A Bayesian Approach to Protein Inference Problem in Shotgun Proteomics

Yong Fuga Li[1], Randy J. Arnold[2], Yixue Li[3], Predrag Radivojac[1], Quanhu Sheng[1,3], and Haixu Tang[1]

[1] School of Informatics, Indiana University, Bloomington, IN 47408, USA
[2] Department of Chemistry, Indiana University, Bloomington, IN 47405, USA
[3] Key Lab of Systems Biology, Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences, Shanghai, China

Abstract. The protein inference problem represents a major challenge in shotgun proteomics. Here we describe a novel Bayesian approach to address this challenge that incorporates the predicted peptide detectabilities as the prior probabilities of peptide identification. Our model removes some unrealistic assumptions used in previous approaches and provides a rigorious probabilistic solution to this problem. We used a complex synthetic protein mixture to test our method, and obtained promising results.

1 Introduction

In shotgun proteomics, a complex protein mixture derived from a biological sample is directly analyzed via a sequence of experimental and computational procedures [1,2,3,4]. After protease digestion, liquid chromatography (LC) coupled with tandem mass spectrometry (MS/MS) is typically used to separate and fragment peptides from the sample, resulting in a number of MS/MS spectra. These spectra are subsequently searched against a protein database to identify peptides present in the sample [5,6]. Many peptide search engines have been developed, among which Sequest [7], Mascot [8] and X!Tandem [9] are commonly used. However, after a reliable set of *peptides* is identified, it is often not straightforward to assemble a reliable list of *proteins* from these peptides. This occurs because some identified peptides, referred to as the *degenerate peptides*, are shared by two or more proteins in the database. As a result, the problem of determining which of the proteins are indeed present in the sample, known as the *protein inference problem* [10], often has multiple solutions and can be computationally intractable. Nesvizhskii and colleagues first addressed this challenge using a probabilistic model [11], but different problem formulations and new solutions have recently been proposed as well [10,12,13].

Previously, we introduced a combinatorial approach to the protein inference problem that incorporates the concept of *peptide detectability*, i.e. the probability of a peptide to be detected (identified) in a standard proteomics experiment, with the goal of finding the set of proteins with the minimal number of *missed peptides* [12]. As in the other combinatorial formulations [13], the *parsimony*

M. Vingron and L. Wong (Eds.): RECOMB 2008, LNBI 4955, pp. 167–180, 2008.
© Springer-Verlag Berlin Heidelberg 2008

condition was chosen only for convenience reasons, without theoretical justification. Furthermore, parsimonious formulations often lead to the *minimum cover set problem*, which is NP-hard. Thus, heuristic algorithms following greedy [12] or graph-pruning strategies [13] are used to solve the protein inference problem without performance guarantee.

In this paper, we address protein inference by proposing two novel Bayesian models that take as input a set of identified peptides from any peptide search engine, and attempt to find a most likely set of proteins from which those identified peptides originated. The basic model assumes that all identified peptides are correct, whereas the advanced model also accepts the probability of each peptide to be present in the sample. Compared with the previous probabilistic models, such as ProteinProphet [11], both of our models differ in two key aspects. First, our approach incorporates peptide detectability [14] since it has been recently shown that even among the peptides that belong to the same protein, some peptides are commonly observed, while some others are not [14,15]. This results in the fact that the peptides not identified by peptide search engines may have significant impact on the final solution. Second, previous models assume that the posterior probability of each peptide is independent of other peptides and can be computed separately. Although this assumption significantly simplifies the computation of the protein posterior probabilities, it is inconsistent with the Bayesian model of a shotgun proteomics experiment (see Materials and Methods). We relax this assumption and adopt Gibbs sampling approach to estimate protein posterior probabilities. The results of this study provide evidence that our models achieve satisfactory accuracy and can be readily used in protein identification.

2 Materials and Methods

To illustrate the challenge of protein inference, we define the *protein configuration graph* (Fig. 1(a)), i.e. a bipartite graph in which two disjoint sets of vertices represent the proteins in the database and the peptides from these proteins, respectively, and where each edge indicates that the peptide belongs to the protein. We emphasize that the protein configuration graph is independent of the proteomics experiment, and thus can be built solely from a set (database) of protein sequences. Therefore, in constrast to the bipartite graph used previously [13], where only the identified peptides and the proteins that contain those peptides were represented, our model also considers the non-identified peptides. Protein configuration graph is partitioned into *connected components*, each representing a group of proteins (e.g. homologous protein families) sharing one or more (degenerate) peptides. If there are no degenerate peptides in the database, each connected component will contain exactly one protein and its peptides. In practice, however, the protein configuration graph may contain large connected components, especially for protein databases of higher animals or those containing closely related species.

