BIOINFORMATICS ORIGINAL PAPER

Vol. 29 no. 7 2013, pages 821–829
doi:10.1093/bioinformatics/btt056

Genome analysis

Advance Access publication February 17, 2013

Non-parametric Bayesian approach to post-translational modification refinement of predictions from tandem mass spectrometry

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

[1]Department of Computer Science, University of Toronto, Toronto, M5S 2E4, [2]Probabilistic and Statistical Inference Group, University of Toronto, Toronto, M5S 3G4, [3]Banting and Best Department of Medical Research, [4]Donnelly Centre for Cellular and Biomolecular Research, University of Toronto, Toronto, M5S 3E1 and [5]Department of Electrical and Computer Engineering, University of Toronto, Toronto, M5S 3G4, Canada

Associate Editor: Alfonso Valencia

ABSTRACT

Motivation: Tandem mass spectrometry (MS/MS) is a dominant approach for large-scale high-throughput post-translational modification (PTM) profiling. Although current state-of-the-art blind PTM spectral analysis algorithms can predict thousands of modified peptides (PTM predictions) in an MS/MS experiment, a significant percentage of these predictions have inaccurate modification mass estimates and false modification site assignments. This problem can be addressed by post-processing the PTM predictions with a PTM refinement algorithm. We developed a novel PTM refinement algorithm, *i*PTMClust, which extends a recently introduced PTM refinement algorithm PTMClust and uses a non-parametric Bayesian model to better account for uncertainties in the quantity and identity of PTMs in the input data. The use of this new modeling approach enables *i*PTMClust to provide a confidence score per modification site that allows fine-tuning and interpreting resulting PTM predictions.

Results: The primary goal behind *i*PTMClust is to improve the quality of the PTM predictions. First, to demonstrate that iPTMClust produces sensible and accurate cluster assignments, we compare it with k-means clustering, mixtures of Gaussians (MOG) and PTMClust on a synthetically generated PTM dataset. Second, in two separate benchmark experiments using PTM data taken from a phosphopeptide and a yeast proteome study, we show that iPTMClust outperforms state-of-the-art PTM prediction and refinement algorithms, including PTMClust. Finally, we illustrate the general applicability of our new approach on a set of human chromatin protein complex data, where we are able to identify putative novel modified peptides and modification sites that may be involved in the formation and regulation of protein complexes. Our method facilitates accurate PTM profiling, which is an important step in understanding the mechanisms behind many biological processes and should be an integral part of any proteomic study.

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

Contact: frey@psi.utoronto.ca

Supplementary information: Supplementary data are available at *Bioinformatics* online

Received on September 18, 2012; revised on January 10, 2013; accepted on January 30, 2013

*To whom correspondence should be addressed.

1 INTRODUCTION

Post-translational modifications (PTMs) are known to play a vital role in the cell and are instrumental in many disease-related studies (Lehninger *et al.*, 1993). A core task in studies involving PTMs is PTM prediction, i.e. identification of peptide sequences and PTMs associated with each modified peptide within a biological sample. A preferred experimental procedure for PTM prediction is tandem mass spectrometry (MS/MS) followed by an analysis with a 'blind' PTM search engine. [See (Cantin and Yates, 2004; Domon and Aebersold, 2006) for reviews]. Blind PTM search engines are commonly used because they can detect both known and novel PTMs. However, PTM predictions produced by blind PTM search engines contain inaccurate modification masses and incorrect modification positions (Keller *et al.*, 2002; Ramakrishnan *et al.*, 2009a,b). The fragmentation process is often incomplete and the presence of labile PTMs may interfere with this process (Mikesh *et al.*, 2006). These issues result in spectra missing peaks that in turn may lead to ambiguous or erroneous modification predictions. Naturally occurring stable isotopes, such as carbon-13, and electronic noise contribute to inaccurate mass measurements. These issues are more prominent in spectra generated from low mass-resolution spectrometers (e.g. ion trap mass spectrometers), which are still commonly used in today's MS studies. Therefore, it is prudent to incorporate PTM refinement as part of a PTM prediction pipeline, as it can significantly improve the quality of PTM predictions. Previous studies demonstrate that post-processing greatly improves the number of positive predictions while reducing the amount of false PTM assignments (Chung *et al.*, 2011; Tanner *et al.*, 2008).

PTM refinement can be classified into two types of approach: one that scores the localization of PTMs and one that refines observed modification masses and modification positions. The first type provides a way to evaluate the quality of predicted modification sites from PTM search engines. The two main strategies for scoring the reliability of modification site localizations are: (i) to calculate the probability that a peak responsible for the site determination is matched at random and (ii) to compute the search engine score difference between predictions with varying site localizations. Methods that use the former strategy include A-score (Beausoleil *et al.*, 2006), PTM Score (embedded in MaxQuant and Andromeda) (Olsen *et al.*, 2006), the

Phosphorylation Localization Score (PLS) in InsPecT (Albuquerque *et al*., 2008), SLoMo (Bailey *et al*., 2009), Phosphinator (Phanstiel *et al*., 2011), PhosphoRS (Taus *et al*., 2011). Examples of the latter scoring strategy are Mascot Delta Score (Savitski *et al*., 2011), the site localization in peptide (SLIP) score in Protein Prospector (Baker *et al*., 2011) and the variable modification localization score in Spectrum Mill (Agilent, 2005). A review of the different modification site scoring localization methods is provided in (Chalkley and Clauser, 2012).

PTM refinement using a modification site localization scoring algorithm can be achieved by reassigning the modification position to the highest scoring position for each modified peptide. However, modification site localization scoring methods are limited when used for general PTM refinement owing to the following three reasons. First, a predefined list of PTMs is required for these scoring methods. Second, these scoring methods assume that input-predicted modification masses are error-free and are mapped precisely to one of the PTMs in the predefined list. Lastly, most of these scoring methods are designed to score only phosphorylated predictions. Consequently, modification site localization scoring methods are ill-suited to analyze PTM datasets generated from blind PTM search engines.

The second type of PTM refinement approach is to refine both observed modification masses and modification positions. Two recently published algorithms that use this type of PTM refinement method are PTMFinder (Tanner *et al*., 2008) and our previous algorithm, PTMClust (Chung *et al*., 2011). PTMFinder takes a peptide-level approach to PTM refinement, 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. Hence, refinement using this method is limited to modified peptides that occur multiple times in the same dataset. As shown in a recent study, a modified peptide is rarely found more than three times even for a large-scale, genome-wide experiment (Chung *et al*., 2011). Furthermore, PTMFinder suffers from favoring high abundance modified peptides and discretizing observed modification masses. In contrast, PTMClust accounts for errors of the observed modification masses and modification positions at the PTM level, and overcomes several issues that PTMFinder has.

