Published in final edited form as:

J Appl Stat. 2019 ; 46(1): 30–46. doi:10.1080/02664763.2018.1454893.

Bayesian Hierarchical Model for Protein Identifications

Riten Mitra¹, Ryan Gill², Sinjini Sikdar³, and Susmita Datta^{3,*}

¹Department of Bioinformatics and Biostatistics, University of Louisville, Louisville, KY 40202.

²Department of Mathematics, University of Louisville, Louisville, KY 40292.

³Department of Biostatistics, University of Florida, Gainesville, FL 32611.

Abstract

In proteomics, identification of proteins from complex mixtures of proteins extracted from biological samples is an important problem. Among the experimental technologies, Mass-Spectrometry (MS) is the most popular one. Protein identification from MS data typically relies on a “two-step” procedure of identifying the peptide first followed by the separate protein identification procedure next. In this setup, the interdependence of peptides and proteins are neglected resulting in relatively inaccurate protein identification. In this article, we propose a Markov chain Monte Carlo (MCMC) based Bayesian hierarchical model, a first of its kind in protein identification, which integrates the two steps and performs joint analysis of proteins and peptides using posterior probabilities. We remove the assumption of independence of proteins by using clustering group priors to the proteins based on the assumption that proteins sharing the same biological pathway are likely to be present or absent together and are correlated. The complete conditionals of the proposed joint model being tractable, we propose and implement a Gibbs sampling scheme for full posterior inference that provides the estimation and statistical uncertainties of all relevant parameters. The model has better operational characteristics compared to two existing “one-step” procedures on a range of simulation settings as well as on two well-studied datasets.

Keywords

Mass-Spectrometry; protein identification; hierarchical model; Gibbs sampling; cluster

1. Introduction

1.1 Mass Spectrometry

One of the main goals of proteomics is the identification of proteins from a biological sample containing mixtures of unknown proteins. The technology of mass spectrometry has become an indispensable tool in attempting to achieve this goal. Some of its recent applications include proteomic biomarker identification for complex diseases using label

*Corresponding author.

Disclosure Statement

No potential conflict of interest.

free shotgun proteomics. In such experiments, proteins are first digested into peptides to generate the first step mass spectra (MS-spectra) [1, 12, 21] containing mass-over-charge (m/z) values on the x-axis and intensities on the y-axis. After preprocessing the spectra, algorithms are used to detect interesting peaks. In order to properly identify the peptides associated with these interesting peaks, they are further fragmented into smaller ions in a second mass spectrometry resulting in MS/MS spectra or tandem mass spectra. Any such spectrum is a collection of certain mass-over-charge and intensity signatures. Based on this, peptide identification follows by means of search algorithms [5, 7, 25]. The basic principle behind the identification techniques is to search a given sequence database for the peptide and then compare the theoretical spectrum with the observed spectrum. Specifically, a peptide-spectrum match (PSM) score is assigned to a peptide that quantitates the strength of the match. There are several PSM algorithms existing in the current literature [2, 4, 8, 9, 13, 17, 19, 26, 30]. Note that, in our article, we will not concern ourselves with the quality of the PSM algorithm for the MS/MS spectrum. However, we do agree that efficient scoring algorithms could largely mitigate the errors involved in the first step and can greatly improve any protein identification procedure. The second step following the peptide identification with PSM is to identify the corresponding proteins as a separate procedure by using the PSM scores. Almost all commonly used protein identification algorithms rely on some form of this “two-step” procedure. These methods identify the peptides first and then use the results in protein identification as a separate step. Some of these methods rely on constructing protein-level p-values from peptide level p-values [3, 29]. A widely used and highly efficient “two-step” method is ProteinProphet [24], which uses a product rule to calculate protein probabilities from peptide probabilities. Most of these methods rely on certain simplistic assumptions (e.g., independence) to connect proteins to peptides. For a “two-step” method, deriving peptide probabilities is the first step and is not influenced by subsequent protein inference.

In more realistic settings, the simplistic assumptions driving these “two-step” methods may not be satisfied. The bias in the peptide presence probability step may influence the protein presence probability and the presence of a protein certainly influences the presence of a peptide. This can cause the bias of estimates obtained from the first step to induce biased standard errors and inflated (or deflated) confidence measures for protein identifications. A toy example explaining the steps of a protein identification process through a typical MS/MS experiment is shown in Figure 1. Here P_1 , P_2 and P_3 denote the proteins which generate the peptides p_1, p_2, p_3, p_4, p_5 and p_6 . The PSM algorithm incorrectly identifies the peptides p_4 and p_5 as p_x and p_y respectively, which result in the incorrect identification of the proteins P_x and P_y .

To alleviate the potential difficulties of the “two-step” procedures some researchers came up with “one-step” models, featuring a feedback loop between the proteins and their constituent peptides in the joint protein peptide likelihood. This phenomenon is described in hierarchical statistical model (HSM) [28] and the nested mixture model (NMM) [18] in great details. “One-step” procedure constitutes a different genre of protein identification in which a joint likelihood of the protein and peptide presence in a sample is considered. Two popular “one-step” methods are the hierarchical statistical model (HSM) [28] and the nested mixture model (NMM) [18]. Both HSM and NMM assume independence of the proteins in the

sample, an assumption which can be relaxed in the context of protein identification problem. Here, in this article we develop a “one-step” protein identification method using a hierarchical statistical model that incorporates the proteins and peptides simultaneously. However, unlike the other two “one-step” methods [18] and [28], where they assume i.i.d prior for all proteins, we assume that proteins present in the same biological pathway are likely to be present or absent together. Additionally, our method deals with “degenerate” peptides unlike [18]. Degenerate peptides are those, which can be present in multiple proteins. These peptides present a challenge in the protein identification using MS/MS data since their true corresponding proteins are ambiguous. The practical effect is that if one peptide has a very high score, and belongs to multiple proteins, then all these proteins will likely be identified as being present, even though only one of them may actually be present in the sample. Additionally, we provide a MCMC based posterior inference to the problem of protein identification, whereas [18] and [28] use approximate maximum likelihood estimates (MLEs) of the parameters using EM algorithm.

