BIOINFORMATICS ORIGINAL PAPER

Vol. 27 no. 6 2011, pages 797–806
doi:10.1093/bioinformatics/btr017

Gene expression

Advance Access publication January 22, 2011

Computational refinement of post-translational modifications predicted from tandem mass spectrometry

Clement Chung[1,2], Jian Liu[3,4], Andrew Emili[3,4] and Brendan J. Frey[1,2,3,5,*]

[1]Department of Computer Science, [2]Probabilistic and Statistical Inference Group, [3]Banting and Best Department of Medical Research, [4]Donnelly Centre for Cellular and Biomolecular Research and [5]Department of Electrical and Computer Engineering, University of Toronto, Toronto, Canada

Associate Editor: Olga Troyanskaya

ABSTRACT

Motivation: A post-translational modification (PTM) is a chemical modification of a protein that occurs naturally. Many of these modifications, such as phosphorylation, are known to play pivotal roles in the regulation of protein function. Henceforth, PTM perturbations have been linked to diverse diseases like Parkinson's, Alzheimer's, diabetes and cancer. To discover PTMs on a genome-wide scale, there is a recent surge of interest in analyzing tandem mass spectrometry data, and several unrestrictive (so-called 'blind') PTM search methods have been reported. However, these approaches are subject to noise in mass measurements and in the predicted modification site (amino acid position) within peptides, which can result in false PTM assignments.

Results: To address these issues, we devised a machine learning algorithm, PTMClust, that can be applied to the output of blind PTM search methods to improve prediction quality, by suppressing noise in the data and clustering peptides with the same underlying modification to form PTM groups. We show that our technique outperforms two standard clustering algorithms on a simulated dataset. Additionally, we show that our algorithm significantly improves sensitivity and specificity when applied to the output of three different blind PTM search engines, SIMS, InsPecT and MODmap. Additionally, PTMClust markedly outperforms another PTM refinement algorithm, PTMFinder. We demonstrate that our technique is able to reduce false PTM assignments, improve overall detection coverage and facilitate novel PTM discovery, including terminus modifications. We applied our technique to a large-scale yeast MS/MS proteome profiling dataset and found numerous known and novel PTMs. Accurately identifying modifications in protein sequences is a critical first step for PTM profiling, and thus our approach may benefit routine proteomic analysis.

Availability: Our algorithm is implemented in Matlab and is freely available for academic use. The software is available online from http://genes.toronto.edu.

Supplementary Information: Supplementary data are available at *Bioinformatics* online.

Contact: frey@psi.utoronto.ca

Received on September 8, 2010; revised on December 14, 2010; accepted on January 5, 2011

*To whom correspondence should be addressed.

1 INTRODUCTION

Proteins are created through a biological process called protein biosynthesis. This process begins with transcription and splicing of genes into messenger RNA (mRNA) molecules, which are later translated into polypeptides. At the time of translation, a protein can either be active or inactive, and its subsequent activity is generally regulated by chemical modifications referred to as post-translational modifications (PTMs). PTMs, which may occur during or after translation, involve an enzymatic addition of a chemical group (e.g. a phosphate) or a larger moiety (e.g. an additional polypeptide such as ubiquitin) onto one or more amino acid side chains. Many PTMs, in particular, phosphorylation on serine (S), threonine (T) or tyrosine (Y), can regulate a protein's function by influencing its folding, stability or physical association with other proteins, thereby activating or suppressing it.

Since PTMs have been shown to dynamically influence a wide range of important processes (e.g. catalysis of biochemical reactions, intracellular cell signaling and cell division), mapping of PTMs in a comprehensive proteome-wide manner remains a critical outstanding research problem. Although the biological importance of certain PTMs is well established, the diversity and the prevalence of PTMs and their targets remain to be fully elucidated. One recently developed approach to discover PTMs on a genome-wide scale is to analyze tandem mass spectrometry (MS/MS) data using an unrestricted (so-called 'blind') PTM search engine. Unlike traditional 'restricted' search methods [Baliban *et al.* (2010); Craig and Beavis (2004); Eng *et al.* (1994); Matthiesen *et al.* (2005); Perkins *et al.* (1999)], blind search engines require no predetermined list of candidate PTMs, with pre-defined delta masses or preferred target residues. This allows blind PTM search engines to be able to consider a large number of potential PTMs at once, representing both previously known PTMs and new ones. A number of blind search engines have been reported that employ various different optimization techniques and sequence prediction approaches [Baumgartner *et al.* (2008); Chen *et al.* (2009); Han *et al.* (2005); Hansen *et al.* (2005); Havilio and Wool (2007); Kim *et al.* (2006); Liu *et al.* (2006, 2008); Na and Paek (2009); Savitski *et al.* (2006); Searle *et al.* (2006); Tanner *et al.* (2005); Tsur *et al.* (2005)].

Reviews of protein mass spectrometry and the detection of PTM by mass spectrometry can be found in Domon and Aebersold (2006) and Witze *et al.* (2007). Briefly, a typical proteomic MS/MS experiment begins with an enzymatic digestion of proteins into peptides. For a complex mixture, it is common to simplify the

© The Author(s) 2011. Published by Oxford University Press.
This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/2.5), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

mixture by separating peptides based on their chemical properties, such as hydrophobicity using liquid chromatography, before they are ionized and injected into a mass spectrometer. Once in the spectrometer, peptides are grouped and isolated based on their mass-to-charge ratio (simply referred to as mass). Ideally, each group will contain one peptide variant. For each group, the peptides are further broken down by a fragmentation method, such as collision-induced dissociation (CID), to produce ion fragments. The mass spectrometer captures the ion fragmentation pattern for each group in a mass spectrum. The presence of PTMs shifts the mass of the corresponding ions, which changes the ion fragmentation pattern significantly. Specialized PTM search engines, like those discussed above, are used to decipher these ion patterns and map each mass spectrum to a peptide sequence that best explains it.

The use of real and decoy proteins is an established practice for estimating false predictions for MS/MS spectra analysis using database search approaches [Kall *et al.* (2007); Kislinger *et al.* (2003); Peng *et al.* (2003)], including blind PTM search methods (Liu *et al.*, 2008). Decoy proteins are generated by reversing the amino acid sequence of real proteins; this ensures that real and decoy proteins have the same distributions of amino acids and protein lengths. When using a protein reference database containing an equal number of real and decoy proteins, a random (false) peptide prediction (modified or unmodified) will have an equal likelihood of choosing a peptide from either a real protein or a decoy protein. This allows the number of decoy peptide hits as an estimate for false detection rate.

