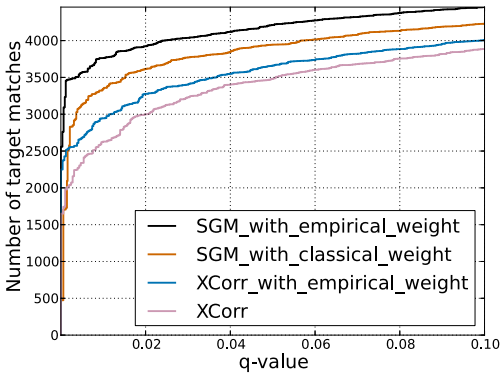


Figure 5: Evaluation of the SGM and XCorr score functions on a subset of the *Plasmodium* data set, with and without using the empirical weighting scheme.

SGM	MS-GF+	p-value	XCorr
8.48×10^{-2}s	8.99×10^{-2}s	6.89×10^{-2}s	$3.39 \times 10^{-3}s$

Table 2: Run time of four methods per spectrum on the *Plasmodium* TMT-10 data set.

5.4 Run time

To evaluate the running time of SGM relative to other search tools, we measured the wall clock time of four search methods on a single Intel Core 2 Quad Q9550 2.83GHz CPU on the *Plasmodium* TMT-10 dataset. The dataset consists of 8841 spectra, each with an average of 365 target peptides and an equal number of decoy peptides. This analysis (Table 5.4) shows that SGM has a comparable run time with MS-GF+ and the Tide p-value. The raw XCorr score calculation is extremely fast because it simply consists of calculating a dot product, with no explicit calibration procedure.

6. CONCLUSION

We have introduced a novel class of score functions for use in tandem mass spectrometry database search. A key advantage of our SGM is that we can model many PSM properties, including long-range interactions among peaks in an observed spectrum, using a rich and powerful framework, namely that of submodularity and various matroid constraints. An additional advantage is that our model runs fast since we may use a simple accelerated greedy algorithm to find the maximum value of the submodular function with a mathematical quality guarantee of $\frac{1}{3}$. We show that our approach achieves statistically significant improvements in performance relative to several state-of-the-art methods according to two different evaluation metrics.

In SGM, we use three hyperparameters, chosen from a grid of possible values (Supplementary Section C). Our empirical analysis (Supplementary Section D) suggests that these hyperparameters generalize well across datasets. Therefore, we do not need to re-tune these hyperparameters to different datasets.

In future studies, we will explore other submodular functions in an attempt to further improve performance. For example, we can explore algorithms that can learn submodular functions and parameters from training data [13, 31]. We can will also consider other generalizations of our framework to still better model the process of spectrum generation.

Acknowledgments.

This work was funded by National Institutes of Health awards R01 GM096306 and R01 CA180777, the National Science Foundation under Grant No. IIS-1162606, and by Google, Microsoft, and Intel research awards.

7. REFERENCES

[1] 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.

[2] A. I. Nesvizhskii, "A survey of computational methods and error rate estimation procedures for peptide and protein identification in shotgun proteomics," *Journal of Proteomics*, vol. 73, no. 11, pp. 2092 – 2123, 2010.

[3] R. Craig and R. C. Beavis, "Tandem: matching

proteins with tandem mass spectra," *Bioinformatics*, vol. 20, pp. 1466–1467, 2004.

[4] D. L. Tabb, C. G. Fernando, and M. C. Chambers, "Myrimatch: highly accurate tandem mass spectral peptide identification by multivariate hypergeometric analysis," *Journal of Proteome Research*, vol. 6, pp. 654–661, 2007.

[5] 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.

[6] N. Zhang, R. Aebersold, and B. Schwikowski, "ProbID: A probabilistic algorithm to identify peptides through sequence database searching using tandem mass spectral data," *Proteomics*, vol. 2, pp. 1406–1412, 2002.

[7] C. D. Wenger, D. H. Phanstiel, M. Lee, D. J. Bailey, and J. J. Coon, "COMPASS: a suite of pre-and post-search proteomics software tools for OMSSA," *Proteomics*, vol. 11, no. 6, pp. 1064–1074, 2011.

[8] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg, "Near-optimal sensor placements: Maximizing information while minimizing communication cost," in *Proceedings of the 5th international conference on Information processing in sensor networks*, pp. 2–10, ACM, 2006.

[9] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1, pp. 105–112, IEEE, 2001.