The principle behind, and the distinguishing feature of, PTMClust is modeling modifications at the PTM level instead of at the peptide level. While peptide-level modeling can benefit from correcting low-level errors, the PTMClust approach has the advantage of accounting for low abundance modified peptides because other peptides with the same underlying PTM can help identify the correct but unknown modification mass and modified amino acid. PTMClust uses a generative model to capture the hidden relationship between factors influencing the PTM mapping process. It uses the expectation-maximization (EM) algorithm and a modified version of the split and merge model selection method to learn and infer an optimal parameter setting for the model. As part of the model selection procedure, a range of models are learned by adjusting a model complexity parameter, and the final model is selected by weighting the trade-off between false positives (determined using decoy peptides) and correct (real) peptides detected. The resulting PTM predictions are of higher quality than those taken from existing blind PTM search engines alone and those post-analyzed with PTMFinder (Chung *et al*., 2011).

Here, we address some limitations of PTMClust: (i) the use of a greedy method for selecting the number of PTM clusters, (ii) the need for manual parameter tuning, and (iii) the lack of a confidence score per modification position. We overcame these limitations by extending the PTMClust model to allow for an unbounded number of mixture components that can account for uncertainties in the quantity and identity of PTMs in the input data. The model makes use of an infinite non-parametric mixture model, so we refer to it as infinite-PTMClust or *i*PTMClust. This extension parallels the conversion from a finite to an infinite standard mixture model (IMM), but the complex nature of the underlying PTMClust model makes this extension non-trivial. We derived and implemented a Gibbs sampling algorithm (Bishop, 2006; Geman and Geman, 1984) and a split-and-merge Metropolis-Hastings algorithm (Jain and Neal, 2000), which enable *i*PTMClust to efficiently infer the groupings of input modified peptides and refine the peptides' modification masses and modification positions. *i*PTMClust achieves the following benefits: (i) it outperforms PTMClust and other PTM refinement algorithms, (ii) it provides a more highly automated model selection method that does not require manual parameter tuning, and (iii) it outputs modification position-level confidence scores that users can use to assess the quality of the result and further refine their analyses. In a series of benchmark experiments on both synthetic and real-world phosphopeptide datasets, we show that *i*PTMClust outperforms PTMClust and other state-of-the-art PTM prediction and PTM refinement algorithms. To ensure broad applicability, we have designed and optimized *i*PTMClust to analyze PTM data generated from both low- and high-resolution MS/MS spectra processed by popular blind PTM search engines. As with PTMClust, the input to *i*PTMClust is a list of PTM predictions consisting of the peptide sequence, modification position and modification mass.

2 METHODS

Similar to PTMClust, *i*PTMClust is based on a generative probability model that describes a process in which observed modified peptides can be generated by modeling interactions between hidden variables that play a role in the protein modification process. Given an observed modification, we assume it comes from one of many PTMs. However, the number and identity of these PTMs are unknown. Our method accounts for this uncertainty by considering a variable unbounded number of PTMs. By defining appropriate priors on the hidden variables, over-fitting can be avoided, and only a finite set of PTM groups are used at anytime during inference. The latter point is important because it makes calculations in the algorithm tractable. During inference, the properties of the active PTM groups are influenced by the input data and the chosen priors. We adapted both the Gibbs sampling and the restricted Gibbs sampling split-merge algorithms to infer the values of the hidden variables and parameters in our model. After inference, these hidden variables and model parameters can be used to deduce the true modification mass and a confidence score per possible modification position for each input peptide sequence, or the maximum *a posteriori* (MAP) estimate of modification masses and positions.

2.1 *i*PTMClust model and conditional distributions

The core of the generative model in *i*PTMClust is the same as in PTMClust: it describes how a pair of observed modification mass and modification position are generated. *i*PTMClust extends PTMClust by

introducing priors on model variables and parameters that govern the choice of active PTM groups from a boundless number of PTM groups. Given the type of PTM (PTM group) chosen from one of the limitless numbers of PTM groups, we can generate the observed modification mass as a noisy version of the mean modification mass, and select an amino acid most likely to contain the modification as the modified amino acid. Given the peptide sequence, we assume that the 'true' modification position is uniformly distributed among those positions that match the modified amino acid. Finally, we assume that the observed modification position is a noisy version of the true modification position.

The structural relationships between variables are shown by the Bayesian network in Figure 1. The top part outlines the priors represented by their corresponding hyper-parameters placed on the model parameters: mixing coefficient, modification mass means, modification mass variances and probability of modification occurring on an amino acid. The bottom portion describes the model for one input peptide and is repeated for N measurements, as indicated by the plate notation. The extension from PTMClust to *i*PTMClust follows similar steps as for a standard mixture model, as given in (Escobar and West, 1994; Rasmussen, 2000). In the following, we outline the rational and intuition behind the different components of our model, and the conditional probabilities used during Markov chain sampling. Details can be found in the Appendix.

In our model, each input peptide sequence S_n, indexed by $n \in \{1, ..., N\}$, where N is the number of peptides in the dataset, has a corresponding discrete peptide length L_n, observed modification position $x_n \in \{1, ..., L_n\}$ and observed modification mass m_n. We denote the amino acid in position j of the input sequence n as $S_n(j)$. The total number of values $S_n(j)$ can take on is $A = 24$, which includes the 20 naturally occurring amino acids and four special characters indicating the beginning and end of proteins and peptides. Additionally, we denote o_k to be the unknown number of input peptides assigned to cluster k. The hidden variable $c_n \in [1, .., \infty]$ denotes the unknown PTM group that peptide sequence n is assigned to. Following from the derivation of IMMs with a Chinese Restaurant Process (Antoniak, 1974; Ferguson, 1973), its probability conditioned on all other c_n's takes into consideration the likelihood

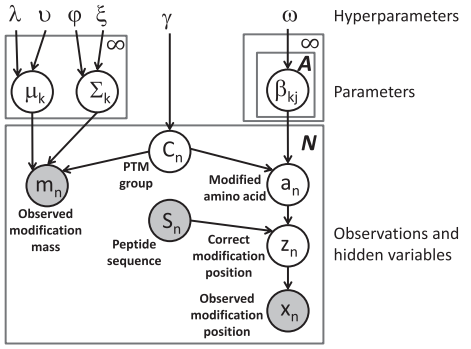

Fig. 1. A Bayesian network describing the generative model for our new algorithm *i*PTMClust, using plate notation. The shaded nodes represent observed variables, the unshaded nodes represent hidden variables and hyper-parameters are shown outside of the plates. The model describes how the observed modification mass and position are generated. The bottom part captures the assumptions about how each observation is generated. The plate notation indicates that there are N copies of the model, one for each measurement. The top portion outlines the structure of the hierarchy of priors and hyper-priors placed on the model parameters, mixing coefficients, modification mass means, modification variances and probability of modified amino acid. The outer plate shows that there is potentially an infinite number of clusters, one for each possible PTM group. The probability of modified amino acid β_{kj} is embedded in two plates signifies that there are $K \times A$ copies, one for each $K \to \infty$ PTM groups and A possible amino acids

of the n-th input modified peptide belonging to the PTM group index is given as

$$P(c_n = k|c_{N\backslash n}, \gamma, \Theta) = \begin{cases} b\frac{o_{-n,k}}{N-1+\gamma} & \text{if } o_{-n,k} > 0 \\ b\frac{\gamma}{N-1+\gamma} & \text{otherwise} \end{cases} \quad (1)$$