In practice, blind PTM search methods suffer from two major sources of error: sequence-dependent uncertainty in the modification position (residue position along the peptide sequence where the modification is deemed to occur) and mass inaccuracy for the modification mass. The fragmentation process is often incomplete and the presence of labile PTMs may interfere with this process (Mikesh *et al.*, 2006). Both issues combined result in MS/MS spectra missing peaks that in turn may lead to ambiguous or erroneous modification predictions. The presences of natural stable isotopes, such as carbon-13, in addition to electronic noise are major contributors to inaccurate mass measurements. This is more prominent in spectra generated from low mass resolution mass spectrometers (e.g. ion trap mass spectrometers), which are still commonly used in today's mass spectrometry studies. Figure 1 shows a diagrammatic representation of the search results obtained from applying the blind PTM search engine SIMS (Liu *et al.*, 2008) to a set of MS/MS spectra previously mapped to phosphopeptides (Beausoleil *et al.*, 2004). Enriched for phosphopeptides using a strong cation exchange-based method, the spectra from the complex peptide mixture in the original study were analyzed by a restricted PTM search method designed to look for phosphorylation and were validated manually. The same dataset has been used in benchmark experiments in previous PTM studies [Liu *et al.* (2008); Tanner *et al.* (2005)]. Phosphorylation is known to occur at ∼80 Da and primarily on the amino acid serine (S) and less frequently on threonine (T). Yet the results show, even for those observed peptide sequences that match to the original reference study, that many of their modification (delta) masses and modified amino acid sites deviate from this reference. A closer look (Fig. 1C) shows that many of the amino acids misplaced are a few residues away from their corresponding reference modification position. In a global-scale PTM survey, these issues can make distinguishing true PTM

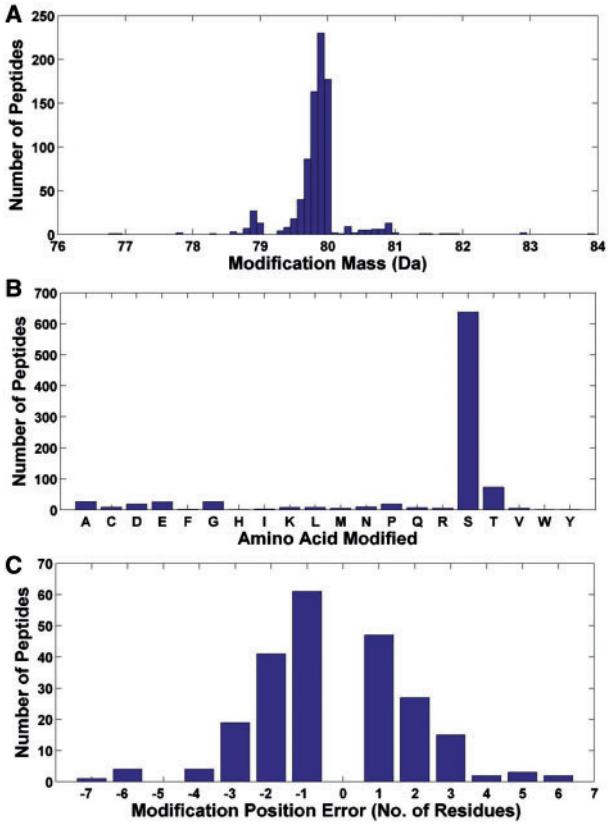

Fig. 1. Histograms of inputs to our algorithm [generated by SIMS (Liu *et al.*, 2008)] for spectra previously determined to be mapped to phosphopeptides (Beausoleil *et al.*, 2004). They show that the statistics for modification mass and modified amino acid deviate from the reference, which determined that the PTM (phosphorylation) occurs at ∼80 Da and on serine (S) and threonine (T). (**A**) The distribution of the measured modification mass. (**B**) Identified amino acids that deviate from S and T. (**C**) The distribution of the distance (in residues) from the identified amino acid to the reference for misplaced modifications; this demonstrates that identified modifications are generally only a few residues away from the reference.

matches from false detections non-trivial; therefore, identifying *bona fide* PTMs confidently remains difficult. While these errors can potentially be reduced by technological improvement in instrumentation (e.g. higher mass accuracy mass spectrometers or using alternate fragmentation mechanisms), we sought to develop an algorithm that can deconvolve errors associated with measuring masses and mapping of modification positions simultaneously to salvage both existing datasets and current experimental platforms.

These two sources of error were acknowledged by Tsur *et al.* (2005), who briefly described a heuristic approach to account for 'shadows' (modifications that are misplaced by a PTM search engine), and later by the same group in the PTM refinement algorithm PTMFinder (Tanner *et al.*, 2008). However, in the former method, their approach favors high abundance modified peptides, since it requires each peptide match to occur multiple times; discretizes observed modification masses, which introduces additional error with the mass measurements; and can handle only one type of error per peptide (namely either a modification mass error of exactly 1 Da or a modification position misplaced by exactly one

residue). The latter method, PTMFinder, takes a machine learning approach where it groups and reanalyzes spectra mapping to the same modified peptide sequence to produce for each spectrum a final peptide sequence with a modification mass and a modification position. This method also suffers from favoring high abundance modified peptides and discretizing observed modification masses. As we show later, it is not always the case that many spectra map to the same modified peptide in a typical genome-wide MS study. These restrictions limit the suitability of both methods' error correction approach to global PTM studies. We note that in addition to correcting for errors with modification mass and modification position estimations, PTMFinder can refine the peptide sequence and provides a *P*-value confidence score for both the reported peptide sequence and modification.

Here we introduce a novel generative probability model (PTMClust) that addresses the aforementioned problems encountered when using blind PTM search engines. It accomplishes a significant boost in PTM prediction accuracy and precision by modeling the hidden relationships between the compositions of amino acids in the peptide sequence, specifically the modification mass, the modification position and the identity of the modified amino acid. Our algorithm iterates between clustering modified peptides with similar modifications to form groups, which we call PTM groups, and finding the most likely modification mass and modification position for each peptide based on the grouping. Our method distinguishes itself from others by modeling modifications at the PTM level instead of at the individual peptide level. By rigorous benchmarking, we show that a number of learned PTM groups correspond to known PTMs and many reported modified peptides match to annotated modifications. In addition, our algorithm simultaneously considers PTMs occurring in either the middle or at the terminal ends of a peptide or protein, which provides additional information missed by blind PTM search techniques [Han *et al.* (2005); Liu *et al.* (2006, 2008); Searle *et al.* (2006); Tanner *et al.* (2005); Tsur *et al.* (2005)]. To ensure broad applicability, we have designed and optimized PTMClust to analyze PTM data generated from low resolution MS/MS spectra processed by popular blind PTM search engines, such as those generated from ion trap mass spectrometers.

2 METHODS

Our proposed algorithm PTMClust consists of a generative model, which captures the hidden relationship between the factors that influence the PTM mapping process, and an algorithm to infer the values of the hidden variables and parameters. It includes a background model to account for spurious data. The input to PTMClust is obtained using a blind PTM search method [e.g. SIMS or InsPecT Tanner *et al.* (2005)]. It consists of a list of modified peptides with the following attributes: peptide sequence, measured modification mass and estimated position of the modification along the peptide sequence (modification position). The output of PTMClust for each input peptide consists of a cluster assignment, a corrected modification position and a corrected modification mass. The identity of the modified amino acid for each peptide can be obtained from its peptide sequence and modification position.

A key component of our algorithm is the model selection method that selects the appropriate number of clusters by adjusting the model complexity 'control knob' α^b. Using the labels of real and decoy peptides, we defined rate of detection (RD) as the number of real peptides that are not assigned to the background model, divided by the total number of real peptides. Similarly, we defined rate of false detection (RFD) as the number of decoy peptides that are not assigned to the background model, divided by the total number of decoy peptides. A setting for α^b was chosen by weighing the tradeoff between the number of decoy peptides allowed and the number of real peptides detected, as described below.

