Published in final edited form as:

J Proteome Res. 2009 September ; 8(9): 4173–4181. doi:10.1021/pr9004794.

False Discovery Rates of Protein Identifications: A Strike against the Two-Peptide Rule

Nitin Gupta^{*,†} and Pavel A. Pevzner[‡]

[†]Bioinformatics Program, University of California San Diego, La Jolla, California 92093

[‡]Department of Computer Science and Engineering, University of California San Diego, La Jolla, California 92093

Abstract

Most proteomics studies attempt to maximize the number of peptide identifications and subsequently infer proteins containing two or more peptides as reliable protein identifications. In this study, we evaluate the effect of this “two-peptide” rule on protein identifications, using multiple search tools and data sets. Contrary to the intuition, the “two-peptide” rule reduces the number of protein identifications in the target database more significantly than in the decoy database and results in increased false discovery rates, compared to the case when single-hit proteins are not discarded. We therefore recommend that the “two-peptide” rule should be abandoned, and instead, protein identifications should be subject to the estimation of error rates, as is the case with peptide identifications. We further extend the generating function approach (originally proposed for evaluating matches between a peptide and a *single* spectrum) to evaluating matches between a protein and an *entire* spectral data set.

Keywords

two-peptide rule; false discovery rate; mass spectrometry; peptide identification; protein identification; decoy database; false positives

1. Introduction

Tandem mass spectrometry (MS/MS) database search tools are routinely used for peptide identifications.¹ The results often include many false positives,² and a common approach for estimating the False Discovery Rate (FDR) of peptide identifications is based on the use of randomized decoy databases.^{3,4} The reported set of peptide identifications is determined by varying the score cutoff to achieve the desired FDR. Intuitively, a protein must be present in the sample if a peptide within it is identified (assuming the same peptide is not present elsewhere in the proteome). However, since peptide identification tools generate some false peptides, many researchers are cautious about the “one-hit-wonders”,^{5–7} making it a common practice to report only proteins with at least two peptides as reliable identifications (proteins with single peptide identifications are often ignored, or delegated to the Supplementary materials). The “two-peptide” rule has done the field a great service by providing a stringent criterion for reporting proteomic data. However, while the “two

peptide” rule seems well-intentioned and reasonable, we are unaware of any *theoretical* studies supporting this rule. Indeed, the question whether two identified peptides with scores x and y (within the same protein) represent a better evidence for expression of this protein than a single peptide with score z (that is larger than x and y) depends on parameters x , y , z (and the protein length) and remains poorly addressed. Below, we show that the “two-peptide” rule is inferior to the “single-peptide” rule that takes into account one-hit-wonders (with appropriately chosen score threshold z). We therefore argue that the “two-peptide” bias should be removed and that protein identifications based on single peptides should be treated *at par* with identifications based on multiple peptides, instead of salvaging them through postprocessing.⁸

Gupta et al.,⁹ estimated that 80% of the one-hit-wonders (proteins with a single identified peptide) in proteogenomics study of *Shewanella oneidensis* MR-1 are likely to be expressed. Comparative analysis of three *Shewanella* species revealed that many of these one-hit-wonders are actually observed as orthologs in multiple species, providing further support for their expression.¹⁰ These observations indicate that the “two-peptide” rule for protein identification is an unsubstantiated heuristic that often results in loss of a large number of protein identifications (20%–25% of all expressed proteins).

Another unsubstantiated assumption often made by proteomics researchers is that maximizing the number of peptide identifications automatically results in maximizing the number of protein identifications. While peptide and protein identification goals are closely related, there are cases when one is more important than the other. Optimizing protein-level FDR is critical in applications such as biomarker discovery or proteome profiling, while optimizing peptide-level FDR is important for label-free quantitations,¹¹ proteogenomics,^{9,10} or peptidomics.^{12,13} We demonstrate that peptide and protein identifications are two different computational problems that should be approached differently: maximizing the number of peptide identifications does not necessarily result in maximizing the number of protein identifications (for example, in cases when most peptides hit a few proteins).

While FDR among *all* protein identifications in a data set can be computed using the decoy database, computing the false positive rate (FPR) of *individual* protein identifications has been an open problem. The existing tools provide probabilistic scores, which may be correlated with the FPR, but do not provide its rigorous estimate taking into account the lengths of the proteins or the size of the spectral data sets. We extended the generating function framework^{14,15} to suggest a different (and simple) approach to the problem of estimating the protein-level FPR. So far, we failed to find any evidence that the previously proposed techniques for evaluating protein identifications (often based on elaborate machine learning models and multiple peptides) improve over a simple “single-peptide” rule combined with the generating function approach.

2. Methods

MS/MS Data Sets

The MS/MS data sets used in this study were obtained from *S. oneidensis* MR-1 (generated in Dick Smith’s laboratory at PNNL) and human (generated in Vivian Hook’s laboratory at UCSD) samples. These data sets are described in the literature.^{9,10,16} The data sets were generated on Thermo LCQ mass-spectrometer for *Shewanella* and Agilent XCT Ultra for human samples. The human data set includes nearly 600 000 spectra, while the *Shewanella* data set has 14.5 million spectra. The *Shewanella* data set was searched against the *Shewanella* protein database (size $\approx$ 1.5 MB, containing 4928 sequences), and the human data set was searched against the IPI database version 3.41 (size $\approx$ 40 MB, containing 72

155 sequences). Decoy databases were generated by randomly shuffling the sequence of each protein in the target database (preserving the background amino acid frequencies for each protein).