where $o_{-n,k}$ is the number of peptides assigned to cluster k that does not consider the n-th peptide sequence, $N\backslash n$ indicates all indices excluding n, $C_{N\backslash n}$ is shorthand notation for $C_i : \forall i \in \{N\backslash n\}$, b is the appropriate normalizing constant so the probabilities sum to one, γ is the hyper-parameter concentration parameter and Θ represents hyper-parameters λ, υ, φ, ξ, γ and ω. Furthermore, we can write the conditional posterior probability for c_n that takes into consideration the likelihood of the n-th input modified peptide belonging to the PTM group index by c_n, using parameter setting from the previous sample iteration, as follows, using $\mathcal{D}$ to indicate the observed data:

$$\begin{aligned} P(c_n = k|c_{N\backslash n}, \gamma, \Theta, \mathcal{D}) \\ \propto \begin{cases} b\frac{o_{-n,k}}{N-1+\gamma} P(a_n, z_n, x_n, m_n|c_n, S_n, \Theta) & \text{if } o_{-n,k} > 0 \\ b\frac{\gamma}{N-1+\gamma} \int P(a_n, z_n, x_n, m_n, \theta|c_n, S_n, \Theta)\partial H_0(\theta) & \text{otherwise} \end{cases} \end{aligned} \quad (2)$$

where θ represents parameters μ, Σ and β, and H_0 indicates the prior distribution placed on μ, Σ and β. The calculations of the conditional posterior probability $P(a_n, z_n, x_n, m_n|c_n, S_n, \Theta)$ and the integral $\int P(a_n, z_n, x_n, m_n, \theta|c_n, S_n, \Theta)\partial H_0(\theta)$ are given later in Equations 13 and 14, respectively. Given a vague inverse gamma prior on γ, $P(\gamma) = \mathcal{IG}(1, 1)$, its conditional posterior can be derived by combining the joint distribution of c_n's with the prior to give

$$P(\gamma|k, N) \propto \left\{\gamma^{k-3/2} \exp\left[(-2\gamma)^{-1}\Gamma(\gamma)\right]\right\}[\Gamma(N+\gamma)]^{-1} \quad (3)$$

We assume that the observed modification mass for each PTM group is normally distributed around the true modification mass, $P(m_n|c_n = k) = \mathcal{N}(\mu_k, \Sigma_k)$, where μ_k and Σ_k are the mean and variance of the modification mass in the k-th PTM group.

For mathematical convenience, we used the conjugate prior normal-inverse gamma distribution with hyper-parameter mean λ, variance υ, shape φ and scale ξ^{-1} for all PTM groups. Broad and vague corresponding priors are given to each hyper-parameter to account for uncertainty in their values (see Supplementary Information for details). Given the combination of prior and hyper-prior distributions used, the conditional posteriors for λ, υ, φ and ξ have the following forms:

$$P(\lambda|\mu_{1-k}, \upsilon) = \mathcal{N}\left(\frac{\mu_*/\sigma_*^2 + \upsilon\sum_{j=1}^k \mu_j}{1/\sigma_*^2 + k\upsilon}, \frac{1}{1/\sigma_*^2 + k\upsilon}\right) \quad (4)$$

$$P(\upsilon|\mu_{1-k}, \lambda) = \mathcal{G}\left\{k+1, \left[(k+1)^{-1}\left(\sigma_*^2 + \sum_{j=1}^k (\mu_j - \lambda)^2\right)\right]\right\} \quad (5)$$

$$P(\varphi|\Sigma_{1-k}, \xi) \propto \Gamma\left(\frac{\varphi}{2}\right)^{-k} e^{-1/2\varphi}\left(\frac{\varphi}{2}\right)^{\frac{k\varphi-3}{2}} \prod_{j=1}^k (\Sigma_j\xi)^{\frac{\varphi}{2}} e^{-\varphi\Sigma_j\xi/2} \quad (6)$$

$$p(\xi|\Sigma_{1-k}, \varphi) = \mathcal{G}\left\{k\varphi+1, \left[(k\varphi+1)^{-1}\left(\sigma_*^{-2} + \varphi\sum_{j=1}^k \Sigma_j\right)\right]^{-1}\right\} \quad (7)$$

where μ_* and σ_*^2 are the mean and variance of the observed modification masses, $\Gamma(\cdot)$ is the gamma function with the form $\Gamma(n) = (n-1)!$ for a positive integer n and $1-k$ is shorthand for $1, ..., k$.

By using conjugate priors on μ_k and Σ_k, both these parameters can be integrated out to give a probability of m_n based directly on the hyper-parameters:

$$\begin{aligned} P(m_n|c_n, \lambda, \upsilon, \varphi, \xi) \\ = \int P(m_n|c_n = k, \mu_k, \Sigma_k)P(\mu_k|\lambda, \upsilon)P(\Sigma_k|\varphi, \xi)\partial\mu_k\partial\Sigma_k \\ = t\left[\hat{\varphi}, \hat{\lambda}, (\hat{\upsilon}+1)(\hat{\upsilon}+o_k)^{-1}\hat{\xi}\right] \end{aligned} \quad (8)$$

where $t(\cdot)$ is the Student's t-distribution, o_k is the number of peptides assigned to cluster k, $\hat{\varphi} = \frac{(\varphi\lambda)+(o_k\bar{m}_k)}{\varphi+o_k}$, $\hat{\lambda} = \lambda + o_k$, $\hat{\upsilon} = \upsilon + o_k$, $\hat{\xi} = \xi + \sum_{i:\forall i,\, c_i=k}(m_i - \bar{m}_k)^2 + \frac{\lambda o_k(\bar{m}_k-\varphi)^2}{\lambda+o_k}$, and $\bar{m}_k = \frac{1}{o_k}\sum_{i:\forall i,\, c_i=k} m_i$ is the average observed modification mass for peptides assigned to the k-th PTM group.

Let $a_n : n \in \{1, ..., N\}$ denote the true (hidden) modified amino acid (i.e. the amino acid that the PTM occurs on) for the n-th peptide sequence. Then, the probability of a_n given that the PTM group is k is modeled as a multinomial distribution with parameters $\beta_{ki} \forall i = 1, ..., A$. A conjugate Dirichlet distribution with hyper-parameter ω is used for β_{ki}. The hyper-parameter ω is given a vague inverse gamma distribution. Because a conjugate prior is used on the β_{ki}'s, we can integrate them out using a standard Dirichlet integral. This leads to the conditional posterior of a cluster assignment given all others:

$$P(a_n = i | c_n = k, a_{-}, z_n, x_n, S_n, \omega) = \frac{o_{-n,ki} + \frac{\omega}{A}}{o_k - 1 + \omega} P(z_n, x_n | a_n, S_n) \quad (9)$$

where $P(z_n, x_n|a_n, S_n)$ can be factorized into $P(z_n|a_n, S_n)P(x_n|s_n)$. Both $P(z_n|a_n, S_n)$ and $P(x_n|s_n)$ are discussed below. The conditional posterior for the hyper-parameter ω can be derived similar to Equation 3 to give

$$P(\omega|A, N) = \left\{\omega^{A-3/2} \exp\left[(-2\omega)^{-1}\Gamma(\omega)\right]\right\}[\Gamma(N+\omega)]^{-1} \quad (10)$$

