Supplemental Material - A Ranking-Based Scoring Function For Peptide-Spectrum Matches

Ari M. Frank

1 A Discriminative Scoring Model For Peptide-Spectrum Matches

The novel PSM scoring function we present is more data-driven than previous approaches. Our goal is not to create an accurate *generative* model $Prob(S|P)$, which is inarguably a difficult task. Instead, we desire a scoring algorithm that performs well on a simpler *discriminative* ranking task: Given a spectrum S and a set of candidate peptides $P_1, \ldots, P_k$, we want the model to be able to assign scores to the peptides according to how well their expected fragmentation pattern matches the observed spectrum S.

We use the RankBoost [9] machine learning algorithm to train our models (see main manuscript). The most important component of our scoring models are the feature functions they use. Our models draw on a diverse set of features, created using domain knowledge, that each in there own way reflect different characteristics that can help distinguish between correct and incorrect PSMs. In total, the models can contain up to 225 features (though not all get selected in each model). We grouped these features into different classes, as described below. For each feature class, we give examples of the most prominent features (the ones that most influence the ranking score). For many of the features described below we also provide a *cumulative weak learner score plot*, which we create as follows. Each feature function f_i can be selected by the algorithm in multiple rounds $t_1, t_2, \ldots, t_k$, to create weak learners $h_{t_1}, h_{t_2}, \ldots, h_{t_k}$, to which the algorithm assign the weights $\alpha_{t_1}, \alpha_{t_2}, \ldots, \alpha_{t_k}$, respectively. To get a clearer picture of how f_i is used in the model we condense these k functions into a single cumulative function $h_i^*(x) = \sum_{i=1}^{k} \alpha_{t_i} h_{t_i}(x)$. The cumulative score plot for a feature f_i is a graphical representation of this function $h_i^*(x)$.

The values for the cumulative score plots depicted below were taken from model trained for scoring de novo sequences of doubly-charged peptides with parent mass 1100-1300 Da (about 9-15 amino acids). Though the features listed below are explained in the context of de novo sequencing, we also use them in models for other scoring tasks besides reranking de novo results, such as scoring tags, and scoring database search results.

Peak Rank Prediction Features

The peak rank prediction features examine how well the peaks in the observed spectrum fit the ranking of a peptide's fragment ions, as predicted by the algorithm we created to solve the peak rank prediction problem (described in Figure 1). For more details on our ranking-based solution to this problem, see the accompanying manuscript [5]. Since our prediction of peak ranks is most accurate for the highest ranked fragments, the features described below mostly focus on these ranks.

Some of the most useful peak rank prediction features we use are:

- **Observed rank for peak with predicted rank X, ($\mathbf{X = 1, \ldots, 7}$) [2a,2b]** - This type of feature examines the difference between the ranks observed for peaks in the spectrum and the

Peak rank prediction problem

Input:

- Peptide sequence $P = p_1 p_2 \dots p_n$, where p_i, $1 \leq i \leq n$, are amino acids.
- Set of fragment ion types $\mathcal{F}$, e.g., $\mathcal{F} = \{b, y, a, y - H_2O, b^{+2}, \dots\}$.

Output:

- A permutation π of the set of all possible fragment peaks ($\mathcal{F} \times \{1, \dots, n\}$), where π is ordered according to decreasing intensity (e.g., $\pi = y_8, y_6, b_3, b_4, b_3 - H_2O, \dots$).

Figure 1: The peak rank prediction problem.

ranks we predicted for them with our ranking-based model [5]. For each predicted fragment peak with rank X, the feature function reports the actual rank observed for that fragment ion's peak in the experimental mass spectrum (a rank of ∞ is given if the peak is not observed in the spectrum). Figures 2*a* and *b* depict the scores assigned to the features that examine the peaks with predicted ranks 1 and 3, respectively. The features give a premium if the observed rank is close to the feature's predicted rank. This premium decreases as the observed rank gets farther from the predicted one. In both cases if the observed rank is above 12, it is treated the same as case of a predicted peak being unobserved in the experimental spectrum.

- **Predicted rank of peak with observed rank X, ($\mathbf{X = 1, \dots, 7}$) [2c,2d]** - This type of feature uses peak rank predictions the other way around, and examines what is the rank that was predicted by the model for the peak observed in the spectrum with rank X. Figures 2*c* and *d* depict the scores assigned to the features that examine the peaks with observed ranks 1 and 3, respectively.

- **Rank of missing peak #X, ($\mathbf{X = 1, \dots, 10}$) [2e,2f]** - This feature examines the theoretical masses of fragment peaks, according to the order of their predicted ranks (starting with the peak predicted to have the highest intensity). The feature notes the rank of the X'th missing peak (i.e., there was no peak detected in the spectrum at the expected mass). Figures 2*e* and *f* depict the features that examine the first and third missing ranks, respectively. The models assign penalties when the ranks of the missing peaks are high (since this indicates a poor fit between the predicted ranks and the observed spectrum).

- **Sum of ranks of missing peaks 1-5,6-10 [2g,2h]** - This type of feature is more general than looking at each rank X individually, since it carries information on the occurrence of multiple missing peaks (which is a strong indication that the peptide is incorrect). Figures 2*g* and *h* depict the features that examine the sum of missing ranks 1-5 and 6-10, respectively.