Peptide Identification

Database searches were carried out using InsPecT¹⁷ and X!Tandem.¹⁸ InsPecT searches were run with the default parameter settings (fragment ion tolerance of 0.5 Da and parent mass tolerance of 2.5 Da). X!Tandem was run using default settings, without any protease specificity (allowing peptides of length up to 40). MS-GeneratingFunction (MS-GF)^{14,15} was run on InsPecT results to evaluate the statistical significance of individual peptide identifications (*spectral probabilities*). We treat the spectral probability as a score, and since MS-GF was used to rescore all InsPecT identifications (including even the very low scoring ones that are typically never reported), InsPecT $\oplus$ MS-GF can be viewed as the third peptide identification tool, besides InsPecT and X!Tandem. MS-GF was recently shown to improve upon InsPecT, X!Tandem, and Sequest/PeptideProphet.^{14,19}

Protein Identification

Protein identifications are inferred by applying score-thresholds to the peptides identified in a protein. In case of the “one-peptide” rule (i.e., allowing one-hit-wonders), any protein that has a peptide scoring above the chosen threshold is considered identified. This is equivalent to using a protein-level scoring scheme where the score of a protein is computed as the score of its highest scoring peptide. Similarly, in case of “two-peptide” rule, any protein that has two or more peptides scoring above the threshold is considered identified. Each point on the ROC curves is generated by changing this threshold, and computing the number of protein identifications by each rule, in the target and the decoy databases.

ProteinProphet was run using the Trans Proteomics Pipeline, v4.0 JETSTREAM rev 2, to process X!Tandem search results on human data set. Both target and decoy protein sequences were included in the X!Tandem search results (without giving this information to ProteinProphet a priori). The final output thus included both target and decoy proteins with a probability-based score for each protein. The ROC curve was computed by varying the value of this score-threshold between 0 and 1. ProteinProphet performs better than the traditional “two-peptide” rule (as expected), but still worse than the one-peptide rule.

3. Results

Comparison of Two-Peptide and Single-Peptide Rule

The intuition behind the “two-peptide” rule is that it may result in a more severe penalty to decoy hits as compared to the target hits. In other words, removing one-hit-wonders should improve the FDR of peptide and protein identifications. Figure 1 shows this trend for peptide identifications in the *Shewanella* data set for InsPecT and MS-GF scoring functions. Indeed, if we discard all peptides representing one-hit-wonders, we observe that the tradeoff between the number of *peptides* identified in the target and the decoy database for different score thresholds shifts in favor of the target hits, compared to the situation when one-hit-wonders are retained.

Common sense suggests that the increased number of peptides, for a given FDR, should also increase the number of protein identifications. Therefore, it seems plausible that the “two-peptide” rule (discarding one-hit-wonders) should perform better than the “single-peptide” rule (retaining all proteins with one or more peptides). Note that, in common practice, the “two-peptide” rule is always used in the context of protein identifications, that is after the peptides have been identified for a chosen FDR at peptide level.

Figure 2 demonstrates that this intuitive conclusion is not substantiated by the data as the simpler “single-peptide” rule has superior FDR as compared to the traditional “two-peptide” approach applied to protein identifications. For example, for the same number (21) of proteins identified in the decoy database, InsPecT¹⁷ identifies 546 proteins in the target human database using the standard “two-peptide” approach and 742 proteins using the “single-peptide” approach, a 36% increase in the number of protein identifications. Similarly, the number of protein identifications with X!Tandem¹⁸ increases from 350 to 414 (Figure 3), while the number of protein identifications with MS-GF¹⁴ increases from 607 to 826. Similar trends are seen for the *Shewanella* data sets (Figure 2b,c).

Besides the “two-peptide” rule, more complex approaches are sometimes used to combine evidence from multiple peptides into protein identifications.^{20–24} We compare our results with ProteinProphet,²¹ a popular tool that can be used to postprocess MS/MS database search results of programs like Sequest and X!Tandem. Benchmarking ProteinProphet (combined with X!Tandem) against the “single-peptide” approach (Figure 3) shows that the “single-peptide” rule has better sensitivity–specificity tradeoff than ProteinProphet, in both human and *Shewanella* data sets. This surprising result indicates that the simple “single-peptide” rule should not be discarded without benchmarking it against seemingly more reasonable (“two-peptide” rule) and complex (ProteinProphet) approaches.

We note that computing FDR using decoy databases (at the protein level) has some limitations because the number of protein identifications in the decoy database may deviate from the number of incorrect protein identifications in the target database, particularly in data sets with high proteome coverage. However, the relative comparison of the “single-peptide” and the “two-peptide” approaches should not be significantly affected by this phenomenon. The next section describes an approach for computing the protein-level error-rates without using a decoy database.

Inference of proteins from identified peptides is complicated by the presence of peptides that are present in multiple proteins. This problem is more pronounced in eukaryotes (since their proteomes have many repeated peptides) as compared to bacteria that have very few repeated peptides. The analysis of *Shewanella* in Gupta et al.⁹ revealed that 98.5% of the identified peptides were unique, but the number can drop down to 40–50% in case of higher eukaryotes when using a sequence database like the human IPI database containing alternatively spliced variants of proteins. A number of approaches have been developed to address this problem, most of which are parsimony-based.^{21,25–27} The computation of FDR in our method can depend on the exact method used for protein inference. For the sake of simplicity, we assign the ambiguous peptides (that map to multiple proteins) to one of the matching proteins randomly. To check that our results are robust to this step, we tried limiting our attention to only the unique peptides that are identified by InsPecT in the human data set. We still find that the “single-peptide” approach works best for protein identifications (Figure 4). This shows that our conclusions are not significantly altered by peptides matching multiple proteins.

Estimating Statistical Significance of Protein Identifications Using Spectral Dictionaries

We say that a protein matches a spectrum with score *Score* if one of its peptides matches the spectrum (with the same *Score*). Consider the following two problems:

Peptide–Spectrum Matching Problem—Given a spectrum *Spectrum* and a score threshold *Threshold* for a spectrum–peptide scoring function, find the probability that a random peptide matches *Spectrum* with score equal to or larger than *Threshold*.