The details of the rest of the model are the same as those in PTMClust given in (Chung *et al.*, 2011). To reiterate, given the peptide sequence S_n and true (hidden) modified amino acid a_n, we assume the distribution over modification position z_n is uniform over those with matching amino acids:

$$P(z_n = j | a_n = i, S_n) = \begin{cases} \frac{1}{\delta_{ni}+1} & \text{if } S_n(j) = i, j > 0 \\ \frac{1}{\delta_{ni}+1} & \text{if } j = 0 \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

where δ_{ni} denotes the number of times amino acid i occurs in sequence n and $z_n = 0$ indicates that the true PTM occurs outside of the given peptide sequence. Given the true modification position z_n, the probability of modification position error $(x_n - z_n)$ for the observed modification position x_n is modeled with a discrete probability distribution, given as

$$P(x_n | z_n = j) = \begin{cases} \phi(x_n - j) & \text{if } j > 0 \\ \phi(L_n) & \text{if } j = 0 \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

where the likelihood function ϕ accounts for the modification position error. This likelihood function is shared across all PTM groups and was inferred from our empirical observation of the yeast PTM dataset as described in (Chung *et al.*, 2011).

Finally, we outline how the conditional posterior probability $P(a, z, x, m|c, S, \Theta)$ and the integral $\int P(a, z, x, m, \mu, \Sigma, \beta|c, S, \Theta)\partial H_0(\mu, \Sigma, \beta)$ used in Equation 2 can be evaluated analytically. Based on the structure of the Bayesian network of our model given in Figure 1, for a given PTM group k, $P(a, z, x, m|S, \Theta)$ can be derived as follows:

$$\begin{aligned} &P(a_n, z_n, x_n, m_n | c_n, S_n, \Theta) \\ &= \int P(a_n, z_n, x_n, m_n, \mu, \Sigma, \beta | c_n, S_n, \Theta)\partial H_{-n,k}(\mu, \Sigma, \beta) \\ &= P(z_n|a_n, S_n)P(x_n|z_n)[Dir(\omega + o_k)]\left\{t\left[\hat{\varphi}, \hat{\lambda}, (\hat{\upsilon}+1)\hat{\xi}/(\hat{\upsilon}+o_k)\right]\right\}, \end{aligned} \quad (13)$$

where $H_{-n,k}$ is the posterior distribution of μ, Σ and β based on their priors and all peptides that are assigned to the k-th PTM group excluding the n-th peptide sequence, and the variables $\hat{\varphi}$, $\hat{\lambda}$, $\hat{\upsilon}$ and $\hat{\xi}$ are defined before in Equation 8. It is easy to see the integral with respect to priors $\int P(a_n, z_n, x_n, m_n, \mu, \Sigma, \beta|c_n, S_n, \Theta)\partial H_0(\mu, \Sigma, \beta)$ for unoccupied clusters is equivalent to setting $o_k = 0$ in Equation 13, which can be written as

$$\begin{aligned} &\int P(a_n, z_n, x_n, m_n, \mu, \Sigma, \beta | c_n, S_n, \Theta)\partial H_0(\mu, \Sigma, \beta) \\ &= P(z_n|a_n, S_n)P(x_n|z_n)[Dir(\omega)]\left\{t\left[\hat{\varphi}, \hat{\lambda}, (\hat{\upsilon}+1)\hat{\xi}/\hat{\upsilon}\right]\right\} \end{aligned} \quad (14)$$

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

$$\begin{aligned} &P(c, a, z, x, m, \mu, \Sigma, \beta, \lambda, \upsilon, \varphi, \xi, \gamma, \omega | S, \Psi) \\ &= P(\gamma|\Psi)P(\lambda|\Psi)P(\upsilon|\Psi)P(\varphi|\Psi)P(\xi|\Psi)P(\omega|\Psi)\prod_{n=1}^{N}[P(c_n|\gamma) \\ &P(m_n|c_n, \lambda, \upsilon, \varphi, \xi)P(a_n|c_n, \omega)P(z_n|a_n, S_n, \Psi)P(x_n|z_n, \Psi)], \end{aligned} \quad (15)$$

where Ψ represents the model hyper-parameters for the hyper-priors placed on γ, λ, υ, φ, ξ and ω.

2.2 Markov chain inference method

The combination of complicated interactions of hidden variables and priors in *i*PTMClust leads to a complex joint distribution over high-dimensional spaces, which is impossible to characterize analytically in its entirety. So, we use an approximate Markov Chain Monte Carlo (MCMC) method (Escobar and West, 1994; Neal, 2000b). MCMC methods are commonly used for IMM models. In MCMC sampling, the model posterior distribution can be sampled to collect instances of parameter and variable settings likely under the model. Given a large enough collection of samples, the posterior can be approximated, and the ideal settings of model parameters and hidden variables can be inferred.

Although the Gibbs sampling algorithm is a commonly used MCMC sampling method for non-parametric Bayesian clustering models such as ours, it can mix poorly and produce poor results when the input dataset is large and complex (Jain and Neal, 2000). Hence, in addition to the Gibbs sampling method, we derived and implemented the restricted Gibbs sampling split-merge algorithm (or the split-merge sampling algorithm for short) (Jain and Neal, 2000) for *i*PTMClust. The split-merge sampling algorithm is designed to mitigate the aforementioned issues with the Gibbs sampling method. We adhered to the recommended settings for running this sampling algorithm given in (Jain and Neal, 2000).

Both of the methods with and without split-merge steps require Gibbs sampling updates. For each Gibbs sampling step, we use the following procedure. First, we sample the set of hidden variables (parameters are considered as hidden variables) associated with input peptide n, where $n \in \{1, ..., N\}$. Starting with the n-th input observation, for each represented (occupied) cluster K_{rep}, we draw a_n according to Equation 9 and z_n using Equation 11. Next, we sample a new cluster assignment c_n from Equation 2. Finally, we update the sufficient statistics associated with assignments for c_n and a_n. We repeat this procedure for each input peptide sequence. At the end of each Gibbs sampling step, we obtain a new value for each hyper-parameter, with the exception for γ, ω and φ, directly by sampling from their conditional posterior distribution given in Equations 4, 5 and 6, which are all distributions of standard form. Although they are not standard distributions, the conditional posteriors for γ, ω and φ are all log-concave, which implies that they are unimodal in the logarithmic domain and have a single global optimum. The log-posteriors of γ, ω and ξ are given as

$$\ln P(\gamma|k, N) = C + \left(k - \frac{3}{2}\right)\ln(\gamma) - \frac{1}{2\gamma} - \ln\Gamma(\gamma) - \ln\Gamma(N+\gamma) \quad (16)$$

$$\ln P(\omega|A, N) = C + \left(k - \frac{3}{2}\right)\ln(\omega) - \frac{1}{2\omega} - \ln\Gamma(\omega) - \ln\Gamma(N+\omega) \quad (17)$$

$$\begin{aligned} &\ln P(\varphi|\Sigma_1, ..., \Sigma_k, \xi) = C - k\ln\Gamma(\varphi/2) - (2\varphi)^{-1} + \\ &(k\varphi - 3/2)\ln(\varphi/2) + \sum_{k=1}^{K_{rep}}(\varphi/2)(\ln\Sigma_k + \ln\xi) - (\varphi\Sigma_k\xi/2) \end{aligned} \quad (18)$$

where C is a normalizing constant. Given the equations for the log-posteriors and their log-concave property, a new value for each hyper-parameter can be efficiently sampled using the efficient, easily implemented slice sampling method (Neal, 2000a).