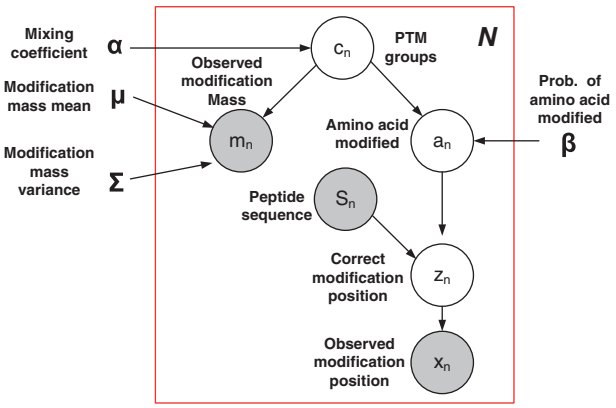

Fig. 2. A Bayesian network describing our generative model, using plate notation (box). The shaded nodes represent observed variables, the unshaded nodes represent latent variables and the variables outside the plate are model parameters. The model describes how the observed modification mass and modification position are generated. Given the type of PTM (PTM group), we can generate the observed modification mass as a noisy version of the modification mass mean, and select an amino acid to be modified. Given the peptide sequence, we can choose a position along it that matches the modified amino acid as the 'true' modification position. We can generate the observed modification position as a noisy version of the 'true' modification position. The plate notation indicates there are N copies of the model, one for each input peptide.

2.1 A generative model for finding PTM groups

By accounting for combinatorial interactions between hidden variables that play a role in the protein modification process, our generative probability model aims to describe how each PTM observation is generated. For a given PTM type (PTM group), the observed modification mass is assumed to be a noisy version of the expected (mean) modification mass, and the modified amino acid is chosen from a distribution over amino acids that may be modified in that type. For example, modifications occur primarily on serine (S) and threonine (T) for phosphorylation. For a given peptide, the true modification position is assumed to be chosen uniformly among occurrences of that amino acid in the peptide. Finally, the observed modification position is assumed to be a noisy version of the true position. Below, we described the components of our model: the probability of choosing each PTM type, the probability of choosing each amino acid to be the modified amino acid given the PTM type, the probability of the true modification position given the modified amino acid and the uncertainty in the observed modification mass and modification position. We then introduce an algorithm for learning the model parameters and inferring the hidden (latent) variables from the input data. Once a model is learned, we can refine the modification for each input peptide sequence by inferring its most likely PTM group, true modification mass and true modification position.

The structural relationships between the variables are shown by the Bayesian network in Figure 2. It describes the model for one input peptide and is repeated for N inputs, as indicated by the plate notation (box in the figure).

In our model, each input peptide sequence S_n, indexed by $n \in \{1,\ldots,N\}$ where N is the number of peptides in the dataset, has a corresponding discrete peptide length L_n, observed modification position $x_n \in \{1,\ldots,L_n\}$

and observed modification mass m_n. $S_n(j)$ is the amino acid in position j of the input sequence n. The total number of values $S_n(j)$ can take on is $A=24$, which includes the 20 naturally occurring amino acids and 4 special tokens indicating the beginning and end of proteins and peptides. The latent variable $c_n \in [1,..,K]$ denotes the unknown PTM group for peptide sequence n, where K is the number of PTM groups and will be adjusted depending on the desired false detection rate, as described later. The prior probability (mixing coefficient) for each PTM group is given as

$$P(c_n=k)=\alpha_k, \tag{1}$$

where it satisfies the constraints $\alpha_k \geq 0$ and $\sum_{k=1}^{K} \alpha_k = 1$, and is inferred from the data (see below for details).

The probability that the PTM occurs on amino acid $i \in \{1,\ldots,A\}$, given that the PTM group is k, is

$$P(a_n=i|c_n=k)=\beta_{ki}, \tag{2}$$

where the latent variable a_n denotes the true (unobserved) modified amino acid and the β's satisfy the constraints $\beta_{ki} \geq 0$ and $\sum_{i=1}^{A} \beta_{ki} = 1$, and is inferred from the data (see below for details).

Given the peptide sequence S_n and the modified amino acid a_n, each occurrence of that amino acid in the peptide sequence has equal probability of being the true (unobserved) modification position z_n. For completeness, our probabilistic model considers the likelihood of cases where an amino acid does not occur in S_n. To do so, we allowed for the event that the true PTM occurs outside of the given peptide sequence,[1] indicated by $z_n=0$, so that $z_n \in \{0,\ldots,L_n\}$. All other positions in the peptide sequence have zero probability of being the true modification position. This can be written as

$$P(z_n=j|a_n=i,S_n)=\begin{cases} \frac{1}{\delta_{ni}+1} & \text{if } S_n(j)=i, j\geq 1, \\ \frac{1}{\delta_{ni}+1} & \text{if } j=0, \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where δ_{ni} denotes the number of times amino acid i occurs in sequence n.

We modeled the modification position error $(x_n - z_n)$ between the observed modification position x_n and the true modification position z_n with a discrete probability distribution, given as

$$P(x_n|z_n=j)=\begin{cases} \phi(x_n-j) & \text{if } j>0, \\ \phi(L_n) & \text{if } j=0, \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where the likelihood function ϕ accounts for the modification position error. This likelihood function is shared across all PTM groups and is inferred from our empirical observation of the yeast PTM dataset as follows (see Section 3 for description of the dataset). We grouped the entries in the dataset by their peptide sequence and modification mass, allowing for mass differences of ± 2 Da. Then, we determined the average modification position for each group (rounded to the nearest position) and computed a histogram of the modification position error. In the above assignment, groups with less than three entries were removed. This threshold was chosen so that a reasonable number of points (1206) were available to estimate the likelihood function, while also filtering out false modified peptides. The frequency of peptides for each group size, shown in Supplementary Figure S1, exhibits a heavy-tail distribution where the majority of modified peptides have low counts. More than 48% of the entries have a group of size exactly three. The resulting likelihood distribution is shown in Figure 3.

Lastly, we accounted for the variation (noise) in the estimated modification mass by assuming the observed modification mass for each PTM group is normally distributed around the true modification mass, given as

$$P(m_n|c_n=k)=\frac{1}{\sqrt{2\pi\Sigma_k}}\exp\left(\frac{-(m_n-\mu_k)^2}{2\Sigma_k}\right), \tag{5}$$

where μ_k and Σ_k are the modification mass mean and variance for the k-th PTM group, and are inferred from the data (see below for details).

[1] $z_n=0$ is needed to avoid numerical issues since our algorithm considers each amino acid as a possible modification target.

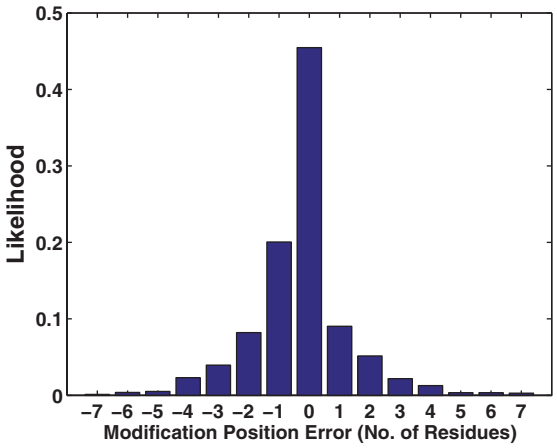

Fig. 3. Distribution of modification position error used by PTMClust. This empirical distribution was derived using yeast PTM data (Krogan *et al.*, 2006) analyzed with SIMS (Liu *et al.*, 2008). A positive (negative) modification position error indicates that the observed modification position is toward the C-terminus (N-terminus) of the expected modification position.