In this article, we will use *f*values (obtained from PeptideProphet) as the PSM scores as a starting value of our procedure. The *f*values are defined as the linear combinations of several parameters, for example *Xcorr* and *deltaCn*, which can be obtained as outputs from SEQUEST [7], a PSM scoring algorithm. In this article, we focus on the joint “one-step” modeling based on these *f*value scores. We note that the *f*value scores play a very different role in our method compared to PeptideProphet. In the latter, the *f*values absolutely determine the final inference for peptide presence through the peptide probabilities and are one to one functions of each another. In contrast, our inference for peptide probabilities, iteratively change with protein probabilities, and only in the first iteration they are deterministically linked. It is to be noted that users are free to use any PSM score such as Tandem scoring [5] for MS/MS data as a starting data value of our method. Some improved versions of such scores are also reported in MacCoss et al. [22].

The main challenge in developing such a model is the complexity of the data-generating mechanism. Integrating this scientific process of data-generation using mass spectrometry with statistical modeling is essential for improved classification - a goal which has not been implemented thus far. This has been our motivation for building a Bayesian hierarchical model (BHM), in the field of protein identification. It offers the generic advantages of Bayesian inference like tractable posterior uncertainties obtained from Markov Chain Monte Carlo (MCMC) simulation.

1.2 Data Structure

The data from a proteomics experiment can be summarized by a list of putative peptides identified from the peptide sequences. These are generated from databases of protein sequences and come with associated PSM scores. Each score represents how well the theoretical peptide spectrum from the database matches the observed MS/MS spectrum. Based on this data, we aim to identifying the subset of truly present proteins from a large set of candidate proteins. This problem of inferring the set of correct proteins from the peptides is the traditional “protein inference problem”. Heuristically, higher scores suggest a higher chance of a peptide being present. Using a similar argument, a protein with a large number

of high scoring peptides is more likely to be present. Keeping these assumptions in view, we propose a “one-step” Bayesian hierarchical model (BHM) which links the proteins and peptides through conditional distributions and assumes a flexible probabilistic mechanism of peptide generation. Also, we incorporate prior information by grouping functionally related proteins. This is motivated by the assumption that proteins sharing the same biological pathway are more likely to co-occur in a proteomics experiment. Proteins sharing the same peptides have been used in Gerster et al. [10] to build intrinsic clusters for improved prediction. Here we take a slightly different approach. We rely on the peptide sharing mechanism as well as external information from known protein pathways. We propose a fully Bayesian MCMC scheme that use the scope of full posterior inference unlike some recent Bayesian formulations [18, 28] for “one-step” protein identification.

The rest of the article is organized as follows. Section 2 describes the model, its underlying motivations, and strategies of inference. In this section, we outline in detail the derivations of the full conditionals required for implementing the posterior inference. In Section 3, we describe a set of simulation experiments to assess the performance of our method. In Section 4, we apply our model to a yeast and *H.influenzae* dataset. We consider these datasets as they are well tested by other protein identification algorithms multiple times. We also discuss here a detailed comparison with two other existing “one-step” approaches and a popular “two-step” protein identification algorithm. Finally, we conclude with a discussion on the importance of our algorithm and its current relevance to proteomics.

2. Bayesian Hierarchical Model

We now proceed to describe each component of the hierarchical model mathematically. For pre-defined protein clusters (indexed by j), we let T_{ij} denote the indicator for the presence of protein i in the j^{th} cluster for $j = 1, 2, \dots, K$ where K is the total number of pathway clusters. We define C_j as the size of the j^{th} cluster. We assume that T_{ij} s are independent Bernoulli random variables with cluster specific parameters ρ_j . That is, we assume

$$P(T_{ij} = 1) = \rho_j; \quad i = 1, 2, \dots, C_j; \quad j = 1, 2, \dots, K$$

The set of all T_{ij} s is denoted by T . We refer to ρ as the collection of all such ρ_j s, i.e., $\rho = \{\rho_1, \rho_2, \dots, \rho_K\}$. We assume that conditioned on ρ , all proteins in a particular pathway cluster share the same probability of occurrence. This allows an implicit borrowing of strength across the proteins in a cluster - a feature that will be evident in the posterior sampling steps for sampling T_{ij} . We shall impose a conjugate Beta hyper prior for each ρ_j . Note that the T_{ij} s are independent binary variables, conditioned on ρ . After integrating out ρ_j s with respect to the Beta prior, they are no longer independent. Specifically, the posterior probability of protein presence would now depend on the cluster specific probability ρ_j and hence indirectly on the presence of other proteins in the cluster.

Next, we assume that a present protein is likely to have greater number of peptide hits than an absent protein. We model this feature by a differential distribution in the number of peptide hits and protein lengths by the equation

$$p\left(n_{ij}|T_{ij}, l_{ij}\right) \sim \begin{cases} h_0(n_{ij}|\lambda_0, l_{ij}) & \text{if } T_{ij} = 0 \\ h_1(n_{ij}|\lambda_1, l_{ij}) & \text{if } T_{ij} = 1 \end{cases} \quad (2.1)$$

where n_{ij} and l_{ij} denote the number of peptide hits and the length of the i^{th} protein in the j^{th} cluster, respectively. We define n as the vector of all peptide hits. Here h_0 and h_1 are the hit distributions corresponding to absent and present proteins, respectively; λ_0 and λ_1 are scaling parameters for the mean for each of the distributions. For our application, we choose the h'_k ($k = 1, 2$) to be Poisson mass functions indexed by λ_k and protein lengths l_{ij} . Thus, for any protein i of length l_{ij} in cluster j

$$h_k\left(n_{ij}|\lambda_k, l_{ij}\right) = \frac{\exp\left\{-\lambda_k l_{ij}\right\}\left(\lambda_k l_{ij}\right)^{n_{ij}}}{n_{ij}!} \quad \text{for } k = 0, 1$$

Note that when the length information is not incorporated (or, equivalently, all lengths are assumed to be equal), this translates to a Gamma posterior for λ_k . For notational convenience, we suppress the explicit conditioning on l_{ij} in the forthcoming discussion.

