GSI: A New Approach to the Protein Inference Problem

Aurélien Berthier
Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000 Nantes, France

Émile Benoist
Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000 Nantes, France

Guillaume Fertin
Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000 Nantes, France

Géraldine Jean
Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000 Nantes, France

Abstract

The protein inference problem, i.e., determining which proteins are present in a biological sample, is key to understanding the roles of proteins and, more broadly, many biological processes. Protein identification is typically achieved by first cleaving proteins into smaller sequences called peptides. Peptides are then identified using tandem mass spectrometry, a process that produces mass spectra, and in which peptide identification consists of associating, via dedicated tools, a mass spectrum to a peptide sequence. Protein inference consists of identifying, from a list of identified peptides, the proteins that most likely produced them, and were therefore present in the original sample.

Usually, peptide identification and protein inference are two separate steps, which are sequentially achieved. However, by proceeding in such a way, a significant amount of potentially useful information contained in the spectra may be discarded in the second step. Moreover, AI-based tools can now predict the likelihood of a peptide's identification when its parent protein is present in the sample.

In this paper, we present the Global Spectrum Interpretation (GSI) model, a protein inference model that integrates all this information to produce more accurate protein identifications. We show that GSI is NP-hard and provide a Mixed Integer Linear Program (MILP) formulation for it. This MILP is then benchmarked against state-of-the-art protein inference models on several datasets. Our results show that GSI's promising and original approach achieves performance comparable to current models and outperforms other widely used ones, while being more explainable.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms; Applied computing → Computational proteomics

Keywords and phrases Tandem Mass Spectrometry, Peptide identification, Protein inference, Optimization, Algorithmic complexity, Mixed Integer Linear Programming

Digital Object Identifier 10.4230/LIPIcs.WABI.2026.3

Supplementary Material *Software (Source Code)*: `https://github.com/BenoistEmile/GSI`

Funding A. Berthier, G. Fertin, and G. Jean are supported by the French National Research Agency (ANR-24-CE45-3296), ANR PeptidOMS.

1 Introduction

Proteomics is the branch of biology that aims at understanding the role of proteins in living beings. An important aspect of proteomics is determining which proteins are present in a given sample. One way to do so is to use tandem mass spectrometry (MS2), in which a sample of previously digested (i.e., cleaved into smaller fragments) proteins is introduced as input, and tandem mass spectra are acquired as output. Each spectrum can be viewed as a list of peaks and represents a *peptide*, i.e., a small fragment of a protein sequence. The goal

is to infer which proteins were present in the sample. Usually, this is done sequentially in two steps: *peptide identification*, which assigns spectra to specific peptide sequences, and *protein inference*, which relies on the identified peptides to determine which proteins were initially present in the sample [20].

Peptide identification

The goal of peptide identification is to associate an amino acid sequence with each mass spectrum obtained during the wet-lab phase of the proteomics pipeline. Such an association is called a *peptide spectrum match* (PSM). Identifying peptides also means computing a confidence score for each PSM, which will subsequently enable statistical validation of the identifications [20].

Different approaches to peptide identification have emerged during the last few decades. *Database search* methods (e.g., X!Tandem [6], Comet [5], SpecPeptidOMS [2]) rely on proteomics databases (such as Uniprot [24]) from which candidate protein sequences are extracted. These sequences are then submitted to a step called *in silico digestion*, during which they are cleaved to generate candidate peptides. For each candidate peptide, a theoretical spectrum will then be generated and compared to the experimental spectra to identify the most likely peptide of origin for each experimental spectrum. Similar to database search methods, *spectral libraries* methods (e.g., Skyline [15]) compare experimental spectra to thoroughly vetted mass spectra collected from previous experiments. Finally, *de novo* methods (e.g., PEAKS [13]) attempt to directly reconstruct peptide sequences from experimental spectra [20].

All these methods face the issues of *scoring* (or *ranking*) and *validating* the PSMs. Although it can be done manually, the volume of data generated during MS2 experiments necessitates statistical methods capable of handling such a quantity. The PSM scores reflect the confidence of the identifications, which is usually achieved by measuring the similarity between the reference and experimental spectra. Several scoring systems exist, such as the *Shared Peaks Count* (SPC), which is the number of common peaks between both spectra, or the *hyperscore*, which refines SPC by integrating spectral peak intensity in its calculation. Probabilistic scores also exist, such as the *expectation value* (E-value), which is the probability that a given PSM is obtained by chance, or the *q-value*, which is a statistical estimate of the false discovery rate (FDR, see after).