Combining the structure of the Bayesian network and the conditional distributions described above, we can write the joint distribution as

$$P(c,a,z,x,m|S,\theta)= \prod_{n=1}^{N}\left(P(c_n|\theta)P(m_n|c_n,\theta)P(a_n|c_n,\theta)P(z_n|a_n,S_n,\theta)P(x_n|z_n,\theta)\right), \tag{6}$$

where θ represents the model parameters (α_k, β_{ki}, μ_k and Σ_k).

The input data are noisy and may contain false positives and modified peptides that do not fit into proper PTM groups. To account for these spurious data points, we included an additional PTM group (background component) that acts as a garbage collection process (background model). In the background component, we assumed there is no specific relationship between the modification mass and modified amino acid. Formally, the background component has a fixed modification mass mean μ^b and variance Σ^b set to be equal to the mean and variance of the data. Additionally, it has a fixed uniform probability over the modified amino acid $\beta_a^b = \frac{1}{A}, \forall\, a=1,\ldots,A$, and a mixing coefficient α^b, which will be used to adjust model complexity (see below).

Inference and learning: The key step in our algorithm is to infer an optimal setting for latent variables and learn the model parameters. However, exact inference and learning of the above model is computationally intractable, because of non-linear relationship between latent variables and parameters. Instead, we used the EM algorithm [Dempster *et al.* (1977); McLachlan and Krishnan (1997)], which alternates between probabilistically filling in the latent variables c_n, a_n and z_n and estimating the parameters α_k, β_{ki}, μ_k and Σ_k. A detailed derivation of the EM algorithm for our model is provided as part of the Supplementary Material.

In the E-step, the posterior probabilities for iteration t and every peptide n are evaluated using the parameters from iteration $t-1$ by conditioning on the observed variables m_n and x_n in (6):

$$Q^{(t)}(c_n,a_n,z_n)=P(c_n,a_n,z_n|m_n,x_n,S_n,\theta^{t-1})= \frac{P(c_n,a_n,z_n,m_n,x_n|S_n,\theta^{t-1})}{\sum_{c_n}\sum_{a_n}\sum_{z_n}P(c_n,a_n,z_n,m_n,x_n|S_n,\theta^{t-1})}. \tag{7}$$

In the M-step, the parameters are reestimated by maximizing the expected complete log likelihood using the current posterior probabilities. This is done by taking the partial derivative of the expected complete log likelihood with respect to each parameter. Lagrangian terms are added to the expected

complete log likelihood to account for the constraints on probabilities α_k and β_{ki}. The updates for the parameters are as follow:

$$\mu_k = \frac{\sum_{n=1}^{N} Q^{(t)}(c_n = k) m_n}{\sum_{n=1}^{N} Q^{(t)}(c_n = k)}, \quad \Sigma_k = \frac{\sum_{n=1}^{N} Q^{(t)}(c_n = k) \left(m_n - \mu_k\right)^2}{\sum_{n=1}^{N} Q^{(t)}(c_n = k)},$$

$$\alpha_k = \frac{1}{N} \sum_{n=1}^{N} Q^{(t)}(c_n = k), \qquad \beta_{ki} = \frac{\sum_{n=1}^{N} Q^{(t)}(c_n = k, a_n = i)}{\sum_{n=1}^{N} Q^{(t)}(c_n = k)}. \tag{8}$$

At the end of each pair of E- and M-steps, we calculated the log likelihood and stop if the difference between the current and previous log likelihood divided by the current log likelihood is smaller than 10^{-5} (this stop criterion is chosen to ensure the EM algorithm terminates).

Recursive merge method for model selection: In our model, the only free parameter is the number of PTM groups (mixture components) K and the probability of the background component. We devised a recursive merge method, similar to the split and merge model selection methods,[2] that will effectively evaluate and identify the optimal free parameters that achieve a desired false detection rate.

Instead of adjusting K directly, we adjusted the mixing coefficient of the background component α^b, which represents the prior probability that a data point belongs to the background model, to adjust model complexity. We gradually increase this parameter and for each specific setting of α^b, our method infers the hidden variables, parameter settings and K. Using maximum likelihood estimation, as α^b increases (we used step size of 0.01), more and more of the loosely clustered peptide sequences are redistributed to other components, including the background component, and the number of non-background components is reduced by merging clusters (reducing K). This is accomplished by pruning away 'empty' components, where we define a component to be empty when it has less than or equal to one peptide sequence assigned to it. In effect, the non-background components are slowly merging with each other and the background component as α^b increases until the non-background components are empty and pruned away, which decreases the model complexity. In our algorithm, we started with a large value for K and a small value for α^b (0.01), and slowly merge the non-background components and the background component, pruning away any empty clusters, by increasing α^b each time. In total, we learn M models where M is the number of different α^b settings. We chose a single model (i.e. a specific setting for α^b) by analyzing the results from our model selection method using the measures RD and RFD. The choice of which model to use depends on the desired RD and RFD.

2.2 Synthetic PTM data generation

To compare PTMClust against standard clustering algorithms on the problem of finding correct groupings of modifications, we generated a synthetic PTM dataset that provides us with ground truth. The dataset consists of five subsets, each having 100 peptides randomly picked from the yeast protein complex dataset, described in Section 3. Here, each set of peptides is assigned to have one of the five arbitrarily chosen modified amino acids: aspartic acid (D), phenylalanine (F), histidine (H), leucine (L) and proline (P). The true modification position for each peptide was randomly chosen to be on one of the instances of the preassigned amino acid for that subset, and the modification positions used as input to the algorithms are set to a noisy version of the true modification positions. The noise (modification position error) added was chosen from a standard normal distribution (see Supplementary Fig. S2). Since the true modified amino acids are predefined, we can use them as labels to evaluate the performance of the algorithms.

The modification mass for each peptide was randomly generated to have Gaussian noise with a small variance (0.2) from the modification mass center for that particular set of peptides (see Supplementary Fig. S3). The distribution of modification masses was chosen to provide significant overlap in modification mass between adjacent sets. The modification mass centers were set to 40.0 Da for peptides with PTMs on D, 41.0 Da for peptides with PTMs on F, 42.0 Da for peptides with PTMs on H, 43.0 Da for peptides with PTMs on L and 44.0 Da for peptides with PTMs on P.

For k-means clustering and mixture of Gaussians (MOG), the format of each input peptide is a vector consisting of the modification mass and the distance between the modification position and the closest instance of each amino acid, i.e. a vector of size 21 with the modification mass as the first element and the 20 amino acid as the next 20 elements (alphabetically ordered). The distance between the true modification position and each amino acid is used to account for our expectation that each PTM occurs on a specific set of amino acids. To interpret the input distances, for each PTM group, the amino acids most likely to be the true modified amino acid will have a small variance and the amino acids that are unlikely to be the true modified amino acid will have a large variance.

[2]Our approach only makes use of merge steps.

3 RESULTS

We conducted two proof-of-concept experiments. First, we compared PTMClust to two standard clustering algorithms, k-means clustering (MacQueen, 1967) and a mixture of Gaussians (MOG), on a synthetically generated PTM dataset. Second, we benchmarked PTMClust against three state-of-the-art blind PTM search engines and a PTM refinement algorithm on a reference phosphopeptide dataset. To show its strengths, we applied our algorithm to process a yeast proteome dataset that contains multiple PTMs.