Next, we connect the proteins and peptides through the following mixture model for the scores of peptides. We let $S_w^{i,j} \in \{1, 0\}$ denote the indicator that protein i in the j^{th} cluster generates the w^{th} peptide. Then, we have

$$\begin{aligned} p\left(S_w^{i,j} = 1\right) &= p_1 \cdot p\left(T_{ij} = 1\right) + p_0 \cdot p\left(T_{ij} = 0\right) \\ p\left(S_w^{i,j} = 0\right) &= \left(1 - p_1\right) \cdot p\left(T_{ij} = 1\right) + \left(1 - p_0\right) \cdot p\left(T_{ij} = 0\right) \end{aligned} \quad (2.2)$$

where, p_1 and p_0 are the probabilities of generating a peptide w given $T_{ij} = 1$ and 0, respectively. The set of all $S_w^{i,j}$ s is denoted by S . Finally, we complete the hierarchy by assigning two separate sampling distributions to the incorrectly and correctly identified peptide scores depending on whether the peptide is absent or present in the sample. These are given by

$$p\left(x_w^{i,j}|S_w^{i,j}\right) \sim \begin{cases} f_u\left(x_w^{i,j}|\theta_u\right) & \text{if } S_w^{i,j} = 0 \\ f_c\left(x_w^{i,j}|\theta_c\right) & \text{if } S_w^{i,j} = 1 \end{cases} \quad (2.3)$$

where $x_w^{i,j}$ denotes the score of the w^{th} peptide generated by the i^{th} protein in j^{th} cluster.

Also, the set of all scores is denoted by x . Here, f_u and f_c denote the distributions of scores for incorrect and correct assignments, respectively; θ_u and θ_c are the corresponding parameters. We chose f_c as the Gamma distribution with $\theta_c = (\alpha, \beta)$ where α and β are the

shape and rate parameters, respectively, and we chose f_u as the Normal distribution with $\theta_u = (\mu, \sigma)$ where μ and σ are the mean and standard deviation, respectively. The mixture model equation for the conditional distribution of scores follows the same formulation used in many peptide-score models, including [10, 18, 28]. The scores used in our case studies were the fvalues obtained from PeptideProphet. The specific forms of the distributions are again mostly based on recommendations from existing literature on protein identification. These could of course vary depending on the peculiarities of the score generation process. Though, we considered certain variations of these distributions in some test simulations, these particular choices worked the best for our case studies.

The distribution of peptide scores conditioned on protein presence can be derived by directly integrating out the peptide indicators. The resultant hierarchical form is

$$p(x_w^{i,j} | T_{ij} = 0) = p_0 f_c(x_w^{i,j}) + (1 - p_0) f_u(x_w^{i,j}) \quad (2.4)$$

$$p(x_w^{i,j} | T_{ij} = 1) = p_1 f_c(x_w^{i,j}) + (1 - p_1) f_u(x_w^{i,j}) \quad (2.5)$$

or equivalently,

$$p(x_w^{i,j} | T_{ij}) = \{p_1 f_c(x_w^{i,j}) + (1 - p_1) f_u(x_w^{i,j})\}^{T_{ij}} \{p_0 f_c(x_w^{i,j}) + (1 - p_0) f_u(x_w^{i,j})\}^{1 - T_{ij}}$$

Next, we impose priors on all the parameters. We assume Gamma priors on $(\lambda_0, \lambda_1) \sim \text{Gamma}(s_1, s_0)$; Beta priors on p_0, p_1, ρ_j and Gaussian priors on $\{\log(\alpha), \log(\beta)\} \sim \mathcal{N}(0, h)$. A prior from an uninformative location-scale family was assigned to μ and σ . Specifically, we assumed the maximal data information prior: $p(u) \propto 1$; $p(\sigma^2) \propto \frac{1}{\sigma^2}$, [31]. The hyper parameters are chosen to reflect prior uninformateness and allow the inference to be guided by data as much as possible. For the specific values of hyper-parameters we refer the readers to Section 4.

2.1 Joint Model

Next, we introduce the following notations to be used in the joint model:

q_1^1, q_0^1 : The number of all true and false peptides generated from true proteins, respectively.

Thus,

$$q_l^1 = \sum_{i,j,w} I(S_w^{i,j} = l, T_{ij} = 1) \text{ for } l = 0, 1$$

q_1^0, q_0^0 : The number of all true and false peptides generated from false proteins, respectively.

m_0 : The mean of spectra scores over peptides w with $S_w = 0$ (absent peptides).

n_0 : The number of peptides with $S_w = 0$.

We can now express the joint likelihood of the data conditioned on the parameters as

$$p(T|\rho) \cdot p(n|T, \lambda_0, \lambda_1, l) \cdot p(S|T, p_0, p_1) \cdot p(x|S, \theta_u, \theta_c) = \prod_l \prod_k p(T_{kl}|\rho_l) \\ \cdot \prod_{u,v} p(n_{uv}|T_{uv}, l_{uv}) \cdot p(S|T, p_0, p_1) \cdot p(x|S, \theta_u, \theta_c)$$

Using the functional forms described above, this reduces to

$$\left(\prod_l \rho_l^{c_l} (1 - \rho_l)^{C_l - c_l} \right) \times \left(\prod_{u,v} h_1(n_{uv}|\lambda_1, l_{uv})^{T_{uv}} h_0(n_{uv}|\lambda_0, l_{uv})^{1 - T_{uv}} \right) \quad (2.6) \\ \times \left(\prod_{i,j; T_{ij}=1} \prod_w p_1^{S_w^{i,j}} (1 - p_1)^{1 - S_w^{i,j}} \right) \times \left(\prod_{i,j; T_{ij}=0} \prod_w p_0^{S_w^{i,j}} (1 - p_0)^{1 - S_w^{i,j}} \right) \\ \times \left(\prod_{i,j,w; S_w^{i,j}=1} f_c(x_w^{i,j}|\theta_c) \right) \times \left(\prod_{i,j,w; S_w^{i,j}=0} f_u(x_w^{i,j}|\theta_u) \right)$$

where c_l is the number of present proteins in cluster l . This joint model is a Bayesian hierarchical description of the data generation process. The spectra scores and the protein-peptide connections form the leaves of the Bayesian hierarchy.

The joint model holds important implications for modeling degeneracy. Consider a toy example with three classes of proteins A) a large group where the corresponding peptides have uniformly high scores, B) a large group where the corresponding peptides have uniformly low scores, and C) where the corresponding set of peptides contain some high scoring peptides which are a subset of peptides mapped in A and also a group of peptides having low scores. Unlike (A) and (B) it is not straightforward to assign classes for proteins of class C since it can be argued that (i) the corresponding peptides are only generated from (A) and the protein contribution is only from peptides with low scores. Or (ii) the protein really generates the high scoring peptides. Note that the MCMC would tend towards assigning the protein indicator as 1 for Class A (with high value of p_1) and 2 for class B with high values of $1 - p_0$. However for class C, the likelihood ratio will depend on a number of factors like the number of subset (high scoring) peptides as well as the number of low scoring peptides. The lesser the former and higher the latter, there is more chance of assignment as false protein and vice versa. Also, it will depend in a complex way on the remaining proteins through the shared information of p_0 and p_1 . It is the tussle played out between the two likelihood components that help in sorting out the classification problem for degenerate peptides.

The conditionals described next will help us in inferring the interconnected latent layers and parameters.

2.2 Inference: Full Conditionals and MCMC Summaries

Previous Bayesian approaches dealing with similar setups [18, 28] focused on finding the marginal posterior mode of the desired parameters and used optimization methods. However, when full conditionals are available, an appropriate Gibbs sampling scheme is able to generate the entire posterior distribution and hence provide uncertainty estimates. In the expressions for full conditionals below, we shall use $[x | y, z]$ to generically indicate a transition probability that samples the parameter x and is indexed by currently imputed values of y and z . In writing the transition probabilities, here and later, we shall include only those conditioning parameters which appear in the kernel.

1. $[T_{ij} | n_{ij}, \rho_j, x, \theta_u, \theta_c] \sim \text{Bernoulli}(\gamma/(1 + \gamma))$ where
$$\gamma = \frac{\rho_j h_1(n_{ij} | \lambda_1) [\prod_w \{p_1 f_c(x_w^{i,j} | \theta_c) + (1 - p_1) f_u(x_w^{i,j} | \theta_u)\}]}{(1 - \rho_j) h_0(n_{ij} | \lambda_0) [\prod_w \{p_0 f_c(x_w^{i,j} | \theta_c) + (1 - p_0) f_u(x_w^{i,j} | \theta_u)\}]}$$
2. $[\lambda_k | n, T] \sim \text{Gamma}(\sum_{i,j} n_{ij} \mathbb{I}(T_{ij} = k) + s_1, \sum_{i,j} \mathbb{I}(T_{ij} = k) + s_0)$ for $k = 0, 1$
The above is the posterior conditional when length is not incorporated. When length is incorporated, no closed form is available and the λ_k 's are sampled via Metropolis-Hastings steps.
3. $[\frac{1}{\sigma} | S, x] \sim \text{Gamma}\left(\frac{n_0 - 1}{2}, \sum_k; S_k = 0 \frac{(x_k - m_0)^2}{2}\right)$
4. $[\mu | S, x, \sigma] \sim \mathcal{N}(m_0, \sigma / \sqrt{n_0})$
5. $[\alpha, \beta | S, x]$ is implemented by a Metropolis-Hastings step. Since $\alpha, \beta > 0$, we apply a change of variables to $\{\log(\alpha), \log(\beta)\}$ and update them via Metropolis-Hastings steps.
6. $[S_w^{i,j} | x, p_0, p_1, T_{ij} = t] \sim \text{Bernoulli}(r)$ where $r = \frac{p_t f_c(x_w^{i,j})}{p_t f_c(x_w^{i,j}) + (1 - p_t) f_u(x_w^{i,j})}$ for $t = 0, 1$
7. $[p_0 | q_1^0, q_0^0] \sim \text{Beta}(q_1^0 + 1, q_0^0 + 1)$
8. $[p_1 | q_1^1, q_0^1] \sim \text{Beta}(q_1^1 + 1, q_0^1 + 1)$
9. $[\rho_j | c_j] \sim \text{Beta}(c_j + 1, C_j - c_j + 1)$

Posterior inference proceeds by iterating over these full conditionals. At the end of each iteration, we obtain imputed values of all the model parameters as well as the latent protein indicators. After a specified number of iterations, we collect the MCMC samples after burn-in and construct posterior summaries. We note that the posterior probability of protein presence is our primary object of inference. We compute this by averaging over the latent protein indicators that appear in the MCMC. The posterior probability for presence of

protein i belonging to cluster j is expressed as $\hat{P}_{ij} = p(T_{ij} = 1 | data) = \frac{\sum_l I(T_{ij}^l = 1)}{n_b}$. Here n_b is

the number of post burn-in samples and T_{ij}^l is the imputed value of T_{ij} at the l^{th} iteration of the MCMC. It can be noted that the same MCMC scheme provides us with posterior uncertainties associated with them. For example, we could estimate the standard error of the posterior estimate by directly computing the standard deviation of the posterior sample $I(T_{ij}^l = 1)$. Inferences based solely on maximization procedures are unable to produce this feature.

3. Simulations

To evaluate the performance of the proposed model, we generated a range of synthetic datasets consisting of hypothetical observations. The simulation scheme consisted of the following steps.

- *Protein generation and pathway assignment:* Fix a sample size m . Each protein is randomly assigned to k_l clusters using a vector of multinomial probabilities. Conditioned on the pathway membership j , the probability of a protein being present is denoted as ρ_j . We simulate protein indicators T_{ij} s as *Bernoulli*(ρ_j) distributions.
- *Connecting peptides and proteins:* Next, we simulate the number of peptide hits for a protein i in pathway j as $h_{ij} \sim \text{Poisson}(\lambda(T_{ij}))$. Conditioned on the set of protein indicators, the peptide generation follows the equations in (2.2). Due to positive mass at zero attained by Poisson distribution, some of the h_{ij} s will be 0. We discard those proteins which had 0 hits after the simulation. We then consider two methods of peptide assignment with (i) degeneracy and (ii) non-degeneracy. The degeneracy is induced by randomly assigning peptides to all non-zero hits. The lesser the number of peptides, the greater is the amount of degeneracy. We evaluate the amount of degeneracy in the dataset by considering two measures - the mean of the number of shared proteins and the proportion of peptides whose set of shared proteins is greater than or equal to 1. The first measure had a mean and median close to 5, and for a majority of simulated datasets, the proportion remained above 0.95. The non-degeneracy setting forces peptides not to be shared. After peptide assignment, the final step is to simulate scores.
- *Generating spectra scores:* The spectra scores are now generated conditioned on peptide indicators by the equations in (2.3).