[10] P. Kohli, M. P. Kumar, and P. H. Torr, "P^3 & beyond: Move making algorithms for solving higher order functions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 9, pp. 1645–1656, 2009.

[11] S. Jegelka and J. Bilmes, "Submodularity beyond submodular energies: coupling edges in graph cuts," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 1897–1904, IEEE, 2011.

[12] H. Lin and J. Bilmes, "A class of submodular functions for document summarization," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 510–520, Association for Computational Linguistics, 2011.

[13] H. Lin and J. Bilmes, "Learning mixtures of submodular shells with application to document summarization," in *Uncertainty in Artificial Intelligence (UAI)*, (Catalina Island, USA), AUAI, July 2012.

[14] H. Lin and J. Bilmes, "Word alignment via submodular maximization over matroids," in *North American chapter of the Association for Computational Linguistics/Human Language Technology Conference (NAACL/HLT-2011)*, (Portland, OR), June 2011.

[15] J. G. Oxley, *Matroid theory*, vol. 3. Oxford University Press, USA, 2011.

[16] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functionsąli," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[17] M. Minoux, "Accelerated greedy algorithms for maximizing submodular set functions," in *Optimization Techniques*, 1978.

[18] B. Diament and W. S. Noble, "Faster SEQUEST searching for peptide identification from tandem mass spectra," *Journal of Proteome Research*, vol. 10, no. 9, pp. 3871–3879, 2011. PMC3166376.

[19] S. Kim and P. A. Pevzner, "MS-GF+ makes progress towards a universal database search tool for proteomics," *Nature Communications*, vol. 5, 2014.

[20] J. J. Howbert and W. S. Noble, "Computing exact p-values for a cross-correlation shotgun proteomics score function," *Molecular & Cellular Proteomics*, pp. mcp–O113, 2014.

[21] J. K. Eng, B. Fischer, J. Grossman, and M. J. MacCoss, "A fast SEQUEST cross correlation algorithm," *Journal of Proteome Research*, vol. 7, no. 10, pp. 4598–4602, 2008.

[22] P. Stobbe and A. Krause, "Efficient minimization of decomposable submodular functions," in *NIPS*, 2010.

[23] M. Conforti and G. Cornuejols, "Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the Rado-Edmonds theorem," *Discrete Applied Mathematics*, vol. 7, no. 3, pp. 251–274, 1984.

[24] 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.

[25] B. N. Pease, E. L. Huttlin, M. P. Jedrychowski, E. Talevich, J. Harmon, T. Dillman, N. Kannan, C. Doerig, R. Chakrabarti, S. P. Gygi, *et al.*, "Global analysis of protein expression and phosphorylation of three stages of plasmodium falciparum intraerythrocytic development," *Journal of Proteome Research*, vol. 12, no. 9, pp. 4028–4045, 2013.

[26] L. Wu, S. I. Candille, Y. Choi, D. Xie, L. Jiang, J. Li-Pook-Than, H. Tang, and M. Snyder, "Variation and genetic control of protein abundance in humans," *Nature*, vol. 499, no. 7456, pp. 79–82, 2013.

[27] C. Y. Park, A. A. Klammer, L. Käll, M. P. MacCoss, and W. S. Noble, "Rapid and accurate peptide identification from tandem mass spectra," *Journal of Proteome Research*, vol. 7, no. 7, pp. 3022–3027, 2008.

[28] J. E. Elias and S. P. Gygi, "Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry," *Nature Methods*, vol. 4, no. 3, pp. 207–214, 2007.

[29] U. Keich, A. Kertesz-Farkas, and W. S. Noble, "Improved false discovery rate estimation procedure for shotgun proteomics," *Journal of Proteome Research*, vol. 14, no. 8, pp. 3148–3161, 2015.

[30] J. D. Storey and R. Tibshirani, "Statistical significance for genome-wide studies," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, pp. 9440–9445, 2003.

[31] S. Tschiatschek, R. Iyer, H. Wei, and J. Bilmes, "Learning mixtures of submodular functions for image collection summarization," in *Neural Information Processing Society (NIPS)*, (Montreal, Canada), December 2014.