Given that the protein configuration graph can be interpreted as a Bayesian network with edges pointing from proteins into peptides, it is straightforward

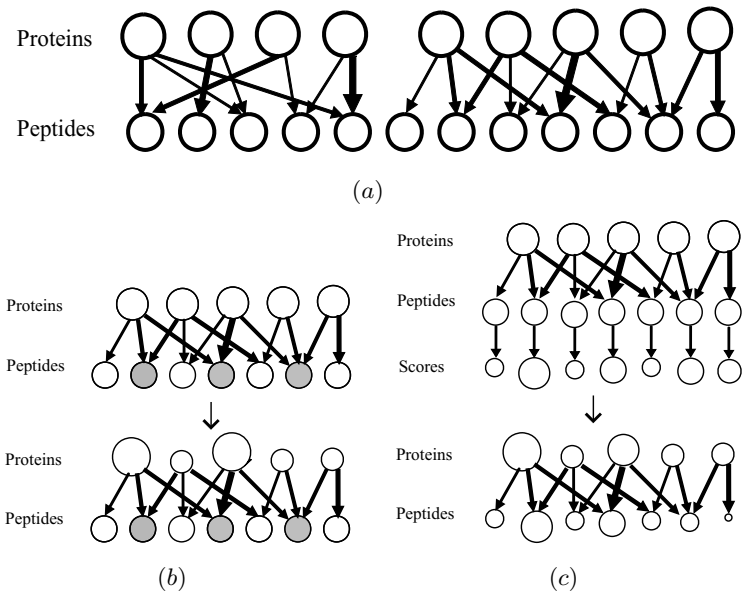

Fig. 1. (a) A protein configuration graph consisting of two connected components; (b) basic Bayesian model for protein inference, in which peptides are represented as a vector of indicator variables: 1 (grey) for identified peptides, and 0 (white) for non-identified peptides; (c) advanced Bayesian model for protein inference, in which each peptide is associated to an identification score (0 for non-identified peptides). Sizes of circles reflect prior/posterior probabilities.

to show that protein inference can be addressed separately for each individual connected component. In this approach, the peptide identification results are first mapped to the protein configuration graph. We use a vector of indicator variables $(y_1, ..., y_j, ..., y_n)$, referred to as the *peptide configuration* to denote a set of identified peptides. Given the peptide configuration, a connected component of the protein configuration graph is called *trivial* if it contains no identified peptides. Clearly, in this case protein inference is simple – none of the proteins should be present in the sample. Therefore, the protein inference problem can be reduced to finding the most likely *protein configuration* $(x_1, ..., x_i, ..., x_m)$ by analyzing *non-trivial* components only. In the basic model, all identified peptides are assigned equal probabilities (= 1) (Fig. 1(b)), whereas in the advanced model different probabilities are considered for different identified peptides depending on the associated identification scores $(s_1, ..., s_j, ..., s_n)$ (Fig. 1(c)). Notation and definitions used in this study are summarized in Table 1.

2.1 Basic Bayesian Model

In the basic probabilistic model, we assume that each identified peptide has an equally high prior probability to be present in the sample and low false discovery rate (FDR) in the results of peptide identification. In practice, even though this

Table 1. Notations and definitions

Notation	Definition
$(1, ..., i, ..., m)$	m proteins within a *non-trivial* connected component of the *protein configuration graph*
$(x_1, ..., x_i, ..., x_m)$	*protein configuration*: indicator variables of proteins' presences
$(1, ..., j, ..., n)$	all n peptides from m proteins being considered
$(Z_{11}, ..., Z_{ij}, ..., Z_{mn})$	indicator variables of peptide j belonging to protein i if peptide j is a peptide from protein i, $Z_{ij} = 1$; otherwise $Z_{ij} = 0$
$(y_1, ..., y_j, ..., y_n)$	*peptide configuration*: indicator variables of peptides' presences if peptide j is present, $y_j = 1$; otherwise $y_j = 0$
$(s_1, ..., s_j, ..., s_n)$	assigned scores of peptides if peptide j is not identified (i.e. $y_j = 0$), $s_j = 0$
$(r_1, ..., r_j, ..., r_n)$	probabilities of peptide being correctly identified also the estimated probabilities of peptides' presences
$(LR_1, ..., LR_j, ..., LR_n)$	likelihood ratio between peptides's presences and absences
$(d_{11}, ..., d_{ij}, ..., d_{mn})$	prior probabilities of peptides to be identified from proteins if $Z_{ij} = 1$, d_{ij} = the detectability of peptide j from protein i; otherwise, $d_{ij} = 0$

assumption does not completely hold, peptide FDRs are usually controlled at a low level (e.g. 0.01) by either a heuristic target-decoy search strategy [16,13,17] or by probabilistic modeling of random peptide identification scores [18,19,20]. In the next section, we extend this basic model to a more realistic model in which we incorporate different probabilities for different identified peptides that are estimated based on the peptide identification scores.

