InsPecT: Fast and accurate identification of post-translationally modified peptides from tandem mass spectra

Stephen Tanner∗ Hongjun Shu † Ari Frank ‡ Marc Mumby § Pavel A. Pevzner ¶ Vineet Bafna ∥

Abstract

Reliable identification of post-translational modifications is key to understanding various cellular regulatory processes. We describe a tool, InsPecT to identify post-translational modifications using tandem mass spectrometry data. The tool is based upon novel algorithms for the following: (a) Constructing tag-based filters based on a novel de novo interpretation algorithm that works in the presence of modifications. The sequence tags filter much of the database while retaining the true peptide; (b) a fast string-based search for scanning the database with sequence tags; (c) a dynamic programming technique to identify candidate peptides with modifications without explicit enumeration of peptides; (d) a scoring algorithm that reflects peptide fragmentation patterns, and is independent of peptide length, and (e) a novel quality score based on several complementary features. The tool was tested on a number of real and simulated data-sets. InsPecT identifies modified and unmodified peptides with better accuracy than other database search tools. We identified a large number of modified peptides, including several novel phosphopeptides in data-sets provided by the Alliance for Cellular Signalling. For these data, InsPecT was one to two orders of magnitude faster than existing search tools.

1 Introduction

Fueled by recent improvements in instrumentation as well as software for interpreting MS data, tandem mass spectrometry has become the tool of choice for protein identification. Database search, typified by tools like Sequest [10], and Mascot [25], is a popular approach to peptide identification. While the underlying algorithms are effective and used extensively for identification, many spectra remain unidentified by these searches. This can be attributed to several factors including poor quality of fragmentation/ionization, and the presence of modifications and mutations that are not explicitly represented in the database.

Another approach to identification is the de novo sequencing approach, in which the peptide is reconstructed solely from the mass spectrum, without the use of a peptide database [1]. The de novo algorithms include Lutefisk [34, 35] (a publicly available tool), SHERENGA [7] (part of the Spectrum Mill by Agilent), and Peaks [20] (Bioinformatics Solutions, Inc.) as well as approaches by Chen et al [5] and Bafna and Edwards [4] (see [19] for a recent review). These algorithms model the problem by generating Prefix Residue
**Mass (PRM)** spectra (scored versions of the MS/MS spectra) and constructing a spectrum graph. Until recently, the application of these algorithms was limited to high-quality spectra, or spectra of peptides that were not already in a database. Recent algorithmic improvements, as well as improvement in instrumentation, have led to a resurgence of interest in these tools. Improvements in *de novo* sequencing notwithstanding, database search algorithms remain the workhorse of peptide identification. The database search approach considers theoretical fragmentation spectra from database peptides, and finds one that best matches the input spectra. In a sense, this approach is not that different from *de novo* sequencing, since the latter can be viewed as a search in the (virtual) database of all peptides. The key in both approaches is a score function that ranks candidate peptides according to their likelihood of generating the spectrum. In the database search scenario, the set of candidates is limited. With incomplete fragmentation and low signal to noise ratio, some spectra may not provide enough information to differentiate (*de novo*) between two or more candidate peptides. In such cases, the database search approach simply selects the one that is present in the database, and thereby obtains the correct identification.

Post-translational modifications level the playing field for the two approaches. Small changes in the peptide sequence due to post-translational modifications and mutations shift spectral peaks significantly. Enumerating all possible mutations and modifications in the database makes the database prohibitively large, but does not seriously affect *de novo* approaches. Consequently, the current approach to identifying modifications is based upon a virtual ‘on-the-fly’ enumeration of modifications of all candidate peptides. This approach is still computationally expensive, so that a search for modifications remains limited to specialized databases, or spectra of particular interest.

The problem is similar to one faced by the genomics community in their search for sequence similarities. In order to find distant homologs, the alignment scoring had to be more sophisticated (and computationally expensive) thus making the database search too expensive to be a routine tool in the laboratory. The problem was solved using database filters that quickly eliminated much of the database, while retaining all true hits (See Figure 1). We argue that filtration is key to identification of modified peptides using database search. At first glance, this is counter-intuitive since there is no apparent connection between reducing the number of candidates and identifying modified peptides. Note, however, that aggressive (but accurate) filtration allows us to apply more sophisticated and computationally intensive scoring to the few remaining candidates. Indeed, if the database is reduced to a few peptides, one can afford to consider a rich set of post-translational modifications (PTMs) for every peptide in the filtered database. Also, the decreased number of candidates reduces the possibility of a high score being achieved by chance.

The notion of filtering has been embraced by the mass spectrometry community. Mann and Wilm [22] were early proponents of using sequence tags in a database search, and this idea has been greatly extended by Tabb et al. [32], Searle et al. [28] Sunyaev et al. [31], Day et al. [8], and others. The main idea is that while correlating masses is expensive, sequence based searches are efficient. This leads to a new paradigm for tandem MS interpretation, as shown in Figure 1. The tandem mass spectrum is interpreted *de novo* to generate partial sequence information. The database is scanned using this information, and candidate peptides that match sequence tags are then correlated with the spectra. Tabb et al. do not focus extensively on identifying modifications, but point out that the tags can be used to find possible candidates with modifications. Sunyaev et al. follow a different path, with an aggressive *de novo* approach to identify much longer tags, followed by a BLAST-like search to identify peptides related (but not identical) to the peptide that generated the query. Searle et al. perform a breadth-first search to match the candidate peptide to the query spectrum allowing for unanticipated modifications. Their search results in candidate peptides that must then be validated.