The MCMC sequence is guaranteed to converge to the exact posterior, eventually. However, in practice, it is common to terminate the sampling procedure after a fixed number of iterations, based on an analysis of how well the chain has mixed according to trace plots of the distribution of posterior probabilities and the number of clusters over time. Furthermore, samples from a burn-in period at the beginning of the chain, before mixing has occurred, are discarded. Again, this is determined using trace plots. We found that 1000 burn-in samples and a total of 15 000 samples are enough to produce a good approximation of the posterior for the large-scale PTM datasets we studied, including the phosphopeptides, yeast proteome and human protein–protein interaction datasets described below. Details of this analysis, including trace plots, are given in Supplementary Information. For simpler datasets, such as the synthetic and phosphorylation datasets, a setting of 100 burn-ins and 6000 total samples was sufficient. A detailed analysis is provided in Supplementary Information. Lastly, to counter auto-correlations amongst the samples, we only use results taken from every fifth sample.

2.3 Background model

Unlike PTMClust, the background model for *i*PTMClust does not explicitly encompass a predefined background component. Instead, it uses a background model consisting of multiple background components learned directly from the input data. Based on the modification mass variance calculated for each PTM group, we define a background component to be a PTM group with a variance ≥ 2.0. This threshold is chosen based on the assumption that valid PTM groups should have well-defined modification masses, and thus have low variance. By allowing for multiple background components instead of one, we observe empirically that the new model is better at capturing spurious data.

3 RESULTS

Our first goal is to demonstrate that *i*PTMClust outperforms existing algorithms both in terms of finding correct clustering assignments and in refining PTMs on a set of noisy modified peptide sequences. To this end, we conducted two experiments. We benchmarked *i*PTMClust against standard clustering algorithms, *k*-means and MOG, as well as our previous algorithm PTMClust. The key observation, given in Supplementary Figure S1, is that *i*PTMClust using the split-merge algorithm attains the most consistent results across different settings and achieves increasingly better results than PTMClust as the problem becomes more complex. Given that the true modification positions are known for simulated data, we can evaluate how well PTMClust and implementations of our new algorithm perform for the task of PTM refinement. Supplementary Figure S2 shows the same trend as above: *i*PTMClust with the split-merge algorithm outperforms the others in almost all cases. We also compared blind PTM search engines SIMS (Liu *et al.*, 2008), InsPecT (Tanner *et al.*, 2005) and MODmap (Na and Paek, 2009), a state-of-the-art PTM refinement algorithm PTMFinder and our algorithms PTMClust and *i*PTMClust on detecting the true modification positions from a well-studied phosphopeptide dataset. In each of the experiments, we use both the split-merge and the Gibbs sampling inference algorithms for *i*PTMClust. We report results based on MAP estimation for *i*PTMClust for both experiments.

The second goal is to directly show the applicability of *i*PTMClust to datasets taken from studies of complex protein solutions. To achieve this goal, we analyze data taken from a genome-wide yeast and a human chromatin-specific protein complex study. We have limited our analyses to only post-processing PTM predictions generated from SIMS with either PTMClust or *i*PTMClust. We include analysis from PTMClust to highlight that *i*PTMClust is producing sensible results.

In these experiments, we initialize *k*-means, MOG and PTMClust with settings that are outlined in (Chung *et al.*, 2011). The settings for the number of burn-in and total samples for *i*PTMClust are described in the Supplementary Information.

3.1 Benchmarking against phosphopeptide predictions

We compare *i*PTMClust against current PTM search engines and PTM refinement algorithms using a real-world dataset enriched for phosphopeptides containing modification sites that are validated (Beausoleil *et al.*, 2004). We will refer to the identities of the known peptide sequences and their modification sites as the reference. The dataset consists of 1655 spectra, but we will focus exclusively on the 1340 spectra mapped and curated as singly modified phosphopeptides (SIMS, InsPecT, PTMClust and *i*PTMClust are limited to one modification per peptide sequence). In this analysis, we define positives (P) as outputs from the base blind PTM search engine that match to the reference considering only their peptide sequence, i.e. disregarding the positions of their modification, and negatives (N) as all other outputs that do not match their corresponding reference peptide sequences. Each blind PTM search engine produces a different number of P and N. For SIMS, PTMClust and *i*PTMClust (SIMS was used as the base blind PTM search engine), there are 895 P and 445 N. Lastly, for PTMFinder, which uses InsPecT as its base unrestricted PTM search engine, there are 860 P and 480 N.

For *i*PTMClust, in addition to the MAP estimate, we considered inference by averaging over samples to produce a confidence score per modification position for each output peptide sequence. By varying the confidence threshold setting, we can adjust the sensitivity and specificity of the PTM predictions. For *i*PTMClust results evaluated using confidence scores, we define a PTM prediction as a peptide sequence and its modification positions with confidence scores above the threshold; peptide sequences that do not have at least one modification position with a confidence score above the threshold are considered to be assigned to the background model. For PTMClust and *i*PTMClust using MAP estimate, a prediction is any peptide sequence not assigned to the background model. All peptide sequences assigned to the background model are removed from the evaluation. Lastly, we consider all outputs as predictions for the other algorithms because they do not use a background model.

Given the true phosphorylation of each peptide are known, we plot the number of correct phosphorylation sites versus the number of incorrect phosphorylation sites identified in Figure 2A for the results of *i*PTMClust, PTMClust, PTMFinder and InsPecT. For this experiment, the blind PTM search engine InsPecT was used to analyze the input spectra, then each of the PTM refinement algorithms, *i*PTMClust, PTMClust and PTMFinder, were used to post-process the PTM predictions output from InsPecT.