Consider m proteins and n peptides from these proteins within a non-trivial connected component of the protein configuration graph. Each protein i is either present in the sample or absent from it, which can be represented by an indicator variable x_i. Therefore, any solution of the protein inference problem corresponds to a vector of indicator variables, $(x_1, ..., x_m)$, referred to as a protein configuration. Given the set of identified peptides from peptide search engines (peptide configuration $(y_1, ..., y_n)$), our goal is to find the *maximum a posteriori* (MAP) protein configuration, that is the configuration that maximizes the posterior probability $P(x_1, ..., x_m|y_1, ..., y_n)$. Using the Bayes' rule, this posterior probability can be expressed as

$$\begin{aligned} P(x_1, ..., x_m|y_1, ..., y_n) &= \frac{P(x_1,...,x_m)P(y_1,...,y_n|x_1,...,x_m)}{\sum_{(x_1,...,x_m)}[P(x_1,...,x_m)P(y_1,...,y_n|x_1,...,x_m)]} \\ &= \frac{P(x_1,...,x_m)\prod_j[1-Pr(y_j=1|x_1,...,x_m)]^{1-y_j}Pr(y_j=1|x_1,...,x_m)^{y_j}}{\sum_{(x_1,...,x_m)}P(x_1,...,x_m)\prod_j[1-Pr(y_j=1|x_1,...,x_m)]^{1-y_j}Pr(y_j=1|x_1,...,x_m)^{y_j}} \end{aligned} \tag{1}$$

where $P(x_1, ..., x_m)$ is the prior probability of the protein configuration. Assuming the presence of each protein i is independent of other proteins, this prior probability can be computed as

$$P(x_1, ..., x_m) = \prod_i P(x_i) \tag{2}$$

$Pr(y_j = 1|x_1, ..., x_m)$ is the probability of peptide j to be identified by shotgun proteomics given the protein configuration $(x_1, ..., x_m)$. Assuming that different proteins are present in the sample independently of one another and ignoring the competition of peptides for ionization and MS/MS fragmentation, we can compute it as

$$Pr(y_j = 1|x_1, ..., x_m) = 1 - \prod_i [1 - x_i Pr(y_j = 1|x_i = 1, x_1 = ... = x_{i-1} = x_{i+1} = ... = x_m = 0)] \tag{3}$$

where $Pr(y_j = 1|x_i = 1, x_1 = ... = x_{i-1} = x_{i+1} = ... = x_m = 0)$ is the probability of peptide j to be identified if only protein i is present in the sample. As we previously showed, for a particular proteomics platform (e.g. LC-MS/MS considered here), this probability, referred to as the *standard peptide detectability* d_{ij}, is an intrinsic property of the peptide (within its parent protein), and can be predicted from the peptide and protein sequence prior to a proteomics experiment [14]. Combining equations above, we can compute the posterior probabilities for protein configurations as

$$P(x_1, ..., x_m|y_1, ..., y_n) = \frac{\prod_i P(x_i) \prod_j \{[\prod_i (1 - x_i d_{ij})]^{1-y_j} [1 - \prod_i (1 - x_i d_{ij})]^{y_j}\}}{\sum_{(x'_1,...,x'_m)} \prod_i P(x'_i) \prod_j \{[\prod_i (1 - x'_i d_{ij})]^{1-y_j} [1 - \prod_i (1 - x'_i d_{ij})]^{y_j}\}} \tag{4}$$

Hence, protein inference is equivalent to finding the MAP protein configuration by maximizing the above function

$$(x_1, ..., x_m)_{MAP} = argmax_{(x_1,...,x_m)} P(x_1, ..., x_m|y_1, ..., y_n) \tag{5}$$

Sometimes, we are also interested in the marginal posterior probability of a specific protein i to be present in the sample, which can be expressed as,

$$P^o(x_i) = P(x_i|y_1, ..., y_n) = \sum_{x_1,...,x_{i-1},x_{i+1},...,x_m} P(x_1, ..., x_m|y_1, ..., y_n) \tag{6}$$

2.2 Advanced Bayesian Model

The basic model described above assumes all identified peptides have equal probability (= 1) of being correctly identified. Here we relax this assumption by introducing a peptide identification score s_j for each peptide j, which is output by peptide search engines. We assume the peptide identification score is highly correlated with the probability of a peptide being correctly identified and their relationship (denoted by $r_j = Pr(y_j = 1|s_j)$) can be approximately modeled using probabilistic methods adopted by some search engines such as Mascot [8] or post-processing tools such as PeptideProphet [18]. Our goal is to compute $P(x_1, ..., x_m|s_1, ..., s_n)$ by enumerating all potential peptide configurations

$$P(x_1,...,x_m|s_1,...,s_n) = \sum_{(y_1,...,y_n)}[P(x_1,...,x_m|y_1,...,y_n)P(y_1,...,y_n|s_1,...,s_n)]$$
$$= \sum_{(y_1,...,y_n)}[\frac{P(x_1,...,x_m)}{P(s_1,...,s_n)}P(y_1,...,y_n|x_1,...,x_m)P(s_1,...,s_n|y_1,...,y_n)] \quad (7)$$