Spectrum Graph Features

The space of all peptides is extremely large, making it inappropriate for an exhaustive case-by-case analysis. Nonetheless, most de novo algorithms are able to consider all likely peptides by modeling the search space for a query spectrum as a spectrum graph [1, 3]. A *spectrum graph* is a directed acyclic graph; it's vertices correspond to putative prefix masses (cleavage sites) of the peptide. Two vertices are connected by a directed edge from the vertex with the lower mass to the one with a

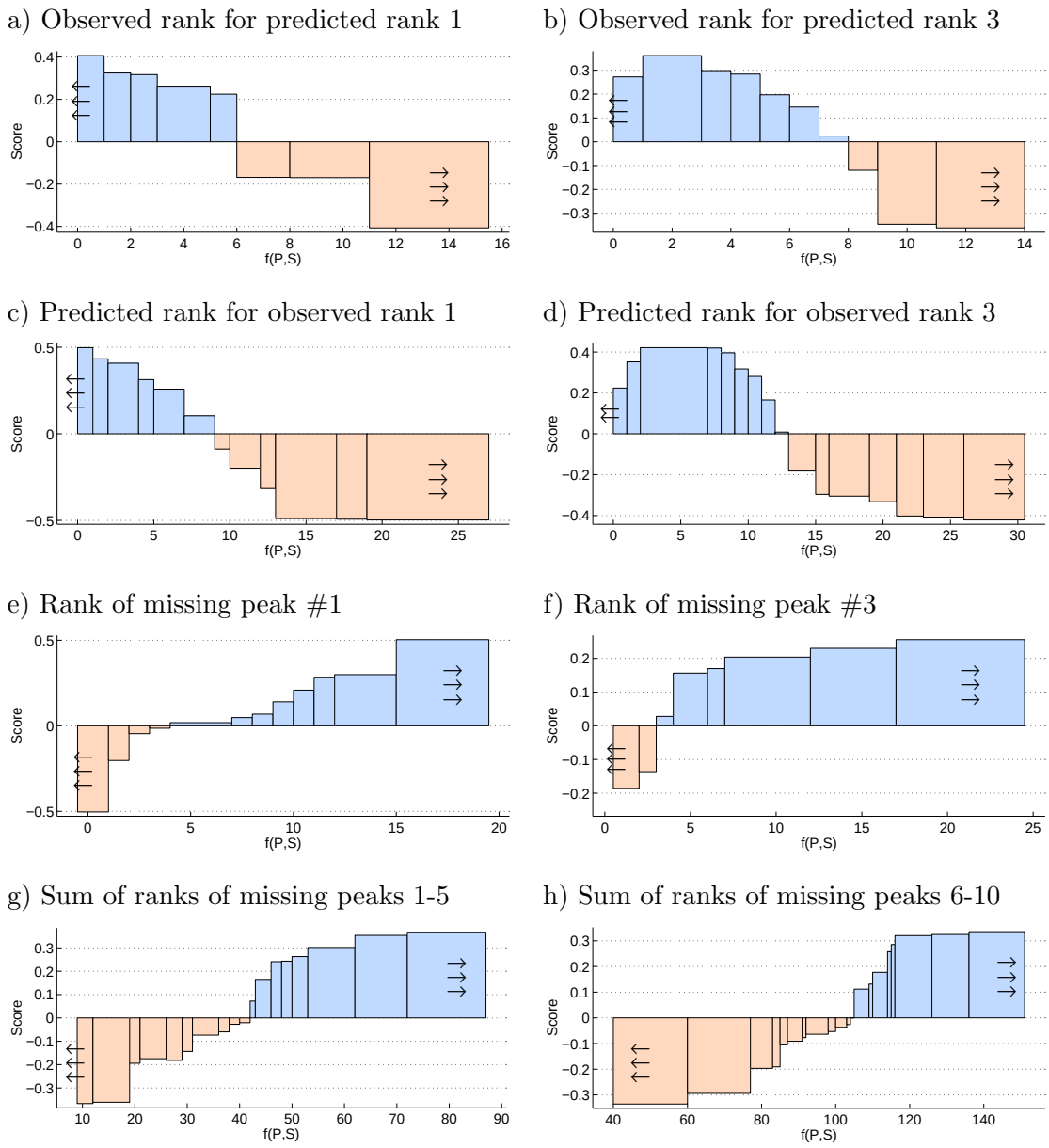

Figure 2: Cumulative weak learner score plots for peak rank prediction features. The x-axis holds feature function values $f(P, S)$ that are computed by the feature function f when matching a peptide P to the spectrum S. The y-axis holds the cumulative score assigned by the ranking model to different values of $f(P, S)$. Arrows on the left (or right) side of the plot indicate that the same score should be given to all feature values smaller (or greater) than that position on the x-axis.

higher mass if the difference between them equals the mass of an amino acid. We use PepNovo's scoring function to score nodes in the graph [6, 7]. It is based on detailed probabilistic models, and considers important factors such as dependencies between fragment ions, the observed peak intensities, the influence of flanking amino acids, and the location of the cleavage site in the peptide. The score assigned to a peptide's path in the spectrum graph is indicative of how likely it is that the observed spectrum was created from the fragmentation of the given peptide. An incorrect PSM is likely to contain more cleavage sites that are either missing detected fragments altogether, or