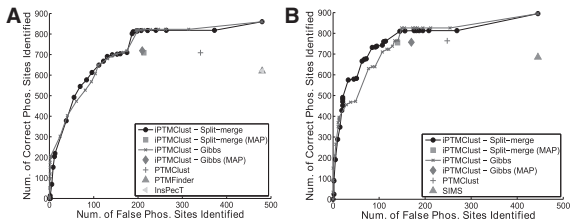

Fig. 2. The figure is a plot of the number of correct phosphorylation sites identified versus the number of false phosphorylation sites identified for *i*PTMClust and state-of-the-art PTM refinement and prediction algorithms a phosphopeptide dataset with known modification sites. (**A**) Compares *i*PTMClust against PTMClust, InsPecT and PTMFinder. The PTM predictions output from InsPecT are used as the baseline and post-processed by *i*PTMClust, PTMClust and PTMFinder. (**B**) Compares *i*PTMClust against PTMClust and SIMS with output from SIMS are used as the baseline. The curves for *i*PTMClust using the split-merge and the Gibbs sampling algorithm are produced by calculating a confidence score per modification position through inference by averaging over samples and varying the confidence score threshold [0,1]. These methods for evaluating the result outperform their counterparts using MAP estimation. More importantly, *i*PTMClust using any one of the inference methods achieves better results than InsPecT, SIMS, PTMClust and PTMFinder

For *i*PTMClust, we included results for both the split-merge and Gibbs sampling methods using either MAP estimation or inference by averaging over samples. We varied the confidence threshold to obtain a series of results for *i*PTMClust using inference by averaging over samples. Four key observations can be made from the result: (i) *i*PTMClust outperforms all other algorithms, including PTMClust and PTMFinder; (ii) *i*PTMClust with inference by averaging over samples produces better results than its MAP estimate counterpart; (iii) a background model is essential for collecting spurious modifications; and (iv) *i*PTMClust with the split-merge inference algorithm performs similarly to *i*PTMClust with the Gibbs sampling algorithm except at the region with low number of incorrect phosphorylation sites identified, where the Gibbs sampling method is performing marginally better than its counterpart. The first observation reinforces our conclusion from the study on synthetic data that *i*PTMClust outperforms other PTM refinement algorithms. The second highlights the advantage of providing a confidence score per modification position, where adjusting for the confidence threshold allows us to achieve improved results over MAP estimation. Finally, the results highlight a major benefit of *i*PTMClust beyond PTM refinement, i.e. removal of noisy data.

Next, we conducted the same experiment by post-processing the output from SIMS with PTMClust and *i*PTMClust to ensure that our algorithms are not biased towards any one particular blind PTM search engine. Similarly, we plot the number of correct phosphorylation sites versus the number of incorrect phosphorylation sites identified in Figure 2B for the results of *i*PTMClust, PTMClust and SIMS. Because PTMFinder is tightly integrated into the InsPecT algorithm, we were not able to decouple PTMFinder from InsPecT to post-process the output from SIMS. Therefore, PTMFinder is omitted from this analysis. The results based on the PTM predictions generated by both SIMS and InsPecT (discussed above) show the same three trends: (i) *i*PTMClust outperforms all other algorithms; (ii) *i*PTMClust with inference by averaging over samples produces better results than its MAP estimate counterpart; and (iii) a background model is essential for collecting spurious modifications. Specific to this experiment based on the output from SIMS, we observe that *i*PTMClust with the split-merge inference algorithm markedly performs better than *i*PTMClust with the Gibbs sampling in two conditions: when MAP estimate is used and at the region between 30 and 145 number of false phosphorylation sites identified when using inference by averaging over samples. The full list of resulting PTM predictions by SIMS and then post-processed with PTMClust and *i*PTMClust (one technical replicate) is given in Supplementary Table S2. The result shows in addition to being able to correctly refine PTM predictions, *i*PTMClust can detect noisy data and rightfully assign them to the background model.

Given enough sampling iterations, *i*PTMClust is able to produce remarkably consistent results. Of the correctly identified phosphopeptides by *i*PTMClust, ~ 98% of them are found by all five technical replications. This is true for either sampling methods. Minor differences are expected and are due to the stochastic nature of the inference methods used in *i*PTMClust. Furthermore, trace plots of log joint probability distributions and numbers of non-background PTM groups identified, shown in Supplementary Figure S3, illustrate that *i*PTMClust using both the split-merge and the Gibbs sampling methods mixes well after 1000 burn-in iterations, and 15 000 iterations are enough to estimate the posterior. Both the trace plots and level of consistency suggest that *i*PTMClust in each run have converged to a high posterior likelihood solution.

The computational time for *i*PTMClust is heavily dependent of the number of samples collected during inference, which were 15 000 iterations for this experiment. Keeping in mind that an MS/MS experiment takes hours, and in the case of a shotgun proteomics (Cantin and Yates, 2004) 12 h, the time it took *i*PTMClust to analyze the phosphopeptide dataset (on average ~50 mins for Gibbs sampling and ~70 mins for Spit-merge sampling) is a reasonable investment for higher-quality PTM predictions. As compared with other tested algorithms, *i*PTMClust took a longer time than InsPect (~2.5 mins) and PTMFinder (~4 mins), and on par with SIMS (~55 mins) and PTMClust (~70 mins). All the algorithms, single process applications, were run on a computer with Intel Xeon 3.0 GHz CPUs and 32 GB of RAM. *i*PTMClust makes a trade-off with longer computational time for superior quality PTM predictions.

3.2 Large-scale PTM analysis of yeast proteome

Through a series of benchmark experiments, we have shown that *i*PTMClust beats state-of-the-art algorithms, including our own PTMClust. Furthermore, we demonstrate that *i*PTMClust using the split-merge sampling method produces improved results over the one using the Gibbs sampling method. Next, we will test *i*PTMClust's versatility in detecting diverse PTM groups by applying it to analyze a large-scale PTM dataset taken from analyses of the yeast proteome (liquid chromatography (LC)-MS/MS spectra only) (Krogan *et al.*, 2006) using SIMS. Briefly, the yeast dataset consists of more than 2 million ion trap MS/MS spectra of which 19 560 putatively modified peptides were identified by SIMS with modification range (0, 200) Da.

The estimated false discovery rate for the predictions made by SIMS is 4.3% based on the number of decoy peptides identified. Here, we present the result taken from our post-analysis using *i*PTMClust with the split-merge algorithm (MAP estimate was used). MAP estimate is used to simplify the analysis and the mapping to known PTMs. The specific setting used for PTMClust is described in (Chung *et al.*, 2011).

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 *i*PTMClust with the split-merge algorithm, *i*PTMClust with the Gibbs sampling algorithm and PTMClust to the set of known yeast modification sites. This table shows the number of peptides and unique sites that are mapped to known PTMs. The results show *i*PTMClust is able to improve on SIMS and significantly outperform PTMClust in a complex dataset.

A summary of commonly known PTMs taken from the Uniprot knowledgebase (Release 2010_11) that are found in our dataset is shown in Table 1. Overall, *i*PTMClust using either the split-merge or the Gibbs sampling method is able to reposition a large portion of modifications to known PTM sites that were missed by SIMS originally (increase of $\sim$ 49% unique PTMs and $\sim$ 34% modified peptides for split-merge, and $\sim$ 55% and $\sim$ 43% for Gibbs). This represents a significant increase over what can be achieved using PTMClust. The most improvement is gained with phosphorylation sites, where post-analysis with *i*PTMClust is able to identify $>$65% more (known) unique sites and almost double the number of instances of phosphopeptide when compared with the result obtained from SIMS. However, *i*PTMClust using the split-merge method incorrectly places a number of peptides with acetylation and other modifications in the background model that SIMS correctly identified. We note that *i*PTMClust using the Gibbs sampling method and PTMClust did not make this mistake. A closer look reveals that many of these instances belong to a few unique peptide sequences. The analysis on the yeast proteome dataset confirms that *i*PTMClust can detect other PTMs such as acetylation and cysteine oxidation (cysteine sulfinic acid) in addition to phosphorylation. Moreover, the results reiterate that *i*PTMClust using either the split-merge or the Gibbs sampling method can refine a greater number of PTMs than PTMClust.