Assuming that s_j is independent of the presences of the other peptides (i.e. $(y_1,...,y_{j-1},y_{j+1},...,y_n)$) and applying Bayes' rule, we have

$$P(s_1,...,s_n|y_1,...,y_n) = \prod_j \frac{P(y_j|s_j)P(s_j)}{P(y_j)} = \prod_j \frac{(1-r_j)^{(1-y_j)}r_j^{y_j}P(s_j)}{P(y_j)} \quad (8)$$

Combining these equations, we can compute the posterior probability of protein configurations as

$$P(x_1,...,x_m|s_1,...,s_n) = \frac{\sum_{(y_1,...,y_n)}\{\prod_i P(x_i)\prod_j\{[\prod_i(1-x_id_{ij})]^{1-y_j}[1-\prod_i(1-x_id_{ij})]^{y_j}\frac{(1-r_j)^{(1-y_j)}r_j^{y_j}}{P(y_j)}\}\}}{\sum_{(x'_1,...,x'_m)(y_1,...,y_n)}\{\prod_i P(x'_i)\prod_j\{[\prod_i(1-x'_id_{ij})]^{1-y_j}[1-\prod_i(1-x'_id_{ij})]^{y_j}\frac{(1-r_j)^{(1-y_j)}r_j^{y_j}}{P(y_j)}\}\}} \quad (9)$$

For most of the work presented here, we do not assume any prior knowledge about the protein presence in the sample. Therefore, in equations 4 and 9, $P(x_i)$ is regarded as constant (i.e. 0.5) for all proteins. In practice, prior knowledge, such as the species which the sample is from, the number of candidate proteins, and known protein relative quantities or protein families that are likely present in the sample, can be directly integrated into our Bayesian models (see results section for a simple demonstration).

Similar to the basic model, we can also compute the posterior probability of a specific protein i present in the sample as

$$P(x_i|s_1,...,s_n) = \sum_{x_1,...,x_{i-1},x_{i+1},...,x_m} P(x_1,...,x_m|s_1,...,s_n) \quad (10)$$

and the marginal probability of a peptide j in the sample (see Appendix).

2.3 Adjustment of Peptide Detectabilities

An adjustment of the predicted peptide detectabilities is necessary when applying them here, since the predicted standard peptide detectabilities (denoted as d_{ij}^0) reflect the detectability of a peptide under a standard proteomics experimental setting, in particular, under fixed and equal abundances (i.e. q^0) for all proteins [14]. Assuming that the abundance of protein i in the sample mixture is q_i instead of q^0, the *effective* detectability of peptide j from this protein should be adjusted to

$$d_{ij} = 1 - (1 - d_{ij}^0)^{q_i/q^0} \quad (11)$$

Although we do not know q_i explicitly, since the total probability of observing a peptide j is given by r_j (or y_j for basic model), we can estimate q_i by solving

the equation $\sum_j d_{ij} = \sum_j Z_{ij} r_j$ for a specific protein i. We note that this adjustment method may immediately lead to a new approach to absolute protein quantification [15]. However, we will address the evaluation of its performance in our future work. Here, our goal is to utilize it to adjust the predicted standard peptide detectabilities based on the estimated protein abundances.

2.4 Gibbs Sampling

Given a protein configuration graph, the peptide detectabilities (d_{ij}) and the probabilities of peptide presence in the sample (r_j), the posterior distribution of protein configurations can be computed directly from equations 4 or 9, depending on which Bayesian model is used. This brute force method, which has computational complexity of $O(2^m)$, is very expensive and only works for small connected components in the protein configuration graph.

Gibbs sampling is a commonly used strategy to rapidly approximate a high dimensional joint distribution that is not explicitly known [21,22]. We adopted this algorithm to achieve the optimal protein configuration with the MAP probability. The original Gibbs sampling algorithm considers one individual variable at a time in the multi-dimentional distribution. It, however, often converges slowly and is easily trapped by local maxima for long time. Several techniques have been proposed to improve the search efficiency of Gibbs sampling algorithm, such as *random sweeping*, *blocking* and *collapsing* [22]. Because in our case each variable x_i to be sampled has small search space (i.e. $\{0, 1\}$), we applied the blocking sampling technique in our Gibbs sampler algorithm.

Without increasing the computational complexity, we adopt a novel *memorizing* strategy that keeps a record of all (as well as the maximum) posterior probabilities (and the corresponding protein configurations) among all configurations we evaluated during the sampling procedure, and report the maximum solution in the end. The memorized posterior probabilities are also used to calculate the marginal posterior protein probabilities in equation 6 and 9. Due to the page limits, the sketch of the block gibbs sampling algorithms and the memorizing approach for the basic and advanced Bayesian models are described in Appendix.

2.5 Datasets

We used two datasets from different sources that are both generated using mixtures of model proteins. Therefore, we know the proteins in these samples. The first dataset is used only for the training of the detectability predictor, while the other dataset was used for testing the protein inference methods. The first dataset from a mixture (Sample A) of 13 standard proteins was prepared at 1 μM final digestion concentration for each protein except human hemoglobin which is at 2 μM, combined with buffer, reduced, alkylated, and digested overnight with trypsin. Peptides were separated by nano-flow reversed-phase liquid chromatography gradient and analyzed by mass spectrometry and tandem mass spectrometry in a Thermo Electron (San Jose, CA) LTQ linear ion trap mass spectrometer.