In our experiments, we initialized our algorithm with number of clusters $K = 150$ (except for the first proof-of-concept experiment); the prior probability of each PTM group $\alpha_k = \frac{1}{K}$, where $k \in \{1, \ldots, K\}$; the probability that the PTM occurs on amino acid $i \in \{1, \ldots, A\}$, given that the PTM group is k, $\beta_{ki} = \frac{1}{A}$; the modification mass mean for each PTM group μ_k to be uniformly distributed across the searched modification mass range (except for the first proof-of-concept experiment); and the variance of modification mass for each PTM group $\Sigma_k = 1$ and limited, during learning, to be no greater than 2. The assumption on the maximum value for Σ_k corresponds to our knowledge that for a PTM group to be physically relevant, it should have a well-defined modification mass.

3.1 Comparison of algorithms on synthetic data

Both the k-means clustering and MOG algorithms are standard methods used when faced with an unsupervised clustering problem. They perform effectively in many cases and are simple to understand and implement. Our algorithm improves upon them by explicitly modeling the hidden relationship between the modification mass, modified amino acid, peptide sequence and modification position. We evaluated the performance of PTMClust against these two algorithms using the synthetic PTM dataset described above, which provides us with ground truth labels for the true modified amino acids, modification positions, modification masses, identities of the PTM groups and cluster assignment for each peptide. The synthetic data was designed to have overlapping modifications, in terms of modification masses and modified amino acids, so that it is non-trivial to identify the PTM groups. The goal of this experiment is to evaluate how well the three algorithms perform with increasing complexity in the input data, so multiple datasets were generated with the number of PTM groups ranging from two to five. Details on how we generated the synthetic data can be found in Section 2.

To test whether each method could identify the PTMs, we fixed the number of clusters (K) for each algorithm. In fact, PTMClust can automatically determine the number of PTM clusters, but we deactivated this feature for this experiment. The initial parameter settings for the modification mass cluster centers, shared for all three algorithms, were initialized randomly within the range of modification masses in the input dataset. For MOG, the variance was initialized to 1 for modification mass (consistent with PTMClust) and distances between the observed modification position and the closest instance of each amino acid (same variance used to generate the data). Theoretically, a large initial variance for modification mass (e.g. 10 in this experiment) can result in data points being falsely assigned to one cluster, because many data points with different labels have the same observed modified amino acids. This has an effect much like our background model. At the other extreme, a small initial variance for modification mass (e.g. 0.1) can result in clusters that explain only a few data points that are near the initial cluster centers. However, due to the small size and simplicity of this dataset, we did not see significant problems in this regard for both MOG and PTMClust when we varied the initial modification mass variance (data not shown). We initialized the other parameters in PTMClust as discussed above. For each method, we performed 30 random restarts and picked the restart with the best joint log-likelihood. To do this, we learned k-means clustering by modifying the MOG, where, after each EM iteration, we set the probability between a data point and its closest cluster center to 1 and 0 for all other cluster centers to that data point.

Using $i \in \{1, \ldots, K\}$ to index each cluster, we evaluated the performance of the algorithms using a criterion that measures how well each ground truth modification was detected. For cluster i, we deemed the largest group of peptides with the same label assigned to it as true positives (TP_i) and all other peptides assigned to it as false positives (FP_i). To evaluate each algorithm, we calculated the correction rate (CR), which we defined as the difference between the total number of true positives and the total number of false positives divided by the total number peptides in the sample, given as

$$\mathrm{CR} = \frac{\sum_i \mathrm{TP}_i - \sum_i \mathrm{FP}_i}{N}, \tag{9}$$

where N is the total number of peptides in the sample. The CR is a measure of the fraction of PTM predictions that are expected to be not due to chance.

Figure 4 shows the result of applying the three algorithms on input datasets with varying number of PTM groups. It shows that our algorithm outperforms both k-means clustering and the MOG. The key observation is that PTMClust performed consistently well, while the performances of the other two algorithms exhibit a significant drop as the complexity of the dataset increases.

3.2 Benchmarking against phosphopeptide predictions

We next examined the abilities of our algorithm to identify PTM groups corresponding to *bona fide* PTMs, fine-tune observed modification masses and correct for misplaced modification positions. We chose to focus this analysis on phosphorylation because it plays a vital role in protein regulation for many different biological processes. As a result, it is well studied and annotated datasets are readily available. Using a dataset of ion trap MS/MS spectra (human HeLa cells) previously mapped and manually validated as phosphopeptides (Beausoleil *et al.*, 2004), we compared

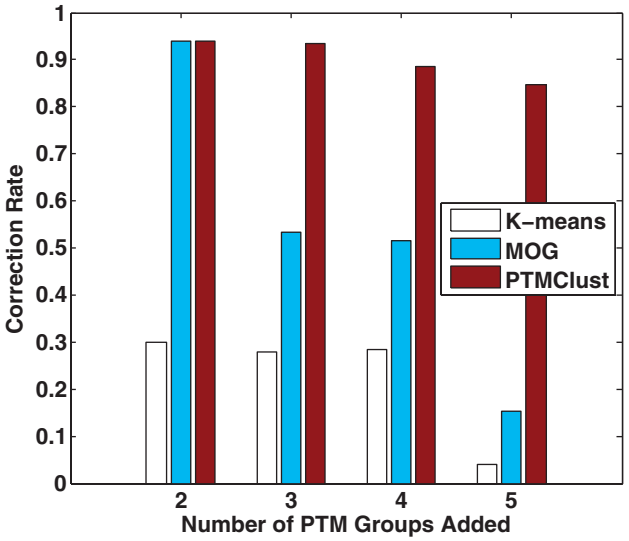

Fig. 4. A comparison of clustering algorithms on a synthetically generated dataset. It shows how each of the three methods, k-means clustering, a mixture of Gaussians (MOG) and PTMClust (our algorithm), performs as more sets of data points with different modifications are added (increasing complexity). Correction rate is a quality measure defined as the difference between the total true positives and the total false positives divided by the total number peptides in the sample; higher correction rate indicates better performance. The result shows PTMClust performs consistently well while the other two algorithms exhibit a significant drop as the complexity of the dataset increases.

the initial PTMs identified from three state-of-the-art blind PTM search engines, SIMS (Liu *et al.*, 2008), InsPecT [3] [Tanner *et al.* (2005); Tsur *et al.* (2005)] and MODmap (Kim *et al.*, 2006; Na and Paek, 2009) to the results after applying PTMClust on each of them individually. Additionally, we compared our results against those obtained by post-processing the result from InsPecT with the PTM refinement algorithm PTMFinder. Knowing the underlying peptide sequence and PTM for each spectrum is critical to allow us to compare the effectiveness of PTMClust.

The dataset consists of 1655 spectra but we focused only on the 1340 spectra mapped and curated as singly modified phosphopeptides (SIMS, InsPecT and PTMClust are limited to one modification per peptide prediction). When searching the spectra, we used the default settings optimized for ion trap instruments for InsPecT, PTMFinder and MODmap, and reference settings described in Liu *et al.* (2008) for SIMS. To simulate a true blind PTM search, an empty list of known PTMs was passed into PTMFinder, which ensures that any corrections made by PTMFinder are not influenced by prior knowledge of known PTMs. Due to the long search time required, which scales linearly with the size of the reference database, a common practice employed by blind PTM search engines [Liu *et al.* (2008); Tanner *et al.* (2005); Tsur *et al.* (2005)] is a two-pass approach (Craig and Beavis, 2003), where a reduced database is generated by filtering the reference database for proteins that are found by an initial analysis of the spectra not considering modifications. The human database from

[3]The MS-alignment algorithm (Tsur *et al.*, 2005), which is part of the InsPecT program, was used to perform blind PTM search.