APPENDIX

A. FURTHER EXPLANATION OF THE CALIBRATION PROCEDURE

To understand why the calibration procedure in Equation 6 is helpful, consider the case when $f_1(E)$ and $f_2(E)$ are both high. The reason can be that this particular spectrum is simply more dense than others. On the other hand, if only one of the two values ($f_1(E)$ or $f_2(E)$) is higher than usual, this cannot be explained by the density of spectrum, we claim that this is real signal. Thus, when we expand $f(A)$, there are four terms

$$\lambda_1\lambda_2 f_1(A)+\bar{\lambda}_1\bar{\lambda}_2 f_2(A)+\lambda_1\bar{\lambda}_2\frac{f_1(E)}{f_2(E)}f_2(A)+\bar{\lambda}_1\lambda_2\frac{f_2(E)}{f_1(E)}f_1(A).$$

The first two terms are non-calibrated so we focus on the last two terms

$$\lambda_1\bar{\lambda}_2\frac{f_1(E)}{f_2(E)}f_2(A) + \bar{\lambda}_1\lambda_2\frac{f_2(E)}{f_1(E)}f_1(A)$$

These can be seen as a function of $\frac{f_2(E)}{f_1(E)}$. We know that a function like $ax+\frac{b}{x}$ will have a higher value if x is close to 0 or 1. So this is exactly what we need. We encourage that $f_1(E)$ and $f_2(E)$ are different, thus improving the calibration.

B. EMPIRICAL WEIGHTS

In this section, we show how we derive our empirical weights that lead to the improved performance demonstrated in Figure 5.

Figure 8 shows the average intensities of observed peaks near b-ions or y-ions in high confidence PSMs ($q = 0.01$) for the worm-01 charge +2 data set. Figure 9 shows the average intensities for *Plasmodium* TMT-10. In the figures, we see strong signals peaks at the b- and y-ions, as well as peaks with offsets of +1 Th peaks (+1 isotope peaks), -17 Th (NH_3-loss), -18 Th (H_2O-loss) and (for b-ions) -28 Th peaks (CO-loss).

The edges are added to E based on these average intensities for b-ion and y-ions separately. The weight of each edge is $w(\{v,u\}) = w'(\{v,u\})x(m_v)$, where m_v is the m/z of v, and x is the preprocessed observed spectrum as introduced in Section 2.1. Recall that in Section 3.1 $w'(\{v,u\})$ is determined by $m_v - m_u$. For low resolution data like yeast and worm, $w'(\{v,u\})$ is read from Table 3. For high resolution data, the $w'(\{v,u\})$ values are then read from Figure 9. In both settings, we calculate $m_v - m_u$ and use the number in the table or intensity in the plot.

For charge +3 data, where the b-ions and y-ions are charge +1 or charge +2, we do similar steps but using the mass-to-charge ratio of the higher charged ion, where $m_{v,c+} = \frac{m_{v,1+}+c-1}{c}$ is the m/z of charge $c+$ ion of v.

C. SELECTION OF SCORE FUNCTION PARAMETERS

Our submodular functions use the hyperparameters λ_1, λ_2 and $\{k_u\}_{u\in U}$. To select values for these hyperparameters, we performed an FDR-based evaluation with cases $k_u \in \{1,2,3,4,5\}$ when $\lambda_1 = \lambda_2 = 1.0$ on worm-01-ch2. The result (Figure C(a)) shows that $k_u = 2$ yields the best performance. Next, we tested cases $\lambda_1 \in \{0.4, 0.6, 0.8, 1.0\}$ on yeast-01-ch2, while fixing $\lambda_2 = 1.0$. We then selected $\lambda_1 = 0.6$ based on these results (Figure C(b)). Next, we tested $\lambda_2 \in \{0.4, 0.6, 0.8, 1.0\}$ while fixing $\lambda_1 = 0.6$. The value $\lambda_2 = 0.8$ had the best performance (Figure C(c)). For α, we calculate the average score of SGM,

$$\alpha_1 = \frac{\sum_{s\in S}\sum_{p\in P_s\cup D_s}\mathfrak{s}(p,s)}{\sum_{s\in S}|P_s\cup D_s|}$$

and the average foreground score of SEQUEST,

$$\alpha_2 = \frac{\sum_{s\in S}\sum_{p\in P_s\cup D_s}\langle p,s\rangle}{\sum_{s\in S}|P_s\cup D_s|},$$

where S is the set of all observed spectra and P_s and D_s are corresponding target and decoy sets of $s \in S$. The value α is then chosen to be $\frac{\alpha_2}{\alpha_1}$. Although the derivation of these parameters was empirical in our study, we hope in future work to develop strategies that can learn these automatically.