The second mixture (Sample Sigma49) was cleaned up by gel electrophoresis, reduced, alkylated, and digested in-gel with trypsin. Tandem mass spectra for doubly-charged precursor ions were obtained from the website at Vanderbilt University website [13] and searched against human sequences in Swiss-Prot using Sequest [7].

3 Results

We implemented two Bayesian approaches described in the Methods section and tested them on the Sigma49 sample. The peptide detectability predictors were trained using Sample A following the method described previously [14]. Similarly as in [13], prior to the protein inference, 13388 MS/MS spectra acquired from Sigma49 sample in one LC/MS experiment were searched against the human proteome in Swiss-Prot database (version 54.2). PeptideProphet [18] was then used to assign a probability score for each identified peptide. For the basic Bayesian model, 152 unique peptides with minimum PeptideProphet probability score 0.95 were retained as identified peptides, while for the advanced model, we retained 443 peptides with minimum probability score 0.05. We tried two methods to set the probability r_j for each peptide identification. In the first method, we directly use the probability for each identified peptide reported by PeptideProphet. Since PeptideProphet does not consider peptide detectability, we implemented the second method which converts the PeptideProphet probability into a likelihood ratio LR_j and then apply our models.

The conversion is done by $LR_j = Pr_{PP}(y_j = 1)/[c \times (1 - Pr_{PP}(y_j = 1))]$, where $Pr_{PP}(y_j = 1)$ is the PeptideProphet probability, and c is the ratio between the prior probabilities of the peptide's presence and absence. For both models, we used block size 3 in the Gibbs sampler.

Table 2 compares the results for the Sigma49 sample from the basic and advanced Bayesian models with that from ProteinProphet [11] and the minimum missed peptide (MMP) approach we proposed previously [12] on the Sigma49 sample. Sigma49 sample was prepared by mixing 49 human proteins, among which 44 proteins contain at least one peptide that can be identified by shotgun proteomics. In addition, 9 keratin proteins and 4 other proteins are categorized as the "keratin contamination" and "bonus" proteins, respectively, and are believed to be present in the sample due to contamination.

From the results, we observed that using detectabilities to adjust PeptideProphet probability improves the performance of the probabilitic models. For example, the advanced model (ABLA) achieved 0.83 and 1.0 for the precision and recall, respectively, whereas directly using ProteinProphet probability (ABPA) achieved 0.66 and 0.98. This indicates that peptide detectability is a useful concept in protein inference. The adjustment of detectability also improves accuracy (see F measures) of the protein inference (e.g. BBA vs. BB or ABLA vs. ABL), implying that the predicted peptide detectabilities need to be adjusted by peptide quantities in real proteomics experiments. We also tried to incorporate a simple method for estimating protein prior probabilisties. In ABLA, we set the

Table 2. Protein inference results on the Sigma49 dataset using minimum missed peptide approach (MMP), ProteinProphet (PP), basic Bayesian model (BB), basic Bayesian model with detectability adjustment (BBA), advanced Bayesian model using raw PeptideProphet probabilities (ABP), ABP after detectability adjustment (ABPA), advanced Bayesian model using converted probability scores (ABL), ABL after detectability adjustment (ABLA), and ABLA with estimated protein prior probabilities (ABLAP). All results are evaluated based on the true positive (TP), false positive (FP) and false negative (FN) numbers of proteins, and the precision (PR), recall (RC) and F-measure (F) in two categories of true proteins in the sample: model proteins, and model proteins plus all contaminations. MAP solutions were used as positive proteins for our probabilistic models; and 0.5 cutoff was used for ProteinProphet.

	MMP	PP	BB	BBA	ABP	ABPA	ABL	ABLA	ABLAP
TP	39/45	41.5/47.5	39/47	37/43	35/39	43/49	37/41	44/50	43/49
FP	6/0	7.5/1.5	16/8	6/0	4/0	22/16	4/0	9/3	6/0
FN	5/12	2.5/9.5	5/10	7/14	9/18	1/8	7/16	0/7	1/8
PR	0.87/1	0.85/0.97	0.71/0.85	0.86/1	0.9/1	0.66/0.75	0.9/1	0.83/0.94	0.88/1
RC	0.89/0.79	0.94/0.83	0.89/0.82	0.84/0.75	0.8/0.68	0.98/0.86	0.84/0.72	1/0.88	0.98/0.86
F	0.88/0.88	0.89/0.90	0.79/0.84	0.85/0.86	0.84/0.81	0.79/0.81	0.87/0.84	0.91/0.91	0.92/0.92