Indeed, while conceptually simple, the filter-based paradigm has a number of technical difficulties that must be overcome before searching for modifications becomes routine. Much work has been done recently on improved *de novo* sequencing and tagging algorithms [12]. However, these algorithms have been tested mainly on high-quality spectra of unmodified peptides, and the possibility of filtering out a true modified peptide remains real. Next, while sequence tags can be used to efficiently filter a sequence database, in practice
most implementations are slower than regular database searching. Also, after matching the tags, identification of candidate peptides is not trivial. Tabb et al. [32] relax the restriction on the flanking masses to include modifications, whereas Searle et al. employ a computationally expensive breadth-first search algorithm to identify candidate peptides. Efficient generation of (modified) peptide candidates remains a challenging part of filter design. Finally, the candidate peptides must be scored against the query spectrum. Much attention has been devoted to scoring [3, 6, 18, 25, 27, 33, 36, 37], and on reliability of the peptide assignments [15, 21, 23]. However, many scoring schemes make the implicit assumption that peptides are of similar lengths. This is reasonable for unmodified peptides as the parent mass restricts the range of peptides that can be scored. However, presence and absence of multiple modifications can change the parent mass of the candidate peptide, making it necessary to normalize for different sequence lengths. Additionally, the presence and absence of modifications can change fragmentation patterns, and the scoring function must be tailored to accommodate such changes. For all these reasons, reliable and automated identification of modified peptides remains a challenging problem.

In this paper, we describe an automated tool, InsPecT (Interpretation of Spectra with PT modifications) that searches large databases for possibly modified peptides by improved algorithms for each of the modules in Figure 1. It has the following features:

- **Tag-based filters** using a novel de novo interpretation algorithm that works in the presence of modifications and poor spectral quality.
- A fast trie-based search for scanning the database with sequence tags.
- A dynamic programming technique to identify candidate peptides with modifications without explicit enumeration of peptides;
- A scoring algorithm that reflects peptide fragmentation patterns, and is independent of peptide length,
- A novel quality score based on several complementary features.
Our tool can search for modified and unmodified peptides significantly faster than other database search tools. We identified a large number of modifications in a search of several large data-sets. Additionally, our approach is modular, along the lines of Figure 1. As each of these areas in active development, we anticipate replacing some of the modules with superior algorithms.

2 Methods

Our approach to peptide identification follows the scheme in Figure 1. Local de novo sequencing is used to generate sequence tags. These tags are used as PTM-aware text-based filters that reduce a large database to a few peptide candidates. We then carry out probabilistic scoring of these candidates, extending previous approaches [3, 7] to both rank the peptide candidates and assess their quality.

Preprocessing: Low-intensity peaks are filtered from spectra in order to remove noise and speed later computations. Each peak is compared with peaks in the neighborhood, centered around itself, with radius 25 Da. Any peak which is not one of the top 6 peaks in its neighborhood is dropped. Parent mass correction is performed, correcting the parent mass by up to two Da so as to maximize b/y ion pairing. On a typical set of spectra (the SimMod_0 data-set, described in Section 3.1), this parent mass correction decreased the mean parent mass error from 0.5 to 0.2 Da.

Tag Generation: Tag generation is a modified (but not simpler) version of de novo spectrum interpretation. We construct a directed acyclic graph where each node corresponds to a prefix residue mass (PRM), and each edge corresponds to a (possibly modified) amino acid. A path in the graph is a possible tag.

Given a peak of mass $M$ for a spectrum of parent mass $P$, we produce a node at mass $M - |H|$ (for b fragments) and at $P - M$ (for y fragments). Two additional “goalpost” nodes are placed at zero mass and at the parent residue mass. Scoring parameters are based upon odds empirically derived from an annotated set of spectra. The PRM range of a spectrum is divided into three equal sectors, and tagging parameters are derived for each sector and parent ion charge individually. A PRM node’s score, $S(N)$, is derived from the following features, scored as log-odds ratios.

1. **Intensity rank:** Peaks are ranked from the most intense to the least. Nodes receive a score based on the odds that a peak of a given rank is a b or y ion, shown in figure 2. These odds differ significantly by sector and charge. For instance, the odds that a rank-3 peak from the first or third sector of a charge-2 spectrum is a b ion are 69% and 14%, respectively. We take the log ratio of these odds with the odds that a randomly chosen peak matches a b or y ion, and obtain score $S_1(N)$.

2. **PRM support:** For each PRM, we enumerate the ion types that support the node by checking for peaks at expected masses. The set of ion types identified is a witness set. Each node receives a score based upon empirical probability that its witness set represents a true peak. The optimal (and largest) witness set includes b, y, b-H$_2$O, y-H$_2$O, b-NH$_3$, y-NH$_3$, a, a-H$_2$O, a-NH$_3$, b$_2$ (doubly-charged b), and y$_2$ (doubly-charged y). Witness set odds are affected by charge and sector. For instance, b$_2$ and y$_2$ ions are more abundant in charge-3 spectra (21% odds of appearing, versus only 8% in charge-2 spectra). We compare witness set odds with the odds that a randomly chosen peak matches a b or y ion; the log odds ratio is $S_2(N)$.