We generated data using the above scheme. (For computational savings and simplicity of exposition, we did not include length information in the generation or the estimation of the model). However, we included length information for real data analyses in Section 4.

The simulation settings were indexed by two settings of sample sizes, 2 distinct values of ρ_j , 2 distinct values of (λ_0, λ_1) , and 2 distinct values of θ_c and θ_u each, under both degeneracy and non-degeneracy. So a total of $2 \times 2 \times 2 \times 2 \times 2 = 32$ scenarios were considered for each

of the two forms of peptide assignments. For each of these scenarios, we generated 10 datasets. The AUC corresponding to each setting were obtained by averaging over AUCs obtained from all 10 datasets. The exhaustive list of AUCs for all the different parameter combinations is reported in Table 1 (no degeneracy) and Table 2 (with degeneracy).

We expected the discriminative power of our model to be boosted with increasing distance between (i) the Poisson densities h_1 and h_0 and (ii) the score distributions f_u and f_c . The largest divergence between h_1 and h_0 was expressed in $(\lambda_1, \lambda_0) = (6, 1)$. Similarly, a significant distance between f_u and f_c were encoded by the settings: $\theta_u = (2, 1.5)$ and $\theta_c = (77, 11)$. This expectation for discriminative power was strongly confirmed by the entries in Tables 1 and 2. The highest AUC (0.98) was indeed obtained under the conditions of highest divergences between $h_1, h_0 \{ \lambda_1 = 6; \lambda_0 = 1 \}; f_u, f_c \{ \theta_u = (2, 1.5); \theta_c = (77, 11) \}$ and variable cluster assignments. For example, with sample size = 175, if both of the pairs (h_1, h_0) and (f_u, f_c) had a large distance between them, the AUC was found to be always greater than 0.95. Similarly, for sample size = 85, when both the pairs were significantly distant, the AUCs were close to 0.9 and above, while for the rest it dropped below. Also, the distance between λ_1 and λ_0 was seen to affect the AUC much more than the other parameters. The values of the AUC remained quite robust irrespective of the sample sizes. Overall, the results suggested a complex interplay of all parameters.

We also compared our values with NMM [18]. The results are added in the rightmost column of Tables 1 and 2. The main difficulty in executing the comparison was that the NMM code supplied by the authors stopped due to errors in the optimization function and in calculating log-likelihood values. So we computed the AUCs by taking an average only over all those simulated datasets which provided an answer. This number was 10 in some settings, and as low as 4 in some. Despite the bias incurred in the procedure, this was the best possible way that we could compare our results to NMM.

The comparison revealed some patterns with respect to degeneracy. First, under no degeneracy, the AUCs for both NMM and BHM are very close to each other in large samples, with NMM clearly showing superior performance (AUC above 0.9) for low samples. BHM for these settings remain in the range of (0.8–0.9). However with degeneracy, the BHM has a clear advantage. For all settings where the AUCs for NMM are greater, the median of the differences is around 0.02, while for those where BHM is superior, the median of the differences is 0.05. For 3 of these settings, it is even greater than 0.1. Except for the second to last entry in Table 2, the BHM seems uniformly superior to NMM whenever $\theta_c = (77, 11)$. However, both NMM and BHM seem to be powerful in the sense that the minimum AUC for both methods remain above 0.7. The minimum AUC across all settings in Tables 1 and 2 for BHM is 0.79 while for NMM it is 0.74. This difference in results is largely due to the way the Bayesian method handles degeneracy. The toy example in Section 2.1 clearly shows the sharing of information between the parameters, especially p_0 and p_1 . This sharing translates to sharing information between peptide classes, and helps in resolving some ambiguity in the protein assignment. NMM on the other hand fixes some of these parameters, like p_0 deterministically. Also we suspect that the feature of mixing of the parameters offered by Markov Chain sampling scheme may offer an advantage. For degenerate data, the joint posterior is very complex. An EM-type algorithm, which is

implemented by NMM, can run into local modes and by doing so miss many global features. BHM on the other hand, traverses the parameter space in a flexible way.

A brief discussion on the effect of using clustering prior in our model is given in the Supplementary pages.

4. Data Analysis

We applied our methods to two different proteomic experiments. To run our model, we required pre-specified clusters. These were obtained from DAVID (the Database for Annotation, Visualization and Integrated Discovery) - a free online bioinformatics resource [14–15]. DAVID classifies the genes/proteins into functionally related groups with enrichment scores. For our datasets, we considered protein clusters based on these enrichment scores. To make the protein clusters mutually exclusive, we considered a protein in that cluster which had the maximum enrichment score. The cluster sizes and the number of clusters are provided in Tables S1 and S2 in the Supplementary. The lengths of the proteins were extracted from the NCBI website (<http://www.ncbi.nlm.nih.gov>).

For both datasets, the MS/MS spectra were matched against a combined search database containing ‘target’ proteins as well as ‘decoy’ proteins. Using decoy databases is now a common and convenient way to assess a protein identification algorithm [6, 16, 23]. Using this strategy, any ‘identified’ protein present as a target in the database can be considered a true positive. Similarly any identified protein corresponding to a ‘decoy’ is termed a false positive. Consequently, our true positive counts could be slightly inflated. However, both are valid estimates in the context of relative comparison between methods. The bias introduced cancels out among the different methods, since they use the same decoy data. A discussion on the merits of decoy databases can be found in [27]. For the following discussion, we still retained the terminology ‘true positives’ while referring to the target proteins in the database.

4.1 Yeast Data

Spectra scores of the yeast (*Saccharomyces cerevisiae*) peptide fragments were collected from a QSTAR mass spectrometer [6]. The scoring algorithm SEQUEST was used to match the generated MS/MS spectra against a combined search database that has both target (yeast) proteins as well as proteins from another species, ‘*C.elegans*’ (*Caenorhabditis elegans*). The latter played the role of ‘decoy’ proteins. This strategy provided us with ready estimates of the empirical false discovery rates, which is obtained simply based on the number of decoy proteins identified by the algorithm.