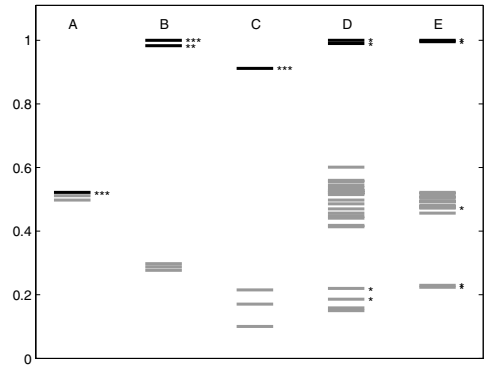

Fig. 2. Protein inference using ABLA method from 5 (A-E) connected components in the protein configuration graph built from Sigma49 dataset. Proteins in the same components are ordered in the same column based on their marginal posterior probabilities. Proteins in the MAP configuration are highlighted in black. The true proteins in Sigma49 sample are labelled by "***" for model proteins, "**" for non-karetin contaminations and "*" for karetin contaminations.

protein prior probabilities as 0.5 for all proteins, whereas in ABLAP, we set it as 0.2, i.e. the ratio of the expected number of proteins in the sample (i.e. 48) and the total number of candidate proteins. Comparing their results, we observed that incorporating protein prior probabilities further improves the performance of protein inference, in which the F measure improves from 0.91 to 0.92. Overall, ABLAP and ABLA model outperforms the other methods. However, it is hard to draw a firm conclusion from the experiments on a relatively simple protein

mixture. Further comparative analysis of these models using more complex (e.g. with hundreds of proteins), but well elucidated samples like Sigma49, is needed.

Fig. 2 illustrates the results of ABLA on 5 connected components in the protein configuration graph built from the Sigma49 dataset. The model proteins and likely contaminant proteins in the sample received higher marginal posterior probabilities than the other proteins, and the MAP configuration contains mostly true proteins. PeptideProphet cannot resolve the correct protein assignment in component A and C. We note that component A consists of three proteins (P51965, Q96LR5 and Q969T4) which share only one identified peptide (often referred to as "single wonders"). ABLA algorithm correctly assigns the true model protein (P51965) as the MAP configuration over the other two proteins.

4 Discussion

In this study we proposed and evaluated a new methodology for protein inference in shotgun proteomics. The two Bayesian models proposed herein attempt to find the set of proteins that is most likely to be present in the sample. The new approach has three advantages over the existing methods: (1) it calculates or, if global optimum is not reached, approximates a MAP solution for the set of proteins present in the sample and can also output the probability of each protein to be present in the sample; (2) it can output the marginal probabilities of the identified peptides to be present in the sample, given the entire experiment; (3) the Gibbs sampling approach used to approximate the posterior probabilities of protein configuration is a proven methodology, and its performance and convergence has been well-studied.

It is common in proteomics for a sample to be analyzed multiple times in order to increase coverage of the proteome as well as to increase confidence in low sequence coverage proteins [23]. While not specifically addressed, the application of the Bayesian models described here adequately accommodates such data since peptide detectability, used to calculate prior probabilities, should assign lower values to those peptides not identified in all the replicate analyses. In addition, higher mammals often contain multiple very similar homologous proteins due to recent gene duplications. These proteins are almost impossible to differentiate using shotgun proteomics, if some but not all of these proteins are present in the sample. As a result, although the MAP protein configuration will contain at least one of these proteins, they each can receive a low marginal probability (e.g. < 0.5). While we have not explicitly addressed this problem here, we note that the proposed models can be easily modified to consider a given set of proteins as a group and then compute the probability of their presence as a whole. We will test this functionality in future implementation of the models.

While we show that the new methodology is accurate and useful, we note that the current detectability predictor, which was trained on a small number of doubly-charged fully tryptic peptide ions, poses a limitation of this approach. Therefore, it does not fully accommodate the results of a shotgun proteomics

experiments, which are known to produce singly- and triply-charged ions as well as peptide ions with missed cleavages that are readily identified. Furthermore, we are currently using an inaccurate method to adjust detectability with different peptide quantities. Future improvements in detectability prediction, peptide confidence estimation, as well as the detectability adjustment may lead to further improvement of the Bayesian models described here.

Acknowledgements

We acknowledge the support of the NCI grant U24 CA126480-01 to F. Regnier, RJA, PR, HT et al. HT and RJA acknowledge the support from NIH/NCRR grant 5P41RR018942. HT is partially supported by NSF award DBI-0642897. PR is partially supported by NSF award DBI-0644017.