Table 3: The empirical weights $w(\{e_v, e_u\})$ used for low resolution data (yeast and worm). For each v and u, we compute the mass-to-charge ratio difference $m_v - m_u$ and read the correspond entry from the table.

b-ion						
$m_v - m_u$	-28	-27	-19	-18	-17	-16
$w'(\{v,u\})$	0.1101	0.0225	0.0121	0.3128	0.2364	0.0784
$m_v - m_u$	-15	-12	-1	0	+1	+2
$w'(\{v,u\})$	0.0112	0.0107	0.0481	0.6122	0.2514	0.0511
y-ion						
$m_v - m_u$	-18	-17	-16	0	+1	+2
$w'(\{v,u\})$	0.1364	0.1179	0.0345	1	0.4253	0.0741

D. HYPERPARAMETER GENERALIZATION

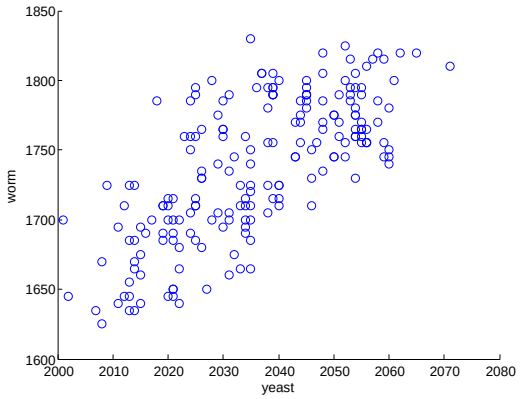
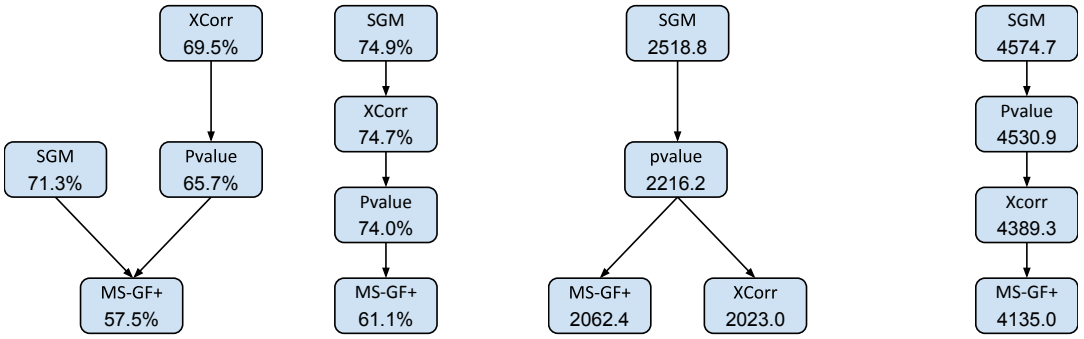


Figure 11: The plot shows the number of high-confidence PSMs ($q \leq 0.01$) obtained by SGM on the yeast data (x-axis) versus the worm data set (y-axis). Every point represents the performance of one combination of parameters.

Our submodular function $f(A)$ contains three hyperparameters. Ideally, these hyperparameters generalize across the different datasets. To test this, we tried all combinations of hyperparameters on one run from the yeast and worm data sets (yeast-01 and worm-01). The results (Figure 11) show that the parameters that work well on one dataset tend also to work well on the other dataset, implying that the parameters generalize well across datasets.



(A) *Plasmodium* TMP (B) human TMP (C) *Plasmodium* accepted PSMs (D) human accepted PSMs

Figure 6: Statistical comparison of methods. Each panel plots, for a single data set, the comparison between four methods in terms of the target match percentage or the number of targets PSMs accepted at $q < 0.01$. A directed edge from A to B means that method A's mean score is significantly larger ($p < 0.05$) than method B's mean score, according to a Wilcoxon signed-rank test. The numbers in the nodes are mean values.