3.3 Analysis of human protein–protein interaction data

Protein complexes and protein–protein interactions studies are a major focal point in the field of proteomics. However, to date, the focus has been mainly on finding complex memberships and interaction partners. Because it is well established that PTMs, such as phosphorylation and acetylation, play a vital role in the formation and regulation of protein–protein interactions, we seek to complement these studies with an emphasis on the identification of PTMs.

The dataset we used consists of high mass-resolution MS/MS spectra (Orbitrap mass spectrometer) from a human protein–protein interaction study searched using SIMS. The study is a collaboration with the Emili lab at the University of Toronto, and the dataset is not yet published at the time of writing. The experimental protocol used is tandem affinity purification (Puig *et al.*, 2001; Rigaut *et al.*, 1999) followed by MS approach. Briefly, the method proceeds by placing a biological tag on all instances of a member of the complex of interest. Next, these tagged proteins are isolated and purified along with their interacting proteins, and finally, the set of purified proteins are subjected to MS analyses.

Here, we chose to focus on three well-studied protein complexes, the Mediator (MED), the RNA Polymerase II (POL2) and the Polycomb Repressive Complex 1 (PRC1). This dataset consists of 17 experiments. These experiments include technical replicates, covering 6 of 12 proteins known to be in the MED complex, two experiments covering 2 of the 32 members of the POL2 complex and three experiments covering 2 of 12 members of the PRC1 complex. There is a total of 13 221 spectra mapped to modified peptides by SIMS (estimated false discovery rate of 13.5% based on the number of decoy peptides identified) with a modification range (0, 300) Da.

Supplementary Table S1 highlights the 48 distinct peptide sequences putatively identified to be modified either with acetylation or phosphorylation. Including duplicates, 114 modified peptides were found. Of the 48 unique peptides, 9 had at least one instance with their modification site corrected by *i*PTMClust using the split-merge sampling algorithm that was originally misplaced by SIMS. These are highlighted in bold. We speculate this smaller number improvement is due to two major reasons: (i) a majority of the listed PTMs are acetylation, which SIMS does a good job with, as seen in the yeast study above, and (ii) spectra are cleaner owing to the use of a high-resolution mass spectrometer. In this list, there are 15 putative novel modification sites and 33 known ones according to the Uniprot knowledgebase (Release 2011_12). Although the complete list is not shown, after removing those assigned to the background model, we have identified 10 409 putative modified peptides such that a

Table 1. Summary of peptides with known PTM sites in the yeast proteome dataset

PTM	Peptides with known PTM sites (number of unique sites)			
	i PTMClust split-merge	*i* PTMClust Gibbs	PTMClust	SIMS
Phosphorylation	196 (103)	185 (101)	115 (66)	100 (61)
Acetylation	59 (5)	78 (9)	75 (9)	72 (8)
Cysteine oxidation (cysteine sulfinic acid)	7 (2)	6 (1)	7 (1)	6 (1)
Others	24 (2)	35 (5)	35 (5)	35 (5)
Total	286 (112)	304 (116)	232 (81)	213 (75)

large portion of them is mapped to regions in their respective proteins that do not contain known PTMs. Similar to those listed putative, novel phosphorylated and acetylated peptides, we believe this list also contains many high-quality new PTM discoveries. Hence, this list represents a filtered list of high-quality candidates for further investigation. We have shown a PTM prediction pipeline comprises a blind PTM search engine, and *i*PTMClust can be fruitful in novel discoveries and should be used routinely.

The use of a high mass-resolution mass spectrometer is expected to reduce errors and potentially remove the need for refinement of measured modification masses. Even so, we noticed many instances where the observed modification mass deviates from our refined modification mass by ~1 Da. This modification mass error is believed to be due to the presence of isotopes. Although heuristics can be used to account for these mass shifts, such methods can be error-prone and cannot adapt to unforeseen mass errors. Our results show that *i*PTMClust can handle such errors and improve the quality of PTM predictions taken from an analysis of a high mass-resolution mass spectrometer.

4 CONCLUSION

Accurately identifying PTMs and their potential roles in clinical studies such as biomarker discovery and drug development is an important task. Although thousands of PTM candidates have been reported using blind PTM search engines (Liu *et al.*, 2006, 2008; Han *et al.*, 2005; Searle *et al.*, 2006; Tanner *et al.*, 2005; Tsur *et al.*, 2005), these blind PTM search algorithms suffer from mass measurement inaccuracy and uncertainty in predicting modification positions, making the findings error prone. The importance of post-processing PTM predictions using a PTM refinement algorithm have been established in (Chung *et al.*, 2011; Tanner *et al.*, 2008). The previous state-of-the-art algorithm, PTMClust, achieves a significantly higher PTM prediction accuracy over blind PTM search engines alone and outperforms existing PTM refinement algorithm, PTMFinder. Despite significant improvements in PTM prediction, PTMClust has three main drawbacks of particular interest: (i) it uses of a greedy-based non-automatic model selection algorithm, (ii) it requires manual parameterization on the maximum number of PTM groups and (iii) it does not provide a confidence score per modification position.

To address these issues, we introduce *i*PTMClust. *i*PTMClust extends PTMClust by using an infinite mixture model approach that achieves the following three benefits: (i) outperforming PTMClust and other PTM refinement algorithms, (ii) providing a fully automated model selection method without the need for any manual parameterization, and (iii) offering modification position level confidence scores that users can use to assess the quality of the results and to greater refine their analyses. Through a series of benchmark experiments using both synthetic and real (phosphopeptides and yeast proteome) data, we demonstrated that *i*PTMClust better models the PTM generative process and outperforms PTMClust, PTMFinder and other blind PTM search engines. In addition, we analyzed data generated from a yeast proteome study using *i*PTMClust in which we reported an improvement over the base blind PTM search algorithm SIMS in detecting annotated PTMs. Thousands of putative PTMs were found in this analysis. Moreover, in our in-depth look at PTM predictions for three human protein complexes, MED, POL2 and PRC1, *i*PTMClust identified numerous validated and putative phosphorylated and acetylated peptides that may be involved in the formation and regulation of protein–protein interactions. Further investigations are warranted, but we believe a number of these putative predictions are valid PTMs and can serve to further our understanding of the complexities involved in protein–protein interactions. To summarize, our new algorithm *i*PTMClust is easy to use, achieves overall greater performance than the state-of-the-art, provides confidence scores at the modification position level that allow for a higher flexibility when evaluating potential PTMs and is designed to be broadly applicable to PTM predictions generated from any blind PTM search engine.