References

1. Aebersold, R., Mann, M.: Mass spectrometry-based proteomics. Nature 422, 198–207 (2003)
2. McDonald, W.H., Yates, J.R.: Shotgun proteomics: integrating technologies to answer biological questions. Curr. Opin. Mol. Ther. 5(3), 302–309 (2003)
3. Kislinger, T., Emili, A.: Multidimensional protein identification technology: current status and future prospects. Expert Rev. Proteomics 2(1), 27–39 (2005)
4. Swanson, S.K., Washburn, M.P.: The continuing evolution of shotgun proteomics. Drug Discov. Today 10(10), 719–725 (2005)
5. Marcotte, E.M.: How do shotgun proteomics algorithms identify proteins?. Nat. Biotechnol. 25(7), 755–757 (2007)
6. Nesvizhskii, A.I.: Protein identification by tandem mass spectrometry and sequence database searching. Methods Mol Biol 367, 87–119 (2007)
7. Yates, J.R., Eng, J.K., McCormack, A.L., Schieltz, D.: Method to correlate tandem mass spectra of modified peptides to amino acid sequences in the protein database. Anal Chem 67, 1426–1436 (1995)
8. Perkins, D.N., Pappin, D.J., Creasy, D.M., Cottrell, J.S.: Probability-based protein identification by searching sequence databases using mass spectrometry data. Electrophoresis 20(18), 3551–3567 (1999)
9. Craig, R., Beavis, R.C.: TANDEM: matching proteins with tandem mass spectra. Bioinformatics 20(9), 1466–1467 (2004)
10. Nesvizhskii, A.I., Aebersold, R.: Interpretation of shotgun proteomic data: the protein inference problem. Mol Cell Proteomics 4(10), 1419–1440 (2005)
11. Nesvizhskii, A.I., Keller, A., Kolker, E., Aebersold, R.: A statistical model for identifying proteins by tandem mass spectrometry. Anal Chem 75(17), 4646–4658 (2003)
12. Alves, P., Arnold, R.J., Novotny, M.V., Radivojac, P., Reilly, J.P., Tang, H.: Advancement in protein inference from shotgun proteomics using peptide detectability. In: PSB 2007: Pacific Symposium on Biocomputing, pp. 409–420. World Scientific, Singapore (2007)
13. Zhang, B., Chambers, M.C., Tabb, D.L.: Proteomic Parsimony through Bipartite Graph Analysis Improves Accuracy and Transparency. J Proteome Res. 6(9), 3549–3557 (2007)

14. Tang, H., Arnold, R.J., Alves, P., Xun, Z., Clemmer, D.E., Novotny, M.V., Reilly, J.P., Radivojac, P.: A computational approach toward label-free protein quantification using predicted peptide detectability. Bioinformatics 22(14), 481–488 (2006)
15. Lu, P., Vogel, C., Wang, R., Yao, X., Marcotte, E.M.: Absolute protein expression profiling estimates the relative contributions of transcriptional and translational regulation. Nat. Biotechnol. 25(1), 117–124 (2007)
16. Elias, J.E., Haas, W., Faherty, B.K., Gygi, S.P.: Comparative evaluation of mass spectrometry platforms used in large-scale proteomics investigations. Nat. Methods 2(9), 667–675 (2005)(Comparative Study)
17. Elias, J.E., Gygi, S.P.: Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. Nat. Methods 4(3), 207–214 (2007) (Evaluation Studies)
18. Keller, A., Nesvizhskii, A.I., Kolker, E., Aebersold, R.: Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. Anal Chem 74(20), 5383–5392 (2002)
19. Wu, F.-X., Gagne, P., Droit, A., Poirier, G.G.: RT-PSM, a real-time program for peptide-spectrum matching with statistical significance. Rapid Commun Mass Spectrom 20(8), 1199–1208 (2006)
20. Bern, M., Goldberg, D.: Improved ranking functions for protein and modification-site identifications. In: Speed, T., Huang, H. (eds.) RECOMB 2007. LNCS (LNBI), vol. 4453, pp. 444–458. Springer, Heidelberg (2007)
21. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans. on Pattern Analysis and Machine Intelligence 6, 721–741 (1984)
22. Liu, J.S.: Monte Carlo strategies in scientific computing. Springer, Heidelberg (2002)
23. Brunner, E., Ahrens, C.H., Mohanty, S., Baetschmann, H., Loevenich, S., Potthast, F., Deutsch, E.W., Panse, C., de Lichtenberg, U., Rinner, O., Lee, H., Pedrioli, P.G.A., Malmstrom, J., Koehler, K., Schrimpf, S., Krijgsveld, J., Kregenow, F., Heck, A.J.R., Hafen, E., Schlapbach, R., Aebersold, R.: A high-quality catalog of the Drosophila melanogaster proteome. Nat Biotechnol. 25(5), 576–583 (2007)

Appendix

Note: In the two algorithms presented below, we have

$$F(x_{v_1},...,x_{v_c},y_{w_1},...,y_{w_d}) = \prod_{i\in v} P(x_i) \prod_{j\in N^+(v)\cup w} P(y_j|x_{N^-(j)}) \prod_{j\in w} \frac{P(y_j|s_{N^+(j)})}{P(y_j)} \tag{12}$$

where $N^+(.)$ and $N^-(.)$ refer to the nodes that the current node(s) are linked to and linked from, respectively; $v = (v_1,...,v_c)$ and $w = (w_1,...,w_d)$ are the block indices for x and y, repectively; w is empty for the basic model; $P(y_j|x_{N^-(j)})$ can be computed by 6; and $Pr(y_j = 1|s_{N^+(j)}) = r_j$.