The statistical validation is usually done using a target-decoy strategy [4], which outputs a false discovery rate (FDR) for each peptide. This strategy consists of adding so-called *decoys* to the search database. The decoys are usually generated by reversing the sequences of the *targets*, which are the peptides originally present in the search database. The FDR is computed with the following formula: $FDR(x) = \frac{N_{decoys}(x)}{N_{tot}(x)}$, where $FDR(x)$ is the FDR at a given score threshold x, $N_{decoys}(x)$ is the number of decoy PSMs whose scores pass the x threshold, and $N_{tot}(x)$ is the total number of PSMs (targets and decoys combined) that pass the x threshold. The rationale is as follows: under the hypotheses that the distribution of randomly positive identifications is the same across targets and decoys, and that all decoys identification are erroneous, the proportion of false positives among the targets at a given score threshold is the same as the proportion of identified decoys at the same threshold. The peptides are considered identified if their FDR is under a given threshold (usually 1% for peptide identification).

Peptide identification is followed by protein inference, during which the presence or absence (and eventually quantity) of candidate proteins is deduced from the set of PSMs obtained during peptide identification. As with PSMs, inferring proteins also means scoring

and validating the protein identifications. The protein inference problem is usually visualized as a bipartite graph that shows which peptides have been identified, and, for each peptide, what its parent proteins are (i.e., the proteins that can produce this peptide after being enzymatically digested). An example of such a graph is shown in the upper part of Figure 1.

Protein inference

The goal of protein inference is then to attribute each PSM to a protein and, from these assignments, to identify which are the proteins whose presence in the sample is proven by the PSMs. An extensive review and classification of protein inference models can be found in [10]. We will focus on several categories that cover the most commonly used protein inference tools. *Bipartite graph models* only use information contained in the bipartite graph, and can be further separated into several categories. *Parsimonious* models (e.g., MaxQuant [25]) try to find the shortest list of proteins that explains the presence of all identified peptides. *Statistical* models (e.g., ProteinProphet [17], EPIFANY [18]) leverage the information contained in the graph to directly compute probabilities of presence for every candidate protein. *Optimistic* models, such as DTASelect [22], mark as present every protein that satisfies a set of conditions (such as a minimal number of identified peptides). Other approaches named supplementary information models integrate additional data compared to bipartite graph models, such as HSM [19] that uses a four-layer model directly relying on spectrum-level information to compute probabilities of presence for peptides and proteins. Finally, the recent development of Graph Neural Networks has been applied to protein inference with interesting results [14]. However, despite the increase in AI use in proteomics, few applications exist in protein inference to date.

As for PSMs, protein identifications need to be statistically validated and, to do so, scored and ranked. The protein inference tools differ in the way they score and rank protein identifications. While most methods directly estimate probabilities of presence for each protein, some do so by relying on all the PSMs associated (in the bipartite graph) with each protein, while others only rely on the one or two best-scoring PSMs. However, doing so propagates the errors made at the PSM-level to the protein identification, which makes these estimates somewhat unreliable [10]. To avoid this pitfall, *entrapment* strategies are used, in which the protein search database is expanded with so-called entrapment protein sequences, known to be absent from the sample (obtained either by shuffling the sequences from the original database, or by using proteins from organisms that are absent from the sample). The whole protein inference pipeline is then performed on the expanded database, and an FDR can be computed in a similar fashion to peptides, using entrapment proteins as decoys [26].

The number of peptides or proteins identified at a given FDR (usually 1% for peptides, and between 1% and 5% for proteins) is a widely accepted metric in the proteomics community for the comparison of peptide identification or protein inference tools.

The connection of the two successive steps (peptide identification followed by protein inference) can raise some issues. Traditionally, for each mass spectrum, only the best PSM is kept in the set transferred to the protein inference tool (and only if it gets through the validation filter). Therefore, the potentially useful information from the spectra contained in the discarded PSMs is lost. Moreover, two close peptide sequences can obtain very close scores, meaning the identification tool may not be very confident about which sequence is the correct one. However, this information would be transmitted in a very binary way to the protein inference tool, which, again, is not desirable, as shown in Figure 1. Indeed, in some cases, changing a spectrum's assignment to another peptide can lead to radically different results.

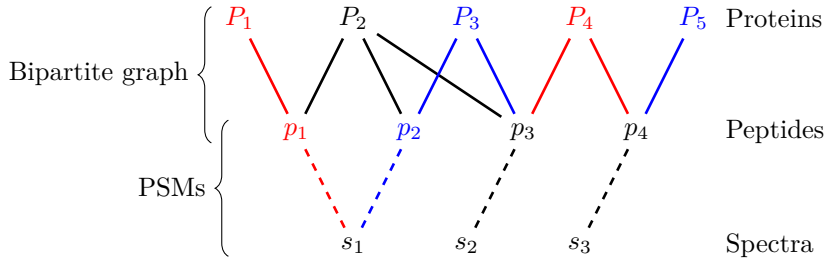

Figure 1 Example of a (fictitious) bipartite proteins-peptides graph, with five proteins from which four peptides are digested and identified. The lower part of the figure, below the bipartite graph, depicts the spectra from which the peptides have been identified (and which peptide was identified from which spectrum). Two protein inference results obtained with a parsimonious approach (that aims at covering every identified peptide, while covering as few unidentified peptides as possible, with a minimum number of proteins) are shown, depending on the assignment of spectrum s_1 to peptide p_1 (in red) or p_2 (in blue). In both cases, spectrum s_2 (resp. s_3) is assigned to peptide p_3 (resp. p_4). The colored protein-peptide edges represent the assignment of each peptide to a protein in the two scenarios. Changing the assignment of a single spectrum from one peptide to another one drastically changes which proteins are inferred (P_1 and P_4 in one case, P_3 and P_5 in the other).

Integrating supplementary information while keeping the useful structure of bipartite graphs could also improve the problem's representation and help disambiguate identifications of close protein sequences. Weighting the graph's edges with *peptide detectability*, a probability (between 0 and 1) depicting how likely the identification of a given peptide is in the presence of a given protein, could be an interesting approach. Indeed, several AI-based tools are available to predict peptide detectability: DbyDeep [21] relies on bidirectional LSTMs to compute detectabilities directly from peptide sequences, and AP3 [7] uses random forest predictors to compute digestibilities (probability of tryptic cleavage) and detectabilities from 588 physicochemical features of the peptides. The accessibility of such tools makes the integration of detectability in a proteomics tool straightforward. Moreover, predicting peptide detectability and the challenges involved are gaining attention within the community, as shown by a recent review [1].

In this paper, we present a new model, called Global Spectrum Interpretation (or GSI), based on a three-layer (proteins - peptides - spectra) graph, whose goal is to simultaneously consider the two steps that are so far usually considered sequentially in proteomics (peptide identification and protein inference), while keeping useful spectral information further away in the proteomics pipeline, by eventually keeping several PSMs for one spectrum. This model can be seen as a hybrid approach between the parsimonious and optimistic models, while integrating additional information (peptide detectability, i.e., the probability that a given peptide will be identified if a given parent protein is present) to compute a more precise protein distribution. We then show that solving an instance of GSI is NP-hard and provide a Mixed Integer Linear Program (MILP) formulation to solve it. Then, we test the MILP implementation on real data from mass spectrometry experiments to evaluate its performance compared to other protein inference models. Doing so, we show that GSI outperforms other protein inference models on several metrics, thus showing encouraging results. We finally discuss the impact of the model's parameter configuration, and defend its explainability compared to other models.

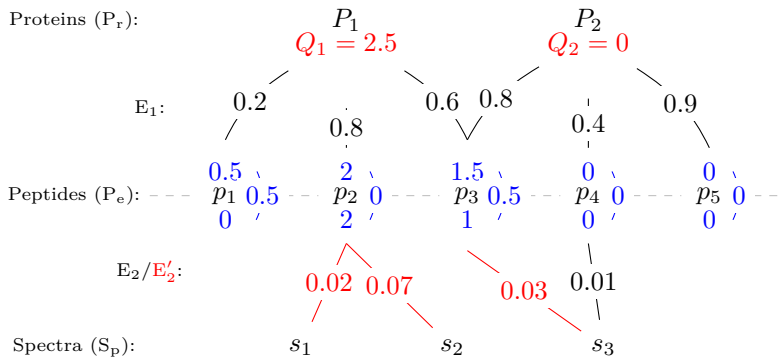

Figure 2 An instance of the GSI problem with two proteins, five peptides, and three spectra. The elements in red present a solution for this instance, which consists of a quantity Q_i per protein and an edge per spectrum (set E'_2). The numbers in blue are the quantities $\overline{q_j}$ (over p_j), $\underline{q_j}$ (under p_j) and $\Delta_j = \overline{q_j} - \underline{q_j}$ (to the right of p_j). For example, $\overline{q_1} = 2.5 \times 0.2 = 0.5$, $\underline{q_1} = 0$, and $\Delta_1 = \overline{q_1} - \underline{q_1} = 0.5$.

2 The integrative Global Spectrum Interpretation model

The GSI model takes as input a graph G, whose three sets of vertices represent proteins (set $\mathrm{P_r}$), peptides (set $\mathrm{P_e}$) and spectra (set $\mathrm{S_p}$) – see Figure 2 for an illustration. G contains two edge sets: the first one, E_1, contains edges connecting each peptide to its parent proteins. These edges are weighted by the detectability of the peptides. The second edge set, E_2, contains edges joining a spectrum to a peptide, weighted by scores that represent PSMs. In the context of proteomics, the edges in E_1 (resp. E_2) and the associated weights can be obtained using prediction tools (resp. peptide identification tools). Thus, an instance of our GSI model consists of a graph $G = ((\mathrm{P_r} \cup \mathrm{P_e} \cup \mathrm{S_p}), (\mathrm{E}_1 \cup \mathrm{E}_2))$, and a weight function w, associating a weight to each edge of the graph.

Let us now describe what a solution for GSI consists of. It can be summarized in two points: (i) choosing for each spectrum which peptide is considered to be the correct one, and (ii) assigning a quantity to the proteins (as it reflects their abundance in the original sample). For (i), we select, for each spectrum, exactly one edge incident to it, and we call this edge set E'_2. For (ii), we assign a quantity Q_i to each protein P_i from $\mathrm{P_r}$. The quality of our solution relies on three elements: (a) first, the discrepancies between the distribution of peptides derived from the inferred protein quantities Q_i and from the assignment of spectra using the selected edges E'_2 should remain low; (b) second, the best PSMs should be selected, and thus the weights of the selected edges in E'_2 should be maximized; (c) third, the list of inferred proteins should be parsimonious, so the number of proteins with non-zero quantity should remain limited.

We present the GSI problem in the form of a minimization problem, whose objective function is a combination of these three goals. For (a), we compute, for every peptide $p_j \in \mathrm{P_e}$ the function Δ_j as follows (see also Figure 2): let us call $\underline{q_j}$ the number of edges incident to p_j in E'_2, and let $\overline{q_j}$ be the sum, over all proteins P_i incident to p_j, of the product $Q_i \cdot w(P_i, p_j)$. We then set $\Delta_j = \overline{q_j} - \underline{q_j}$, and aim at minimizing the sum of the Δ_js under the constraint that each Δ_j must be positive. This constraint is justified by the need to explain, for each peptide identified by a spectrum, why it is present in the solution. We therefore consider that a solution unable to explain the presence of a peptide is incorrect. Concerning (b), we convert the PSM scores in such a manner that the edges in E_2 corresponding to the best PSMs have weights closer to 0. We then aim at minimizing the global weight of the edges

selected in E_2'. Finally, concerning (c), we aim at minimizing the size of the set $P_r' \subseteq P_r$ of proteins with non-zero quantity (i.e., identified proteins). To reduce these three goals into a single-objective optimization problem, we combine them in the form of a weighted sum with 3 strictly positive coefficients Ψ_1, Ψ_2, and Ψ_3 (meant to be user-tunable). We can now formally define the GSI problem.

> Global Spectrum Interpretation (GSI)
> **Instance**: A graph $G = (P_r \cup P_e \cup S_p, E_1 \cup E_2)$, a weight function $w : (E_1 \cup E_2) \to \mathbb{R}^{*+}$, three parameters $\Psi_1, \Psi_2, \Psi_3 \in \mathbb{R}^{*+}$
> **Solution**: A quantity $Q_i \in \mathbb{R}^+$ for every protein $P_i \in P_r$, and a set of edges $E_2' \subseteq E_2$ so that each spectrum of S_p is incident to exactly one edge of E_2' and for every $p_j \in P_e$, $\Delta_j \geq 0$ where $\Delta_j = \overline{q_j} - \underline{q_j}$, $\overline{q_j} = \sum_{(P_i,p_j)\in E_1} w((P_i,p_j)) \cdot Q_i$ and $\underline{q_j} = \big|\{s_k | (p_j, s_k) \in E_2'\}\big|$
> **Measure**: minimize $\Psi_1 \cdot \big(\sum_{p_j \in P_e} \Delta_j\big) + \Psi_2 \cdot \big(\sum_{(p_j,s_k)\in E_2'} w((p_j,s_k))\big) + \Psi_3 \cdot |P_r'|$, where $P_r' = \{P_i \in P_r | Q_i > 0\}$

Our first result is the fact that GSI is NP-hard.

▶ **Theorem 1.** *The GSI problem is NP-hard for any set of positive parameters Ψ_1, Ψ_2 and Ψ_3, even in the following conditions: (i) each vertex from P_r is of degree at most 3, (ii) each vertex from P_e is of degree at most 4, (iii) each vertex from S_p is of degree at most 3 and (iv) all edge weights are equal.*

Proof. The proof is by reduction from Exact Cover by 3-Sets (or X3C), which, starting from a set $\mathcal{U} = \{e_1, e_2 \dots e_m\}$ of elements (with $m = 3q$), takes as input a set $\mathcal{S} = \{S_1, S_2 \dots S_n\}$ of subsets of $\mathcal{U}$ where each S_i is of cardinality 3, and asks whether it is possible to find a subset $\mathcal{S}' \subseteq \mathcal{S}$, of size q, such that each element e_k is present exactly once in $\mathcal{S}'$. X3C is known to be NP-hard, even if every element e_k appears at most 3 times in the S_is [8, 11].

Starting from any instance $I_1 = (\mathcal{U}, \mathcal{S})$ of the X3C problem, we will show how to transform it into an instance $I_2 = (G = (P_r \cup P_e \cup S_p, E_1 \cup E_2), w)$ of the GSI problem. See Figure 3 for an illustration.

For each element e_k of $\mathcal{U}$, we create a vertex s_k in S_p; for each set S_j of $\mathcal{S}$, we create a vertex p_j in P_e that we connect to all vertices s_k for which $e_k \in S_j$. Next, for each vertex p_j constructed in this way, we add to the graph what we call an "extension" (see the blue rectangle in Figure 3 for an example). More precisely, each vertex p_j is connected by an edge to a vertex P_j^1 belonging to P_r. P_j^1 is itself connected to a vertex p_j^1 belonging to P_e. Three vertices s_j^1, s_j^2, and s_j^3 are then added to S_p, all being neighbors of vertex p_j^1. Next, three new vertices p_j^2, p_j^3, and p_j^4 are added to P_e, respectively neighbors of s_j^1, s_j^2, and s_j^3. To complete this extension, the vertices p_j^2, p_j^3, and p_j^4 are connected to a unique vertex P_j^2, belonging to P_r. All the edges in the above-described extension belong to sets E_1 and E_2 according to the vertices they connect. Finally, the weight of every edge is set to 1.

We will now show there is a solution for GSI on instance I_2 with score at most $\Psi_2 \cdot (|\mathcal{U}| + 3 \cdot |\mathcal{S}|) + \Psi_3 \cdot |\mathcal{S}|$, iff I_1 is a YES-Instance for X3C. We begin by recalling that the score of a solution to an instance of GSI is divided in three parts: (i) the sum of the Δ_js, (ii) the sum of the scores of the edges in E_2' and (iii) the number of inferred proteins $|Pr'|$. Concerning score (ii) in instance I_2, it will be the same in any solution, as edge weights are all set to 1, leading to a sum of weights equal to $|E_2'| = |\mathcal{U}| + 3 \cdot |\mathcal{S}|$. Hence, this part of the score is $\Psi_2 \cdot (|\mathcal{U}| + 3 \cdot |\mathcal{S}|)$.

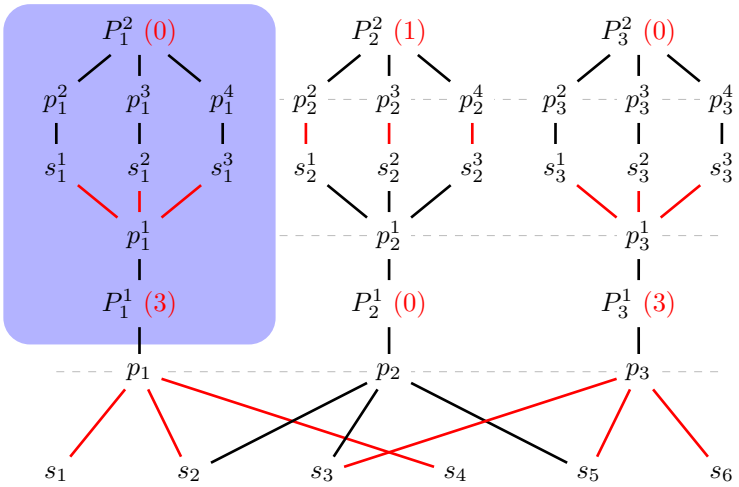

Figure 3 Transforming the following X3C instance into an instance of GSI: $\mathcal{U} = \{e_1, e_2, e_3, e_4, e_5, e_6\}$, $\mathcal{S} = \{S_1, S_2, S_3\}$ with $S_1 = \{e_1, e_2, e_4\}$, $S_2 = \{e_2, e_3, e_5\}$, and $S_3 = \{e_3, e_5, e_6\}$. All edges have weight 1. The elements in red (edges in E_2' and protein quantities) indicate an optimal solution, related to solution $\mathcal{S}' = \{S_1, S_3\}$ of X3C.

Otherwise stated, the scores from parts (i) and (iii) are the ones on which we will rely for our proof.

($\Leftarrow$) Assume that instance I_1 is a YES-Instance. We construct a solution for instance I_2 as follows: for each set $S_j = \{e_{k_1}, e_{k_2}, e_{k_3}\}$ selected in the solution $\mathcal{S}'$ for instance I_1, we select in E_2' the edges going from p_j to the three vertices s_{k_1}, s_{k_2}, and s_{k_3} in $\mathrm{S_p}$. By doing this, the quantities q_j are equal to 3 in the case where S_j belongs to $\mathcal{S}'$, and to 0 otherwise. We then set all quantities Q_j^1 (concerning proteins P_j^1) so that the Δ_js are equal to 0: more precisely, $Q_j^1 = 3$ if $S_j \in \mathcal{S}'$, and 0 otherwise. We then select the edges incident to vertices s_j^1, s_j^2, and s_j^3: if $S_j \in \mathcal{S}'$, then we select in E_2' the edges going from vertices s_j^1, s_j^2, and s_j^3 to vertex p_j^1; otherwise, E_2' will contain the edges going from vertices s_j^1, s_j^2, and s_j^3 to, respectively, vertices p_j^2, p_j^3, and p_j^4. Having done this, we ensure that $\Delta_j^1 = 0$ for all j. Finally, if $S_j \in \mathcal{S}'$, then we set $Q_j^2 = 0$, otherwise we set it to 1. This last step ensures that the Δ_j^2 are all equal to 0 as well. We also note that since $Q_j^1 = 3$ and $Q_j^2 = 0$ when $S_j \in \mathcal{S}'$ and $Q_j^1 = 0$, $Q_j^2 = 3$ otherwise, we have $|\mathrm{P_r}'| = |\mathcal{S}|$. The score due to the number of inferred proteins is therefore equal to $\Psi_3 \cdot |\mathcal{S}|$. Since all Δ_js are equal to 0, we have $\Psi_1 \cdot \left(\sum_{p_j \in \mathrm{Pe}} \Delta_j\right) = 0$, and the global score of the constructed solution is equal to $\Psi_2 \cdot (|\mathcal{U}| + 3 \cdot |\mathcal{S}|) + \Psi_3 \cdot |\mathcal{S}|$.

($\Rightarrow$) We now assume that there exists a solution to instance I_2 of the GSI problem whose score is less than or equal to $\Psi_2 \cdot (|\mathcal{U}| + 3 \cdot |\mathcal{S}|) + \Psi_3 \cdot |\mathcal{S}|$. We will show that instance I_1 of the X3C problem is thus a YES-Instance. First, as already discussed, note that the part of the score due to the selection of edges in $\mathrm{E_2}$ is necessarily equal to $\Psi_2 \cdot (|\mathcal{U}| + 3 \cdot |\mathcal{S}|)$. Furthermore, for each extension of the graph, either the edges going from s_j^1, s_j^2 and s_j^3 to p_1^1, or the edges going from the same vertices to, respectively, p_j^2, p_j^3 and p_j^4 have to be selected. The constraint $\Delta_j \geq 0$ implies that if an edge from E_2 incident to p_1^1 (resp. p_j^2, p_j^3, p_j^4) is selected, then $Q_1^1 \geq 1$ (resp. $Q_1^2 \geq 1$). