Protein inference: a review

Ting Huang, Jingjing Wang, Weichuan Yu and Zengyou He

Submitted: 24th March 2011; Received (in revised form): 29th January 2012

Abstract
Assembling peptides identified from tandem mass spectra into a list of proteins, referred to as protein inference, is a critical step in proteomics research. Due to the existence of degenerate peptides and 'one-hit wonders', it is very difficult to determine which proteins are present in the sample. In this paper, we review existing protein inference methods and classify them according to the source of peptide identifications and the principle of algorithms. It is hoped that the readers will gain a good understanding of the current development in this field after reading this review and come up with new protein inference algorithms.

Keywords: Shotgun proteomics; peptide identification; protein inference; supplementary information model; false discovery rate

INTRODUCTION
Mass spectrometry-based proteomics is the large-scale study of proteins expressed in an organism. It can provide information that is not readily available from genomic sequence or RNA expression data [1]. An explicit goal of proteomics is the identification of all proteins expressed in a cell or tissue.

Shotgun proteomics is a method of identifying complex protein mixtures using a combination of high-performance liquid chromatography and mass spectrometry. As shown in Figure 1, proteins are enzymatically digested and optionally fractionated from their biological source. The resulting peptide mixtures are then ionized and scanned by tandem mass spectrometry (MS/MS) to obtain a set of MS/MS spectra. Finally, peptides and proteins are identified by computational analysis of the acquired MS/MS spectra [2, 3]. The data generated by shotgun proteomics experiments are highly redundant in the sense that a subset of the peptides is repeatedly and preferentially selected for fragmentation and tandem MS scanning. And those peptides derived from low abundance proteins are more difficult to detect. As a consequence, a large number of fragment ion spectra have to be acquired to increase the identification coverage rate [4–6].

There are three key computational problems in the protein identification process: peptide identification, protein inference and result evaluation. In peptide identification, fragment ion spectra are assigned to peptide sequences to generate a set of Peptide-Spectrum Matches (PSM) using database search.
engines, such as Mascot [7], SEQUEST [8] and X!Tandem [9]. In protein inference, identified peptide sequences are assembled into a set of confident proteins [10, 11]. Finally, we need to assess the reliability of these identifications.

Over the past decade, researchers paid much attention to peptide identification. But it is not straightforward to generate a reliable list of proteins from those identified peptides. This occurs mainly due to the existence of degenerate peptides and ‘one-hit wonders’. Degenerate peptides denote the same list of identified peptides shared by multiple proteins in the database. The corresponding ambiguity cannot be easily resolved. ‘One-hit wonders’ denote some proteins that only have one single identified peptide. Theoretically, it is difficult to infer proteins based on one single peptide identification due to the possibility of false-positive identification. As a result, the problem of determining which protein is indeed present in the sample often has multiple solutions and can be computationally intractable. ProteinProphet [12], a probabilistic model, was first proposed to address this challenge. After that, we have witnessed the increase of new problem formulations and new solutions (e.g. [13, 14]). But an up-to-date and comprehensive review is still lacking.

In this article, we aim at providing a comprehensive review of existing protein inference approaches. To achieve this objective, we present two frameworks that can categorize all existing solutions to the protein inference problem. More precisely, we use the dependence on the peptide search engine and the underlying algorithmic technique as the classification criteria.

The remainder of the article is organized as follows. In ‘The Protein Inference Problem’ section, we discuss the key difficulties in solving the protein inference problem. In ‘Peptide Identification Post-processing’ section, we list common methods for peptide identification post-processing. In ‘Peptide Assembly’ section, we summarize available protein inference methods and describe different classes of methods in detail. We discuss how to assess the reliability of protein identifications in ‘Result Validation for Protein Identification’ section. We conduct a comprehensive evaluation and comparison of five typical protein inference methods in ‘Experiment’ section. According to the experiment results, we discuss the future development of protein inference problem in ‘Future Perspective’ section. Finally, we conclude this article in ‘Conclusion’ section.

THE PROTEIN INFEERENCE PROBLEM

In shotgun proteomics, the peptide-centric strategy is widely adopted as a means of identifying proteins from biological samples. Since researchers are more interested in which proteins are present in the sample, peptide identification is only an intermediate step for protein identification. After gathering all
peptide identifications, we need to infer the existence of proteins in the sample. However, protein inference is a difficult problem due to the existence of ‘one-hit wonders’ and degenerate peptides.

One-hit wonders refer to those proteins who have only one identified peptide. Formally, we use \( u_i \) to denote the number of identified peptides that could be digested from protein \( i \). Protein \( i \) is a one-hit wonder if \( u_i = 1 \). Since current peptide identification algorithms are still far from perfect, an identified peptide may be a false positive regardless of its uniqueness. That is, the only evidence for confirming the presence of a one-hit wonder is not reliable, even if such peptide is unique throughout the protein sequence database. Thus, compared to those proteins having two or more identified peptides, it is not easy to determine the existence of these one-hit wonders.

Degenerate peptides are peptides that can be shared by multiple proteins. Mathematically, let \( v_j \) denote the number of proteins that can generate an identified peptide \( j \). We call peptide \( j \) a degenerate peptide if \( v_j > 1 \). Compared to unique peptides that only occur in one protein, degenerate peptides pose computational challenges since it is hard to know which protein or protein group indeed generates the degenerate peptides. This is often caused by the existence of homologous proteins in the database. In this context, it is difficult to distinguish between two possibilities: (i) All of the related proteins are present in the sample; (ii) Only some proteins are truly present.

We can divide the analysis efforts of protein inference into different stages to reduce its complexity. As shown in Figure 2, protein inference consists of three phases: peptide identification post-processing, peptide assembly and result validation. In the forthcoming sections, we will review existing solutions to these three sub-problems listed in Figure 2. Note that the step of peptide assembly is of central importance to the success of the whole procedure. We provide two general categorization frameworks to organize existing assembly approaches. The first one is application-oriented, which categorizes assembly algorithms according to their dependence on the output of some specific peptide search engines. The second one mainly uses the underlying algorithmic technique as the classification criterion. To understand the rationale behind our second framework, we need to first investigate the computational nature of such assembly process.

One can model the relationship between the identified peptides and the proteins in the database as a bipartite graph, as shown in Figure 3. This is the standard input setting for most assembly algorithms. The general problem of inferring correct proteins from this bipartite graph is difficult to solve due to the existence of ‘one-hit wonders’ and degenerate peptides. For example, the first protein and the second protein in Figure 3 have the same set of identified peptides: Peptide 1 and Peptide 2, which are degenerate peptides. If there is no other supporting information, we cannot determine which protein is indeed present in the sample. Furthermore, Protein 3 is a one-hit wonder that has Peptide 3 as its single peptide identification. Compared with Protein 4 having two identified peptides, it is not easy to determine the existence of Protein 3. In addition, the quality of peptide identification also influences the final protein identification results. But if we make some reasonable assumptions, it is possible to reveal the computational nature of such assembly problem.

First, we can assume that all \( m \) peptides are true positives. Under this assumption, we can derive the upper bound and lower bound on the number of possible proteins in the sample. Since there are at most \( n \) proteins that are associated with all identified peptides, it is easy to see that the upper bound of the number of proteins is \( n \). In other words, it is impossible for us to identify extra proteins under the
current problem setting. To date, some existing protein inference methods adopt the strategy of returning all potential proteins without filtering [15]. Obviously, the underlying assumption behind such a strategy is that the sample contains a large portion of homologous proteins.

It is not straightforward to calculate the lower bound of the number of proteins. Recalling that all identified peptides are assumed to be correct, the set of proteins reported by any assembly algorithm should contain all identified peptides. In other words, it is not allowed to miss any identified peptides in the final protein list. Based on above observations, we can formulate the assembly task as a set covering problem [16, 17]: finding a minimum subset of proteins that cover all identified peptides. Here one peptide is said to be covered by some protein if there is an edge between them in the bipartite graph shown in Figure 3. Therefore, the lower bound is the number of proteins in the optimal solution of the corresponding set covering problem. It is well known that the set covering problem is NP-complete, making it infeasible to obtain the optimal solution in practice. Therefore, algorithms in this category [18, 19] adopt a parsimony philosophy and assume that one peptide is seldom shared by multiple proteins.

Overall, the analysis on upper bound and lower bound provides some hints on the nature of assembly problem: any algorithm under this problem setting is to seek a trade-off between two extreme cases. On the one hand, reporting all $n$ proteins has the risk of including many false positives. The reason is that we search the protein database to obtain all the possible proteins associated with all identified peptides. However, some results may be statistical artifacts that match the identified peptides only by chance. To better clarify this point, here we will use an example to give a comprehensive explanation. Suppose we conduct an experiment using a synthetic mixture of known proteins, e.g. 18 proteins. There can be a protein $T$ in these 18 proteins and another protein $F$ in the database such that they share the same peptide sequence, even though $F$ is not present in the sample. Such cases will occur more frequently in real data with the increase of protein number in the target sample. When the candidate protein is only associated with a peptide that is unique throughout the database, the probability of such a protein being present is very high if the identification confidence of corresponding unique peptide is high.

In fact, in many protein inference tools such as ProteinProphet, the protein identification probability is reduced to the peptide identification probability if the protein has only one unique peptide identified [see Equation (1) in the article]. That means the status of the protein (present or absent) is determined by whether its unique peptide is a statistical artifact. Even if such unique peptide is confirmed by more than one MS/MS spectrum, it just indicates that its probability of being absent in the sample is extremely low. High identification confidence cannot be interpreted absolutely as ‘there are no statistical errors’. Hence, it is very likely to include many false positives without any filtering. On the other hand, retaining only a minimum subset of proteins may lose many true positives. Therefore, we have to face this dilemma in the design of effective assembly algorithms.

Figure 4 shows that most current peptide assembly algorithms find a subset of proteins whose size is between min (the lower bound) and max (the upper bound).

The above analysis is based on the two assumptions. One assumption is that all identified peptides are correct. If we relax this strong assumption and suppose that there are still certain false-positive peptide identifications, the number of proteins may be less than the current lower bound. In Figure 4,
we use $P$ to denote the new lower bound when not all $m$ peptide identifications are correct. If we know the exact fraction of incorrect peptide identifications, this new lower bound will be the number of proteins in the optimal solution of a partial set covering model [20]. In addition, it is impossible to extend the upper bound if we use only information in the bipartite graph. To improve the identification coverage, we have to borrow additional information from other data sources. That is, we change the problem setting by introducing extra information as input to the assembly procedure. To date, researchers have used raw MS/MS spectra, single-stage MS data, peptide expression profiles, mRNA expression data, protein interaction network and gene model as additional input to improve the coverage. As a result, it is possible to obtain a new upper bound $E$ on the number of possible proteins, as shown in the right part of Figure 4. More importantly, we can categorize available peptide assembly methods according to different kinds of extra information they have exploited.

Another assumption is that all peptides have the same likelihood of being detected. However, a protein’s peptide fragments are not observed with equal probability. Indeed, for any given protein, only a few ‘proteotypic’ peptides are repeatedly and consistently identified using a particular proteomic platform [6, 21, 22]. Proteotypic peptides are widely used in quantitative proteomics. Peptide detectability, a new concept that is based on proteotypic peptides, was introduced into protein inference. Peptide detectability is defined as the probability of observing a peptide in a standard sample by a standard proteomics routine [23, 24]. In other words, peptide detectability can be considered as an intrinsic property of a peptide that is mainly decided by its primary sequence and its parent protein sequence. So a degenerate peptide now can be assigned to each of its parent proteins with a corresponding probability assuming that it comes from that protein. This concept can well explain the assignment of degenerate peptides in principle, compared with the use of weight [12], which presumes that degenerate peptides only can come from one protein or the use of peptides grouping [25] that puts peptide sequences with indistinguishable predicted spectra into the same group.

In summary, the protein inference problem can be divided into three steps: peptide identification post-processing, peptide assembly and result validation. Since peptide assembly is of central importance to the success of the whole procedure, we further investigate its computational nature. Through the analysis, researchers may know the limitations of existing methods and develop new solutions.

**PEPTIDE IDENTIFICATION POST-PROCESSING**

Before peptide assembly, it is necessary to validate the peptide identification results. One simple approach is to use decoy (reversed, randomized or shuffled) sequence database [26]. An alternative approach is the probabilistic model (such as PeptideProphet [27]).

Zhang et al. [28] use a multivariate non-linear discriminate function to control the quantity of database search results. A Bayesian non-parametric (BNP) model [29] incorporates a probability framework into the randomized database searching method. Both methods are based on the target-decoy database search strategy. The BNP model can incorporate more features into a linear discriminate function if needed. Another robust and non-parametric method [30] uses a novel indicator, the probability ratio, to consider the statistical information provided by the first and second best scores. The probability ratio is a conditional probability indicator that uses the score given by the second match (or subsequent matches) to determine spectral quality. Error rates associated with peptide identification are calculated by a fully automated, non-parametric algorithm.

PeptideProphet [27] automatically validates peptide assignments to MS/MS spectra made by database search programs such as SEQUEST. PeptideProphet first combines multiple scores from a standard search engine (SEQUEST, Mascot, etc.) into a single score, which is called ‘discriminant score’. Such a discriminant score is usually generated from the linear combination of raw scores, in which the weight coefficients are empirically tuned parameters for different search engines. After Peptide Prophet calculates the discriminant scores for all the spectra in a sample, it learns distributions of the discriminant scores among correct and incorrect peptides from each dataset, and uses those distributions to compute for each result a probability that it is correct.

Recently, Choi et al. [31] presented two flexible methods, the variable component mixture model
and the semiparametric mixture model, to remove the restrictive parametric assumptions in the mixture modeling approach of PeptideProphet. The most recent version of PeptideProphet provides an option to use non-parametric modeling with target-decoy database searches.

At the same time, Choi et al. [32] propose a semi-supervised framework that combines probabilistic modeling approach of PeptideProphet with the decoy strategy. It presents a semi-supervised expectation–maximization (EM) algorithm for constructing a Bayes classifier for peptide identification using the probability mixture model, extending PeptideProphet to incorporate decoy peptide matches.

**PEPTIDE ASSEMBLY**

To date, there are many methods for peptide assembly. These methods can be classified into different categories according to different criteria. It is very important to have a systematic framework that is expandable to cover all available methods. With these considerations in mind, we propose two hierarchical frameworks in Figures 5 and 6, respectively. In Figure 5, peptide assembly methods are divided into two classes according to their independence on the peptide search engines. In Figure 6, peptide assembly methods are categorized according to the underlying algorithmic techniques. In the following subsections, we will discuss each category in detail. Furthermore, we provide a comprehensive list of all available methods from two different angles in Table 1.

Note that our main objective in this article is to summarize available peptide assembly methods from an algorithmic aspect. Though some systems that
automate data management and processing in MS-driven proteomics analysis do provide the functionality of protein inference, they generally re-use existing methods rather than develop new algorithm. For example, ms_lims [33] reports all the matching proteins for each identified peptide and selects a representative protein as the primary protein hit according to the algorithm published in [34]. Therefore, we will omit these papers in our review.

**Search engine-based classification**

There are many well-known programs for peptide identification, including database search programs (such as SEQUEST, Mascot, X!Tandem and OMSSA [35]), de novo sequencing programs (such as PEAKS [36] and PepNovo [37]) and hybrid tools (such as GutenTag [38] and InsPecT [39]). Among these methods, SEQUEST and Mascot are the most widely used peptide identification approaches.

<table>
<thead>
<tr>
<th>Assembly method</th>
<th>Algorithm</th>
<th>Search engines</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBParser [18]</td>
<td>Parsimonious Model</td>
<td>Mascot</td>
<td>NA</td>
</tr>
<tr>
<td>IDPicker [9, 45]</td>
<td>Parsimonious Model</td>
<td>Search engine-independent</td>
<td><a href="http://fenchurch.mc.vanderbilt.edu/">http://fenchurch.mc.vanderbilt.edu/</a></td>
</tr>
<tr>
<td>LDFA [24]</td>
<td>Parsimonious Model</td>
<td>Search engine-independent</td>
<td><a href="http://darwin.informatics.indiana.edu/applications/ProteinInference/">http://darwin.informatics.indiana.edu/applications/ProteinInference/</a></td>
</tr>
<tr>
<td>EBP [46]</td>
<td>Non-parametric Model</td>
<td>Search engine-independent</td>
<td><a href="http://bioinf.itmat.upenn.edu/ebp/">http://bioinf.itmat.upenn.edu/ebp/</a></td>
</tr>
<tr>
<td>PANORAMICS [25]</td>
<td>Non-parametric Model</td>
<td>Mascot</td>
<td>NA</td>
</tr>
<tr>
<td>PROVA LT [43]</td>
<td>Non-parametric Model</td>
<td>Mascot</td>
<td><a href="http://kiwi.rcr.uga.edu/tcprot">http://kiwi.rcr.uga.edu/tcprot</a></td>
</tr>
<tr>
<td>MSBayesPro [49]</td>
<td>Non-parametric Model</td>
<td>Search engine-independent</td>
<td><a href="http://darwin.informatics.indiana.edu/yonli/ProteinInfer/">http://darwin.informatics.indiana.edu/yonli/ProteinInfer/</a></td>
</tr>
<tr>
<td>Qscore [42]</td>
<td>Parametric Model</td>
<td>Search engine-independent</td>
<td>NA</td>
</tr>
<tr>
<td>PROT_PROBE [51]</td>
<td>Parametric Model</td>
<td>Search engine-independent</td>
<td>NA</td>
</tr>
<tr>
<td>A nested model [56]</td>
<td>Raw MS/MS Data</td>
<td>Search engine-independent</td>
<td>NA</td>
</tr>
<tr>
<td>HSM [13]</td>
<td>Raw MS/MS Data</td>
<td>Search engine-independent</td>
<td>NA</td>
</tr>
<tr>
<td>Barista [57]</td>
<td>Raw MS/MS Data</td>
<td>Search engine-independent</td>
<td><a href="http://noble.gs.washington.edu/proj/crux">http://noble.gs.washington.edu/proj/crux</a></td>
</tr>
<tr>
<td>Scaffold [61]</td>
<td>Raw MS/MS Data</td>
<td>Search engine-independent</td>
<td><a href="http://www.proteomesoftware.com/">http://www.proteomesoftware.com/</a></td>
</tr>
<tr>
<td>MS and MS/MS combined method [60]</td>
<td>Single-stage MS data</td>
<td>Search engine-independent</td>
<td>NA</td>
</tr>
<tr>
<td>PIPER [52]</td>
<td>Peptide Expression Profile</td>
<td>Search engine-independent</td>
<td>NA</td>
</tr>
<tr>
<td>MSNet [54]</td>
<td>Protein Interaction Network</td>
<td>Search engine-independent</td>
<td><a href="http://aug.csres.utexas.edu/msnet">http://aug.csres.utexas.edu/msnet</a></td>
</tr>
<tr>
<td>CEA [33]</td>
<td>Protein Interaction Network</td>
<td>Search engine-independent</td>
<td><a href="http://bioinfo.vanderbilt.edu/cea">http://bioinfo.vanderbilt.edu/cea</a></td>
</tr>
<tr>
<td>MSpresso [55]</td>
<td>mRNA Expression Data</td>
<td>Search engine-independent</td>
<td><a href="http://www.marcottelab.org/MSpresso/">http://www.marcottelab.org/MSpresso/</a></td>
</tr>
</tbody>
</table>

The first function computes a preliminary score (Sp) by simply counting the number of common peaks between experimental and theoretical spectrum, which is used to rapidly determine a small subset of peptide candidates for each spectrum. The second function computes a normalized correlation score (Xcorr), which is a scalar dot product between the acquired and theoretical spectrum with a correction factor.

Mascot [8] incorporates a probability-based implementation of the Mowse algorithm [40]. By calculating the distribution of tryptic peptide lengths across the entire search database, a probability $P$ that the observed match between the experimental data and the database sequence is a random event can be calculated [41]. Then, the ion score for an MS/MS match is defined as: $-10 \log_{10}(P)$, which is used as the primary score in Mascot.

Due to the popularity of SEQUEST and Mascot, some peptide assembly methods can only handle identification results from SEQUEST or Mascot.
Therefore, we divide existing assembly algorithms into two groups: search engine-dependent approaches and search engine-independent approaches. As shown in Fig. 5, current search engine-dependent approaches are specific to either SEQUEST or Mascot.

**Search engine-dependent approach**

**SEQUEST-based approach** Such approach mainly uses XCorr and Sp scores as input to perform protein inference. Typical examples include DTASelect [15] and Qscore [42].

- DTASelect collates the peptide identifications for each protein, and then applies user-defined criteria (such as thresholds for XCorr scores) to select identification results. DTASelect groups together proteins with identical sets of identified peptides and uses a similarity score to describe the relationship between proteins with overlapped peptide identifications. It removes proteins for which the observed evidence is a subset of the peptides observed for another protein.

- Qscore is presented specifically for database search results from SEQUEST. But it can be extended to search results from other database search programs. SEQUEST returns a best match peptide as long as at least one peptide from the database falls within the peptide mass tolerance. Up to this point, the quality of this match is not able to be ensured. Therefore, Moore et al. [42] develop a statistically based algorithm to determine the goodness of a protein match. It resembles an approximation to a binomial distribution with protein size, peptide match quality, number of peptides matching a protein and size of spectral data set as parameters.

**Mascot-based approach** Since Mascot reports probability-based score for each identified peptide, the Mascot-based assembly algorithms mainly concentrate on how to manipulate these scores to obtain desired output. PROVALT [43], DBParser [18] and PANORAMICS [25] fall into this category.

- PROVALT chooses a sequence of Mascot score thresholds for proteins that match different numbers of peptides. For example, a protein could be accepted with three scores of at least 30, two scores of at least 35 or one score of at least 50. The thresholds are chosen to meet a user-specified false discovery rate (FDR) by using deliberate false positives, such as reverse sequences [44].

- DBParser is a web-based application. It is designed to parse Mascot search results. DBParser reports proteins in six hierarchical categories: distinct, differentiable, subsumable, superset, subset and equivalent. Parsimony analysis is employed to derive a sufficient protein list to account for the observed peptide identifications.

- PANORAMICS is a probability model for peptide assembly that handles peptide identifications provided by Mascot. The algorithm groups proteins having the same sets of matched peptides and calculates reasonable probabilities for grouped proteins with respect to the search database. The probabilities are in line with false positive estimates that can be approximated by reverse database searching.

**Search engine-independent approach**

When a peptide assembly method, such as ProteinProphet [12], IDPicker [19, 45] and MIPGEM [14], can handle the result from any search engine, we consider it search engine independent and include it in this category. Such independence is very important since we often need to perform protein inference using peptide identification results from different search engines. Note that some of the methods use probabilities rather than raw scores as input, such as ProteinProphet [12] and EBP [46]. Such requirements can be accomplished by the peptide identification post-processing step, as we have discussed in ‘Peptide Identification Post-processing’ section. Since most assembly methods are independent of peptide search engines, we will discuss them in the following subsection in detail.

**Algorithm-based classification**

Different from search engine-based classification, the algorithm-based classification focuses on the underlying algorithmic techniques used in peptide assembly tools. As we have discussed in ‘The Protein Interference Problem’ section, the standard bipartite graph model deserves certain drawbacks: Given a set of identified peptides, it is impossible to make the protein identification coverage higher than the upper bound if no extra information is included. Therefore, researchers begin to investigate the possibility of utilizing different kinds of supplementary information in protein inference. Figure 6 summarizes these methods.
Since generally there are more than one method in each category, we choose one typical method for each category to describe the mathematical characteristics.

**Bipartite graph model**

**Parsimonious model** In the bipartite graph model, one simple solution for peptide assembly is to employ the principle of parsimony. It applies Occam’s razor [47] to deal with homologous proteins and degenerate peptides by assuming that only a small subset of proteins should be sufficient to explain all identified peptides.

Here, the objective is to find a minimum subset of proteins that cover all identified peptides. Due to the NP-Hardness of such problem, a fast greedy algorithm is often used to find a subset of protein vertices that cover all of the peptide vertices (see the top panel of Figure 3). The greedy algorithm iteratively chooses a protein vertex that connects to the largest number of uncovered peptide vertices. When all peptides have been covered, the remaining proteins are eliminated. Proteins that have been selected as the set cover constitute the parsimonious protein list.

DBParser [18], IDPicker [19, 45], MassSieve [48] and LDFA [24] use the parsimony principle to solve the peptide assembly problem.

IDPicker first filters peptide identifications using a user-chosen false-positive rate, which can be estimated from the percentage of decoy matches at every potential cut point. Then, it selects candidate proteins using the greedy algorithm according to the number of peptides matched to each protein. At last IDPicker reports the protein and peptide information via HTML or other formats. IDPicker combines multiple scores on a per sample basis, groups together duplicate peptide identifications. If the user is compiling a report from multiple runs, IDPicker can apply an appropriate hierarchy to organize the data.

DBParser first classifies proteins into six hierarchical categories: distinct, differentiable, subsumable, superset, subset and equivalent. Then parsimony analysis is employed to derive a protein list sufficient to account for all identified peptides. That is, DBParser reports all of the distinct proteins first (distinct and differentiable proteins), and then nondistinct proteins (non-redundant superset and non-redundant equivalent proteins). The same method is also applied to MassSieve [48] for redundancy removal of protein identifications.

LDFA [24] incorporates peptide detectability into the set cover problem formulation. LDFA employs a simple greedy algorithm to assign the peptide with the lowest detectability to the corresponding protein first in order to solve the minimum missed peptide problem, which can be reduced to the set cover problem.

**Statistical model** Although deterministic approaches that use parsimony principle are easy to understand, they are not as informative as statistical methods. The statistical methods can assemble large quantities of weak evidence to estimate the probability that a given protein is present. The basic idea is to perform statistical analysis using peptide scores and other related information. These statistical tools can be divided into two categories: non-parametric model and parametric model.

**Non-parametric model** Non-parametric (or distribution-free) methods make no or few assumptions about the probability distributions of the variables being assessed. Therefore, they are easier to use and have greater robustness than the parametric methods.

ProteinProphet [12] is the most widely used method for solving the peptide assembly problem. It employs an iterative procedure to estimate the protein probabilities. More precisely, it computes protein probabilities with the assumption that distinct peptide matches are independent, and then recomputes peptide probabilities conditioned on the protein probabilities. The above iteration process continues until convergence.

Mathematically, the protein probability is calculated as:

\[
P_n = 1 - \prod_i (1 - w_i^n p(+|D_i, NSP^n_i))).
\]  

where \(P_n\) is the probability that protein \(n\) is present in the sample and \(w_i^n\) is the weight of peptide \(i\) actually corresponding to protein \(n\). If peptide \(i\) has \(N_i\) parent proteins, then

\[
w_i^n = \frac{P_n}{\sum_{k=1}^{N_i} P_k}.
\]  

We iteratively compute the protein probability \(P_n\) and the weights \(w_i^n\) using Equations (1) and (2).
In addition,
\[
p(+|D_i, \text{NSP}_i^p) = \frac{p(+|D_i)p(\text{NSP}_i^p)+}{p(+|D_i)p(\text{NSP}_i^p)+ + p(-|D_i)p(\text{NSP}_i^p)-}
\]  
\tag{3}
\]
denotes the probability of correct identification of peptide \(i\) given its assignment information \(D_i\) and the number of its sibling peptides \(\text{NSP}_i^p\) in protein \(n\). The number of sibling peptides (NSPs) of peptide \(i\) is defined as
\[
\text{NSP}_i^p = \sum_{[m|m \neq i]} w_{i,m}p(+|D_m),
\]  
\tag{4}
\]
where \(p(+|D_m)\) is the probability of peptide \(m\) from the same protein as peptide \(i\) given its information \(D_m\). Information \(D\) is provided by searching engines, which may include useful information such as matching scores and the number of missed cleavages.

Then, the NSP distributions among correct and incorrect peptide assignments, \(p(\text{NSP}_i^p)+\) and \(p(\text{NSP}_i^p)-\), are calculated for each bin \(B\) in a similar way:
\[
p(\text{NSP}_i^p)+ = \frac{1}{Np(+)} \sum_{[i|\text{NSP}_i^p \in B]} \sum_n w_{i,n}p(+|D_n, \text{NSP}_i^p),
\]  
\tag{5}
\]
where \(N\) is the total number of peptide assignments and \(p(+)\) is the prior probability of a correct peptide assignment.

\[
P(x_1, \ldots, x_m|y_1, \ldots, y_n) = \frac{P(x_1, \ldots, x_m)P(y_1, \ldots, y_n|x_1, \ldots, x_m)}{\sum_{(x_1', \ldots, x_m')} P(x_1', \ldots, x_m')P(y_1, \ldots, y_n|x_1', \ldots, x_m')} = \frac{P(x_1, \ldots, x_m) \prod_j [1 - Pr(y_j = 1|x_1, \ldots, x_m)]^{1-y_j} Pr(y_j = 1|x_1, \ldots, x_m)^{y_j}}{\sum_{(x_1', \ldots, x_m')} P(x_1', \ldots, x_m') \prod_j [1 - Pr(y_j = 1|x_1', \ldots, x_m')]^{1-y_j} Pr(y_j = 1|x_1', \ldots, x_m')^{y_j}},
\]  
\tag{7}
\]

The probabilities are in line with false-positive estimations that can be approximated by reverse database searching. In PANORAMICS, even peptides with low probabilities are considered in the assembly process. This is in direct contrast to popular assembly methods whereby peptides with probabilities below a threshold are excluded from a protein assembly. Thus, PANORAMICS provides a more accurate measure to assess the confidence of protein identifications. Moreover, compared to popular assembly methods, PANORAMICS has a shorter computing time.

MSBayesPro is proposed for estimating protein probabilities in [49], which places particular emphasis on the peptide degeneracy problem.

All the shared peptides and unique peptides assigned to a group of proteins form a peptide configuration \((y_1, \ldots, y_j, \ldots, y_m)\), where \(y_i = 1\) when the peptide \(i\) is identified, otherwise \(y_i = 0\). Then, the protein inference problem is reduced to find the maximum a posterior (MAP) protein configuration \((x_1, \ldots, x_i, \ldots, x_m)\), which maximizes the conditional probability \(P(x_1, \ldots, x_m|y_1, \ldots, y_n)\)
\[
(x_1, \ldots, x_m)_{MAP} = \arg\max_{(x_1, \ldots, x_m)} P(x_1, \ldots, x_m|y_1, \ldots, y_n),
\]  
\tag{6}
\]
where \(x_i = 1\) if protein \(i\) is present and \(x_i = 0\) otherwise.

The conditional probability is
\[
P(x_1, \ldots, x_m) = \prod_i P(x_i)
\]  
\tag{8}
\]
and
\[
Pr(y_i = 1|x_1, \ldots, x_m) = 1 - \prod_j [1 - x_j Pr(y_j = 1|x_1, x_j = 0, k \neq i, 1 \leq k \leq m]],
\]  
\tag{9}
\]
where \(P(y_j = 1|x_1, x_k = 0, k \neq i, 1 \leq k \leq m)\) is the probability of peptide \(j\) being identified if only
protein $i$ is present in the sample. This probability is the detectability of peptide $j$ when it comes from protein $i$, referred to as the standard peptide detectability $d_i$. Substituting Equations (8) and (9) into Equation (7) leads to

$$P(x_1, \ldots, x_m | y_1, \ldots, y_m) = \frac{\prod_i P(x_i) \prod_j (\prod_i (1 - x_i d_j))^{1-y_j} \prod_j (1 - \prod_i (1 - x_i d_j))^{y_j}}{\sum_{(x'_1, \ldots, x'_m)} P(x'_1) \prod_j (\prod_i (1 - x'_i d_j))^{1-y_j} \prod_j (1 - \prod_i (1 - x'_i d_j))^{y_j}}. \quad (10)$$

In addition to this basic model, the authors also propose an advanced model which incorporates the peptide identification scores into the Bayesian model.

Recently, Serang et al. \[50\] introduce a novel Bayesian method named Fido for computing discriminative and interpretable posterior protein probabilities. The authors build a model using a few relatively simple assumptions and develop fast algorithms. The model uses three parameters, $\alpha$, $\beta$, $\gamma$, where $\alpha$ is the probability of generating associated peptides from present proteins, $\beta$ is the probability of creating peptides from a noise model, and $\gamma$ is the prior probability of each protein. With respect to the peptide degeneracy problem, this approach automatically apportions information from degenerate peptides during the marginalization procedure, rather than requiring an ad hoc adjustment. They also describe a series of heuristics including partitioning, clustering and pruning, which substantially increase the efficiency of computing posterior probabilities. In contrast to sampling, marginalizing yields an exact, closed-form solution in a finite period of time.

**Parametric model** Parametric model assumes that data come from a probability distribution and makes inference about the parameters of the distribution. Since parametric methods make more assumptions than non-parametric methods, they produce more accurate protein probability estimation if those extra assumptions are correct.

Q-score \[42\] applies a statistical algorithm resembling an approximation to a binomial distribution and uses parameters such as protein size, peptide match quality, number of peptide matches to a protein and size of spectral data set.

**PROT_ROBE** \[51\] uses binomial distribution when appropriate probabilities for database matches are available and develops a multinomial model that can be used with any database search results for the general case. In the following, we will mainly introduce the binominal model.

First, PROT_ROBE assumes that protein identification follows a binomial model and each database search result is a random Bernoulli event. The trial has two outcomes: a protein is either identified or not. Two binomial distributions are determined for protein assignment. The probability of the protein identification at each Bernoulli event is determined either from the relative length of the protein in the database (null hypothesis, $H_0$) or from the hypergeometric probabilities of peptides (an alternative hypothesis, $H_1$).

$H_1$ states that the peptide match distribution is governed by the hypergeometric probabilities determined by PEP_ROBE results. If there are $N$ spectra in the data set and $K$ matches (each with probability $p_i$) to a protein $A$, then the probability of all combinations leading to $K$ matches (with specified probabilities $p_i$) is estimated as

$$P_1 = \frac{N!}{K!(N-K)!} (1 - P(A))^{N-K} \prod_{i=1}^{K} p_i, \quad (11)$$

where $P(A)$ is

$$P(A) = \sqrt{K \prod_{i=1}^{K} p_i}. \quad (12)$$

The model can be interpreted as a Bernoulli trial where the probability of matching a protein $A$ is $P(A)$, and the probability of missing this protein is $(1 - P(A))$.

An alternative hypothesis, $H_0$ states that the distribution of peptide matches to a protein is governed by the binomial distribution determined from the database. The binomial probability of $K$ matches in $N$ trials is

$$P_0 = \frac{N!}{K!(N-K)!} P(A)^K (1 - P(A))^{N-K}, \quad (13)$$

where $P(A)$ is the ratio of the number of amino acids of protein A to the total number of amino acids in the database.
The maximum likelihood ratio, LR, is the ratio of $P_1$ to $P_0$. Then, PROT_PROBE assigns $-\log(LR)$ as a score to every protein.

ComByne [1] takes a $p$-value approach, modeling the probability that the peptide identifications would arise by chance alone. It makes use of information such as protein lengths, retention times and spectrum-to-spectrum correlation. ComByne considers the peptide assembly problem from the viewpoint of multiple hypothesis testing, in which the possibility of each protein being present in the sample is assessed. This is different from those methods whose aim is to optimally divide the proteins into two groups: present and absent.

**Optimistic model**

In contrast to parsimonious model, optimistic model adopts the strategy of returning all potential proteins that meet some simple criterion without further filtering. Obviously, the underlying assumption behind such a strategy is that the sample contains a large portion of homologous proteins.

As a typical example of optimistic model, DTASelect [15] reports proteins that have a sufficient number of different peptides or have at least one peptide showing up several times. It groups together proteins with identical sets of identified peptides and removes proteins for which the observed evidence is a subset of the peptides observed for another protein.

**Supplementary information model**

In the bipartite graph model, the output can not be further improved no matter how ideal the algorithm is, since the input information is limited. In order to improve the identification coverage and accuracy, extra information is employed. Borrowing extra information changes the input of protein inference problem: in addition to the standard input, supplementary information is used as part of the input, such as raw MS/MS data and single-stage MS data. These algorithms use supplementary information to identify proteins that may not be identifiable with high confidence by MS/MS evidence alone, but are nevertheless highly likely to be present as demonstrated by the combination of MS/MS evidence with supplementary information. Based on the type of the extra information, the peptide assembly methods can be divided into two categories: MS-generated data (peptide expression profile, raw MS/MS data, single-stage MS data) and data from other sources (protein interaction network, mRNA expression data, gene model).

Different supplementary information models have their own characteristics. Here, we compare them from two aspects: availability and reliability. Methods that incorporate MS-generated data can be applied to the analysis of any sample since such data are always available. In contrast, approaches that borrow data from other sources can only work when the required supplementary information exists. With respect to reliability, gene model is superior to other kinds of supplementary information exploited in protein inference.

We also have another classification criterion: the data fusion strategy. Standard input and supplementary information are combined based on different fusion strategies: the sequential data fusion strategy and the parallel data fusion strategy: either at data level or at decision level. As shown in Figure 7,

- **Figure 7:** The sequential fusion strategy. Some methods such as CEA, MSNet, PIPER introduce an additional stage to reprocess the protein list identified from the standard input.
the sequential fusion strategy firstly obtains a list of proteins identified from the standard input and then uses the supplementary data to adjust the result. For instance, PIPER [52], CEA [53], MSNet [54] and MSpresso [55] adopt the sequential fusion strategy. As shown in Figure 8, the parallel data fusion strategy handles standard input and supplementary information simultaneously. Such data fusion strategy can be further categorized as low or high-level fusion depending on the processing stage at which fusion takes place. Low-level fusion (fusion at data level) combines standard input and supplementary information to produce new raw data as new input. Existing methods such as the nested model [56], HSM [13], Barista [57], PSC [20] and PeptideClassifier [58, 59] fall into this category. High-level fusion (fusion at decision fusion) combines identification results coming from both standard input and supplementary information to obtain a consensus protein list. One typical example is the peptide mass fingerprinting method using MudPIT-based MS data [60].

**MS-generated data**

**Raw MS/MS data**  The MS/MS model mainly takes advantage of the raw MS/MS spectra information. Many algorithms designed for inferring proteins from a collection of PSMs divide the problem into two stages: assessing the quality of the PSMs and then inferring the protein set. Subdividing the protein identification problem in this fashion may result in a significant loss of information during the second stage of the analysis. For example, only a subset of spectra are assigned to a peptide during the peptide identification stage. Thus, information about the unassigned spectra is not available to the peptide assembly algorithms. Furthermore, suppose at most one peptide is assigned to each spectrum, and if for a particular spectrum that assignment happens to be incorrect, then information about the second-ranked, possibly correct peptide is not available during the protein identification stage. Also, if the quality of the match between a peptide and a spectrum is summarized using a single score, such as the probability assigned by PeptideProphet, then detailed information on how the peptide matches the spectrum is lost. In contrast, the raw MS/MS model described below, directly optimizes the number of identified proteins, taking into account all available information to obtain the best possible result.

HSM [13] is a typical example of using raw MS/MS data. It constructs a hierarchical statistical model to assess the confidence of peptides and proteins.
served variables and model parameters are estimated

HSM derives the probability distribution by incorporating multiple scores of the same peptide and considering three types of unobserved binary variables: presence/absence status for proteins and peptides, and the correctness of each peptide assignment. HSM starts with the identified proteins and peptides with at least one peptide hit and considers three types of unobserved binary variables: presence/absence status for proteins and peptides, and the correctness of each peptide assignment. HSM derives the probability distribution by incorporating multiple scores of the same peptide and peptides with at least one peptide hit and 0 otherwise. Then they use an EM algorithm to infer model parameters over all connected components.

More precisely, suppose \( N \) is the number of proteins with at least one peptide hit and \( M \) is the number of peptides assigned to at least one spectrum. \( Y_i \) is a binary indicator such that \( Y_i = 1 \) if protein \( i \) is in the sample and 0 otherwise. \( V_i \) is a binary variable such that \( V_i = 1 \) if the number of peptides identified for protein \( i \) is beyond a threshold and 0 otherwise. \( Z_j \) is the binary indicator such that \( Z_j = 1 \) if peptide \( j \) is present in the digested sample and 0 otherwise. \( W_{jk} \) is a binary indicator such that \( W_{jk} = 1 \) if the \( k \)th assignment of peptide \( j \) to a spectrum is correct and 0 otherwise, with corresponding matching score \( S_{jk}(k = 1, 2, \ldots, T_j) \). \( C_j \) is the set of proteins that could potentially generate peptide \( j \).

The model is:

\[
[Y, Z, W, S, V] = [Y][Y][Z][Y][W][Z][S][W]
\]

\[
=N \prod_{i=1}^{N} [Y_i] \prod_{i=1}^{N} [V_i][Y_i] \prod_{j=1}^{M} [Z_j][Y_j] \prod_{j=1}^{M} \prod_{k=1}^{T_j} [W_{jk}][S_{jk}][W_{jk}].
\]

(14)

Then, the \( Y, Z \) and \( W \) are treated as the unobserved variables and model parameters are estimated via EM algorithm. Finally, the confidence of peptides and proteins can be calculated conditioned on \( S \) and \( V \).

A nested model is proposed in Ref. [56], which can also estimate the protein probability and peptide probability simultaneously. It allows for appropriate evidence feedback between proteins and their constituent peptides. The nested model is built on several reasonable assumptions but ignores the peptide degeneracy problem entirely. Similar to HSM, Barista [57] formulates the protein identification problem as an integrated optimization problem. In Barista, the protein identification problem can be represented as a tripartite graph, with layers corresponding to spectra, peptides and proteins (Figure 10).

The input to the problem is the tripartite graph, with a fixed set of features assigned to each peptide-spectrum match. In this method, they represent each PSM using 17 features that collectively describe properties of the spectrum and the peptide, as well as the quality of the match between the observed and theoretical spectra. This model uses a machine learning method to directly optimize the total number of proteins identified by the experiment. Furthermore, it does not filter any PSMs at any stage of the analysis, with the observation that low scoring PSMs can imply the protein’s presence when other PSMs from the same protein are available.

Scaffold [61] uses a peptide-spectra-protein graph structure for peptide assembly. Under this scheme, the original mass spectral evidence can be assigned to multiple possible peptides in a ‘peptide group’. That peptide group is linked to multiple proteins by their associated peptide sequences. Then Scaffold uses a

![Figure 9: Four layers of the HSM. Layer 1 is a marginal distribution and the other three layers are conditional distributions.](image-url)

![Figure 10: Barista. The tripartite graph represents the protein identification problem, with layers corresponding to spectra (bottom), peptides (middle) and proteins (top). Barista computes a parameterized non-linear function on each PSM feature vector. The score assigned to a peptide is the maximum PSM score associated with it. The score assigned to a protein is a normalized sum of its peptide scores.](image-url)
greedy algorithm to assign the peptide group to the most confidently identified proteins.

**Single-stage MS data** While MS-based methods provide wider coverage than MS/MS-based methods, their identification accuracy is lower since MS data have less information than MS/MS data. It is natural to consider the combination of MS data and MS/MS data in a unified model such that the identification performance can be further improved.

Peptide mass fingerprinting (PMF) is a technique used to identify proteins by matching observed peptide masses to theoretical peptide masses [40]. The presumption of PMF is that every protein has a set of unique peptides and thus masses of these peptides can form its fingerprinting [62–64]. Initially, PMF is proposed to identify single purified proteins separated by 2D gel electrophoresis [65] and is only applicable to data generated from high accuracy instruments. Recently, PMF is extended to identify protein mixtures from shotgun proteomics data [20, 66].

Some methods [20, 60] are proposed to combine the MS information and MS/MS data together to identify the proteins in the sample so that the identification performance is further improved, especially for the ‘one hit wonders’.

**Peptide expression profile** Proteomics discovery platforms generate both peptide expression information and peptide identification information. In label-free quantitative proteomics studies, peptide expression information, such as peptide intensity information, has been widely used [67]. Recently, researchers also introduce it into protein identification.

PIPER [46] assumes that peptides derived from the same protein are correlated with expression profiles. The peptide-to-protein assignments along with the peptide expression profiles are submitted to PIPER for expression correlation filtering. The output is a list of differentially expressed proteins along with an estimate of the false-positive error rate.

Spectral counting [68, 69] is also a good method for label-free qualification. It involves measuring the abundance of a given protein based on the number of tandem mass spectral observations for all its constituent peptides. Thus, it offers a practical alternative to peak intensity measurements, which relies heavily on computational efforts for chromatogram alignment and peak detection [70]. Using spectral counting for peptide expression estimation to enhance protein identifications would be an interesting problem in future research.

**Data from other sources** Besides MS-generated data, we can also use data from other sources to facilitate protein identification. For example, protein interaction network information has been used to enhance protein identification.

**Protein interaction network** Protein–protein interactions indicate two or more proteins binding together, when carrying out some biological functions. In most peptide assembly methods, proteins are considered as independent entities. Nevertheless, accumulating evidence suggests that most biological functions are carried out from interactions among proteins, and a discrete biological function is rarely be attributed to an individual protein. Thus, some methods start to take advantage of protein–protein interaction networks.

CEA [53] is a clique enrichment approach to rescue eliminated proteins by incorporating the relationship among proteins as embedded in a protein interaction network. The model is based on the general concept that proteins involved in the same biological process or pathway tend to be close to one another in the protein interaction network [71]. After peptide identification and peptide assembly, proteins in the maximal protein list are grouped into confident proteins and non-confident proteins, and then mapped to the protein interaction network. All maximal cliques are enumerated from the network and evaluated for the enrichment of confident proteins.

MSNet [54] improves protein identification in shotgun proteomics experiments by utilizing protein–protein interaction network information from gene functional network. In MSNet, each protein receives a revised identification score with contributions both from direct MS-based evidence and MS evidence of neighboring proteins in the gene functional network. In this way, MSNet can rescue proteins which are present in the sample but were eliminated in the MS/MS-based protein identification.

**mRNA expression data** At the stage of transcription, genes are differentially transcribed due to the chromatin arrangement and the activity of transcription factors. Thus, the stability and distribution of the different transcripts need to be regulated. The regulatory processes include capping, splicing, addition of tail poly(A) and RNA editing. Then regulated mRNA is decoded by the ribosome to produce a
specific amino acid chain, or polypeptide, that will later fold into an active protein.

This mRNA information can be applied to peptide assembly process. For example, most of peptide assembly methods assume that all proteins are equally likely to be present. In reality, other information may be readily available to help infer the probability of protein’s presence when evidence from the MS/MS experiment is weak. For example, raw MS/MS identification score of protein $A$ falls below a given confidence threshold but its mRNA abundance is high enough. In this case, we have a reason to believe that protein $A$ is present in the sample.

MSpresso [55] re-examines protein identification scores with respect to their mRNA abundance. It boosts the protein identification scores given sufficient mRNA concentration. Proteins are then labeled present if their MSpresso probability is larger than a newly determined cutoff.

More formally, $K$ is a Bernoulli variable where $K = 1$ is the event that the protein is present in the sample, and $P(K = 1)$ is the probability of that event. The MSpresso score, $P(KS = s, M = m)$, is the posterior probability that a protein is present in the sample given its associated mRNA abundance $M = m$, and its raw MS/MS protein identification score $S = s$. Using Bayes’ law,

$$P(K | S, M) \propto P(K)P(S|K)P(M|K)$$

$$\propto P(K)\left(\frac{P(K|S)P(S)}{P(K)}\right)\left(\frac{P(M|K)P(M)}{P(K)}\right)$$

$$\propto \frac{1}{P(K)}\left(\frac{P(K|S)P(S)}{P(K)}\right)\left(\frac{P(M|K)P(M)}{P(K)}\right),$$

where $P(K|S)$ and $P(K|S)$ are the posterior probabilities of a protein existing in the sample, given only its mRNA abundance $M$ and primary identification score $S$, respectively. $P(K)$ is the prior probability of the protein being present. The formula can be rewritten as the following:

$$P(K | S, M) = \sum_{K=0,1} P(K|S)P(K|M)/P(K).$$

**Gene model** Compared to the methods using protein interaction network and mRNA expression data, it is more appealing to employ the gene model because gene model is a relative mature field and it is easier to accurately and quickly obtain the information. In the translation process, alternative transcriptional start sites and alternative splicing sites make mRNA sequences produced from the same DNA different and further lead to different proteins. In other words, a DNA segment can generate multiple mRNA and proteins (only for eukaryotes), which increases the number and variety of the proteins. But the obtained proteins are homologous and difficult to distinguish. Moreover, the degenerate peptides generated by homologous proteins have become a great challenge for protein inference.

According to the gene model–protein sequence–protein identifier (identified peptides) relationships, Grobei et al. [58] classify each peptide sequence. This allows shared peptides to be further distinguished depending on whether the implied proteins could be encoded either by the same or by distinct gene models. They distinguish five peptide evidence classes for Arabidopsis pollen. The same method is applied to PeptideClassifier [59]. But for prokaryotes PeptideClassifier reports three peptide evidence classes and for eukaryotes, to capture potential alternative splice isoforms, it considers three additional evidence classes. Both methods facilitate seamless integration with transcriptomics data.

The recent application of gene model is Markovian Inference of Proteins and Gene Models (MIPGEM) [14]. It is a statistical model that addresses the problem of protein and gene model inference from shotgun proteomics data. In particular, MIPGEM deals with dependencies among peptides and proteins using a Markovian assumption on $k$-partite graphs (Figure 11) which states that only the neighboring proteins matter in the conditional distribution for the peptides.

MIPGEM first divides the bipartite graph into different connected components and assumes that different components are independent. For instance, in Figure 3, the first component has two peptides $(1,2)$ and two proteins $(1,2)$. Then the probability distribution of the peptide scores can be modeled as

$$P\{p_i; i \in \xi_1\} = P\{p_i; i \in \xi_1\} \cap \cdots \cap P\{p_i; i \in \xi_R\}$$

$$= \prod_{r=1}^{R} P\{p_i; i \in \xi_r\},$$

where $\xi_r$ (with $r = 1, 2, \ldots, R$) is the set of peptides of the $r$-th connected component of the bipartite graph. It denotes with $z_j = 1$ or 0 whether a protein $j$ is present or absent in the sample of interest, respectively, and denotes with $p_i$ the peptide probability or score for the presence of peptide $i$. 

Furthermore, let $\xi$ be the index set of all peptides and $\{j; j \in \eta\}$ be the list of candidate proteins.

The factors in the product in Equation (17) can be rewritten as

$$P(p_i; i \in \xi) = \sum_{z_j \in \{0,1\}, j \in \mathbb{R}(\xi)} \prod_{j \in \mathbb{R}(\xi)} P(p_i(z_j; j \in \mathbb{N}(i))) \cdot \prod_{j \in \mathbb{R}(\xi)} P(z_j),$$

(18)

where $\mathbb{R}(\xi) = \{j; j \in \eta\}$ and there exists an edge between $i$ and $j$ for at least one $i \in \xi$, which is the range of $\xi$. $\mathbb{N}(i)$ are the neighbors of the peptide $i$, that is, the set of all the proteins $j$ having an edge to the peptide $i$.

In addition, $P(p_i(z_j; j \in \mathbb{N}(i)))$ is defined as:

$$P(p_i(z_j; j \in \mathbb{N}(i))) = \begin{cases} \frac{1}{\alpha} & \text{if } \sum_{j \in \mathbb{N}(i)} z_j = 0 \\ f_i(p_i) & \text{if } \sum_{j \in \mathbb{N}(i)} z_j > 0 \end{cases}$$

(19)

with

$$f_i(x) = \begin{cases} b_1(x - l) & l \leq x \leq m \\ (b_1 + b_2)(x - m) + b_1(m - l) & m \leq x \leq u \end{cases}$$

(20)

where $b_1 \geq 0$, $b_2 \geq 0$ are unknown parameters and $l = \min(p_i)$, $m = \text{median}(p_i)$, $u = \max(p_i)$. The density function $f_i(x)$ must fulfill

$$\int_{l}^{u} f_i(x) dx = 1.$$

(21)

One of the parameters $b_1$ or $b_2$ has to be estimated. The second one can then be computed with the constraint on the integral.

The probability that a protein $j$ is present given the peptide scores is computed as:

$$P[Z_j = 1|p_i; i \in \xi] = \frac{A(1)}{A(0) + A(1)}$$

(22)

with

$$A(z) = \sum_{z_j \in \{0,1\}, k \neq j} \left[ P(p_i; i \in \mathbb{N}(i))|Z_j = z, Z_k = z_k \right]$$

\cdot P(Z_j = z) \cdot \prod_{k \neq j, k \in \mathbb{R}(\xi, i)} P(Z_k = z_k),$$

(23)

where $d_i$ is the index of the connected component holding the protein $j$.

In addition, MIPGEM scores the encoding gene models to address the problems of shared peptides and ambiguous proteins. The principle of gene model is the following:

$$P[\text{gene model occurs}] = 1 - P[\text{none of its proteins occur}]$$

RESULT VALIDATION FOR PROTEIN IDENTIFICATION

After a protein inference model is developed, how to assess their performance is still a problem. The most straightforward strategy for assessing the performance of different protein identification methods is to generate some synthetic datasets in which the ground-truth proteins are known in advance. To date, there are already some standard data sets, e.g., the standard mix database of 18 proteins [72]. However, such benchmark data sets usually contain no more than 100 proteins, providing nothing like the complexity of most real data problems. As a
result, these standard mixtures inevitably provide only a very limited comparison of the performance of different methods. Thus, it is necessary to build some standard data sets that have much more ground-truth proteins for performance test in future research.

An alternative strategy is to use simulation data that is reasonably close to reality and provides a fair testing ground for different methods. To date, there are already some simulation software for LC-MS and LC-MS/MS experiments (e.g. [73]). However, the characteristics of such synthetic data depend heavily on the underlying assumption of simulation model, making it difficult to provide an unbiased comparison.

Another solution is to create an expertly curated large-scale reference data set as a substitute when real benchmark data (with thousands of proteins known in advance) are not available. We call it ‘the reference set approach’. However, building such reference set is not an easy task since it needs to conduct many experiments to obtain a confident protein list with human intervention.

Thus, in most cases, we can only estimate the reliability of protein identifications. At the peptide level, current estimation methods can be classified into FDR approaches and p-value-based approaches.

FDR approaches
FDR [74], the expected fraction of false-positive assignments, has become a widely used measure for assessing PSMs. FDR for PSMs can be estimated by means of decoy database search strategies in which the acquired tandem mass spectra are searched against a target–decoy protein database. As shown in Figure 12, such target–decoy database contains all (target) protein sequences possibly present in the sample and an equal number of decoy sequences by reversing or reshuffling target protein sequences.

The FDR can be estimated as the ratio of the number of decoy matches to that of target matches. Researchers usually filter peptide identifications using a chosen FDR threshold (say FDR of 5%), which can be estimated from the percentage of decoy matches at every potential cut point of search algorithms. FDR approaches are particularly appealing since they constitute a generic and independent approach to validate PSMs generated by any type of identification strategy.

Unfortunately, most available FDR methods focus on quality control on the level of PSMs. Deriving FDR for protein identifications is more difficult than determining FDR for PSMs. This is because protein identifications are derived from peptide assembly procedure, errors determined at the PSM level may propagate to the protein identification level. Controlling quality at the level of PSMs does not ensure quality at the level of protein identifications. This issue has so far not been appropriately appreciated since the distinction between PSMs and protein identifications is frequently ambiguous in the literature [75]. The error rate estimation of protein identification has to account for the fact that false and true PSMs distributing differently across the protein database. While false PSMs comparably distribute over all entries in the database [76], true PSMs map exclusively to the smaller subset of proteins being present in the biological sample. As a result, protein identification FDR in practice is larger than the PSM FDR [77]. Therefore, optimizing FDRs for peptides and proteins must be considered as different problems that are best addressed by different approaches. Gupta et al. [78] addressed this issue using single-peptide rule. Recently, Hather et al. [79] proposed a method for estimating the FDR and local FDRs (LFDRs) for both peptide and protein identifications based on randomized database matches and isotonic regression.
However, none of these approaches reliably quantifies the confidence of protein identifications in very large, integrated data sets.

To fulfill this void, a new approach called MAYU [75] is presented to quantify the uncertainty of protein identifications in the context of large-scale data sets. MAYU extends the well-established target–decoy strategy from PSM level to protein level (Figure 13). It assembles a user-defined set of PSMs, e.g. the set of PSMs at FDR = 0.01, to a list of protein identifications and estimates the expected number of false-positive identifications mapping to the target database.

MAYU refers to the set of all protein identifications as $H_t$, the subset mapping to the target database $P_t$ as $H_t$ and its complement as $H_d$. MAYU distinguishes three types of protein identifications, i.e. (i) true positive (TP) identifications, which is denoted by $H_{tp}$. A protein identification is considered to be TP, if it contains at least one TP PSM. Similarly, a protein identification is considered to be a false positive (FP), if all its PSM are FPs. The second type (ii) covers the set $H_{fp}$ of FP protein identifications mapping to $P_t$, the complementing set with its identifications projecting to the decoy database $P_d$ equals $H_d$. As the third type (iii) MAYU introduces the set $H_{cf}$ that is composed of all protein identifications in $P_t$, each containing FP PSM. Note that elements of $H_{cf}$ can be TP as well as FP. Then, $E[h_{y|t}, h_{y|f}, \theta_{exp}]$ is the number of FP protein identifications given the proteomics experiment characterized by parameters $\theta_{exp}$ and its outcome $h_t$, $h_{cf}$, $\theta_{exp}$ particularly includes parameters related to the target protein database, such as the number of protein entries $N$. By application of Bayes formula and by assuming $P(h_{y|t}, h_{y|f}, \theta_{exp})$ and $P(h_t|h_{y|f}, \theta_{exp})$ to be uniform and $h_{d|f} = h_{y|f}$, $E[h_{y|t}, h_{d|f}, \theta_{exp}]$ evaluates as follows.

$$
E[h_{y|t}, h_{y|f}, \theta_{exp}] = \sum_{h_{y|t}} \sum_{h_{y|f}} P(h_{y|t}|h_{t}, h_{f}, \theta_{exp}) \\
= \sum_{h_{y|t}} \sum_{h_{y|f}} P(h_{y|t}|h_{t}, h_{f}, \theta_{exp}) \cdot P(h_{y|t}|h_{y|f}, \theta_{exp}) \\
= \sum_{h_{y|f}} \sum_{h_{y|t}} P(h_{y|f}|h_{y|t}, h_{y|f}, \theta_{exp}) \cdot P(h_{y|t}|h_{t}, h_{f}, \theta_{exp}) \\
= \sum_{h_{y|f}} \sum_{h_{y|t}} P(h_{y|f}|h_{y|t}, h_{y|f}, \theta_{exp}) \cdot \theta_{exp} \\
= \sum_{h_{y|f}} P(h_{y|f}|h_{y|t}, h_{y|f}, \theta_{exp}) \cdot \frac{N - h_{d|f} + 1}{N + 1},
$$

(24)

where $h_{y|f} = h_t - h_{y|t}$.

Then protein identification FDR is computed as the ratio of expected number of false-positive protein identifications to the total amount of protein identifications mapping to the target database ($h_t$).

In Ref. [75], MAYU is compared with two other FDR estimation approaches, ProteinProphet and a naïve target–decoy strategy. They find that estimation results derived with ProteinProphet are too ‘optimistic’ while estimation results from a naïve target–decoy approach are too ‘pessimistic’. The naïve target–decoy strategy estimates protein identification FDR analogously to PSM FDR, i.e. by approximating the expected number of false positive (FP) protein identification by the number of decoy protein identification.
**P-value-based approaches**

Contrary to FDR approaches which apply a single FDR threshold to all spectra from the dataset, p-value-based approaches report p-values specific to individual peptide-spectrum matches. These methods are primarily based on fitting the distribution of scores with a specific parametric model and use information like properties of spectra and peptide length to increase the accuracy. OMSSA [35] approximates the scores of each peptide-spectrum match with Poisson distribution. Klammer et al. [80] fit the distribution of SEQUEST Xcorr scores with Weibull distribution. However, the approaches based on fitting specific parametric models normally cannot be generalized to other platforms and they assess p-values at peptide level rather than at protein level. To address this issue, Spirin et al. [81] developed a method independent of instrumentation or software scoring method. It assigned a spectrum-specific p-value to each peptide-spectrum match and combined these p-values to assess statistical significance of protein identifications.

First, the probability $P$ that the database search with spectrum $i$ will result in a false identification with score $s$ is given by:

$$P = 1 - F(s) = 1 - \exp \left( -\alpha_i \exp \left( -\left( s - \mu_i \right) / \beta_i \right) \right).$$

(25)

where $F(s)$ is the cumulative extreme value distribution function. The parameter $\alpha_i$ depends on the relative database size. Parameters $\mu_i$ and $\beta_i$ are specific to the spectrum and depend only weakly on the relative databases size in the asymptotic regime.

Then p-values of individual peptides matching the same protein are combined to cumulative protein p-value using Stouffer’s method [82]. For the peptide match with the $k$-th lowest p-value among all peptides from the same protein, the method converts the p-value $P_i$ into the standard normal deviates $Z_i$ using Z-test. The sum of these deviates, divided by the square root of the number of tests $k$

$$Z = \frac{\sum_{i=1}^{k} Z_i}{\sqrt{k}}$$

(26)

can be shown to follow the standard normal distribution. The minimal value of this probability over all $k$ peptides is the p-value for the protein identification.

The protein p-value strategy outlined by Equation (26) is based on the assumption that p-values resulting from different peptide identifications are independent. Considering peptide correlations when evaluating protein p-value is a very important and open problem.

While the proteomics community often takes great care in evaluating peptide-level error rates, the protein-level result validation is often ignored. It is very important to report protein-level error rate along with protein identification results. Lacking of this checkpoint raises concerns about the validity of studies, such as biomarker discovery, where the number of identified proteins as well as the reliability of each individual identification is of critical importance.

**EXPERIMENT**

We have summarized existing peptide assembly methods using two systematic and expandable frameworks. Then, one may ask: Do we still need to develop new peptide assembly algorithms? To answer this question, we have to conduct a comprehensive evaluation and comparison to assess the strengths and weaknesses of current available algorithms. More precisely, we examine the overlap among the results of five typical methods to show that the set of proteins identified from the same benchmark data set can vary significantly. This indicates that the performance of current peptide assembly algorithms is still far from satisfactory. Meanwhile, we use two different evaluation methods in the performance comparison: one protein-level FDR estimation method (MAYU) and the reference set approach. In addition, we also describe how protein inference is performed from start to finish using Trans-Proteomic Pipeline (TPP) software [83] as an example. It shows that the final identification performance can change dramatically if one uses different parameter combinations in the analysis pipeline.

In fact, there are already some research efforts that compare the performance of different peptide assembly algorithms. For instance, Xue et al. [84] evaluated the performance of four algorithms: ProteinProphet [12], PROT_PROBE [51], Poisson model [85] and two-peptide hits. They also validated that the probabilities of identified proteins were strongly correlated with several experimental factors including spectra number, database size and protein abundance distribution. However, they only evaluated peptide assembly algorithms from bipartite graph model, leaving supplementary information
model uncovered. Here, we compare five typical methods: IDPicker (Parsimonious Model) [45], ProteinProphet (Non-parametric Model) [12], two-peptide hits (Optimistic Model), MSNet (Protein Interaction Network) [54], Peptide Classifier (Gene Model) [59] from both bipartite graph model and supplementary information model. All the five methods we use are search engine-independent. We do not choose DTASelect as the representative of optimistic model since it can only handle peptide identification results generated from Sequest. Alternatively, we use two-peptide hits in the performance evaluation since it returns all potential proteins matching at least two peptides without any further filtering.

**Data sets and evaluation methods**

We use two data sets that have been used in MSNet [54] in our experiments. The raw data are available at [http://www.marcottelab.org/users/MSdata/Data_02/](http://www.marcottelab.org/users/MSdata/Data_02/) and [http://www.marcottelab.org/MSdata/Data_04/](http://www.marcottelab.org/MSdata/Data_04/), respectively. Since both data sets are generated from samples that consist of yeast proteins, we use Yeast_D1 and Yeast_D2 to denote them in the remaining parts of this section.

The motivation behind choosing these two data sets is that both of them have a reference set, enabling us to conduct performance comparison with multiple evaluation criteria. An identified protein is labeled as a true positive if it is present in the corresponding reference set. An identified protein is labeled as a true positive if it is present in the corresponding reference set. In Yeast_D1, the protein reference set is prepared from four MS-based proteomics data sets and three non-MS-based data sets. We use a list of 4265 proteins observed in either two or more MS-data sets or any of non-MS-data sets as the reference set, which can be obtained from [http://www.marcottelab.org/MSdata/gold_yeast.html](http://www.marcottelab.org/MSdata/gold_yeast.html). In Yeast_D2, a list of 593 proteins that is composed of known ribosomal, translation and ribosome biogenesis proteins is used as the reference set.

In the experiment, we use both MAYU [75] and the number of identified proteins from the reference set to compare different methods. MAYU is strongly dependent on the number of decoy peptides in its FDR estimation procedure. Decoy peptides/proteins are not truly present entities so that they have no corresponding biological counterparts. Thus, some existing methods that utilize supplementary information almost can eliminate all decoy entries in the peptide assembly stage. As a result, target–decoy based FDR estimation methods like MAYU may over-estimate the performance of MSNet and PeptideClassifier in the comparison. Based on this observation, we only use MAYU to evaluate the results from IDPicker, ProteinProphet and two-peptide hits.

**Database search and post-processing**

We use a sequence file downloaded from [http://aug.csres.utexas.edu/msnet/](http://aug.csres.utexas.edu/msnet/) as the target database, which contains 7248 proteins and 33 contaminants. All tandem mass spectra are searched against the protein sequence database and its randomized version (forward and reverse) with X!Tandem search engine (v2009.10.01.1). We use default search parameters wherever possible, assuming that parameters have already been optimized. Some important parameter specifications are listed in the following:

- Fragment monoisotopic mass error: 0.4 Da
- Parent monoisotopic mass error: 100 ppm
- Minimum peaks: 15
- Minimum fragment m/z: 150.

We use a tool included in the TPP (v4.4) software to convert tandem result files to pepXML files. If necessary, PeptideProphet is used as a second step to analyze the pepXML files. The input of each peptide assembly method is listed as follows:

- IDPicker: the pepXML files
- ProteinProphet: the output of PeptideProphet
- Two-peptide hits: the output of PeptideProphet
- MSNet: the proteins and their probabilities reported by ProteinProphet
- PeptideClassifier: the output of PeptideProphet.

The peptides with PeptideProphet probability >0.05 are considered as candidate peptides, and the proteins containing at least one candidate peptide are considered as candidate proteins. The probability threshold is set to be 0.8 and 0.6 in ProteinProphet and MSNet, respectively.

**Comparison of peptide assembly algorithms**

The protein identification results on two data sets are summarized in Table 2. The number of reported proteins by each algorithm is indicated in bold text. The figures in brackets are the number of correctly identified proteins in the reference set.
First of all, the results in Table 2 show that the identification performance can vary significantly if we use one method versus another with respect to the number of (correctly) identified proteins. To further investigate such performance variation, we use Venn diagram to describe the overlap among five methods in terms of the number of identified proteins in Figures 14 and 15. For Yeast_D1 (Fig. 14), out of a possible 1888 proteins from the five algorithms (union), 766 proteins are identified by all five algorithms (intersection), while 489 proteins are identified by a single algorithm. For Yeast_D2 (Fig. 15), 117 proteins are identified based on a consensus of all five algorithms (intersection), while 489 proteins are identified by one or more algorithms (union). Clearly, this poor concordance indicates that more research efforts are still needed to develop effective peptide assembly algorithms. Compared to the methods from bipartite graph model, MSNet and PeptideClassifier consistently exhibit good performance. This indicates that the use of supplementary information can boost the identification performance.

Another interesting observation is that IDPicker (Parsimonious Model) can report more proteins than other methods such as two-peptide hits (Optimistic Model) for Yeast_D1. This is because there are a large number of one-hit wonders (proteins that have only one identified peptide) in Yeast_D1. For instance, there are 724 one-hit proteins in the output of PeptideProphet for Yeast_D1, which is ~40% of the number of total candidate proteins. In addition, IDPicker uses the output from X!Tandem rather than the output of PeptideProphet as the input. That means the input of IDPicker contains more identified peptides than the input used in two-peptide hits.

**Table 2:** The performance of five methods

<table>
<thead>
<tr>
<th>Data</th>
<th>IDPicker</th>
<th>ProteinProphet</th>
<th>MSNet</th>
<th>PeptideClassifier</th>
<th>Two-peptide hits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yeast_D1</td>
<td>1293 (1152)</td>
<td>1530 (1254)</td>
<td>1376 (1319)</td>
<td>1008 (977)</td>
<td>1112 (978)</td>
</tr>
<tr>
<td>Yeast_D2</td>
<td>154 (116)</td>
<td>322 (162)</td>
<td>291 (166)</td>
<td>232 (145)</td>
<td>317 (133)</td>
</tr>
</tbody>
</table>

Two protein samples undergo MS/MS analysis to generate ten lists of proteins identified by five peptide assembly algorithms. The figures in brackets indicate the number of proteins present in the reference set.

**Figure 14:** Five-way Venn diagram showing the overlap among five peptide assembly algorithms for Yeast_D1. The number of identified proteins by one or more algorithms is indicated, e.g. 766 proteins are identified based on a consensus of all five algorithms (intersection), whereas 1888 proteins are identified by one or more algorithms (union).
We also use the FDR value estimated with MAYU to compare three methods of bipartite graph model: IDPicker, ProteinProphet and two-peptide hits. Table 3 indicates that for Yeast_D2 IDPicker achieves the lowest FDR, while two-peptide hits gives the highest FDR and for Yeast_D1 the opposite is true. We can derive similar results for Yeast_D2 from Table 2. This is because Yeast_D2 has more proteins identified by more than two peptides. Reporting any protein which has more than two identified peptides without filter will greatly increase the number of potential false positives. On the other hand, for Yeast_D1, Tables 2 and 3 show different results. This fact indicates that current performance evaluation methods are not perfect and need further improvement. Meanwhile, it also shows that statistical methods perform most consistently with respect to FDR.

### Protein inference analysis pipeline

In this review, we have described three steps of protein inference problem. Here, we conduct some experiments to provide the reader a sense of how protein inference is performed step by step. More importantly, it shows that the use of different parameters in each step may change the performance significantly. Consequently, such uncertainty will propagate to subsequent analysis step. The experiment is completed on TPP software (v4.4), which is a collection of integrated tools for MS/MS proteomics.

We use PeptideProphet as the tool for post-processing peptide identification results from X!Tandem, and use ProteinProphet as the peptide assembly method. The overall number of identified peptide hits and protein identifications from Yeast_D1 are listed in Table 3 when PeptideProphet is executed under some typical parameter combinations. Since there are more than 10 input parameters for PeptideProphet, we only choose four typical parameters in the experiment. To test the influence of PeptideProphet on the final results, we fix the parameters of ProteinProphet using their default values. The results in Table 4 show that the use of different parameter settings in PeptideProphet has a great impact on the identification results. We also adjust two parameters for ProteinProphet when the output of PeptideProphet is obtained from the 5th experiment in Table 4. As shown in Table 5, the parameter adjustment for ProteinProphet can also affect the number of identified proteins.