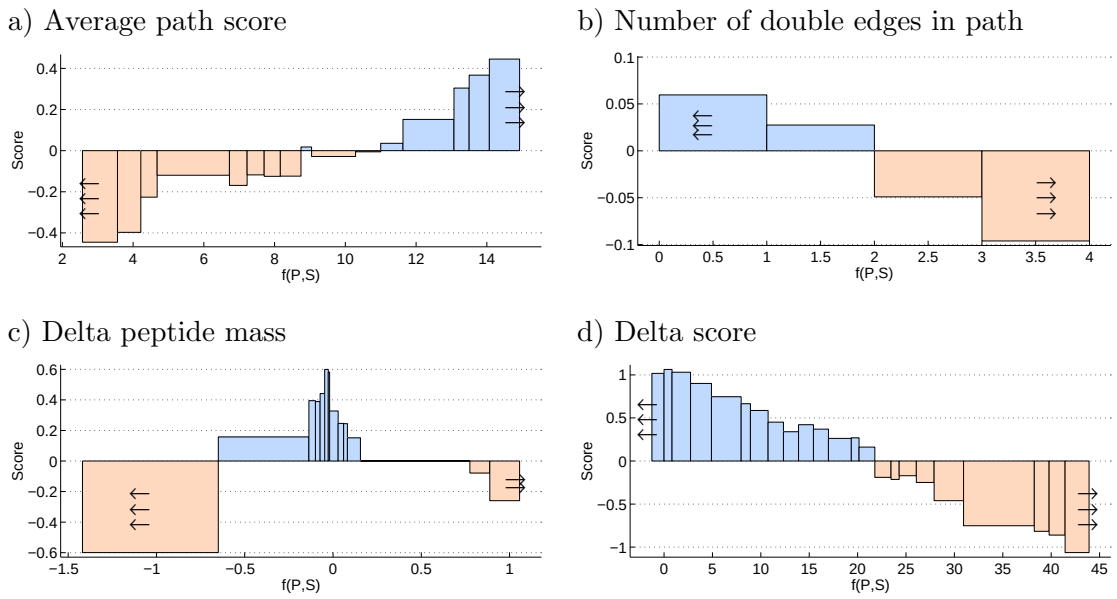

Figure 3: Cumulative weak learner score plots for spectrum graph features. The x-axis holds feature function values $f(P, S)$ that are computed by the feature function f when matching a peptide P to a spectrum S. The y-axis holds the cumulative score assigned by the ranking model to different values of $f(P, S)$. Arrows on the left (or right) side of the plot indicate that the same score should be given to all feature values smaller (or greater) than that position on the x-axis.

have combinations of observed ions that are less likely, and are thus scored poorly in the spectrum graph.

To capture this type of information, we examine several aspects of the spectrum graph scores that can be informative. Since we use the same model to compare peptides that can have different lengths, we cannot use a fixed set of features that is length dependant (e.g., score at cleavage 1, score at cleavage 2, etc.), rather we use features that are invariant to length such as the total score, or average cleavage score, etc. Below is a list of the most prominent spectrum graph-based features used in the models:

- **Total and average path score [3a]** - The score of a peptide's path in the spectrum graph is computed using a likelihood ratio score [6]. On average we can expect the path of the correct PSM to be higher than the path of an incorrect PSM. To avoid biases that are due to the predicted peptide's length, such as when comparing a partial peptide prediction to a full one, it is also beneficial to look at the average path score (total score divided by the number of amino acids in the predicted peptide). Figure 3a depicts the plot of the average path score feature function. It shows a mostly monotonically increasing reward for having a high average path score.

- **Minimal cleavage scores** - Usually the top scoring de novo sequences are quite similar to the correct sequence, but make suboptimal short "detours" in the spectrum graph. In such cases they are likely to score lower at certain cleavage sites. An informative feature can be to look at the minimal (and second and third lowest) scores assigned to cleavage sites in the peptide's path.

- **Number of double edges used in the path [3b]** - Most peptide bond cleavages produce fragment ions that are detected as peaks in the spectrum. However, there are often cases

where a bond's cleavage does not result in detectable peaks, which could lead to incomplete paths in the graph. To address this problem, our graphs typically allow the use of double edges. However, excessive use of double edges in a peptide's path usually indicates that the path belongs to an incorrect peptide. In Figure $3b$ we see the cumulative score plot for the function that reports the number of double edges used in the peptide's path. Having no double edges receives a premium of +0.05 to the rank score, while having 3 or more double edges reduces the score by 0.1.

- **Number of forbidden node pairs** - Forbidden node pairs occur when a single peak is assigned to more than one fragment (e.g., it is considered to be both a b_8 and a y_4). This phenomenon is especially common in de novo sequencing, where is leads to the formation of incorrect "symmetric" paths. If one or more such a cases are detected, the score for the PSM receives a penalty of -0.5.

- **Delta of peptide mass [3c]** - The spectrum graph is constructed while allowing a certain error tolerance for peak masses (typically we used 0.5 Da.). Such mass errors can accumulate as we traverse along the peptide's path. However with correct peptides, the typical difference between the sum of the mass of the peptide's amino acids and the mass of the path is not great (the mass of a path is defined as the mass of the last node minus the mass of the first node). Figure $3c$ shows that while having a delta mass near 0 yields a premium of +0.6, having a negative delta mass beyond 0.65 Da is not common with correct PSMs, and incurs a large penalty of -0.6

- **Delta rank** - When scoring de novo sequences, we are given a list candidate peptides that can be ranked according to their paths' scores. Since often the highest scoring de novo paths belong to correct peptides, knowing the path ranks is also helpful.