Given the rapid advancement of mass spectrometer technology, how applicable is *i*PTMClust going forward? We explore this question by analyzing data generated from high mass-resolution mass spectrometers, e.g. from an Orbitrap. Although mass accuracy has improved with the use of high mass-resolution mass spectrometers, the presence of isotopes, for example, can result in deviations in observed modification masses. In addition, higher mass accuracy does not necessarily equate to errorless modification site determination. We have shown in our analysis of the human protein complex data that problems with mass measurement inaccuracy and uncertainty predicting modification positions continue to exist for data generated from high mass-resolution mass spectrometers, such as an Orbitrap used in the experiment. Our results demonstrate that *i*PTMClust can improve on PTM predictions taken from data with high mass accuracy, and continue to be a vital component of a genome-wide PTM study.

In designing *i*PTMClust, we have developed two different inference algorithms, the split-merge Metropolis-Hastings and the Gibbs sampling algorithm. While the Gibbs sampling method is the standard inference algorithm for Bayesian mixture models and IMMs, the split-merge method is shown to perform better when dealing with complex datasets (Jain and Neal, 2000). Results of our synthetic data experiments also exhibit this trend. *i*PTMClust using the Gibbs sampling method displays a performance drop, while its counterpart using the split-merge sampling algorithm perform consistently well as the data used gets more complex. It is important to note that when there are few PTMs within a small modification mass window, say 1–2 Da (e.g. when two or three PTM groups added in our experiment with the synthetic data), our algorithm using the Gibbs sampling performs at par with or slightly worse than its counterpart using the split-merge method. Furthermore, *i*PTMClust using the Gibbs sampling runs faster per iteration (~ 50% quicker) than *i*PTMClust using the split-merge method. Hence, for large datasets, where running time can be overwhelmingly long, we recommend running *i*PTMClust with the Gibbs sampling method as it provides a good trade-off between quality of the result and computational cost.

Despite outperforming its competitors, *i*PTMClust has a number of limitations. Similar to PTMClust, it is unable to handle more than one modification per input peptide sequence, and PTM groups identified can contain multiple PTMs if their

modification masses are similar. The latter problem is less of an issue when working with high mass-resolution data. Moreover, *i*PTMClust does not consider the underlying spectrum when refining a PTM prediction. The presence of certain peaks in the spectrum can add to support to a residue along the peptide sequence as being the modification position. Its limitations notwithstanding, *i*PTMClust is shown to outperform both PTMClust and previous state-of-the-art in our benchmark tests using both synthetic and real-world PTM data.

ACKNOWLEDGEMENTS

We thank Jonathan Olsen, Jian Liu, 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 in parts by funds from a John C Polanyi Award (to B.J.F.); an Operating Grant from the Canadian Institutes of Health Research (to B.J.F.); and an Ontario Ministry of Research and Innovation grant (to A.E.).

Conflict of Interest: none declared.

REFERENCES

Agilent (2005) Spectrum mill. http://www.chem.agilent.com/en-US/Products-Services/Software-Informatics/Spectrum-Mill/pages/default.aspx.

Albuquerque,C.P. *et al.* (2008) A multidimensional chromatography technology for in-depth phosphoproteome analysis. *Mol. Cell Proteomics*, **7**, 1389–1396.

Antoniak,C.E. (1974) Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Ann. Stat.*, **2**, 1152–1174.

Bailey,C.M. *et al.* (2009) SLoMo: automated site localization of modifications from ETD/ECD mass spectra. *J. Proteome Res.*, **8**, 1965–1971.

Baker,P.R. *et al.* (2011) Modification site localization scoring integrated into a search engine. *Mol. Cell Proteomics*, **10**, M111.008078.

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

Beausoleil,S.A. *et al.* (2006) A probability-based approach for high-throughput protein phosphorylation analysis and site localization. *Nat. Biotechnol.*, **24**, 1285–1292.

Bishop,C.M. (2006) *Pattern Recognition and Machine Learning*. Springer, NY, USA.

Cantin,G.T. and Yates,J.R. (2004) Strategies for shotgun identification of post-translational modifications by mass spectrometry. *J. Chromatogr.*, **1053**, 7–14.

Chalkley,R.J. and Clauser,K.R. (2012) Modification site localization scoring: strategies and performance. *Mol. Cell Proteomics.*, **11**, 3–14.

Chung,C. *et al.* (2011) Computational refinement of post-translational modifications predicted from tandem mass spectrometry. *Bioinformatics*, **27**, 797–806.

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

Escobar,M.D. and West,M. (1994) Bayesian density estimation and inference using mixtures. *J. Am. Stat. Assoc.*, **90**, 577–588.

Ferguson,T.S. (1973) A Bayesian analysis of some nonparametric problems. *Ann. Stat.*, **1**, 209–230.

Geman,S. and Geman,D. (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intel.*, **PAMI-6**, 721–741.

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

Jain,S. and Neal,R. (2000) A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *J. Comput. Graph. Stat.*, **13**, 158–182.

Keller,A. *et al.* (2002) Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Anal. Chem.*, **74**, 5383–5392.

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

Lehninger,A.L. *et al.* (1993) *Principles of Biochemistry*, 2nd edn. Worth Publishing, New York, NY.

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 searching of global MS/MS datasets for unexpected post-translational modifications. *Anal. Chem.*, **80**, 7846–7854.

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

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

Neal,R. (2000a) Slice sampling. *Ann. Stat.*, **31**, 705–767.

Neal,R.M. (2000b) Markov chain sampling methods for Dirichlet process mixture models. *J. Comput. Graph. Stat.*, **9**, 249–265.

Olsen,J.V. *et al.* (2006) Global, in vivo, and site-specific phosphorylation dynamics in signaling network. *Cell*, **127**, 635–648.

Phanstiel,D.H. *et al.* (2011) Proteomic and phosphoproteomic comparison of human ES and iPS cells. *Mol. Cell Proteomics*, **8**, 821–827.

Puig,O. *et al.* (2001) The tandem affinity purification (tap) method: a general procedure of protein complex purification. *Methods*, **24**, 218–229.

Ramakrishnan,S. *et al.* (2009a) Mining gene functional networks to improve mass-spectrometry based protein identification. *Bioinformatics*, **25**, 2955–2961.

Ramakrishnan,S.R. *et al.* (2009b) Integrating shotgun proteomics and mRNA expression data to improve protein identification. *Bioinformatics*, **25**, 1397–1403.

Rasmussen,C.E. (2000) The infinite Gaussian mixture model. In: Solla,S.A., Leen,T.K. and Müller,K.-R. (eds.) *Advances in Neural Information Processing Systems 12*. MIT Press, Cambridge, MA, USA, pp. 554–560.

Rigaut,G. *et al.* (1999) A generic protein purification method for protein complex characterization and proteome exploration. *Nat. Biotechnol.*, **17**, 1030–1032.

Savitski,M.M. *et al.* (2011) Confident phosphorylation site localization using the mascot delta score. *Mol. Cell Proteomics*, **10**, M110.003830.

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,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.

Taus,T. *et al.* (2011) Universal and confident phosphorylation site localization using phosphors. *J. Proteome Res.*, **10**, 5354–5362.

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