Table 1. Results for SIMS, InsPecT, MODmap and PTMFinder with and without application of our method, PTMClust

	No. of correct modification position matches (% improvement over base algorithm)	No. of misplaced modification position matches (% improvement over base algorithm)	Total correct peptide sequence matches
SIMS	685	267	952
SIMS with PTMCIust	791 (∼15%)	161 (∼40%)	952
InsPect	621	239	860
InsPect with PTMCIust	712 (∼15%)	148 (∼38%)	860
PTMFinder	620	242	862
PTMFinderwith PTMCIust	711 (∼15%)	151 (∼38%)	862
MODmap	97	28	125
MODmap with PTMCIust	108 (∼11%)	17 (∼39%)	125

A reference set of MS/MS spectra previously mapped to phosphopeptides (Beausoleil *et al.*, 2004) was analyzed by SIMS (Liu *et al.*, 2008), InsPecT (Tanner *et al.*, 2005; Tsur *et al.*, 2005), MODmap (Na and Paek, 2009), and InsPecT followed by PTMFinder (Tanner *et al.*, 2008), a PTM refinement method. Using the reference peptide sequences and modifications as the truth, the table shows the number of correct peptide sequence matches, and the correct and misplaced modification positions before and after applying PTMClust (our algorithm) to the output from the four methods. PTMClust was able to correct for a significant portion of the modification position errors made by the four methods and the improvements are consistent across different methods. Furthermore, PTMClust is able to correct errors that PTMFinder missed, significantly outperforming it in terms of refining PTMs.

the National Center for Biotechnology Information was used as the initial reference database in this two-pass approach. A reduced reference database of 1827 real proteins appended with the same number of decoy proteins and a common modification range [−20, 300] Da was used for all algorithms.

Among the 952 outputted peptide sequences matching to the reference for SIMS, 267 had their modification misplaced. Similarly, InsPecT result matched 860 reference sequences but misplaced 239 modification positions. Using the default settings MODmap produced a peptide sequence for only 157 spectra, which resulted in 125 peptide sequences matching to the reference with 28 of those having misplaced modification positions. Lastly, post-processing InsPecT outputs with PTMFinder produced a change to five peptide predictions: two peptide sequence changes resulted in a match to the reference but both cases failed to identify the correct modification position; an incorrect modification position change on a previously correct prediction; and two incorrect modification position changes on previously mismatch modification positions (i.e. no positive effect). In summary, we observed 242 of the 862 peptide sequences matching to the reference with a misplaced modification position.

We initialized our algorithm as described above. Weighting the tradeoff between maximizing RD and minimizing RFD, we settled on a model complexity setting of $\alpha^b = 0.90$, which resulted in a RD of 0.76 and a RFD of 0.27 for SIMS; $\alpha^b = 0.94$ with a RD of 0.72 and a RFD of 0.34 for InsPecT, $\alpha^b = 0.94$ with a RD of 0.72 and a RFD of 0.34 for PTMFinder; and $\alpha^b = 0.45$ with a RD of 0.701 and a RFD of 0 for MODmap.

As shown in Table 1, PTMClust was able to correct a significant portion of the misplaced modifications identified by SIMS, InsPecT, MODmap and PTMFinder. Across the board, PTMClust performed consistently well. More specifically, for SIMS, PTMClust decreased the number of misplaced modifications by ∼40% (106 fewer misplaced modification positions) to produce 791 correct matches, an increase of ∼15%. Similarity, for InsPecT, our algorithm reduced the number of misplaced modification positions by ∼38% (91 fewer modification position misplacement) to produce 712 correct predictions, an increase of ∼15%. PTMClust obtained improvement on par with others for MODmap with a ∼39% decrease in the number of misplaced modifications (11 fewer misplaced modification positions) and a ∼11% increase of correct predictions (108). Given PTMFinder had little effect on the result from InsPecT for this dataset, we experienced similar improvements to those for InsPecT where we obtained ∼38% (91) fewer modification position misplacement to produce 711 correct predictions, an increase of ∼15%.

Importantly, a breakdown of the results show that our algorithm made very few mistakes (19 for SIMS, 26 for InsPecT, 1 for MODmap and 26 for PTMFinder) where it incorrectly changed modification positions that were correctly identified by SIMS, InsPecT, MODmap or PTMFinder, while making a large number of improvements (125 for SIMS, 117 for InsPecT, 12 for MODmap and 117 for PTMFinder). A closer examination of the models learned (for all four algorithms) shows that the majority of the reference phosphopeptides were assigned to a PTM group with modification mass ∼79.87 Da and high likelihood for S (∼0.94) and T (∼0.06): this corresponded correctly to our knowledge about phosphorylation. A listing of the search results from all algorithms are provided in Supplementary Table S1.

Next, we examined the overlap between the results from SIMS, InsPecT and MODmap, and the corrected results after applying our algorithm. PTMFinder is omitted here since its result is nearly identical to InsPecT. It has been reported that a significant portion of the results from SIMS and InsPecT do not match (Liu *et al.*, 2008), and this observation is widely believed to be true for many pairs of blind PTM search methods. Our analysis shows that many of the mismatches are due to incorrect modification position assignments: 229 of the 790 spectra that both SIMS and InsPecT, mapped to the same peptide sequence have mismatched modification position. After post-processing with our algorithm, ∼41% (93) of the mismatches were corrected, which significantly improved the overlap between the results from the two algorithms. We observed similar improvements when we include MODmap in the analysis: 25 of 106 spectra have mismatching modification position with ∼44% (11) improvement between InsPect and MODmap, and 25 of 98 spectra have mismatching modification position with ∼48% (12) improvement between all three algorithms (SIMS, InsPecT and MODmap). Due to the small number of observed mismatched modification positions among the overlaps between SIMS and MODmap (14 of 119 matching peptide sequences (∼12%)), we did not observe any improvement post-processed with PTMClust. PTMClust consistently, with the exception of SIMS versus MODmap, is able to improve on the overlap of the identified modified peptides between the different algorithms. These results provide additional evidence that our algorithm is producing sensible results.

3.3 Large-scale PTM analysis of yeast proteome

To test its versatility in detecting diverse PTM groups in a more complex biological context, we next applied PTMClust to analyze a large-scale PTM dataset taken from analyses of yeast protein complexes (LC-MS/MS spectra only) (Krogan *et al.*, 2006) using

Table 2. Summary of known modifications in the yeast proteome dataset

PTM	PTMClust		SIMS	
	Known PTM sites (% improvement over SIMS)	Peptides with known PTM sites (% improvement over SIMS)	Known PTM sites	Peptides with known PTM sites
Phosphorylation	66 (∼8%)	115 (∼15%)	61	100
Acetylation	9 (∼13%)	75 (∼42%)	8	72
Cysteine oxidation (Cysteine sulfinic acid)	1 (∼0%)	7 (∼17%)	1	6
Others	5 (∼0%)	35 (∼0%)	5	35
Total	81 (∼8%)	232 (∼9%)	75	213

The known set of modifications was taken from Uniprot (Release 2010_11). We matched the sets of modified peptides produced by SIMS and post-processed with PTMClust to the set of known yeast modification sites. The results show PTMClust is able to identify and refine PTMs in a complex dataset.

SIMS. Briefly, the yeast dataset consists of over 2 million ion trap MS/MS spectra of which 19 560 putatively modified peptides (estimated false discovery rate of 4.3% based on the number of decoy peptides identified) were identified by SIMS with modification range [0, 200] Da. In this experiment, we used a model complexity setting of $\alpha^b = 0.92$, which resulted in a RD of 0.58 and a RFD of 0.16.

Analysis with our algorithm was able to identify 121 PTM groups. The complete list of modified peptide predictions are provided in Supplementary Table S2 and a summary of the frequent PTMs observed are listed in Supplementary Table S3. Within the list of PTM groups are naturally occurring PTMs such as phosphorylation, acetylation and oxidation, and *in vitro* artificial modifications such as oxidized methionine and sodium/potassium salt adduct. Among them are many modified peptides not previously annotated to contain these modifications. In addition to those listed, there are a number of putative novel modifications types that have not been previously reported.

To validate that our approach is generally applicable to any PTM, we compare the results before and after applying PTMClust to known modified yeast proteins taken from the Uniprot Knowledgebase (Release 2010_11). A breakdown of our findings is shown in Table 2. For this analysis, we determined the modification sites (positions in the corresponding protein where the modifications occur) for each modified peptide in our results and matched them against the list of known modification sites from Uniprot. We found 213 modified peptide matches consisting of 75 unique known modification sites before and 232 modified peptide matches and 81 unique modification sites after applying PTMClust, for an overall improvement of ∼9%. In addition to phosphorylation, PTMClust was able to detect and refine other known PTMs, such as acetylation and cysteine oxidation (cysteine sulfinic acid).

A novel feature of PTMClust is the ability to consider modifications at the ends of proteins and peptides. Examples are modified peptides that exhibit N-terminus glycosylation (modification mass ∼162 Da) (Tanner and Lehle, 1987). This modification is a PTM that adds sugar molecules to proteins and is known to play a vital role in proteolytic resistance, protein solubility, stability, local structure, lifetime in circulation and immunogenicity (Lis and Sharon, 1993). Although the original distribution of modified amino acids did not show any pattern with modifications mainly found on alanine (A), isoleucine (I), leucine (L) and valine (V), PTMClust was able to recognize that all the modifications occur close to the N-terminus of the peptide. This observation is unlikely to be explained by simple amino acid substitutions or artifacts since they have a similar initial modification mass and their modifications were initially observed to occur on different amino acids. In terms of where the modifications occur, they all share the commonality that their modifications occur near the N-terminus, which PTMClust is able to capture.

4 CONCLUSION

Accurate identification of protein modifications in protein sequences is a critical first step in any PTM study, and thus it may benefit the utility of proteomic profiling to address research problems in basic biology, as well as biomarker discovery and drug development in the clinical domain. A recently developed approach for PTM discovery is to analyze MS/MS data using a blind PTM search method. Genome-wide studies using SIMS, InsPecT and other blind PTM search engines have reported numerous PTM candidates (Han *et al.*, 2005; Liu *et al.*, 2006, 2008; Searle *et al.*, 2006; Tanner *et al.*, 2005; Tsur *et al.*, 2005). However, these search methods suffer from two problems: mass measurement inaccuracy and uncertainty in predicting modification positions, which limit their accuracy and precision. We developed a novel machine learning algorithm called PTMClust for post-processing the results of blind PTM search engines and improving prediction performance, by simultaneously identifying the positions of the most likely modified amino acids and grouping peptides with similar modification mass and modified amino acid side chains. We demonstrated that PTMClust improved on both true positives (correct modification position predictions) and false positives (misplaces modification positions) when applied to the outputs of SIMS, InsPecT, MODmap, and InsPecT post-processed with PTMFinder, a PTM refinement algorithm. The results showed that our algorithm was able to detect a number of previously annotated naturally occurring and artificially induced PTMs, most notably phosphorylation, but also acetylation (lysine), oxidization (methionine) and even the formation of non-covalent adducts (e.g. sodium/potassium salts). In addition, our algorithm facilitates the identification of terminal modifications, which is a feature not currently found in common blind PTM search engines. To our knowledge, this algorithm is the first technique that systematically and objectively addresses sequence-dependent variation in the PTM dataset at the PTM level, which can improve the reliability of individual PTM identification.

For the task of PTM refinement, we have shown that PTMClust outperforms PTMFinder on the dataset of phosphopeptides. PTMFinder failed here because only ∼4% (69) of spectra map to modified peptides already detected in the dataset. This is expected since it is known that only a small portion of spectra in an experiment map to modified peptides (Liu *et al.*, 2008; Tanner *et al.*, 2008) and current MS experimental protocols for genome-wide studies are designed to sample as many different peptides as possible (through the use of an exclusion list in the mass spectrometer). Moreover, many instances of the same modified peptide either share the same modification position (for both correct and misplaced cases) or have vastly different modification positions that point to different phosphorylation sites in the peptide. The former can be explained

since some missing peaks due to incomplete fragmentation are generally not detected for different instances of a peptide and blind PTM search algorithms produce the same modified peptide prediction for similar looking spectra. For blind PTM searches, PTMFinder only works when there are multiple instances of the same modified peptide. On the other hand, our method, PTMClust is successful even for low abundance modified peptides as long as there are multiple instances of the same underlying PTM.

We believe PTMClust is complementary to and can benefit from technological improvements in mass spectrometer instrumentation. Two of the more prominent advancements in recent years are high mass accuracy and alternate fragmentation mechanisms. For high mass accuracy mass spectrometers, such as an Orbitrap (Hu *et al.*, 2005), mass errors are significantly reduced and peak intensity signal-to-noise ratios are greatly improved in the observed MS/MS spectra if they are acquired in high resolution mode. However, currently the common practice for experiments using Orbitrap is to generate MS/MS spectra in low resolution mode due to its higher scan rate. Distinguishing features of electron-transfer dissociation (ETD), a recently introduced fragmentation mechanism, are its abilities to preserve the localization of labile PTMs and produce near complete ion fragmentation (Mikesh *et al.*, 2006). However, it is limited to peptides with charge state greater than +2 and can identify significantly less peptides than other fragmentation methods. To address these issues, a current approach is to use a mass spectrometer equipped with ETD and another fragmentation method, such as CID, and switch between them depending on the properties of the peptides to be fragmented (Hogan *et al.*, 2005; Molina *et al.*, 2008). These technological advancements can help reduce the issue of misplaced modification position due to missing peaks and noisy spectra but can still benefit from using PTMClust in its analysis. Given input data with higher mass resolution and fewer misplaced modifications due to cleaner ion fragmentation signals, PTMClust can improve upon its abilities to refine modification positions and find meaningful PTM groups. Our algorithm could be used to analyze modified peptides processed from spectra generated by both low- and high-resolution mass spectrometers using a variety of fragmentation methods [e.g. CID, ETD and high-energy collision dissociation (HCD)].