3. **Isotope pattern:** For each atomic mass up to 1750, we compute the expected relative intensity of the first heavy isotopic peak based upon isotope frequencies of atoms C, H, O, N, and S. A node’s score is augmented if the source peak has a plausible secondary isotopic peak, and penalized if the source peak appears to be a secondary isotopic peak. Using isotopic peaks in a path is undesirable, particularly since there are pairs of amino acids whose masses differ by roughly 1 Da. The log odds ratio for this feature gives us score $S_3(N)$.

4
Figure 2: Odds that a peak is a b or y fragment, based on its intensity rank, for peaks in the center sector (left plot) and right sector (right plot) of charge-2 spectra. The most intense spectral peaks are very likely to represent y ions, and these odds drop rapidly as the rank falls. Low-intensity peaks in the center sector are unlikely to be b or y ions, as illustrated by the more rapid tailing off of the curves on the left.

We connect two nodes of mass \(m_1\) and \(m_2\) with an edge if the distance \((m_2 - m_1)\) is within \(\epsilon\) (by default, \(\epsilon = 0.5\)) of a legal jump. Legal jumps include all amino acid masses, and (if PTMs are allowed) all singly-modified amino acids. We assign a score, \(S(E)\), to each such edge based upon its skew (the difference between the edge length and the expected mass).

Tags are scored additively. Given a path connecting nodes \(N_0 \ldots N_n\) via edges \(E_0 \ldots E_{n-1}\), with node scores \(S(N_k)\) and edge scores \(S(E_k)\), assign the tag a score of \(\sum_{k=1}^{n} (S_1(N_k) + S_2(N_k) + S_3(N_k)) + \sum_{k=1}^{n-1} S(E_k)\). Currently, we generate tags of length three, and retain up to fifty top-scoring tags for subsequent database searching. We will show in the results section that this is effective. However, we plan to augment tags by optimizing the relative weights of features [12].

2.1 Database Search & Candidate Generation

We search the database for sequence tags using a trie-based data structure [2]. After a rapid construction of the trie automaton, all matches to any tag in the trie can be computed in a single scan of the database, independent of the number of tags in the automaton. We note that this is one of many ways to search for tags. For instance, the database could be indexed by pre-computing the locations of each tag. In practice, however, this approach has problems. The index must be recomputed each time the database or parent mass tolerance is changed, and the index size grows explosively with the number of PTMs considered. With the automaton, we are forced to perform at least one scan of the entire database. However, the cost is easily amortized by combining tags from multiple spectra into a single automaton. Each node of our trie contains zero or more tags; leaf nodes all contain at least one tag. We preprocess the database to facilitate fast scanning, and are working to distribute processing to further speed the search. InsPecT scans the database with the trie, looking for any peptide that matches a tag.

When such a peptide is found (an initial match, or tag hit), we attempt to extend it to a full match by finding flanking sequences which match the tag’s prefix and suffix masses, allowing for potential modifications. Different combinations of allowed modifications generate many different molecular masses. To search efficiently, we use the following approach: Define a decoration to be a possible set of post-translational modifications. For instance, if our search allows up to two phosphorylations and two methylations, we generate a total of nine decorations (including the “empty decoration”), each with a specific mass delta. We order the decorations according to increasing mass, and generate an array \(DM\) to store the mass values. This is done once, at the beginning of the search. Consider a tag with suffix value \(S\). The tag matches a peptide in the database
with decoration $d$ if the peptide contains the tag, and has a suffix of mass $R$, such that $|S - (R + DM[d])| < \epsilon$. Figure 3 describes an efficient algorithm to search for candidate peptides.

Extend_tag_hit(S,n)

(* Find a peptide sequence that matches suffix mass S, starting at position n and allowing feasible modifications *)

\[ p = n \]
\[ d = \text{MaxMod} \]
\[ R = 0 \]

repeat

while \((R > S + DM[d] + \epsilon)\)

\[ d = d - 1; \]
if \((d < 0)\) EXIT;
if \(|R - S - DM[d]| < \epsilon\)

MATCH

\[ p = p + 1 \]
\[ R = R + M[p] \]

until \((d < 0)\)

Figure 3: Algorithm to check if the suffix of a tag-hit is a candidate, allowing for modifications

Prefix extension is performed similarly. Successful extensions are filtered to remove unfeasible decorations (for instance, oxidation requires the presence of a methionine). Next, given a candidate peptide, the optimal attachment positions of PTMs on the flanking regions are determined by dynamic programming. This technique allows us to determine the optimal attachment positions without explicit enumeration and scoring of all possibilities. We consider a set of attachment sites to be optimal if the resulting set of PRM nodes has the highest possible score, as assigned during tag generation.

We select attachment sites by computing $S[d,j]$, which is the optimal score attainable by attaching decoration $d$ to the first $j$ residues $r_1, \ldots, r_j$ of the candidate peptide. For an amino acid $r$, let $D(r)$ represent the set of all possible decorations that can be applied to that residue. Let $M[d,j]$ be the PRM created by attaching decoration $d$ to the first $j$ residues. Then, $S[d,j]$ is computed by the following recurrence

\[ S[d,j] = \text{NodeScore}(M[d,j]) + \max_{d' \leq d, d' \in D(r_j)} \{S[d-d', j-1]\} \quad (1) \]

The corresponding d.p. is used to compute the optimal attachment points. NodeScore($M[d,j]$) equals the score $S(N_k)$ of the PRM node at that mass, or a penalty if no PRM node was generated nearby. Multiple PTMs at one amino acid - such as double oxidation of a methionine residue - are permitted where chemically reasonable. Note that $D(r)$ remains a small set. The peptide, with appropriately attached modifications, forms a candidate peptide that must now be scored.

2.2 Candidate Peptide Scoring

The candidate peptides must be scored and ranked according to their relative likelihoods of generating the mass spectrum. Clearly, scoring is the mainstay of all database search algorithms, and the focus of intensive research [3, 6, 18, 25, 27, 33, 36, 37]. When post-translational modifications are considered, the score function must be modified. A key issue here is normalization for peptides of different lengths. In the absence of modifications, all candidate peptides have identical mass, and therefore a small variation in peptide length. When putative modifications are considered, the variation in length can be considerable, particularly if multiple
modifications must be considered. As longer peptides have more putative fragments that can match, as well as more fragments that might be missing, certain score functions (such as the total ion current explained by the peptide) have a peptide-length bias. Here, we describe a score function that extends earlier approaches [3, 7] and also normalizes for peptide length. Dynamic programming is used to compute the score.

We assume a probabilistic model in which a peptide generates fragments which show up as spectral peaks. Peaks can possess skew in m/z due to instrument error, and have an intensity proportional to the likelihood of the fragment ion being seen. We also consider a null model, in which the same number of spectral peaks are generated at random m/z values. Consider a spectrum \( S' = S_1, S_2, \ldots, S_k \) with the \( k \) spectral peaks sorted by mass-charge ratio. For a candidate peptide \( P \), let \( \mathcal{F}(P) = f_1, f_2, \ldots, f_n \) denote the candidate fragments (theoretical peaks) sorted by mass-charge ratio. We will define a probability \( \Pr(S|\mathcal{F}(P)) \), that the spectrum \( S \) is generated by the peptide \( P \). We compare this to \( \Pr(S|n) \), the probability that the spectrum was generated from the null model. The score for the candidate peptide is the log-odds function

\[
\log \left( \frac{\Pr(S|\mathcal{F}(P))}{\Pr(S|n)} \right)
\]

(2)

In order to compute \( \Pr(S|\mathcal{F}(P)) \), we use a mapping between theoretical ions and spectral peaks. The (hardware-dependent) probability \( \Pr(f_j|P) \) that a theoretical ion \( f_j \) was generated by a peptide \( P \) is computed based on empirical observations of peptide composition. For example, the probability of seeing a neutral \( H_2O \) loss is higher in the presence of acidic residues. The probability of not seeing a fragment ion \( f_j, \Pr(\bar{f}_j|P) \), is simply \( 1 - \Pr(f_j|P) \). Ion probabilities are computed separately for each sector, reflecting the richer sets of peaks in the center of typical spectra. The probability of matching \( S_i \) with \( f_j \) also depends upon the difference in m/z values (skew), and is modeled by \( \Pr(S_i \simeq f_j) \). In the null model, the probability \( \Pr(f_j|n) \) that a randomly generated fragment matches a fragment \( f_j \) is simply a function of the number of peaks in its sector and the mass tolerance allowed. Similarly, \( \Pr(\bar{f}_j|n) \) equals \( 1 - \Pr(f_j|n) \).

An unassigned spectral peak \( S_i \) is assumed to be noise, and is generated with a probability \( \Pr(S_i \simeq \phi) \), which depends on the intensity of the peak. In the null model, the probability \( \Pr(N|n) \) that a peak is noise (unaccounted for by the \( n \) 'true' nodes) depends somewhat on peak count, but is generally near 1.

Since many assignments of peaks to ions are possible, we choose one that maximizes the log odds score. This can be done using a dynamic programming computation under the reasonable assumption that assigned peaks do not cross. In other words if peaks \( S_{i_1}, S_{i_2} \), with \( S_{i_1} < S_{i_2} \) are assigned to \( f_{j_1}, f_{j_2} \), then \( f_{j_1} \leq f_{j_2} \). Let \( S_i \) represent the first \( i \) spectral peaks \( S_1, \ldots, S_i \), and \( \mathcal{F}_j(P) \) denote the first \( j \) theoretical fragments. Denote

\[
\psi(i, j) = \log \left( \frac{\Pr(S_i|\mathcal{F}_j(P))}{\Pr(S_i|n)} \right)
\]

Clearly, it is sufficient to compute \( \psi(i, j) \) for all \( i, j \). The computation is given by the recurrence

\[
\psi(i, j) = \max \left\{ \begin{array}{ll}
\psi(i - 1, j) + \log \left( \frac{\Pr(S_i \simeq \phi)}{\Pr(N|n)} \right) & \text{Peak } i \text{ is noise} \\
\psi(i - 1, j - 1) + \log \left( \frac{\Pr(f_j|P)}{\Pr(f_j|n)} \right) + \log(\Pr(S_i \simeq f_j)) & \text{Peak } i \text{ is matched with fragment } j \\
\psi(i, j - 1) + \log \left( \frac{\Pr(f_j|P)}{\Pr(f_j|n)} \right) & f_j \text{ is not generated}
\end{array} \right.
\]

This algorithm returns the score of the optimal assignment, as well as the mapping from theoretical fragments to spectral peaks. The main advantage of this model is that the fragmentation probabilities can be recomputed for different instrumentation types, and the effects of PTMs on fragmentation (as in phosphopeptides) can be explicitly modeled. The main disadvantage is that we do not explicitly handle dependencies between different fragments. Therefore, we refine the initial interpretation, primarily by removing unacceptable ion interpretations, where secondary ions are present but a primary ion is not. For instance, no theoretical peak of
ion type b-H$_2$O can be assigned to a spectral peak unless the corresponding b peak has been assigned. Any such spectral peaks are reinterpreted as noise. A second refinement step rescues peaks that are interpretable as isotopes of other (non-noise) peaks. Finally, depending on search options, a small bonus is added for matching protease digestion rules (no missed cleavage, and the ends match the protease digestion specificity). In ongoing work, we plan to replace these refinements with more sophisticated scoring algorithms that include fragment dependencies [11].

### 2.3 P-value computation

The scoring procedure outlined above ranks the candidate peptides. This is not sufficient as the top-scoring peptide might not be the correct one even if the rank order of peptides is correct, if the true peptide is not in the database. This is a common issue with database search algorithms. Different criteria, both automated (such as the Sequest Xcorr score) and manual, can be applied to separate true positives from high-scoring spurious matches. Based on optimization of several features, we devised a match quality score to estimate the probability that the top match is correct, and not just the best of various poor alternatives. Besides the candidate score, we considered several features: explained intensity ($I$), explained peaks ($P$), $\delta$-score, and b/y ion score ($B$). The explained intensity of a candidate is the fraction of total ion current belonging to annotated spectral peaks. Similarly, the explained peak score is the fraction of the (filtered) peaks that are annotated. The $\delta$-score is the difference in score between a candidate peptide and the next-best candidate. The b/y ion score is the fraction of b and y ions found in the spectrum. As shown in Figure 4a-d, these measures help distinguish true peptides and complement the score computation. The $\delta$-score is affected by database size (a small database will give artificially large $\delta$-score values to poor matches). Therefore, we do not consider it in computing p-value, but report it separately. In the future, we plan to incorporate it into a more sophisticated confidence calculation calibrated to an MS run [23].

We compute an optimal linear combination of the remaining four factors, to obtain a quality score of a match as

$$Q = w_1S + w_2I + w_3P + w_4B$$

(3)

The scores $S$, $I$, $P$, and $B$ are normalized by subtracting the mean and dividing by standard deviation. The weights $w_i$ have been chosen to optimally discriminate correct matches from incorrect matches. Let $T$ be the set of true matches whose score $Q$ is below 1, and let $F$ be the set of false matches whose score $Q$ is above -1. (For a perfect scoring scheme, $T$ and $F$ would both be empty sets). We used an objective function which penalizes these poorly-resolved cases:

$$\sum_{m \in T} B(1 - Q_m)^2 + \sum_{m \in F} (1 + Q_m)^2$$

The constant $B$ is the ratio of total incorrect to correct matches, used so that false negatives and false positives will carry the same weight. Assignment of weights was carried out using the Nelder-Mead Simplex Method [17]. Weights were tuned on a training set consisting of 612 correct and 6,471 incorrect (but top-scoring) matches on the ISB data-set, and tested on a separate data-set containing 590 correct, and 6,373 incorrect spectra. Repeated runs from different starting conditions produced the same optimized weights, indicating that the optimum is global. The optimized weights indicate that success in finding b and y ion series, and annotating many of the peaks of the spectrum, is more indicative of a true match than explaining a large fraction of the total ion current. The $p$-value for a candidate peptide is computed by comparing the match quality score to the distribution of quality scores for incorrect matches. Figure 5(a) shows the performance of the optimal quality score in separating between the two sets for both training and test data. We use the distribution of quality scores on the incorrect set as a $p$-value (Figure 5(b)). This $p$-value was tested on the second set of spectra, and sensitivity and specificity were as good. In the future, we plan to reconsider the $p$-value computations when searching (as is usual) a full run of spectra as a batch.
Figure 4: Other score features that complement the candidate-score in distinguishing a correct peptide from a false one. (a) Explained intensity versus Score. (b) b/y score versus score. (c) Explained peaks versus score (d) b/y score versus explained peaks. The measures all help distinguish correct top matches from incorrect top matches, and complement each other.

Figure 5: (a) Histogram of match quality scores, separated into correct and incorrect matches, showing the performance of quality scores in separating true matches from false matches. (b) Empirically generated p-values. False matches rarely attain quality scores, and so higher quality scores give low p-values.
3 RESULTS

3.1 Data

The following data-sets were used:

**ISB:** The ISB data-set [16] is a well-known collection of MS/MS spectra from 22 separate LC-MS runs on a ThermoFinnigan ESI-ITMS. Two mixtures were prepared by combining a set of purified proteins. This list of pure proteins, together with known contaminants like keratin, constitute a valid set. We created an ISB-search database containing 93000 human proteins from the nr database, together with the valid proteins, for a total size of 31Mb. As a first approximation, an identification is considered correct if the top match in the search database is a valid protein, and incorrect otherwise. Also, we consider a peptide match to be correct if it substitutes I for L, or Q for K (or vice versa).