Protein–Spectra Matching Problem—Given a spectral data set $\mathcal{S}pectra$ and a score threshold $Threshold$ for a spectrum–peptide scoring function, find the probability that a random protein matches a spectrum in the set $\mathcal{S}pectra$ with score equal to or larger than $Threshold$.

Recently, Kim et al.¹⁴ suggested a generating function approach (MS-GF) for solving the Peptide–Spectrum Matching Problem and thereby evaluating the FPR of Peptide–Spectrum Matches (PSM). In contrast to FDRs empirically computed by the target-decoy approaches (that lack the ability to evaluate the statistical significance of *individual* PSMs), MS-GF rigorously evaluates individual PSMs using *spectral probabilities*. However, Kim et al.¹⁴ defined the spectral probabilities for a *single* peptide and spectrum, while MS/MS searches compare *multiple peptides* (e.g., all peptides present in a protein database) against *multiple* spectra. Therefore, accurate estimation of protein-level FPRs requires a solution to the yet unsolved Protein–Spectra Matching Problem. In this section, we address this problem by extending the generating function framework from *Peptide–Spectrum* matches to *Protein–Spectra* matches.

We start by reviewing the terminology related to spectral probability and spectral dictionaries^{14,15} (see Table 1). Given a scoring function for PSMs, $Dictionary(Spectrum, Threshold)$ is defined as the set of all possible peptides with $Score(Peptide, Spectrum) \geq Threshold$. The spectral dictionary framework transforms the (difficult) problem of evaluating the statistical significance of a PSM with score $Threshold$ into a (simple) problem of evaluating the statistical significance of matches between a set of strings ($Dictionary$) and a random peptide. Assuming probabilities of all amino acids in a random peptide equal to $1/20$, the probability that a $Dictionary$ contains a random peptide (more precisely, a prefix of a sufficiently long random string of amino acids) is given by $SpectralProbability(Dictionary) = \sum_{Peptide \in Dictionary} 20^{-|Peptide|}$, where $|Peptide|$ stands for the length of $Peptide$. The dictionary corresponding to a PSM ($Dictionary(Peptide, Spectrum)$) is defined as $Dictionary(Spectrum, Score(Peptide, Spectrum))$. Similarly, the spectral probability of a PSM is defined as $SpectralProbability(Peptide, Spectrum) = SpectralProbability(Dictionary(Peptide, Spectrum))$. This allows one to convert an arbitrary (additive) scoring function $Score(Peptide, Spectrum)$ into a new scoring function represented by $SpectralProbability(Peptide, Spectrum)$. Kim et al.¹⁴ demonstrated that this new scoring function (varying between 0 and 1) results in a better sensitivity-specificity tradeoff than all other scoring functions they evaluated. Note that, in this new scoring function, the lower scores (spectral probabilities) represent “better” PSMs. As such, given a $Spectrum$, one can define $Dictionary(Spectrum, SPThreshold)$, for a spectral probability threshold $0 \leq SPThreshold \leq 1$, as the set of all peptides with $SpectralProbability(Peptide, Spectrum) \geq SPThreshold$ when $SpectralProbability$ serves as a scoring function. We alert the reader that the term dictionary refers to both dictionaries defined by the original scoring function $Score$ and the new scoring function $SpectralProbability$. We further define $SpectralxProbability(Spectrum, SPThreshold)$ as $SpectralProbability(Dictionary(Spectrum, SPThreshold))$. It is easy to see that the $Spectral Probability$ score (in contrast to the original $Score$) satisfies the important property:

$$SpectralProbability(Spectrum, SPThreshold) = SPThreshold$$

In the remainder of this paper, we assume that $Spectral Probability$ is the scoring function being used.

The probability that a $Dictionary$ contains a peptide from a random protein of length n can be computed as $1 - \bar{P}(Dictionary, n)$, where $\bar{P}(Dictionary, n)$ is the probability that none of

the peptides in the *Dictionary* are contained in a random string of length n . Computing $\bar{P}(\text{Dictionary}, n)$ is a nontrivial problem that was solved by Guibas and Odlyzko²⁸ using the generating function approach. While this approach allows one to compute $\bar{P}(\text{Dictionary}, n)$ precisely, the resulting expressions and recurrences are rather difficult to analyze due to *correlations* between different strings in the *Dictionary* (see ref 28 for details). We therefore prefer to use an approximation (that ignores correlations), a reasonable assumption for a rather large alphabet of 20 amino acids (the effect of correlations is reduced with the increase in the alphabet size²⁸). Under this assumption, $\bar{P}(\text{Dictionary}, n)$ is approximated as $(1 - \text{SpectralProbability}(\text{Dictionary}))^n$.

Therefore, the probability that a *Dictionary* contains a peptide from a random protein *Database* can be approximated as $1 - (1 - \text{SpectralProbability}(\text{Dictionary}))^{|\text{Database}|} \approx \text{SpectralProbability}(\text{Dictionary}) \cdot |\text{Database}|$ (under the condition that $\text{SpectralProbability}(\text{Dictionary}) \cdot |\text{Database}| \ll 1$).¹⁴ This condition is satisfied in practice since otherwise one ends up with spectral identifications that feature an unacceptably high FPR. Since we can represent a PSM by a *Dictionary*(*Peptide*, *Spectrum*), the probability of a Protein–Spectrum Match (with a random protein) can be similarly computed as $\text{SpectralProbability}(\text{Dictionary}) \cdot |\text{Protein}|$. Therefore, if one accepts peptides with spectral probability score $SP_{\text{Threshold}}$ and below ($0 \leq SP_{\text{Threshold}} \leq 1$), then the probability of a match with a random *Protein* (spectral probability of Protein–Spectrum Matches) is approximated as $SP_{\text{Threshold}} \cdot |\text{Protein}|$.

Accounting for Protein Length—The score of a protein identification was previously computed as the score of its best scoring PSM, without taking into account the length of the protein. This is an oversimplification since longer proteins are more likely to have spurious matches than shorter ones. The spectral probability of Protein–Spectrum Matches ($\text{SpectralProbability}(\text{Dictionary}) \cdot |\text{Protein}|$) suggests a natural normalization for computing protein-level FPR (note that while this normalization is applicable to the *SpectralProbability* score, it is not necessarily valid for other scoring functions). Figure 5 compares this Protein–Spectrum score (normalized by the protein length) with the original Peptide–Spectrum score (spectral probability before normalization) and shows that the proposed normalization makes sense (for both human and *Shewanella* data sets). While this normalization does not result in a large change in the sensitivity–specificity tradeoff (since most proteins have similar lengths), the normalized Protein–Spectrum scoring function does show a modest increase in the number of protein identifications in the target database. For example, for 20 proteins identified in the decoy database (human data set), the Peptide–Spectrum scoring identifies 838 proteins in target database, while the Protein–Spectrum scoring (normalized by the protein length) identifies 865 proteins.

Accounting for the Size of Spectral Data Set—We now extend the spectral dictionary approach from ref 15 to evaluate the statistical significance of matches between entire spectral data set *Spectra* and a protein. Like protein length, the size of the spectral data set also affects the statistical significance of a protein identification. While large spectral data sets are more likely to produce spurious identifications, some protein identification tools do not correct for the spectral data set size. When computing the probability of a match below the score threshold $0 \leq SP_{\text{Threshold}} \leq 1$ for a *single* spectrum (we remind the reader that smaller *SpectralProbability* scores correspond to better Peptide–Spectrum matches), one could correct for the size of the *Protein* as $SP_{\text{Threshold}} \cdot |\text{Protein}|$. However, when estimating the number of hits for *multiple* spectra, a similar correction $SP_{\text{Threshold}} \cdot |\mathcal{S} \text{pectra}|$ does not apply because of the correlations between the spectra within a typical spectral data set (e.g., multiple spectra of the same peptide). To address this problem, we introduce the notion of a *spectral lexicon* below. We start by defining the dictionary of a spectral data set as

$$Dictionary(\mathcal{S}pectra, Threshold) = \bigcup_{Spectrum \in \mathcal{S}pectra} Dictionary(Spectrum, Threshold)$$

A string from *Dictionary* is called *redundant* if its proper substring also belongs to *Dictionary* (e.g., PEPTIDE is redundant if PEPTID, or PTIDE, or any of the shorter substrings belong to the dictionary). We define *Lexicon*(*Dictionary*($\mathcal{S}pectra, Threshold$)), represented more succinctly as *Lexicon*($\mathcal{S}pectra, Threshold$), as the set obtained by removal of all redundant strings from *Dictionary*($\mathcal{S}pectra, Threshold$). While *Lexicon*($\mathcal{S}pectra, Threshold$) combines dictionaries of all spectra in the data set, for all practical purposes it can be treated as a dictionary of a single (virtual) spectrum. Moreover, the probability that a *Lexicon* contains a random peptide (i.e., that a spectrum from $\mathcal{S}pectra$ matches a random peptide with a score equal to or better than *Threshold*) is again given by *SpectralProbability*(*Lexicon*). Similarly, the probability that a *Lexicon* contains a peptide from a random *Protein* (i.e., the probability of a Protein- $\mathcal{S}pectra$ Match) can be approximated as $1 - (1 - SpectralProbability(Lexicon))^{|Protein|} \approx 1 - e^{-SpectralProbability(Lexicon) \cdot |Protein|}$. Again, if $SpectralProbability(Lexicon) \cdot |Protein| \ll 1$, this probability can be approximated as $SpectralProbability(Lexicon) \cdot |Protein|$. We remark that while the expression $1 - (1 - SpectralProbability(Lexicon))^{|Protein|}$ is an approximation (the exact formula is given in²⁸), it nevertheless leads to a reasonable estimate of Protein- $\mathcal{S}pectra$ FPR in practice (see next section). While this simplified formula involves multiple levels of approximations, it remains reasonable for small values of spectral probability (such as 10^{-11} or lower), for the typical sizes of spectral data sets (10^6) and protein lengths (10^3). For analyzing larger data sets, we recommend using more stringent (smaller) thresholds for spectral probability.

While *SpectralProbability*(*Lexicon*) evaluates the statistical significance of Protein- $\mathcal{S}pectra$ matches (protein-level FPR), it remains unclear how to compute it. To address this problem, we define the notion of *Compression* of a spectral data set as a way to analyze dependencies between spectra. Loosely speaking, *Compression* is the ratio of the spectral probability of the lexicon (of a spectral data set) to the sum of spectral probabilities of individual dictionaries (of each spectrum). The ratio will often be less than 1 due to removal of redundant peptides from the lexicon and to the fact that many individual dictionaries share the same peptides. More precisely,

$$Compression(\mathcal{S}pectra, SPThreshold) = \frac{SpectralProbability(Lexicon(\mathcal{S}pectra, SPThreshold))}{\sum_{Spectrum \in \mathcal{S}pectra} SpectralProbability(Dictionary(Spectrum, SPThreshold))}$$

Since we use *SpectralProbability* as the scoring function,

$$SpectralProbability(Dictionary(Spectrum, SPThreshold)) = SPThreshold$$

and therefore,

$$Compression(\mathcal{S}pectra, SPThreshold) = \frac{SpectralProbability(Lexicon(\mathcal{S}pectra, SPThreshold))}{|\mathcal{S}pectra| \cdot SPThreshold}$$

Two spectra are called *independent* if their dictionaries do not overlap and the union of their dictionaries does not contain redundant peptides. If all spectra in the spectral data set were independent, then *Compression* = 1 and the quantity $|\mathcal{S}pectra| \cdot SPThreshold$ would provide a

rigorous solution to the problem of the statistical significance of *Peptide-Spectra* Matches. In reality, however, spectra of the same and related peptides (e.g., peptides that represent subpeptides of other peptides) that are typically present in the spectral data set reduce *Compression*. While it can be explicitly computed for small spectral data sets, its efficient evaluation for large spectral data sets remains an open problem.

Spectral Compression and Spectral Clustering—To estimate *Compression* of a spectral data set, we selected all high-accuracy (FT) *Shewanella* spectra with parent mass varying between 1090 and 1100 Da resulting in the data set *Spectra* with 12 617 spectra. For different values of *SPT*threshold, *Dictionary(Spectrum, SPT*threshold) were generated for each spectrum in *Spectra* and combined into *Lexicon(Spectra, SPT*threshold). Since all peptides in these dictionaries have similar parentmass (and thus do not have redundant peptides), *Lexicon* can be generated by simply taking a union of spectral dictionaries of individual spectra. We find that *Compression* values vary between 0.6 and 0.8 depending on the spectral probability threshold (Table 2). This experiment indicates that, while the expression $|\mathcal{Spectra}| \cdot SPT$ threshold overestimates FPR, it is still within ≈ 60 –80% of the correct estimate, a reasonable approximation.

Below we describe an alternative approach to estimating *Compression* via *spectral clustering*. We assume that a spectral sample *Spectra* represents the set of peptides *Peptides*. While the set *Peptides* is unknown, its size $|\mathcal{Peptides}|$ can be estimated by MS-Clustering tool²⁹ as the number of spectral clusters. Under the “ideal” scenario, the spectra of the same peptide (belonging to one cluster) have identical spectral dictionaries, spectra of different peptides do not overlap, and all spectra are independent. Under these assumptions,

$$SpectralProbability(Lexicon(\mathcal{Spectra}, SPTthreshold)) = |\mathcal{Peptides}| \cdot SPTthreshold$$

Therefore,

$$Compression = \frac{|\mathcal{Peptides}| \cdot SPTthreshold}{|\mathcal{Spectra}| \cdot SPTthreshold} = \frac{|\mathcal{Peptides}|}{|\mathcal{Spectra}|}$$

In reality, dictionaries of spectra of the same peptides are not identical, dictionaries of spectra of different peptides may overlap, and spectra may be dependent. As a result, the parameter $|\mathcal{Peptides}|/|\mathcal{Spectra}|$ typically underestimates *Compression*. For example, for the data set *Spectra* consisting of spectra with parent mass between 1090 and 1100 Da, MSclustering²⁹ found 3584 clusters from 8689 spectra, as shown in Table 3 (3928 out of 12 617 spectra were discarded as low-quality spectra). The estimated *Compression* = 3584/8689 ≈ 0.41 is lower than the values observed in Table 2. Table 4 shows that large clusters have *Compression* much larger than $1/Cluster\ Size$ and illustrates that clustering underestimates *Compression*.

While computing the FPR of Protein-*Spectra* matches enables one to evaluate the statistical significance of *individual* protein identifications, it also allows one to estimate the expected number of false identifications among all identifications, without requiring a decoy database. The following computational experiment shows that it is feasible to estimate the number of protein identifications in a decoy database without actually using a decoy database. The human spectra were searched with InsPecT $\oplus$ MS-GF against a database of 10 000 randomly generated proteins of length 500 each, with equal probability of each amino acid at each position. A total of 286 717 spectra (*Spectra*) got some InsPecT/MS-GF score against this database and were included in the analysis. Table 5 shows the *observed* number of proteins

that have any peptide exceeding the score threshold ($SPThreshold$). The *expected* number of protein identifications, for different values of $SPThreshold$, is given by

$$10000 \cdot (1 - e^{-SpectralProbability(Lexicon) \cdot |Protein|})$$

where $SpectralProbability(Lexicon)$ is estimated as $Compression \cdot |\mathcal{S}pectra| \cdot SPThreshold$. This requires an estimate for the value of $Compression$ for the spectral data set that we describe below.

Suppose that the spectral data set $\mathcal{S}pectra$ is partitioned into nonoverlapping $\mathcal{C}lusters$ (i.e., $\mathcal{S}pectra = \bigcup_{Cluster \in \mathcal{C}lusters} Cluster$), where each $Cluster$ represents the set of $|Cluster|$ spectra originating from the same peptide. If we assume that $Compression$ within each $\mathcal{S}pectra$ is $1/|Cluster|$, then

$$SpectralProbability(Lexicon(\mathcal{S}pectra, SPThreshold)) = |\mathcal{C}lusters| \cdot SPThreshold$$

However, Table 4 reveals the limitations of this estimate. A more accurate estimate is given by

$$SpectralProbability(Lexicon(\mathcal{S}pectra, SPThreshold)) \approx \sum_{Cluster \in \mathcal{C}lusters} SpectralProbability(Lexicon(Cluster, SPThreshold)) \approx \sum_{Cluster \in \mathcal{C}lusters} |Cluster| \cdot Compression(Cluster, SPThreshold) \cdot SPThreshold$$

Therefore,

$$Compression(\mathcal{S}pectra, SPThreshold) \approx \frac{\sum_{Cluster \in \mathcal{C}lusters} |Cluster| \cdot Compression(Cluster, SPThreshold)}{|\mathcal{S}pectra|}$$

While the parameter $Compression(Cluster, SPThreshold)$ can be explicitly computed for each cluster, Table 3 shows that this parameter depends largely on the cluster size (and $SPThreshold$). Therefore, we can approximate $Compression(Cluster, SPThreshold)$ by using the average value over the previously analyzed clusters of the same size. Using the cluster partitioning of the human data set determined by MS-Clustering, the expected number of proteins was estimated using the above formula by plugging in average $Compression(Cluster, SPThreshold)$ values from Table 3. Table 5 compares this expected number of protein identifications with the number of proteins actually identified in the decoy database search and demonstrates that this approach allows one to get a reasonably close estimate of the number of false protein identifications without searching a decoy database.