- **Delta Score [3d]** - A de novo search often results in many high-scoring, but similar, spectrum graph paths, that differ from each other by only one or two amino acids. In such cases, the correct peptide might have a relatively low rank, but its score will not be much lower than score of the highest ranked peptide. It is therefore useful to have a feature that relies on the difference in score, rather than the difference in rank (as does the "Delta rank" feature mentioned above). Figure $3d$ shows that being close to the optimal score is a characteristic of many correct PSMs. There is a premium of approximately +1 when the path score is up to 3 away from the optimum, which monotonically decreases, and turns into a penalty once the score difference exceeds 21.

The relatively high weight assigned to the "Delta Score" feature indicates the importance of the original ranking of the de novo results (according to PepNovo's score). In essence, PepNovo's output is ordered solely according to this feature. All the other features described in this section serve to refine the ordering induced by this feature, and increase the number of cases in which correct lower-scoring peptides are ranked above incorrect higher-scoring ones.

Peak annotation features

The spectrum graph scores evaluate combinations of fragments that involve specific cleavage sites. It is also beneficial to take a global look at how well the peptide explains the spectrum's peaks, like in the case of the aforementioned peak rank prediction features. With the peak annotation features, we look at more basic statistics that examine the quality of PSMs using functions that

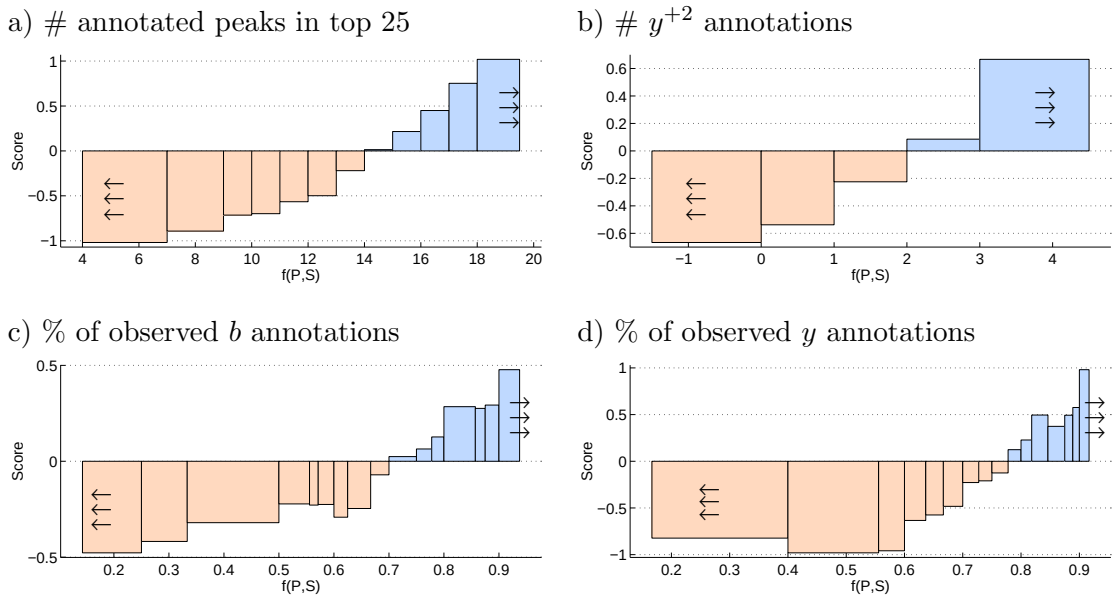

Figure 4: Cumulative weak learner score plots for peak annotation features. The x-axis holds feature function values $f(P, S)$ that are computed by the feature function f when matching a peptide P to a spectrum S. The y-axis holds the cumulative score assigned by the ranking model to different values of $f(P, S)$. Arrows on the left (or right) side of the plot indicate that the same score should be given to all feature values smaller (or greater) than that position on the x-axis.

simply count a peptide's matched peaks. This type of information is not easily conveyed when additive scoring functions are used.

The most useful peak annotation features are:

- **# Annotated peaks in top 25,50 peaks [4a]** - A correct peptide should typically explain many of the strongest peaks in the spectrum. Figure 4*a* depicts the cumulative scores assigned by the model to this feature. A good match tends to explain a large proportion of the top 25 peaks.

- **% Explained intensity** - This feature measures how much of the spectrum's total peak intensity can be explained by the peptide's fragment ions. Generally, we expect a good match to explain a large proportion of the experimental spectrum's intensity.

- **# of peak annotations for fragment $\mathbf{X} = \mathbf{b}, \mathbf{y}, \mathbf{a}, \mathbf{y^{+2}}, \mathbf{y - H_2O}, \ldots$ [4b]** - Correct peptides are likely to explain many types of fragments. Since the spectrum graph score looks at individual cleavage sites, it cannot detect events that are probable for any single cleavage site, but less probable for a whole peptide. For example, even though with doubly charged tryptic peptides, the probability of observing a y^{+2}-ion at any given cleavage is less than 50%, it is quite unlikely not to detect any y^{+2}-ions at all. Figure 4*b* shows that such cases are penalized by subtracting 0.65 from their scores. However, peptides for which we find 3 or more y^{+2}-ion peaks receive a large premium.

- **% of peak annotations for fragment $\mathbf{X} = \mathbf{b}, \mathbf{y}, \mathbf{a}, \mathbf{y^{+2}}, \mathbf{y - H_2O}, \ldots$ [4c,4d]** - This feature is similar to the feature above, however gives values that are normalized according to the peptide's length. Figures 4*c* and *d* depict the scores given to the features measuring the proportion of annotated b- and y-ions, respectively.

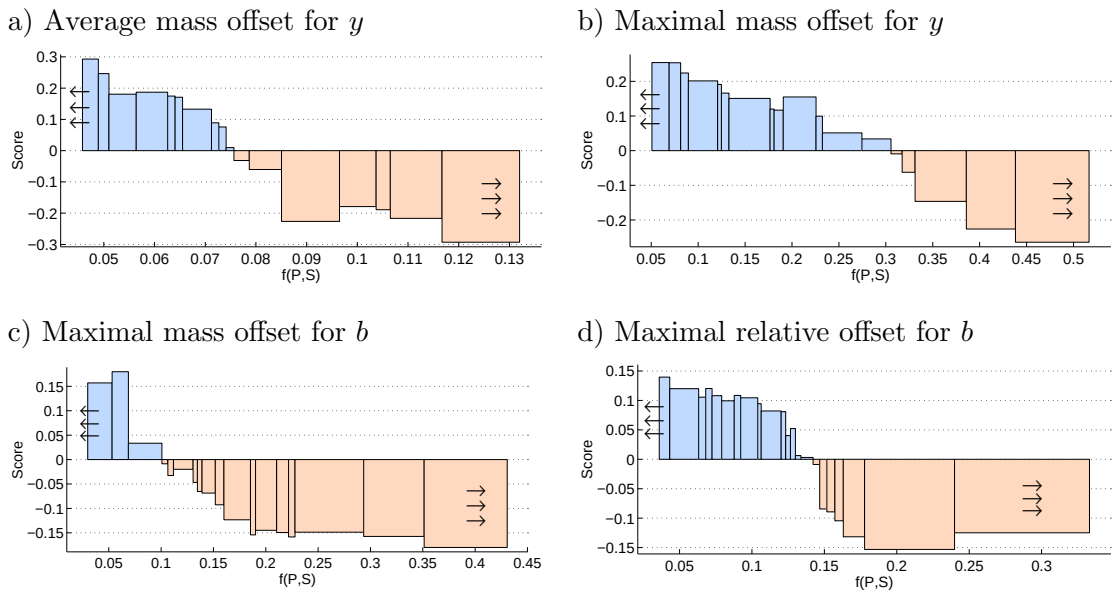

Figure 5: Cumulative weak learner score plots for peak offset features.The x-axis holds feature function values $f(P, S)$ that are computed by the feature function f when matching a peptide P to a spectrum S. The y-axis holds the cumulative score assigned by the ranking model to different values of $f(P, S)$. Arrows on the left (or right) side of the plot indicate that the same score should be given to all feature values smaller (or greater) than that position on the x-axis.

Peak offset features

When annotating fragment ions, we generally tolerate a mass differences of up to 0.5 Da between the expected mass of a fragment, as computed from the peptide sequence, and the actual mass observed in the spectrum. However, most of the true fragment peaks observed in spectra are much closer to their expected mass, usually being less than 0.1 Da away. A peptide that has many fragment peaks with a relatively large offset from their expected mass is likely to be relying on spurious opportunistic peak matches, and is therefore more likely to be incorrect. This type of peak offset information is most useful with the most abundant fragment ions, which are b, y, so offset related features focus only on them. Following are the peak offset features we use:

- **Average mass offset for fragment b/y [5a]** - This feature looks at the average mass offset of all identified b (or y) peaks. Figure 5a depicts the scores assigned by the model to the average offset measured for the peptide's y-ion fragments. Typical correct PSMs have an average peak offset of less than 0.085 Da; larger offsets are penalized.

- **Maximal mass offset for fragment b/y [5b,5c]** - Often a bad PSM contains an opportunistic use of a single peak (this is especially true in de novo sequencing). Often these peaks are not close to the expected mass. Looking at the maximal offset observed for a fragment, rather than the average, can be more discriminating in these cases. Figures 5b and c depict the scores given to the maximal offset features for y- and b-ions, respectively.

- **Maximal relative offset for fragments b/y [5d]** - Sometimes spectra contain systematic biases in the peak locations (e.g., there is a fixed offset to most of the peak masses or an offset that increases with peak mass). In such cases the absolute peak offset might be relatively high, but we still can detect good peak matches by examining the mass of *successive* fragment

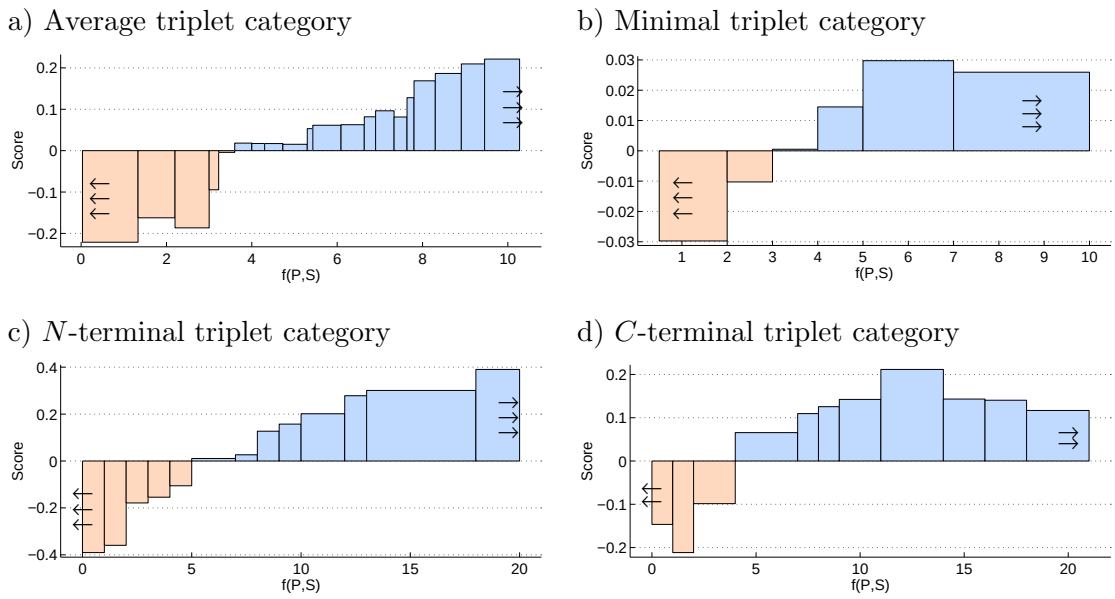

Figure 6: Cumulative weak learner score plots for sequence composition features. The x-axis holds feature function values $f(P, S)$ that are computed by the feature function f when examining a peptide P (the spectrum S does not play an active role in the sequence composition features). The y-axis holds the cumulative score assigned by the ranking model to different values of $f(P, S)$. Triplets of amino acids were assigned to categories according to their frequency in proteotypic peptides; ranging from 1, for the most rare triplets, to 20 for the most frequent. Arrows on the left (or right) side of the plot indicate that the same score should be given to all feature values smaller (or greater) than that position on the x-axis.

ions. For example the offset of two successive b separated by amino acid A is computed as $b_n - b_{n-1} - mass(A)$. Figure 5d depicts the score assigned by the model to the feature examining the maximal relative offset of b-ions.

Sequence Composition Features

Proteins are not random sequences of amino acids. They often contain conserved, or characteristic patterns that are responsible for inducing a specific spacial conformation or for providing certain function. In addition, certain amino acid patterns are more likely to be ionized and detected using MS/MS than others (e.g., basic amino acids are usually required for effective peptide ionization). These observations gave rise to the notion of proteotypic peptides [2, 12, 13], peptides that are most likely to be confidently identified by MS/MS methods. Maintaining a list of proteotypic peptides is of course not suitable for an unrestricted general-purpose scoring function. However, many of the characteristics of proteotypic peptides can be captured using simple features that pertain to the peptide's amino acid composition.

We focused our efforts on amino acid triplets. These are relatively short sequences, and thus could not be trained to fit specific peptide sequences. We examined a large set of proteotypic peptide sequences [2], and computed frequency statistics for all possible amino acid triplets. We then divided the triplets into 20 categories according to their frequency. Category 1 contains the least frequent amino acid triplets (e.g., WKW, KCR, YRM), and category 20 contained the most frequent triplets (e.g., GGG, ELL, ALA). Similar tables were constructed for the first triplet (the first three amino acids on the N-terminal side) and the last triplet (the last three amino acids on

the C-terminal sides), which can have different frequencies due to the specificities of the enzymatic digestion. The features that used the information of these tables, which we included in our models are:

- **Average/minimal triplet category [6a,6b]** - For a peptide of length n, there are $n-2$ triplets for which we compute the average and minimum triplet category values. Figures 6a,6b give examples of the cumulative scores assigned by the models to these feature functions.
- **Category of the amino acid triplets on the N-/C-terminal sides [6c,6d]** - Figures 6c,6d give examples of the cumulative scores assigned by the models to these two feature functions.

We also examined the composition by creating features of the type

$$f_{\#X}(P) = \#\text{ of amino acids X in the peptide P.}$$

Using such simple features helped correct biases in the de novo sequencing scoring. For instance, if a peptide had one glutamine there was a score penalty of -0.4 and if there were two or more glutamines, there was a score penalty of -0.7. The reason this amino acid received these penalties is that glutamine has the same mass as alanine + glycine, and it often wrongfully replaces these two amino acids in the de novo sequencing results. Likewise, if the peptide contains tryptophan (W), there is a penalty -0.31. W has the same mass as A+D,E+G, and V+S, so it too is likely to cause sequencing mistakes.

Summary

Our rank models can contain most of the 225 scoring features, but not all feature are necessarily used. One of the benefits of the RankBoost algorithms is that it only incorporates into its models features that are useful (i.e., they help it make fewer ordering mistakes on the training data). RankBoost also has a high tolerance to uninformative features; we can supply it with a large pool of candidate features that can be correlated, and some might not even be relevant to the objective we are trying to achieve, nevertheless, in the training process, the algorithm automatically performs its own "feature selection", and incorporates only features that advance the goal [9]. This property makes it easy for us to design a single set of feature functions ("one size fits all") that incorporates all features that *might* be useful for peptide identification, without needing to consider wether the features are important for a specific model. For example, a feature that looks at the number of y^{+2} annotations might be very important for a model for scoring large triply charged peptides, but only represent noise in a model that scores singly charged peptides. With RankBoost we do not have to evaluate each model and decide which of the possible features are relevant for it, the algorithm does that automatically for us.

Of the features that get included in a model, not all functions carry the same weight. As depicted in Figures 3-6, some feature functions are assigned high scores, even reaching ± 1, while other feature functions are assigned much lower scores. All features are important for optimal ranking (otherwise they would not have been incorporated into the model by the learning algorithm). It is true that a small set of features that posses high scores can perform most of the coarse ranking process; moving the good PSMs up in the ranks and the bad PSMs down. However, for close calls, such as correctly ranking very similar peptides obtained by de novo sequencing, the models rely on the many other features that have small score values (e.g., peak offset features, composition features, etc.), to perform the fine tuning and give the correct PSM a slightly higher score, which is sufficient to push it ahead to the top of the PSM list.

Table 1: Database search results for the new ISB Standard Protein Mix (18 Proteins) [11]. The table holds results of 4 experiments in which the protein mixture was generated and processed. The 4 MS/MS datasets were generated using a Thermo Electron LTQ mass spectrometer. All peptides were identified with a 2.5% FDR. (*) results taken from ref. [11]

	#Peptides identified			
Search Strategy	Mix 1	Mix 2	Mix 3	Mix 4
SEQUEST + Peptide Prophet*	738	1033	646	468
InsPecT	891	1313	687	618
SEQUEST + Rescoring	821	1149	651	568
InsPect + Rescoring	883	1341	697	629

2 Additional Benchmark Experiments

This supplemental material contains additional benchmark results that were not included in the published manuscript. We first describe experiments with a standard protein mixture that show that the search tool InsPecT outperforms SEQUEST. We also describe experiments with peptide sequence tag generation in which we demonstrate that using our ranking-based score improves on previous results.

2.1 Benchmark Experiments With Standard Protein Mix

We start off the benchmark experiments with a small standard dataset that has recently been released, the new ISB 18 protein mixture [11]. These experiments were conducted to demonstrate that using our scoring function along with InsPecT does not reduce the number of peptide identifications that can be made by standard approaches like SEQUEST. From the ISB data we took four sets of MS/MS spectra of the protein mixtures that were acquired on a Thermo Electron LTQ mass spectrometer, along with the search results from running SEQUEST [4] followed by Protein Prophet [10]. We also searched the MS/MS spectra of each of the mixtures with InsPecT [14]. We rescored the raw results obtained from the SEQUEST and InsPecT searches with our ranking-based scoring model. This gives four different search strategies to compare. The results for each search were post-processed to maintain a 2.5% peptide false discovery rate, using a decoy database of protein sequences of *H. influenzae* [11].

1 summarizes the results of the four search strategies on four mixture datasets. InsPecT performs better than SEQUEST with this data, identifying between 6% to 32% more peptides in each mixture than SEQUEST. Due to the small size of the decoy database ($\sim$0.5 million amino acids), our rescoring function does not significantly improve over InsPecT's results (though it does improve over the results obtained by SEQUEST). Below we demonstrate that with more challenging search domains, it is necessary to have a strong scoring function for optimal results.

2.2 Benchmark Results For Tag Generation

Our scoring function is not restricted to ranking long de novo sequences. It can also be used to rerank lists of tags, and as we show below, can be quite useful in creating covering sets of tags. Since the characteristics of short sequence tags are much different than longer de novo sequences,

Table 2: Benchmark results for tag generation. The table compares the sets of tags generated using PepNovo and ranking with tags generated without ranking (the "LocalTag+" algorithm [8]). Each algorithm generated sets of 1-500 tags of length 3-6 amino acids. The test set consisted of 685 spectra of double-charged peptides from the ISB dataset [8].

Algorithm (tag length)	Number of tags								
	1	3	5	10	25	50	100	250	500
LocalTag+ (3)	0.752	0.828	0.853	0.893	0.927	0.945	0.959	0.965	0.974
PepNovo + ranking (3)	0.772	0.886	0.909	0.933	0.949	0.962	0.968	0.985	0.985
LocalTag+ (4)	0.676	0.772	0.804	0.844	0.891	0.914	0.930	0.950	0.959
PepNovo +ranking (4)	0.728	0.850	0.872	0.892	0.915	0.940	0.949	0.956	0.964
LocalTag+ (5)	0.578	0.670	0.707	0.782	0.844	0.866	0.879	0.915	0.930
PepNovo+ranking (5)	0.663	0.793	0.828	0.850	0.880	0.893	0.908	0.927	0.940
LocalTag+ (6)	0.502	0.603	0.657	0.724	0.784	0.806	0.828	0.850	0.872
PepNovo+ranking (6)	0.587	0.720	0.750	0.803	0.840	0.872	0.880	0.893	0.902

we created special ranking model for each specific tag length from 3 to 6 amino acids. We generated tags in similar fashion to the LocalTag+ method [8]. To generate x tags of a given length, we first ran PepNovo and extracted $4x - 6x$ tags. A small number of these tags came from parsing the globally-optimal long de novo sequences, while the majority were locally-optimal short tags directly extracted from the spectrum graph. We then used the ranking models to score and rerank the lists of tags, and returned the updated list of the x highest-scoring tags.