The combined database consisted of 6473 entries of yeast (*S.cerevisiae*) and 22,437 entries of *C.elegans*. In total, 9272 MS/MS spectra were assigned to 4148 unique peptides. This dataset was previously analyzed in [28]. We followed the pre-processing steps of [28], and excluded 13 charge +1 peptides. The remaining 4135 peptides consisted of 3516 yeast and 619 *C.elegans* peptides. These peptides corresponded to 1011 yeast proteins and 876 *C.elegans* proteins. We initialized the parameters as follows: $(\lambda_1, \lambda_0) = (0.05, 0.02)$, $(p_1, p_0) = (0.2, 0)$. To initialize θ_c and θ_u , we partitioned the peptides into two sets D_1 and D_0 based on

whether the corresponding spectra scores was above or below the median score. The parameters μ and σ were initialized to the Gaussian MLEs for D_0 . The parameters α and β were initialized to MLEs from a shifted Gamma distribution. The MLE of the shift parameter was fixed across the entire span of MCMC iterations.

A similar unbiasedness was maintained with respect to priors. We set the hyper-parameters to make them as uninformative as possible. Specifically, the hyper parameters for $\log(\alpha)$ and $\log(\beta)$ were assigned priors $\mathcal{N}(0, 10^{10})$. The probability parameters (p_1, p_0, p_j) had $Uniform(0, 1)$ priors. The parameters μ and σ were provided the conventional uninformative location-scale priors [31]. Next, we executed the steps outlined in Section 2 iteratively for 6000 iterations. Posterior samples were collected with a post burn-in sample of 5950. The posterior summaries were highly robust to the number of iterations and number of burn-in samples. We compared our performance on this dataset with HSM and NMM. Such analyses were not possible unfortunately with ProteinProphet, since neither the raw posterior probabilities nor the raw data files were available to us. The only relevant results for ProteinProphet we had was selected values at some specific probability thresholds, which was insufficient to produce a TP/FP curve for full blown comparison. As discussed in [18], the processed version lacks key processing steps by PeptideProphet and ProteinProphet, making a fair comparison untenable. Finally, to show the effect of the clustering prior, we applied a poorer version of the BHM that suppressed the clustering information. This was done by constraining the cluster-specific probabilities to be equal in the updating steps.

For a comprehensive comparison, we plotted the number of true detected proteins, or true positives (TPs) against false positives (FPs) in Figure 2. Note that, here ‘BHM’ represents the proposed Bayesian Hierarchical Model. The aim was to investigate the long-range dynamics of these methods across a wide range of posterior probabilities. We refer to this curve as the TP/FP curve.

Each point in the three curves of the plot corresponded to true and false proteins detected at a fixed posterior threshold for the three methods. The horizontal axis shows the FP count attained by our algorithm as we varied the posterior threshold from 0 to 1. In this range of probabilities, the minimum false positive count for our method is 6 and the maximum is 30. Since we were unable to run the code supplied to us by the authors of [28] for HSM, we extracted the points for the curve from Figure 8 in [18].

We found that our method outperformed both HSM and NMM. That is, at each fixed number of false positives, our method picks up more true positives than other methods across the considered range of cutoffs. Though our method is uniformly better over the range of recorded values in Figure 2, it comes closest to the performance of NMM. HSM ranks still lower, with number of TPs ranging only from 358 to 390. The other two algorithms, in comparison, show a smoother ascent, implying that TPs and FPs are sensitive to respective posterior model probabilities, which is a feature indicative of a good model calibration. The BHM (without clustering) dominates the performance of HSM, but is clearly dominated by the complete version of the proposed BHM algorithm, indicating the effect of clustering.

4.2 *H.influenzae* Data

Haemophilus influenzae is an anaerobic bacterium responsible for a wide range of localized and invasive infections. *H.influenzae* proteins were digested with multiple enzymes (including trypsin) and profiled using LC-MS/MS. The resultant spectra were subsequently searched with SEQUEST against a database that contained *H.influenzae* targets along with a decoy database of human proteins. This dataset is identical to the one used in [27] and was provided to us by [24]. The *H.influenzae* database has 476 proteins, of which 269 are true proteins, and 207 are decoy. These proteins correspond to 1011 unique peptides. No further preprocessing was done on the data obtained.

Similar to the yeast dataset, a full analysis used the length information of the proteins obtained from the same database search. However, unlike the yeast dataset, some protein lengths were extremely high. Overall, we found a highly skewed distribution for these lengths. To negate the influence of extreme observations, we capped the protein lengths at 6000. Also, as in yeast dataset, we compared our performance to that of a version that retains all the features of BHM except that it does not use clustering prior.

In practice with realistic examples such as this one, there is a tradeoff between detecting true positives and avoiding false positives, and the performance of each method will depend on an arbitrary selection of criteria for controlling the false positives. From Figure 3 we see that our method (BHM) performs superior to both ProteinProphet and NMM in the number of true positives detected when the number of false positives exceeds 2. Again, the BHM with clustering prior clearly outperforms a version that does not use clustering information, indicating the usefulness of a clustering prior.

Interestingly, Serang et al. [27] analyzed the same data and their results are shown graphically in Figure 4a of [27]. We observe that their curve almost overlaps with that of ProteinProphet for this dataset. This result along with superiority of our method over ProteinProphet indicates that our proposed model performs well compared with the other algorithms. The analyses of these two datasets are suggestive of the robustness and applicability of our method to very different proteomes experiments.

A brief discussion on the convergence behavior of all the model parameters for both the yeast and *H.influenzae* datasets are given in the Supplementary pages.

5. Discussion

Protein identification is a complex stochastic problem demanding a unified statistical framework. In this paper, we proposed a Bayesian hierarchical model to address some key aspects of the protein generation mechanism. Naive probabilistic approaches assume that the peptides generated from a protein contribute independently to the presence of proteins. The hierarchical model modifies this approach to provide joint posterior probabilities of proteins and peptides. Our approach does share certain aspects of a hierarchical model with HSM and NTT. However, we can deal with degeneracy unlike NTT. We improve upon the independence assumption of the proteins by incorporating the pathway information unlike HSM and NTT. We allow conditional independence of peptides given the proteins and that

of spectra given the peptides, and we use prior distributions to model the binary indicators of protein and peptide presence. Additionally, we perform full Bayesian inference, which could be the potential reason for substantial differences in the final output from these algorithms. Among other novel features, our algorithm can produce valid uncertainty estimates of the identifications. The algorithm is also computationally stable as it is robust to the number of iterations. The results for this paper also show very good mixing behavior removing the concern of posterior convergence.