**SimMod:** The Open Proteomics Database (OPD) is a public repository of mass spectrometry data[26]. We selected high-quality E. coli spectra from the OPD, and combined these with spectra from the ISB data-set to obtain a total of 1112 annotated spectra for test purposes (charge 2, Sequest Xcorr score > 2). We constructed a data-set of modified spectra by adding feasible modifications to each peptide, and shifting the spectral peaks appropriately. Given a theoretical peak at mass $m_1$ that shifts in mass by $\delta$, we shift all peaks in the neighborhood of $m_1$ by $\delta$. We also swap peaks in a neighborhood of $m_1 + \delta$ down by $-\delta$; carrying out swaps rather than shifts avoids introducing gaps in the spectrum. Swaps of major ion types (b and y) take precedence over swaps of secondary ion types (neutral losses). SimMod$_i$ refers to the data-set with $i$ modifications randomly selected from the set of feasible modifications. Thus SimMod$_0$ is the original data-set, while SimMod$_1$, and SimMod$_2$ represent real spectra with 1 and 2 modifications. In the absence of curated data-sets with real modifications, the SimMod data-sets are useful in testing methods. The set of allowable modifications was hydroxylation of proline or lysine, sulfation of tyrosine, and oxidation of methionine. To search the SimMod data-sets, we used the ISB-search database, augmented with E. coli proteins used for the OPD data-set.

**Mus-IMAC data-set:** The Mus-IMAC data-set represents partially annotated spectra of peptides from the Protein Chemistry Laboratory of the Alliance for Cellular Signaling. Murine RAW 264.7 cells were treated with the serine/threonine phosphatase inhibitor calyculin-A and mass spectra of enriched phosphopeptides obtained as described previously[29]. Proteins were extracted using the Tripure reagent (Roche Applied Science), digested with trypsin, and subjected to immobilized metal affinity chromatography (IMAC) to enrich for phosphopeptides. The enriched samples were analyzed by LC-MS/MS using a ThermoFinnigan LCQ Deca mass spectrometer. The data-set consists of three mass spectrometry runs (14,061 spectra)[30]. An nr-mus database was created by extracting proteins that matched the keywords “mouse”, “mus musculus”, or “m. musculus” from the current NCBI non-redundant database. The nr-mus database contained 70000 proteins (24 Mb).

**Hardware:** Timing statistics were gathered on a desktop PC with 2.8GHz Intel processor with 1GB of RAM. The operating system was RedHat Linux, kernel version 2.6.5.

3.2 Tagging

The tagging accuracy of InsPecT was tested on the SimMod data-sets, and is shown in Figure 6. All results are for tags of length 3, along with flanking masses that are correct to a pre-specified tolerance. Figure 6(a) plots the fraction of spectra with a correct tag against the number of tags generated. The data is tabulated in Figure 6(b) As the number of tags predicted goes from 1 to 100, the fraction of spectra tagged correctly rises to about 90%. While this might seem low (we will never identify 10% of the spectra), a manual examination
Figure 6: Tagging performance of InsPecT. The Tagging accuracy is defined as the % of spectra with at least one correct tag in the top $n$ predictions. (a) Comparison with GutenTag. (b) Performance on spectra with simulated modifications. Clearly, the accuracy does not decrease by adding modifications.

![Graph of Tag Generation](image)

<table>
<thead>
<tr>
<th>Number of Tags</th>
<th>Tagging Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SimMod$_0$GutenTag 40.0</td>
</tr>
<tr>
<td>10</td>
<td>SimMod$_0$InsPecT 53.3</td>
</tr>
<tr>
<td>20</td>
<td>SimMod$_1$GutenTag 27.5</td>
</tr>
<tr>
<td>50</td>
<td>SimMod$_1$InsPecT 50.8</td>
</tr>
<tr>
<td>100</td>
<td>SimMod$_2$GutenTag 18.7</td>
</tr>
<tr>
<td></td>
<td>SimMod$_2$InsPecT 49.4</td>
</tr>
</tbody>
</table>

Table 1: (a) InsPecT search speed in the presence of post-translational modifications. Trie-based searching allows the identification of many spectra in one pass through the database. (b) InsPecT peptide identification accuracy in the presence of post-translational modifications. Search was carried out against the 5.7Mb database.

<table>
<thead>
<tr>
<th>Database Size (MB)</th>
<th>Speed (sec.)</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Top candidate</td>
<td>Top ten</td>
</tr>
<tr>
<td>GutenTag</td>
<td>SimMod$_0$GutenTag 83.5</td>
<td>91.7</td>
</tr>
<tr>
<td>SimMod$_0$</td>
<td>SimMod$_0$InsPecT 90.6</td>
<td>92.0</td>
</tr>
<tr>
<td>SimMod$_1$</td>
<td>SimMod$_1$InsPecT 79.6</td>
<td>81.0</td>
</tr>
<tr>
<td>SimMod$_2$</td>
<td>SimMod$_2$InsPecT 75.8</td>
<td>78.0</td>
</tr>
</tbody>
</table>

reveals that many of the spectra for which tags are missed are of low quality. Note that these tests are on spectra in which the parent mass can be off by up to 2 Da, and fragment ions may be off by 0.5Da. With improvements in instrumentation w.r.t fragmentation, and mass accuracy, we should be able to search with longer and fewer tags, greatly improving the efficiency. In practice, we use the top 50 tags to launch the search. Clearly, the performance of InsPecT does not get worse after adding modifications. The ability of InsPecT to incorporate modifications into a tripeptide tag is essential for high accuracy. GutenTag does not generate such tags. When modifications are restricted to the N- and C-termini of a peptide, so that more unmodified valid tags are attainable, GutenTag’s performance improves (from 62% to 71% accuracy in the presence of one PTM).

Next, we tested the speed and accuracy of InsPecT (combined tagging, filtering, and scoring) in identifying peptides in the SimMod data-sets. Tables 1a,b describe the results. Note that the speed decreases only modestly upon increasing the allowed modifications. As mentioned earlier, we test accuracy by searching a larger database. A search is considered correct if the top hit is a valid protein, and incorrect otherwise. As expected, the accuracy falls upon increasing allowed modifications, mainly because of a loss of tags. GutenTag was not designed to produce candidate peptides with PTMs, so we report its database results only for SimMod$_0$. 

11
Table 2: Peptides found in oxidized form in the ISB data-set. Relative degree of oxidation varies between peptides and between runs. Although the oxidations in this data-set represent chemical change rather than biological regulation, identification of oxidized peptides is important for identification and quantification.

3.3 ISB data-set

We searched the set of ISB spectra against the database using InsPecT. The top match was valid for a total of 2978 spectra, as compared with 2765 originally found by SEQUEST. Our search found 79 previously unannotated spectra containing an oxidized methionine residue. Of these spectra, 72 match a peptide whose unmodified counterpart is also matched by some other valid spectrum, thus providing additional validation (See Table 2). Figure 7 shows such a pair of annotated spectra with and without oxidation. The average search time was 0.7 seconds per spectrum. In comparison, the SEQUEST runtime on this data-set/database (with modifications) was 2.7 seconds per spectrum. A total of 644 new annotations were gained, while 422 were lost. Many of the differences from SEQUEST results are explained by the greater emphasis our scoring scheme places on peak intensity. Tag-based filters are robust against low-intensity noise, but have difficulty with low-quality spectra where the b and y peaks have low intensity relative to noise. The average explained intensity for the new annotations is 52%, while for the lost SEQUEST annotations it is only 28%. Those spectra where InsPecT and SEQUEST were both able to annotate are particularly high quality, with average explained intensity of 57%. Of the spectra correctly identified by SEQUEST, the majority were missed due to errors in tagging which filtered out the correct peptide. Refinements to our tagging algorithm may decrease the number of such cases by improving filter sensitivity.

The average number of initial tag hits per spectrum was 263,000. The number of candidate peptides per spectrum (produced by successful tag extension) was 700. Non-tryptic peptides were not excluded from scoring (and indeed, many non-tryptic matches were found). Given our initial database size of 31 Mb, and the fact that peptides of various lengths are considered, the filtration efficiency due to tags is roughly $2 \times 10^{-6}$.

As additional validation, in 40 cases, plain and oxidized forms of a peptide appeared in the same run. We compared the elution time of the pairs. In general, oxidized forms elute earlier than non-oxidized peptides (data not shown). This is probably due to an slight increase in polarity. There is also some evidence that oxidation of methionine influences fragmentation patterns. Breakage immediately downstream (C-terminal) of an oxidized methionine appears to be more likely. The odds of seeing either a b or y fragment for this break were 64%, as compared with 58% on the N-terminal side, and 59% overall.

These particular oxidation events represent chemical damage rather than cellular regulation. However, identifying oxidized peptides is important: not doing so might result in peptide identifications being missed. Additionally, in MS based quantification, the relative intensity of peptides in different samples (ex: diseased versus normal) is a proxy for relative expression levels of the proteins. Unknown differences in oxidation...
levels in the two samples might lead to spurious conclusions on relative protein expression.

3.4 Mus-IMAC data-set

The Mus-IMAC spectra were searched against the mus-nr database (70,000 proteins, 24 Mb). The search revealed a large number of modified and unmodified hits with reliable (p-value < .05) annotations on 1700 spectra. These high-confidence annotations include a total of 281 phosphorylated and 455 non-phosphorylated peptides, from 611 proteins in all. A total of fifteen top-scoring phosphorylated peptides were verified as valid hits. These included ten of eleven previously discovered phosphopeptides from these MS runs, and five novel ones. The searches were conducted allowing a maximum of four modifications (≤ 2 oxidations, ≤ 2 phosphorylations), a parent mass tolerance of 2.5 Da, non-tryptic peptides, and missed cleavages. Even with these permissive settings, the search took an average of 1.0 seconds per spectrum using 25 tags. Using these same settings with Sequest led to a search time of 99 seconds per spectrum.

Among the novel phosphopeptides, we find phosphothreonine and phosphoserine sites in histones, including PAAPAAPAPVEKpTPVKK and PNIQAVLLPKKTEpSHHK. (See Table 3, and Figure 8). Phosphorylation has been described as a key mechanism for regulation of Histones H1 and H3 [13, 14, 24]. A large number of unmodified histone peptides were matched as well (six of the top ten peptides come from histone proteins), indicating that histones are plentiful in the cell extract. This gives us further confidence in the identification of the histone phosphorylations. Many of the novel peptides include missed cleavage sites or non-tryptic end-points, indicating that efficiency considerations may have forced SEQUEST to overlook them in a more restrictive search.

Phosphorylations are among the most widespread and biologically significant post-translational modifications. In addition to shifting peak masses, PTMs (and phosphorylations in particular) affect ion fragmentation patterns. As the data-set of phosphorylated spectra grows, we can mine it to obtain empirical estimates of fragmentation probabilities, which in turn will improve the score function, leading to more identifications. Our search identified a total of 77 spectra matching the manually verified phosphopeptides (25 phosphoserine spectra, 50 phosphothreonine, and 3 phosphotyrosine). Even with this limited data-set, we can start computing fragmentation probabilities. See Table 4 summarizes our findings. For example, it has been suggested [9] that the loss of neutral ion $H_3PO_4$ is very common for phosphoserine. For multiply charged peptides, this
Figure 8: Phosphorylated peptides from the Tripure data-set. (a-c) are novel, and (d) is a known peptide. Bolded segments of the b and y ion ladders indicate the tripeptide tag used in to locate the candidate.
Table 3: Phosphopeptides identified on the Mus-IMAC data-set.

<table>
<thead>
<tr>
<th>Novel?</th>
<th>Occurrence</th>
<th>p-value</th>
<th>Peptide</th>
<th>Protein (GI number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>8</td>
<td>0.0022</td>
<td>PAAPAAPVEKP/TPVKK</td>
<td>Histone H1d (gi</td>
</tr>
<tr>
<td>Yes</td>
<td>1</td>
<td>0.0033</td>
<td>RPpSVYLPTR</td>
<td>1500034J01Rik protein (gi</td>
</tr>
<tr>
<td>Yes</td>
<td>1</td>
<td>0.0031</td>
<td>MASNIfGp1PEENPPSWAK</td>
<td>(gi</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.0042</td>
<td>DLLHpSPEEEK</td>
<td>40S ribosomal protein S27 (gi</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>0.0057</td>
<td>DTGKnTPVEPEVAHR</td>
<td>40S ribosomal protein S20 (gi</td>
</tr>
</tbody>
</table>

Table 4: Percentage frequency of b and y ions, as well as phosphate losses, in several environments. pre-pS refers to ions formed by breaking the upstream (N-terminal) peptide bond of a phosphorlated residue, post-pS to the downstream (C-terminal) bond. Phosphate loss peaks are particularly prevalent in the presence of phosphoserine, even moreso than standard b and y peaks. Phosphate losses are particularly prevalent for breaks adjacent to the phosphorylation site.

<table>
<thead>
<tr>
<th>Phosphorylation</th>
<th>b</th>
<th>b-98</th>
<th>b-80</th>
<th>y</th>
<th>y-98</th>
<th>y-80</th>
</tr>
</thead>
<tbody>
<tr>
<td>Containing pS</td>
<td>53</td>
<td>55</td>
<td>49</td>
<td>63</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>Containing pT/pY</td>
<td>45</td>
<td>37</td>
<td>50</td>
<td>43</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>pre-pS</td>
<td>70</td>
<td>n/a</td>
<td>70</td>
<td>99</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>post-pS</td>
<td>30</td>
<td>70</td>
<td>80</td>
<td>n/a</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>pre-pTY</td>
<td>93</td>
<td>n/a</td>
<td>57</td>
<td>93</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>post-pTY</td>
<td>71</td>
<td>93</td>
<td>100</td>
<td>n/a</td>
<td>n/a</td>
<td></td>
</tr>
</tbody>
</table>

4 Discussion

We demonstrate that it is possible to identify modifications in peptides from complex mixtures using desktop computer resources. We focus extensively on search efficiency using sequence tag filters. Efficient searching is often not at the forefront of proteomic research, as one can compensate by “adding more computers”. In practice, however, efficiency considerations often dictate which peptides get identified. As PT modifications would lead to a dominant $M - H_3PO_4$ ion in the spectrum, as well as neutral losses on fragment ions. The effect is much less pronounced for phosphothreonine and phosphotyrosine.

Additionally, a b-prefix ion, formed by a break immediately upstream of the phosphorylation site, has an 83% chance of being observed (as compared with 54% for b ions overall). A y suffix ion, formed by a break immediately downstream of the phosphorylation site, has a 92% chance of being observed (as compared with 63% for y ions overall). This reflects the fact that breakages are particularly likely to leave positive charge on the side without the (negatively charged) phosphate group. This is similar to the favored breakage on the N-terminal (rather than C-terminal) side of proline residues. However, the counterpart ions (b suffix, and y suffix) are not significantly suppressed in phosphopeptides. As we train our model of fragmentation preferences, our ability to identify PTMs will improve.
lead to a combinatorial explosion, they are often searched for under very restrictive conditions. For example, one might limit the search to proteins from which unmodified peptides have been found, or forbid multiple modifications or missed trypsin cleavages. We demonstrate that it is feasible to search efficiently without such restrictions.

Our approach is modular, and will improve with improvements in the modules for tagging, filtering, and scoring. With improvements in instrument mass accuracy, and de novo sequencing algorithms, sequence tags should become the dominant filter, identifying all but a few of the true peptides. As other results, and our own results on the ISB data-set show, the tag-based approaches do very well for high intensity spectra, where the signal peak intensities dominate. We also exploit the full power of sequence-based search using tags by combining the tags from multiple spectra in a single scan. With a linear time pre-processing of the tags to construct a trie-automaton [2], a single pass through the database handles many tags and spectra. Correspondingly, our search is faster than other database search algorithms, even after allowing for modifications. With improvements in tag generation, we can use longer (and fewer) tags, leading to further speed improvements. Finally, a dynamic programming technique allows us to generate modified peptide candidates without explicit enumeration of all modifications. In future work, we intend to extend our algorithm to handle “blind” modification search (for modifications of unspecified size), as well as mutation-tolerant search. Our tool is available by contacting the authors.

Acknowledgments

We would like to thank Edward Marcotte and his colleagues at the University of Texas for making the Open Proteomics Database publicly available. We would also like to thank Lien Chung for running GutenTag benchmarks. We benefitted greatly from discussions with Helge Weissig and Nuno Bandeira. We are grateful to Alexei Nesvizhskii for a careful reading of this paper and for many useful discussions.

References