2 holds results of benchmark experiments in which we compared the performance of our new tagging with the LocalTag+ algorithm. The ranking procedure shows a clear superiority for all lengths examined, though for the shorter lengths the advantage diminishes somewhat when we look at large sets of tags. The table also shows that if one is concerned about the tagging efficiency, using larger tags can be quite advantageous in reducing the number of database hits. For instance, a tag of length 6 is about 400 times more efficient for filtration than a tag of length 4. However, using 250 tags of length 6 gives an 89.3% chance that the predicted set of tags contains a correct sequence, while using a single tag of length 4, which is less efficient as a filter, has only a 73% chance of being correct. The problem with relying solely on long tags is that many peptides have poor fragmentation patterns. In these cases, the spectrum graph often does not contain a correct tag of the desired length, or the correct path has such a poor score, that it does not get into the initial set of tags. This is evident in the table where the results for tags of length 3 and 4 reach 96%-98% while the tags of length 6 reach only 90%.

We found that in order to make tags both efficient and accurate we should use a mixture of tags. For instance, by default, InsPecT generates 25 tags of length 3 for each query spectrum. Since increasing the tag length by one amino acid gives about 20 times higher filtration efficiency, using 100 tags of length 5 would make the database filtration about 100 times more efficient. According to the table, this gives correct results in 88% of the test cases. However, we can get superior results if we select tags with several lengths. For instance, if we use a mixture of tags 3 of length 4, 35 of length 5, and 100 of length 6, these too increases the filtration efficiency by 100-fold, but generate a correct set of tags in 93.1% of the cases. Note that when we select this mixed set of tags, we need to eliminate redundancies that arise when we use a long tag that is "covered" by a shorter one.

LocalTag+ has an advantage over the new ranking-based method when it comes to running time. Due to its simpler scoring method, LocalTag+ needs 0.05-0.1 seconds to generate tags, while tags obtained by ranking take about 10 times longer. This means that for simple searches (e.g., small databases or searches that do not consider post-translational modifications), using LocalTag+ will probably give the fastest overall running time. However, when the time required for tag generation is dwarfed by the database scanning time, such as when performing blind searches [15] or searches involving large genomes, it can be quite beneficial to use the longer tags generated by PepNovo with ranking.

References

[1] Bartels, C. (1990). Fast algorithm for peptide sequencing by mass spectroscopy. *Biomedical and Environmental Mass Spectrometry 19*(6), 363–8.

[2] Craig, R., J. Cortens, and R. Beavis (2005). The use of proteotypic peptide libraries for protein identification. *Rapid Commun. Mass Spectrom 19*, 1844–1850.

[3] Dancík, V., T. Addona, K. Clauser, J. Vath, and P. Pevzner (1999). De novo peptide sequencing via tandem mass spectrometry. *J. Comput. Biol. 6*, 327–342.

[4] Eng, J., A. McCormack, and J. Yates, III (1994). An Approach to Correlate Tandem Mass-Spectral Data of Peptides with Amino Acid Sequences in a Protein Database. *J. Am. Soc. Mass Spectrom. 5*, 976–989.

[5] Frank, A. (2008). Predicting intensity ranks of peptide fragment ions. submitted.

[6] Frank, A. and P. Pevzner (2005). Pepnovo: De novo peptide sequencing via probabilistic network modeling. *Anal. Chem. 77*, 964–973.

[7] Frank, A., M. Savitski, M. Nielsen, R. Zubarev, and P. Pevzner (2007). De novo peptide sequencing and identification with precision mass spectrometry. *J. Proteome Res. 6*, 114–123.

[8] Frank, A., S. Tanner, V. Bafna, and P. Pevzner (2005). Peptide sequence tags for fast database search in mass-spectrometry. *J. Proteome Res. 4*, 1287–95.

[9] Freund, Y., R. Iyer, R. Schapire, and Y. Singer (2003). An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research 4*, 933–969.

[10] Keller, A., A. Nesvizhskii, E. Kolker, and R. Aebersold (2002). Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Anal. Chem. 74*, 5383–5392.

[11] Klimek, J., J. Eddes, L. Hohmann, J. Jackson, A. Peterson, S. Letarte, P. Gafken, J. Katz, P. Mallick, H. Lee, A. Schmidt, R. Ossola, J. Eng, R. Aebersold, and D. Martin (2008). The standard protein mix database: A diverse data set to assist in the production of improved peptide and protein identification software tools. *J. Proteome Res. 7*(1), 96–103.

[12] Mallick, P., M. Schirle, S. Chen, M. Flory, H. Lee, D. Martin, J. Ranish, B. Raught, R. Schmitt, T. Werner, K. B., and R. Aebersold (2007). Computational prediction of proteotypic peptides for quantitative proteomics. *Nature Biotech. 25*, 125–131.

[13] Tang, H., R. Arnold, P. Alves, Z. Xun, D. Clemmer, M. Novotny, J. Reilly, and P. Radivojac (2006). A computational approach toward label-free protein quantification using predicted peptide detectability. *Bioinformatics 22*, e481–488.

[14] Tanner, S., H. Shu, A. Frank, M. Mumby, P. Pevzner, and V. Bafna (2005). Inspect: Fast and accurate identification of post-translationally modified peptides from tandem mass spectra. *Anal. Chem. 77*, 4626–4639.

[15] Tsur, D., S. Tanner, E. Zandi, V. Bafna, and P. Pevzner (2005). Identification of post-translational modifications via blind search of mass-spectra. *Nature Biotech. 23*, 1562–2567.