$$\begin{aligned} P(y_j) &= \textstyle\sum_{(x_1,...,x_m)} [P(y_j|x_1,...,x_m)P(x_1,...,x_m)] \\ &= [1 - \textstyle\prod_i (1 - Pr(x_i = 1)d_{ij})]^{y_j} [\textstyle\prod_i (1 - Pr(x_i = 1)d_{ij})]^{(1-y_j)} \end{aligned} \tag{13}$$

Algorithm 1. Gibbs sampler for protein inference using the basic model

Input : Peptide configuration $(y_1,...,y_n)$ and peptide detectabilities $\{d_{ij}\}$
Output: MAP protein configuration $(x_1,...,x_m)$

Initialize $(x_1,...,x_m)$ randomly ;
$MaxPr \leftarrow 0$;
Normalizing factor $T \leftarrow 1$;
while *not converge* **do**
  $v = (v_1,...,v_t) \leftarrow$ a random t-block from $(1,...,m)$;
  $T \leftarrow \frac{Value(x_1,...,x_m)}{F(x_{v_1},...,x_{v_t})}$;
  for *all* $(v_1,...,v_t)$ **do**
    Compute $F(x_{v_1},...,x_{v_t})$;
    $Value(x_1,...,x_m) \leftarrow F(x_{v_1},...,x_{v_t}) \times T$;
    if $Value(x_1,...,x_m) > MaxPr$ **then**
      $MaxPr \leftarrow Value(x_1,...,x_m)$;
      $(x_1^{Max},...,x_m^{Max}) \leftarrow (x_1,...,x_m)$;
      $(x_{v_1}^{Max},...,x_{v_t}^{Max}) \leftarrow (x_{v_1},...,x_{v_t})$;
    end
  end
  Sample $(x'_{v_1},...,x'_{v_t})$ from normalized $F(x_{v_1},...,x_{v_t})$;
  $(x_{v_1},...,x_{v_t}) \leftarrow (x'_{v_1},...,x'_{v_t})$;
end
Report $MaxPr$, $(x_1^{Max},...,x_m^{Max})$, and compute marginal probabilities ;

Algorithm 2. Gibbs sampler for protein inferencing using the advanced model

Input : Peptide prior probabilities $(r_1, ..., r_n)$ and peptide detectabilities $\{d_{ij}\}$
Output: MAP protein configuration $(x_1, ..., x_m)$

Initialize $(x_1, ..., x_m)$ and $(y_1, ..., y_n)$ randomly ;
$MaxPr \leftarrow 0$;
Normalizing factor $T \leftarrow 1$;
while *not converge* **do**
  $c \leftarrow$ a random number between 0 and t ;
  $(v_1, ..., v_c) \leftarrow$ a random c-block from $(1, ..., m)$;
  $d \leftarrow t - c$;
  $(w_1, ..., w_d) \leftarrow$ a random d-block from $(1, ..., n)$;
  Compute normalizing factor $T \leftarrow \frac{Value(x_1,...,x_m;y_1,...,y_n)}{F(x_{v_1},...,x_{v_c},y_{w_1},...,y_{w_d})}$;
  for *all* $(x_{v_1}, ..., x_{v_c})$ *and* $(y_{w_1}, ..., y_{w_d})$ **do**
    Compute $F(x_{v_1}, ..., x_{v_c}; y_{w_1}, ..., y_{w_d})$;
    memorizing: $Value(x_1, ..., x_m, y_1, ..., y_n) \leftarrow F \times T$;
    if $Value(x_1, ..., x_m, y_1, ..., y_n) > MaxPr$ **then**
      $MaxPr \leftarrow Value(x_1, ..., x_m, y_1, ..., y_n)$;
      $(x_1^{Max}, ..., x_m^{Max}) \leftarrow (x_1, ..., x_m)$;
      $(x_{v_1}^{Max}, ..., x_{v_c}^{Max}) \leftarrow (x_{v_1}, ..., x_{v_c})$;
      $(y_1^{Max}, ..., y_n^{Max}) \leftarrow (y_1, ..., y_n)$;
      $(y_{w_1}^{Max}, ..., y_{w_d}^{Max}) \leftarrow (y_{w_1}, ..., y_{w_d})$;
    end
  end
  Sample $(x'_{v_1}, ..., x'_{v_c}; y'_{w_1}, ..., y'_{w_d})$ from normalized $F(x_{v_1}, ..., x_{v_c}; y_{w_1}, ..., y_{w_d})$;
  $(x_{v_1}, ..., x_{v_c}) \leftarrow (x'_{v_1}, ..., x'_{v_c})$;
  $(y_{w_1}, ..., y_{w_d}) \leftarrow (y'_{w_1}, ..., y'_{w_d})$;
end
Report $MaxPr$, $(x_1^{Max}, ..., x_m^{Max})$, and compute marginal probabilities ;