Our current version of PTMClust has a small number of weaknesses, which can potentially be solved. Although RD and RFD can provide a confidence estimate for the overall result, we have not explored how our algorithm can be used to provide a confidence score per peptide and per modification, which is a feature that can be found in PTMFinder. However, since our method is based on a probability model, such a score can be computed. Additionally, our method cannot detect PTMs that occur only once in the data, since multiple instances are needed for model building. Moreover, our method is currently unable to handle multiple modifications per input sequence. Lastly, depending on the mass resolution in the input data, PTM groups identified by our algorithm may contain multiple PTMs with similar modification mass. Despite these limitations, we were able to obtain results that significantly exceeded the performance of the state of the art. A noteworthy extension would be to combine blind search algorithms with our algorithm to jointly analyze MS data for modified and unmodified peptides. This would enable the algorithm to take into account ion fragmentation patterns directly. One advantage to this extension is that it might be able to handle cases where multiple, equally likely modification positions are present in the peptide but the modification was originally misplaced. An example would be multiple serines appearing side by side in the peptide and the modification (phosphorylation) having been misplaced on one of the serines. We believe that the utility, reliability and generality of our approach in refining PTMs indicate that our probability model and extensions of it can be used to produce higher-quality datasets and facilitate novel biological discoveries in the future.

ACKNOWLEDGEMENTS

We thank Johannes Hewel, Yoseph Barash and Leo Lee for valuable conversations and Vincent Fong for his help with collecting the experimental data and generating the SIMS and InsPecT results.

Funding: This research was supported by Steacie, Discovery and NET grants from the Natural Sciences and Engineering Research Council of Canada (to B.J.F.); an Operating Grant from the Canadian Institutes of Health Research (to B.J.F.); and two Operating Grants from the Ontario Ministry of Research and Innovation, (to A.E. and B.J.F.).

Conflict of Interest: none declared.

REFERENCES

Baliban,R.C. *et al.* (2010) A novel approach for untargeted post-translational modification identification using integer linear optimization and tandem mass spectrometry. *Mol. Cell. Proteomics*, **9**, 764–769.

Baumgartner,C. *et al.* (2008) Semop: a new computational strategy for the unrestricted search for modified peptides using lc-ms/ms data. *J. Proteome Res.*, **7**, 4199–4208.

Beausoleil,S. *et al.* (2004) Large-scale characterization of hela cell nuclear phosphoproteins. *Proc. Natl Acad. Sci. USA*, **101**, 12130–12135.

Chen,Y. *et al.* (2009) Ptmap - a sequence alignment software for unrestricted, accurate, and full-spectrum identification of post-translational modification sites. *Proc. Natl Acad. Sci. USA*, **106**, 761–766.

Craig,R. and Beavis,R.C. (2003) A method for reducing the time required to match protein sequences with tandem mass spectra. *Rapid Commun. Mass Spectrom.*, **17**, 2310–2316.

Craig,R. and Beavis,R. (2004) Tandem: matching proteins with tandem mass spectra. *Bioinformatics*, **20**, 1466–1467.

Dempster,A. *et al.* (1977) Maximum likelihood from incomplete data via the em algorithm. *J. R. Stat. Soc.*, **39**, 1–38.

Domon,B. and Aebersold,R. (2006) Mass spectrometry and protein analysis. *Science*, **312**, 212–217.

Eng,J. *et al.* (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.

Han,Y. *et al.* (2005) Spider: software for protein identification from sequence tags with de novo sequencing error. *J. Bioinformatics Comput. Biol.*, **3**, 697–716.

Hansen,B.T. *et al.* (2005) P-mod: An algorithm and software to map modifications to peptide sequences using tandem ms data. *J. Proteome Res.*, **4**, 358–368.

Havilio,M. and Wool,A. (2007) Large-scale unrestricted identification of post-translation modifications using tandem mass spectrometry. *Anal. Chem.*, **79**, 1362–1368.

Hogan,J. *et al.* (2005) Complementary structural information from a tryptic n-linked glycopeptide via electron transfer ion/ion reactions and collision induced dissociation. *J. Proteome Res.*, **4**, 628–632.

Hu,Q. *et al.* (2005) The orbitrap: a new mass spectrometer. *J. Mass Spectrom.*, **40**, 430–443.

Kall,L. *et al.* (2007) Assigning significance to peptides identified by tandem mass spectrometry using decoy databases. *J. Proteome Res.*, **7**, 29–34.

Kim,S. *et al.* (2006) Modi: a powerful and convenient web server for identifying multiple post-translational peptide modifications from tandem mass spectra. *Nucleic Acids Res.*, **34**, W258–W263.

Kislinger,T. *et al.* (2003) Prism: a generic large-scale proteomics investigation strategy for mammals. *Mol. Cell. Proteomics*, **2**, 96–106.

Krogan,N. *et al.* (2006) Global landscape of protein complexes in the yeast saccharomyces cerevisiae. *Nature*, **440**, 637–643.

Lis,H. and Sharon,N. (1993) Protein glycosylation. structural and functional aspects. *Eur. J. Biochem.*, **218**, 1–27.

Liu,C. *et al.* (2006) Peptide sequence tag-based blind identification of post-translational modifications with point process model. *Bioinformatics*, **22**, e307–e313.

Liu,J. *et al.* (2008) Sequential interval motif search: unrestricted database surveys of global ms/ms data sets for detection of putative post-translational modifications. *Anal. Chem.*, **18**, 7849–7854.

MacQueen,J. (1967) Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*. Vol. 1, Berkeley, CA, pp. 281–297.

Matthiesen,R. *et al.* (2005) Vems 3.0: Algorithm and computational tools for tandem mass spectromtry based identification of post-translational modifications in proteins. *J. Proteome Res.*, **4**, 2338–2347.

McLachlan,G. and Krishnan,T. (1997) *The EM Algorithm and its Extensions*. Wiley, San Francisco, CA.

Mikesh,L. *et al.* (2006). The utility of etd mass spectrometry in proteomic analysis. *Biochim. Biophys. Acta*, **1764**, 1811–1822.

Molina,H. *et al.* (2008) Comprehensive comparison of collision induced dissociation and electron transfer dissociation. *Anal. Chem.*, **80**, 4825–4835.

Na,S. and Paek,E. (2009) Prediction of novel modifications by unrestrictive search of tandem mass spectra. *J. Proteome Res.*, **8**, 4418–4427.

Peng,J. *et al.* (2003) Evaluation of multidimensional chromatography coupled with tandem mass spectrometry (lc/lc-ms/ms) for large-scale protein analysis: the yeast proteome. *J. R. Stat. Soc.*, **2**, 43–50.

Perkins,D. *et al.* (1999) Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, **20**, 2551–2567.

Savitski,M.M. *et al.* (2006) Modificomb, a new proteomic tool for mapping substoichiometric post-translational modifications, finding novel types of modifications, and fingerprinting complex protein mixtures. *Mol. Cell. Proteomics*, **5**, 935–948.

Searle,B. *et al.* (2006) Identification of protein modifications using ms/ms de novo sequencing and the opensea alignment algorithm. *J. Proteome Res.*, **4**, 546–554.

Tanner,W. and Lehle,L. (1987) Protein glycosylation in yeast. *Biochim. Biophys. Acta*, **906**, 88–99.

Tanner,S. *et al.* (2005) Inspect: identification of posttranslationally modified peptides from tandem mass spectra. *Anal. Chem.*, **77**, 4626–4639.

Tanner,S. *et al.* (2008) Accurate annotation of peptide modifications through unrestrictive database search. *J. Proteome Res.*, **7**, 170–181.

Tsur,D. *et al.* (2005) Identification of post-translational modifications by blind search of mass spectra. *Nat. Biotechnol.*, **23**, 1562–1567.

Witze,E.S. *et al.* (2007) Mapping protein post-translational modifications with mass spectrometry. *Nat. Methods*, **4**, 798–806.