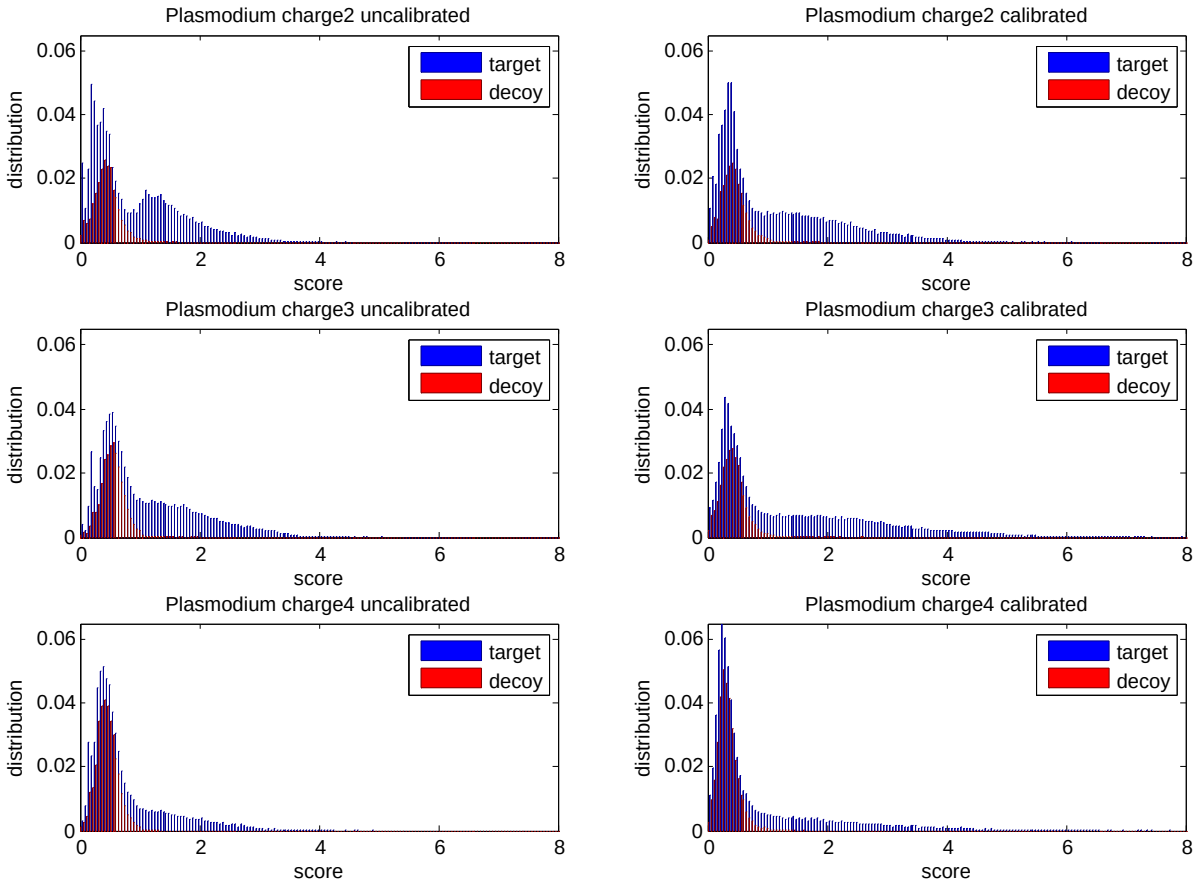


Figure 7: Score calibration of SGM. Each panel plots, for a single charge state, the SGM score distribution of top-scoring PSMs, separated into target and decoy distributions. Panels on the left are uncalibrated scores, and panels on the right are calibrated. In each plot, the x-axis is normalized so that the score threshold at $q = 0.01$ equals 1.