This modeling scheme is naturally equipped to integrate various sources of extraneous information and will lead to interesting future work. With advances in experimental proteomics, more extraneous information will be readily available.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

Funding statement

This research was supported by the National Institute of Health (NCI-NIH) grant CA170091.

References

- [1]. Aebersold R, and Mann M, Mass spectrometry-based proteomics, *Nature*. 422 (2003), pp. 198–207. [PubMed: 12634793]
- [2]. Alves G, Ogurtsov AY, and Yu YK, RAId_DbS: peptide identification using database searches with realistic statistics, *Biol. Direct* 2 (2007), pp. 25. [PubMed: 17961253]
- [3]. Alves G, and Yu YK, Mass spectrometry-based protein identification with accurate statistical significance assignment, *Bioinformatics* 31 (2015), pp. 699–706. [PubMed: 25362092]
- [4]. Bafna V, and Edwards N, SCOPE: a probabilistic model for scoring tandem mass spectra against a peptide database, *Bioinformatics* 17 (2001), pp. S13–S21. [PubMed: 11472988]
- [5]. Craig R, and Beavis RC, TANDEM: matching proteins with tandem mass spectra, *Bioinformatics* 20 (2004), pp. 1466–1467. [PubMed: 14976030]
- [6]. Elias JE, and Gygi SP, Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry, *Nat. Methods* 4 (2007), pp. 207–214. [PubMed: 17327847]
- [7]. Eng JK, McCormack AL, and Yates JR, An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database, *J. Am. Soc. Mass Spectrom* 5 (1994), pp. 976–989. [PubMed: 24226387]
- [8]. Feng J, Naiman DQ, and Cooper B, Probability model for assessing proteins assembled from peptide sequences inferred from tandem mass spectrometry data, *Anal. Chem* 79 (2007), pp. 3901–3911. [PubMed: 17441689]
- [9]. Geer LY, Markey SP, Kowalak JA, Wagner L, Xu M, Maynard DM, Yang X, Shi W, and Bryant SH, Open mass spectrometry search algorithm, *J. Proteome Res* 3 (2004), pp. 958–964. [PubMed: 15473683]
- [10]. Gerster S, Qeli E, Ahrens CH, and Bühlmann P, Protein and gene model inference based on statistical modeling in k-partite graphs, *Proc. Natl. Acad. Sci. USA* 107 (2010), pp. 12101–12106. [PubMed: 20562346]
- [11]. Geweke J, [Statistics, Probability and Chaos]: Comment: Inference and Prediction in the Presence of Uncertainty and Determinism, *Statist. Sci* 7 (1992), pp. 94–101.

- [12]. Gygi SP, Rist B, Gerber SA, Turecek F, Gelb MH, and Aebersold R, Quantitative analysis of complex protein mixtures using isotope-coded affinity tags, *Nat. Biotechnol* 17 (1999), pp. 994–999. [PubMed: 10504701]
- [13]. Howbert JJ, and Noble WS, Computing exact p-values for a cross-correlation shotgun proteomics score function, *Mol. Cell Proteomics* 13 (2014), pp. 2467–2479. [PubMed: 24895379]
- [14]. Huang da W, Sherman BT, and Lempicki RA, Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists, *Nucleic Acids Res.* 37 (2009), pp. 1–13. [PubMed: 19033363]
- [15]. Huang da W, Sherman BT, and Lempicki RA, Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources, *Nat. Protoc* 4 (2009), pp. 44–57. [PubMed: 19131956]
- [16]. Käll L, Storey JD, MacCoss MJ, and Noble WS, Assigning significance to peptides identified by tandem mass spectrometry using decoy databases, *J. Proteome Res* 7 (2008), pp. 29–34. [PubMed: 18067246]
- [17]. Kim S, Gupta N, and Pevzner PA, Spectral probabilities and generating functions of tandem mass spectra: a strike against decoy databases, *J. Proteome Res* 7 (2008), pp. 3354–3363. [PubMed: 18597511]
- [18]. Li Q, MacCoss MJ, and Stephens M, A nested mixture model for protein identification using mass spectrometry, *Ann. Appl. Stat* 4 (2010) pp. 962–987.
- [19]. Li Q, Eng JK, and Stephens M, A likelihood-based scoring method for peptide identification using mass spectrometry, *Ann. Appl. Stat* 6 (2012), pp. 1775–1794.
- [20]. Li YF, Arnold RJ, Li Y, Radivojac P, Sheng Q, and Tang H H, A Bayesian approach to protein inference problem in shotgun proteomics, *J. Comput. Biol* 16 (2009), pp. 1183–1193. [PubMed: 19645593]
- [21]. Link AJ, Eng J, Schieltz DM, Carmack E, Mize GJ, Morris DR, Garvik BM, and Yates JR 3rd, Direct analysis of protein complexes using mass spectrometry, *Nat. Biotechnol* 17 (1999), pp. 676–682. [PubMed: 10404161]
- [22]. MacCoss MJ, Wu CC, Liu H, Sadygov R, and Yates JR 3rd, A correlation algorithm for the automated quantitative analysis of shotgun proteomics data, *Anal. Chem* 75 (2003), pp. 6912–6921. [PubMed: 14670053]
- [23]. Moore RE, Young MK, and Lee TD, Qscore: an algorithm for evaluating SEQUEST database search results, *J. Am. Soc. Mass Spectrom* 13 (2002) pp. 378–386. [PubMed: 11951976]
- [24]. Nesvizhskii AI, Keller A, Kolker E, and Aebersold R, A statistical model for identifying proteins by tandem mass spectrometry, *Anal. Chem* 75 (2003), pp. 4646–4658. [PubMed: 14632076]
- [25]. Perkins DN, Pappin DJ, Creasy DM, and Cottrell JS, Probability-based protein identification by searching sequence databases using mass spectrometry data, *Electrophoresis* 20 (1999) pp. 3551–3567. [PubMed: 10612281]
- [26]. Sadygov RG, and Yates JR 3rd, A hypergeometric probability model for protein identification and validation using tandem mass spectral data and protein sequence databases, *Anal. Chem* 75 (2003), pp. 3792–3798. [PubMed: 14572045]
- [27]. Serang O, MacCoss MJ, and Noble WS, Efficient marginalization to compute protein posterior probabilities from shotgun mass spectrometry data, *J. Proteome Res* 9 (2010), pp. 5346–5357. [PubMed: 20712337]
- [28]. Shen C, Wang Z, Shankar G, Zhang X, and Li L, A hierarchical statistical model to assess the confidence of peptides and proteins inferred from tandem mass spectrometry, *Bioinformatics* 24 (2008), pp. 202–208. [PubMed: 18024968]
- [29]. Spirin V, Shpunt A, Seebacher J, Gentzel M, Shevchenko A, Gygi S, and Sunyaev S, Assigning spectrum-specific p-values to protein identifications by mass spectrometry, *Bioinformatics* 27 (2011), pp. 1128–1134. [PubMed: 21349864]
- [30]. Tabb DL, Fernando CG, and Chambers MC, MyriMatch: highly accurate tandem mass spectral peptide identification by multivariate hypergeometric analysis, *J. Proteome Res* 6 (2007), pp. 654–661. [PubMed: 17269722]
- [31]. Yang R, and Berger JO. A catalog of noninformative priors. Institute of Statistics and Decision Sciences, Duke University, 1996.