#### Table 3: The FDR values estimated with MAYU

<table>
<thead>
<tr>
<th>Data</th>
<th>IDPicker (%)</th>
<th>Protein Prophet (%)</th>
<th>Two-peptide hits (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yeast_D1</td>
<td>6.1</td>
<td>5.4</td>
<td>3.2</td>
</tr>
<tr>
<td>Yeast_D2</td>
<td>0</td>
<td>3.3</td>
<td>5.75</td>
</tr>
</tbody>
</table>

![Figure 15: Five-way Venn diagram showing the overlap among five the peptide assembly algorithms for Yeast_D2.](image)


FUTURE PERSPECTIVE

In shotgun proteomics, one important problem is to identify all proteins present in the sample accurately. Unfortunately, such protein inference problem is only partially solved since three technical challenges still remain unsolved: the identification coverage problem, the identification ambiguity problem and the identification validation problem.

- **Identification coverage problem:** the protein identification task is subdivided into two stages: first identifying a collection of peptides with a low FDR, and then inferring the protein set from the resulting collection of peptides. However, due to the complexity of MS data and the limitations of current peptide identification algorithms, up to 80–90% of MS/MS spectra in a typical liquid chromatography tandem mass spectrometry experiment cannot be interpreted with high confidence. As a result, only a subset of peptides present in the sample can be detected. This may miss those proteins that have no constituent peptides being identified in protein inference.

- **Identification ambiguity problem:** with the advance of MS instruments and peptide identification algorithms, it is reasonable to expect that we are capable of identifying at least one constituent peptide for each protein present in the sample in the near future. Even we have already achieved this objective, we have to face another difficult challenge in protein inference: the identification ambiguity problem. As we have discussed in previous sections, degenerate peptides and one-hit wonders are the main sources for the ambiguity issue. It is generally very difficult to determine which proteins are truly present in the sample if they share the same set of peptides or have only one constituent peptide being detected.

- **Identification validation problem:** assessing the correctness of reported proteins is critical to the success of proteomics applications. For instance, only truly present proteins that are differentially expressed between case and control are of practical interest in biomarker discovery. If we are unable to accurately judge the correctness of reported proteins, some random artifacts that have different expression levels may be returned to the user. Though researchers have realized the importance of identification validation, there is still no consensus of the best evaluation measure so far.

As summarized in this survey, researchers have proposed many solutions from different angles to tackle above technical challenges. However, both our own performance comparison and similar studies from other groups have shown that the identification results of different methods can disagree with each other significantly. This indicates that the performance of current available protein inference methods is still far from satisfactory in practice. Therefore, more research efforts are still needed towards this direction. Here, we would like to discuss some future perspectives along with on-going specific efforts and present our view on potentially promising area and the development of future tools.

First of all, researchers have proposed many strategies to increase the protein identification coverage. We can divide on-going efforts from the peptide identification perspective into three classes.

### Table 4: Performance of protein inference under different parameter combinations of PeptideProphet

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Only use Expect Score as the discriminant</td>
<td>Yes</td>
</tr>
<tr>
<td>Use gamma distribution to model the negatives</td>
<td>Yes</td>
</tr>
<tr>
<td>Use decoy hits to pin down the negative distribution</td>
<td>No</td>
</tr>
<tr>
<td>Force the fitting of the mixture model</td>
<td>No</td>
</tr>
<tr>
<td>The number of identified peptides</td>
<td>4641</td>
</tr>
<tr>
<td>The number of identified proteins</td>
<td>529</td>
</tr>
</tbody>
</table>

### Table 5: Performance of protein inference under different parameter combinations of ProteinProphet

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Normalize NSP for Protein Length</td>
<td>Yes</td>
</tr>
<tr>
<td>Use Expected Number of Ion Instances</td>
<td>No</td>
</tr>
<tr>
<td>The number of identified proteins</td>
<td>1530</td>
</tr>
</tbody>
</table>
• Developing more accurate peptide identification algorithm. Fundamental to any protein inference method, a set of confident and complete peptide identifications are of primary importance. New peptide identification algorithms continue to be developed very year since current software generally failed to interpret most tandem MS spectra generated in a single experiment.

• Combining results from different peptide identification algorithms. Since each peptide identification method uses a different algorithm, which proceeds from a different view of how the spectrum should be interpreted. Thus, the peptide identification results for one spectrum from various algorithms may differ significantly. Generally, it is feasible to identify more confident peptides if we find complementary identification algorithms and combine the results in an effective way [86].

• Combining results from multiple replicates. The protein and peptide mixtures are typically still too complex to allow the mass spectrometer to acquire tandem mass spectra for all peptides in a single LC-MS/MS experiment. Consequently, LC-MS/MS experiments are usually repeated extensively, in order to increase the number of peptides for which tandem mass spectra are acquired [87]. Hence, we may increase the peptide identification coverage by merging results from multiple replicate experiments.

Recently, the supplementary information model begins to receive much more attention in which extra information from other data sources is used to facilitate the identification of more proteins. For instance, the use of single-stage MS data makes it possible to find some extra proteins whose peptide digestion results are not covered by MS/MS data [66]. We believe such supplementary information model is the most promising strategy for improving the identification coverage of protein inference. Though it has been demonstrated that we can obtain very encouraging results by incorporating different kinds of supplementary information, the following questions remain open and need further investigation in future research.

• To date, researchers have investigated the feasibility of using raw MS/MS data, single-stage MS data, peptide expression profiles, mRNA expression, protein interaction network and gene model. Then, a natural question is: Are there other kinds of supplementary information that can be utilized in protein inference? If the answer is yes, we need to further study how to incorporate it in the inference process.

• Different supplementary information models have different characteristics. Therefore, one may be interested in the following questions: Which supplementary information is more reliable and easy to use? Which supplementary information is more effective in improving the protein identification performance?

• It is also possible to further extend the identification coverage if we can use all kinds of supplementary information simultaneously. Then, the question becomes: Can we provide an integrated inference model that incorporates all extra information seamlessly? Furthermore, we need to build a unified database and provide corresponding software for storing and manipulating different kinds of supplementary information to facilitate protein inference.

To handle the identification ambiguity problem, researchers have developed various methods according to different principles. On the one hand, we can borrow supplementary information such as gene model, protein interaction network and peptide detectability to distinguish truly present proteins from false positives even they share the common set of peptides. On the other hand, we may use
sophisticated statistical models built on the peptide–protein relationship network to perform inference. In our opinion, the identification ambiguity problem is even much harder than the identification coverage problem since some protein sequences are almost identical. To solve the protein identification ambiguity problem, some potentially promising strategies are listed below.

- Since different protein inference approaches use different algorithms to solve the identification ambiguity problem from different viewpoints. As a result, the protein inference results from the same set of identified peptides may vary significantly. Therefore, a natural idea is to combine the identification results from multiple inference algorithms into a consensus protein list.

Researchers have developed various validation measures to evaluate the correctness of identification proteins. Most of them are based on empirical database-dependent estimates of error rates using the target–decoy search strategy (e.g. MAYU). However, the reliance on decoy database deserves certain drawbacks as we have discussed in the performance comparison section. Some analytically derived and database-independent error rate estimation methods are also available. Unfortunately, there is still no consensus on the best measure so far, urging on us the need for developing new validation methods in future research.

Overall, the issue of validating protein identification results is only partially solved. We have the following suggestions regarding the standard validation protocol:

- Using different data. To obtain an unbiased and comprehensive performance evaluation, it is necessary to include test data of different kinds of characteristics: standard synthetic mixture of known proteins, simulation data, practical real data.
- Using multiple validation measures if possible. Inevitably, each validation method has some bias. Using multiple evaluation measures in the test may provide more convincing comparison results.

**CONCLUSIONS**

This review categorizes protein inference using two general frameworks. Protein inference is very important, from both theoretical perspective and practical aspect. More research efforts should be devoted to this challenging topic. We hope this article can provide a starting point for those who are interested in this area. In addition to the references cited, readers may find the review articles [88–90] and thesis [91] very useful.

**Key points**

- Protein inference is an important step in proteomics research. More attention should be paid to this problem.
- There are at least two sources that cause the difficulties in solving protein inference problem: degenerate peptides and one-hit wonders.
- We divide the analysis procedure of protein inference into different stages: peptide identification post-processing, peptide assembly and result validation.
- For existing peptide assembly methods, we propose two hierarchical classification frameworks. According to the dependence on the peptide search engine, peptide assembly methods are categorized into two classes: search engine-dependent approaches and search engine-independent approaches. According to the underlying algorithmic techniques, peptide assembly methods are categorized into two classes: bipartite graph model and supplementary information model.

**Acknowledgements**

The comments and suggestions from the anonymous reviewers greatly improved the article. We thank Dr Smriti R. Ramakrishnan for providing us the protein reference set used in the experiments.

**FUNDING**

This work was partially supported by the Natural Science Foundation of China under Grant No. 61003176, the Fundamental Research Funds for the Central Universities of China (DUT10JR05 and DUT10ZD110), and a research competition award (RPC10EG04) from the Hong Kong University of Science and Technology.

**References**


discovery rates for peptide and protein identification using
80. Klammer AA, Park CY, Noble WS. Statistical calibration of
2106–13.
spectrum-specific p-values to protein identifications by
MS analysis platform utilizing open XML file formats. Mol
84. Xue X, Wu S, Wang Z, et al. Protein probabilities in shot-
gun proteomics: evaluating different estimation methods
using a semi-random sampling model. Proteomics 2006;
6(23):6134–45.
analytical techniques for the characterisation of the human
serum proteome in HUPO Plasma Proteome Project. Proteomics
tification confidence by combining search methods. J Proteome Res
87. Claassen M, Aebersold R, Buhmann JM. Proteome cover-
age prediction with infinite Markov models. Bioinformatics
88. Ma B. Challenges in computational analysis of mass spectr-
107–23.
89. Shi J, Wu F-X. Protein inference by assembling peptides
identified from tandem mass spectra. Curr Bioinform 2009;
90. Nesvizhskii AI. A survey of computational methods and
error rate estimation procedures for peptide and protein
identification in shotgun proteomics. J Proteomics 2010;
91. Claassen M. Design and Validation of Proteome Measurements.