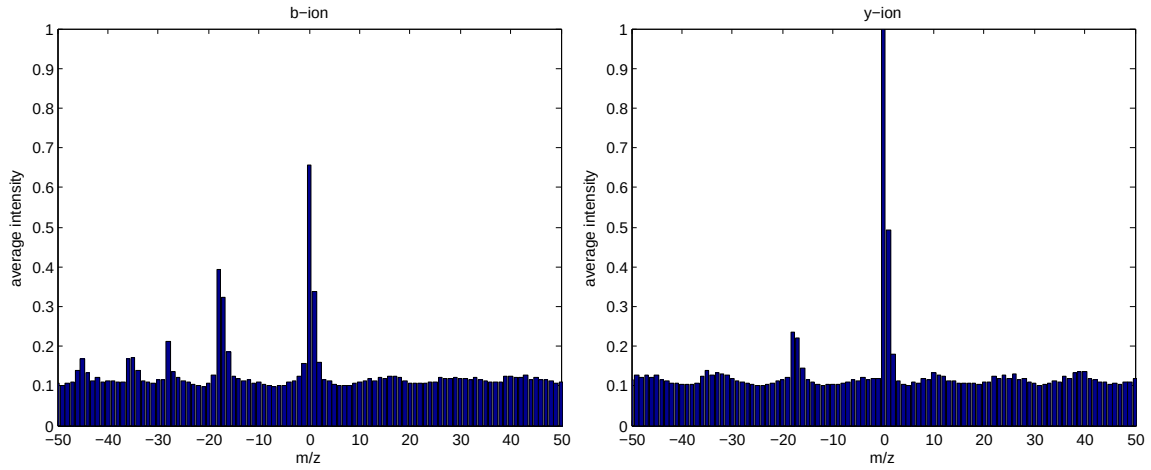


Figure 8: Average intensity near b-ion and y-ion peaks from high-confidence ($q < 0.01$) PSMs in the worm-01 dataset. We see strong signals at m/z=0 (central peak), m/z=+1 (+1 isotope), m/z=-17 (NH_3 loss), m/z=-18 (H_2O loss) and m/z=-28 (CO loss for b-ion only).

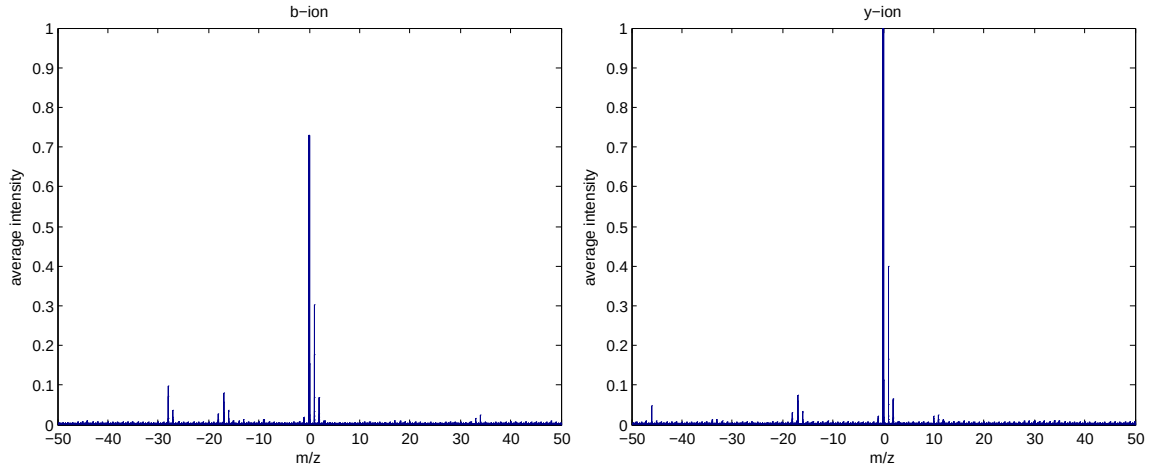
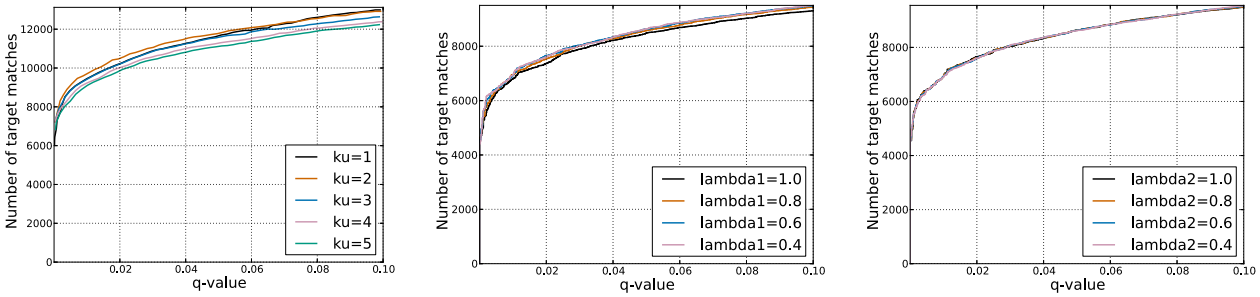


Figure 9: Average intensity near b-ion and y-ion from high-confidence ($q < 0.01$) PSMs in the *Plasmodium* TMT-10 dataset. We see strong signals at m/z=0 (central peak), m/z=+1 (+1 isotope), m/z=-17 (NH_3 loss), m/z=-18 (H_2O loss) and m/z=-28 (CO loss for b-ion only). The intensities in this figure are also used for the empirical weights $w(\{e_v, e_u\})$ for high resolution data for both *Plasmodium* and human. For each v and v, we compute the mass-to-charge ratio difference $m_v - m_u$ and then use the corresponding peak intensity.



k_u λ_1 λ_2

Figure 10: FDR-based evaluation of SGM using different hyperparameters.