Figure 1:

A toy example showing the protein identification process through an MS/MS experiment.

Figure 2:
Comparative performances among BHM, NMM, HSM, and BHM (without clustering) using yeast proteomics experiment including length information.

Figure 3:
Comparisons among BHM, NMM, ProteinProphet and BHM (without clustering) using *H.influenzae* proteomics experiment.

Table 1:

Comparison of BHM and NMM using the average AUC values corresponding to several choices of the model parameters and the sample size under no degeneracy

Sample Size	ρ_j	(λ_1, λ_0)	θ_u	θ_c	Average AUC	
					BHM	NMM
175	(0.6,0.8,0.4,0.3)	(6,1)	(2.8,0.9)	(77,11)	0.980	0.982
				(40,9.3)	0.975	0.979
			(2,1.5)	(77,11)	0.981	0.984
				(40,9.3)	0.974	0.980
		(4,1)	(2.8,0.9)	(77,11)	0.941	0.939
				(40,9.3)	0.923	0.918
			(2,1.5)	(77,11)	0.933	0.937
				(40,9.3)	0.924	0.924
	(0.7,0.7,0.7,0.7)	(6,1)	(2.8,0.9)	(77,11)	0.980	0.973
				(40,9.3)	0.975	0.980
			(2,1.5)	(77,11)	0.981	0.978
				(40,9.3)	0.975	0.979
		(4,1)	(2.8,0.9)	(77,11)	0.917	0.931
				(40,9.3)	0.848	0.918
			(2,1.5)	(77,11)	0.904	0.929
				(40,9.3)	0.889	0.920
85	(0.6,0.8,0.4,0.3)	(6,1)	(2.8,0.9)	(77,11)	0.975	0.979
				(40,9.3)	0.969	0.975
			(2,1.5)	(77,11)	0.976	0.982
				(40,9.3)	0.973	0.977
		(4,1)	(2.8,0.9)	(77,11)	0.920	0.910
				(40,9.3)	0.908	0.914
			(2,1.5)	(77,11)	0.924	0.920
				(40,9.3)	0.898	0.918
	(0.7,0.7,0.7,0.7)	(6,1)	(2.8,0.9)	(77,11)	0.916	0.979
				(40,9.3)	0.919	0.971
			(2,1.5)	(77,11)	0.911	0.978
				(40,9.3)	0.926	0.968
		(4,1)	(2.8,0.9)	(77,11)	0.897	0.942
				(40,9.3)	0.829	0.930
			(2,1.5)	(77,11)	0.878	0.946
				(40,9.3)	0.879	0.930

Table 2:

Comparison of BHM and NMM using the average AUC values corresponding to several choices of the model parameters and the sample size under degeneracy

Sample Size	ρ_j	(λ_1, λ_0)	θ_u	θ_c	Average AUC	
					BHM	NMM
175	(0.6,0.8,0.4,0.3)	(6,1)	(2.8,0.9)	(77,11)	0.967	0.898
				(40,9.3)	0.974	0.963
			(2,1.5)	(77,11)	0.968	0.948
				(40,9.3)	0.965	0.961
		(4,1)	(2.8,0.9)	(77,11)	0.913	0.810
				(40,9.3)	0.859	0.881
			(2,1.5)	(77,11)	0.885	0.839
				(40,9.3)	0.878	0.856
	(0.7,0.7,0.7,0.7)	(6,1)	(2.8,0.9)	(77,11)	0.965	0.810
				(40,9.3)	0.963	0.967
			(2,1.5)	(77,11)	0.962	0.918
				(40,9.3)	0.963	0.959
		(4,1)	(2.8,0.9)	(77,11)	0.860	0.802
				(40,9.3)	0.859	0.879
			(2,1.5)	(77,11)	0.865	0.864
				(40,9.3)	0.846	0.849
85	(0.6,0.8,0.4,0.3)	(6,1)	(2.8,0.9)	(77,11)	0.953	0.900
				(40,9.3)	0.947	0.965
			(2,1.5)	(77,11)	0.955	0.921
				(40,9.3)	0.946	0.960
		(4,1)	(2.8,0.9)	(77,11)	0.867	0.740
				(40,9.3)	0.883	0.911
			(2,1.5)	(77,11)	0.860	0.803
				(40,9.3)	0.878	0.750
	(0.7,0.7,0.7,0.7)	(6,1)	(2.8,0.9)	(77,11)	0.934	0.864
				(40,9.3)	0.937	0.957
			(2,1.5)	(77,11)	0.925	0.910
				(40,9.3)	0.938	0.953
		(4,1)	(2.8,0.9)	(77,11)	0.795	0.738
				(40,9.3)	0.807	0.875
			(2,1.5)	(77,11)	0.790	0.767
				(40,9.3)	0.798	0.858