Using FPR of Protein Identifications—This paper recommends discarding the commonly used “two-peptide” rule and instead supports reporting protein identifications (including one-hit-wonders) according to their rigorous statistical significance (FPR). To illustrate this, Supplementary Table 1 provides a ranked list of protein identifications in the human data set, sorted by increasing FPRs. Denote the FPR of the i th ranked protein as $FPR(i)$. Below we estimate the number of protein identifications in a decoy database (and thus FDR of protein identifications) if only proteins with FPRs equal to or exceeding $FPR(i)$ are reported.

If N is the total number of proteins in the sequence database ($N = 72\,155$ for the human IPI database used here), one can estimate the expected number of false positives among top i protein identifications as $\text{False}(i) = N \cdot \text{FPR}(i)$, and therefore, the estimated FDR at that stringency level is $\text{FDR}(i) = (\text{no. proteins in decoy database})/(\text{no. proteins in target database}) = (N \cdot \text{FPR}(i))/i$. As i is increased from 1 to ≈ 600 in this data set, $\text{False}(i)$ increases from 0 to 0.2 only, indicating that these top 600 protein identifications are almost error-free. It is worth noticing that 160 of these extremely reliable protein identifications are one-hit-wonders, a large fraction that may be missed by the traditional approaches favoring multiple peptides. When i is increased from 600 to 800, $\text{False}(i)$ increases to 21, indicating that $1/10$ ($\approx 21/200$) of the protein identifications in this range may be incorrect. Increasing i beyond 800 rapidly increases $\text{False}(i)$, eventually at a rate higher than the rate of increase of i (see the last few rows in the table), thereby essentially reducing the number of true identifications. Therefore, one can choose to select the top ≈ 800 proteins in this list (at an FDR of $21/800 \approx 2.5\%$) as a reasonable set of protein identifications. This analysis shows how the knowledge of protein-level FPRs allows one to make informed judgment calls in selecting reliable protein identifications from many hits obtained in database searches.

4. Discussion

We have demonstrated that the commonly used “two-peptide” rule jeopardizes the sensitivity–specificity tradeoff in protein identifications. This counterintuitive observation points out that we are more likely to get a protein with two mediocre peptide hits in the decoy database (by chance) than a single high-scoring peptide hit. Some software tools (e.g., Protein-Prophet,²¹ Panoramics)²² identify proteins if the combined score from all peptides exceeds the threshold, and thus allow scoring some single-hit proteins higher than proteins with multiple hits. However, these approaches are dependent on specific scoring models and require further theoretical justification to be suitable for use with all protein identification tools. In particular, our results indicate that the simpler “single-peptide” rule results in better sensitivity/specificity tradeoff than ProteinProphet.

One might expect that, since proteins are inferred from peptides, optimizing peptide identifications also optimizes protein identifications. We have provided evidence that this intuition does not hold ground. In reality, when one attempts to maximize the number of peptides in the target database, the additional peptides often come from the already covered proteins (i.e., from proteins with more than 1 match) and thus do not increase the overall number of protein identifications. However, the corresponding increase in the number of peptides in the decoy database significantly raises the number of decoy protein identifications (and thus increases FDR). Therefore, we emphasize that optimizing FDRs for peptides and proteins must be considered as different problems that are best addressed by different approaches.

This study does not recommend accepting *all* proteins with single peptide hits but instead argues that the “single-peptide” approach must be used in conjunction with control of the FDR. While it may be surprising that the “single-peptide” approach generates a larger set of identified proteins than the seemingly more reliable “two-peptide” approach (without sacrificing FDR), our results indicate that it is the case. We demonstrated that for any set of proteins identified by the “two-peptide” approach (with peptide score threshold x), there is a larger set of protein identified by the “single-peptide” approach with the same FDR (with a more stringent peptide score threshold $x + \epsilon$). Therefore, one has to choose the peptide-level score thresholds carefully to ensure that the “single-peptide” approach and “two-peptide” approach are being used for the same level of FDR.

We also discussed how to estimate the FPR of individual Protein–Spectra matches using the generating function framework. Spectral clustering appears as a promising, albeit an indirect, approach for approximating spectral compression and should be explored further.

While the proteomics community often takes great care in evaluating peptide-level error rates, the protein-level FDRs and FPRs are rarely computed. We suggest that publications reporting protein identifications should also report protein-level FPRs and/or FDRs. Lack 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.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

This work was supported by NIH 5R01RR016522-05 and 1-P41-RR024851-01 grants. We are indebted to Yong Li, Nuno Bandeira, and Sangtae Kim for many helpful discussions. We are grateful to Sangtae Kim for helping with MS-Dictionary, Ari Frank for helping with MS-Clustering, Seungjin Na and Jongwha Joo for help in running ProteinProphet, and Richard D. Smith, Vivian Hook, and Steven Bark for making MS/MS data sets generated in their laboratories available.

References

1. Aebersold R, Mann M. Mass spectrometry-based proteomics. *Nature*. 2003; 422:198–207. [PubMed: 12634793]
2. Cargile BJ, Bundy JL, Stephenson JL Jr. Potential for false positive identifications from large databases through tandem mass spectrometry. *J. Proteome Res.* 2004; 3:1082–1085. [PubMed: 15473699]
3. Elias JE, Gygi SP. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nat. Methods*. 2007; 4:207–214. [PubMed: 17327847]
4. Kall L, Storey JD, MacCoss MJ, Noble SW. Assigning significance to peptides identified by tandem mass spectrometry using decoy databases. *J. Proteome Res.* 2008; 7:29–34. [PubMed: 18067246]
5. Omenn GS, States DJ, Adamski M, Blackwell TW, Menon R, Hermjakob H, Apweiler R, Haab BB, Simpson RJ, Eddes JS. Overview of the HUPO Plasma Proteome Project: results from the pilot phase with 35 collaborating laboratories and multiple analytical groups, generating a core dataset of 3020 proteins and a publicly-available database. *Proteomics*. 2005; 5:3226–3245. [PubMed: 16104056]
6. Carr S, Aebersold R, Baldwin M, Burlingame A, Clauser K, Nesvizhskii A. The Need for Guidelines in Publication of Peptide and Protein Identification Data: Working Group on Publication Guidelines for Peptide and Protein Identification Data. *Mol. Cell. Proteomics*. 2004; 3:531–533. [PubMed: 15075378]
7. Bradshaw RA, Burlingame AL, Carr S, Aebersold R. Reporting protein identification data: the next generation of guidelines. *Mol. Cell. Proteomics*. 2006; 5:787–788. [PubMed: 16670253]
8. Higdon R, Kolker E. A predictive model for identifying proteins by a single peptide match. *Bioinformatics*. 2007; 23:277–280. [PubMed: 17121779]
9. Gupta N, Tanner S, Jaitly N, Adkins J, Lipton M, Edwards R, Romine M, Osterman A, Bafna V, Smith RD, Pevzner PA. Whole proteome analysis of post-translational modifications: applications of mass-spectrometry for proteogenomic annotation. *Genome Res.* 2007; 17:1362–1377. [PubMed: 17690205]
10. Gupta N, Benhamida J, Bhargava V, Goodman D, Kain E, Nguyen N, Ollikainen N, Rodriguez J, Wang J, Lipton MS, Romine M, Bafna V, Smith RD, Pevzner PA. Comparative proteogenomics: combining mass spectrometry and comparative genomics to analyze multiple genomes. *Genome Res.* 2008; 18:1133–1142. [PubMed: 18426904]

11. Lu P, Vogel C, Wang R, Yao X, Marcotte EM. Absolute protein expression profiling estimates the relative contributions of transcriptional and translational regulation. *Nat. Biotechnol.* 2007; 25:117–124. [PubMed: 17187058]
12. Boonen K, Landuyt B, Baggerman G, Husson SJ, Huybrechts J, Schoofs L. Peptidomics: the integrated approach of MS, hyphenated techniques and bioinformatics for neuropeptide analysis. *J. Sep. Sci.* 2008; 31:427–445. [PubMed: 18266258]
13. Falth M, Skold K, Norrman M, Svensson M, Fenyo D, Andren PE. SwePep, a database designed for endogenous peptides and mass spectrometry. *Mol. Cell. Proteomics.* 2006; 5:998–1005. [PubMed: 16501280]
14. Kim S, Gupta N, Pevzner PA. Spectral probabilities and generating functions of tandem mass spectra: a strike against decoy databases. *J. Proteome Res.* 2008; 7:3354–3363. [PubMed: 18597511]
15. Kim S, Gupta N, Banderia N, Pevzner PA. Spectral dictionaries: Integrating de novo peptide sequencing with database search of tandem mass spectra. *Mol. Cell. Proteomics.* 2009; 8:53–69. [PubMed: 18703573]
16. Gupta N, Bark SJ, Lu WD, Taupenot L, O'Connor DT, Pevzner PA, Hook V. Evaluation of alternative neuropeptide processing in human and bovine dense-core secretory granules by mass spectrometry-based neuropeptidomics. 2009 Manuscript in preparation.
17. Tanner S, Shu H, Frank A, Wang LC, Zandi E, Mumby M, Pevzner PA, Bafna V. InsPecT: identification of posttranslationally modified peptides from tandem mass spectra. *Anal. Chem.* 2005; 77:4626–4639. [PubMed: 16013882]
18. Craig R, Beavis RC. TANDEM: matching proteins with tandem mass-spectra. *Bioinformatics.* 2004; 20:1466–1467. [PubMed: 14976030]
19. Keller A, Nesvizhskii AI, Kolker E, Aebersold R. Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Anal. Chem.* 2002; 74:5383–5392. [PubMed: 12403597]
20. Kall L, Canterbury JD, Weston J, Noble WS, MacCoss MJ. Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nat. Methods.* 2007; 4:923–926. [PubMed: 17952086]
21. Nesvizhskii AI, Keller A, Kolker E, Aebersold R. A statistical model for identifying proteins by tandem mass spectrometry. *Anal. Chem.* 2003; 75:4646–4658. [PubMed: 14632076]
22. Feng J, Naiman DQ, Cooper B. Probability model for assessing proteins assembled from peptide sequences inferred from tandem mass spectrometry data. *Anal. Chem.* 2007; 79:3901–3911. [PubMed: 17441689]
23. Moore RE, Young MK, Lee TD. Qscore: an algorithm for evaluating SEQUEST database search results. *J. Am. Soc. Mass Spectrom.* 2002; 13:378–386. [PubMed: 11951976]
24. Weatherly DB, Atwood JA, Minning TA, Cavola C, Tarleton RL, Orlando R. A Heuristic Method for Assigning a False-discovery Rate for Protein Identifications from Mascot Database Search Results. *Mol. Cell. Proteomics.* 2005; 4:762–772. [PubMed: 15703444]
25. Nesvizhskii AI, Aebersold R. Interpretation of shotgun proteomic data: the protein inference problem. *Mol. Cell. Proteomics.* 2005; 4:1419–1440. [PubMed: 16009968]
26. Alves, P.; Arnold, RJ.; Novotny, MV.; Radivojac, P.; Reilly, JP.; Tang, H. Advancement in protein inference from shotgun proteomics using peptide detectability; *Pac. Symp. Biocomput.* 2007. p. 409–420.
27. Li, YF.; Arnold, RJ.; Li, Y.; Radivojac, P.; Sheng, Q.; Tang, H. Research in Computational Molecular Biology. Berlin: Springer; 2008. A Bayesian approach to protein inference problem in shotgun proteomics; p. 167–180.
28. Guibas LJ, Odlyzko AM. String overlaps, pattern matching, and nontransitive games. *J. Comb. Theory, Ser. A.* 1981; 30:183–208.
29. Frank AM, Bandeira N, Shen Z, Tanner S, Briggs SP, Smith RD, Pevzner PA. Clustering millions of tandem mass spectra. *J. Proteome Res.* 2008; 7:113–122. [PubMed: 18067247]

Figure 1. Identification of peptides in the *Shewanella* data set using different approaches and scoring functions. Each point in the curves is generated by varying the scoring threshold and computing the number of hits in the target and the decoy database exceeding the threshold.

Figure 2.

(a) Identification of proteins in the human data set using different approaches and scoring functions. (b) Similar plot as in panel (a) for *Shewanella* data set.

Figure 3.

(a) Protein identification in the human data set using X!Tandem search results with different scoring approaches at the protein level. (b) Similar plot as in panel (a) for an arbitrarily selected subset of *Shewanella* data set containing 1.25 million spectra.

Figure 4. Identification of proteins, using the unique peptides only (peptides that are not shared between multiple proteins), in the human data set using InsPecT search results with different approaches.

Figure 5.

(a) Identification of proteins in the human data set using MS-GF scores, without and with length correction. (b) Similar plot as in panel (a) for *Shewanella* data set.

Table 1An Intuitive Description of Some Frequently-Used Terms^a

Term	Description
$\text{Dictionary}(\text{Spectrum}, \text{Threshold})$	A set of peptides that match the <i>Spectrum</i> with scores greater than or equal to the <i>Threshold</i>
$\text{SpectralProbability}(\text{Dictionary})$	The probability that a random peptide matches a sequence contained in the <i>Dictionary</i>
$\text{Lexicon}(\mathcal{S} \text{ ectra}, \text{Threshold})$	Union of the dictionaries of all spectra in the set $\mathcal{S} \text{ ectra}$ (for the given <i>Threshold</i>), after removing redundant peptides
$\text{Compression}(\mathcal{S} \text{ ectra}, \text{Threshold})$	Ratio of the spectral probability of the <i>Lexicon</i> to the sum of the spectral probabilities of each spectral dictionary $\text{Dictionary}(\text{Spectrum}, \text{Threshold})$ for all spectra from the set $\mathcal{S} \text{ ectra}$

^aPlease refer to the text for details.

Table 2

Compression for Different Values of Spectral Probability Threshold (*SPTThreshold*) on a Small Spectral Data Set (*Spectra*) Consisting of 12 617 Spectra, with Parent Mass between 1090 and 1100 Da^a

<i>SPTThreshold</i>	$\sum_{S \in \mathcal{S}_{\text{spectra}}} \text{Dictionary}(S, \text{SPTThreshold}) $	$ \text{Lexicon}(\mathcal{S}_{\text{spectra}}, \text{SPTThreshold}) $	<i>Compression</i>
1×10^{-12}	444027	314347	0.752
1×10^{-11}	2709887	1708317	0.737
1×10^{-10}	16919910	9782247	0.689

^aThe second column represents the sum of the sizes of the dictionaries of individual spectra and the third column represents the size of the combined Lexicon.

Table 3

Summary of Spectral Clusters Produced by MS-Clustering on the Set of 12 617 Spectra with Parent Mass between 1090 and 1100 Da.^a

cluster-size	no. clusters	no. spectra	average Compression			
			1 × 10 ⁻¹⁰	1 × 10 ⁻¹¹	1 × 10 ⁻¹²	1 × 10 ⁻¹²
1	2875	2875	1	1	1	1
2	234	468	0.96	0.97	0.98	0.98
3	115	345	0.88	0.90	0.92	0.92
4	71	284	0.87	0.88	0.90	0.90
5	47	235	0.83	0.93	0.95	0.95
6	36	216	0.80	0.86	0.88	0.88
7	24	168	0.76	0.81	0.85	0.85
8	20	160	0.75	0.84	0.95	0.95
9	17	153	0.66	0.73	0.87	0.87
10	17	170	0.68	0.80	0.84	0.84
11-15	40	506	0.71	0.81	0.83	0.83
16-20	27	475	0.60	0.65	0.75	0.75
21-30	23	584	0.47	0.55	0.65	0.65
31-40	11	380	0.48	0.65	0.77	0.77
41-50	8	370	0.28	0.41	0.50	0.50
>50	19	1300	0.23	0.47	0.45	0.45
Total	3584	8689				

^a A total of 8689 spectra were clustered into 3584 clusters, while the remaining 3928 spectra were discarded by MS-Clustering due to low spectral quality. The last three columns indicate the average values of *Compression* for various cluster-sizes, for three different values of spectral probability threshold (*SPThreshold*).

Table 4

Compression Computed for the 10 Largest Clusters in Table 3, Using Dictionaries Generated with the Spectral Probability Threshold of 1×10^{-10}

rank (by size)	size	Compression
1	125	0.246
2	82	0.211
3	77	0.128
4	75	0.126
5	74	0.080
6	72	0.495
7	72	0.066
8	71	0.235
9	70	0.237
10	66	0.361

Table 5

Comparison between the Observed and the Expected Number of Protein Identifications for Different Values of Spectral Probability Threshold (*SPThreshold*), When Searching the Human Spectral Data Set against a Randomly Generated Decoy Database Consisting of 10 000 Proteins of Size 500 Each^a

<i>SPThreshold</i>	<i>Compression</i>	expected no. IDs	observed no. IDs
1×10^{-10}	0.88	125.36	80
1×10^{-11}	0.91	13.04	14
1×10^{-12}	0.92	1.32	1

^aThe values of *Compression* were estimated using the average values for different cluster-sizes from Table 3.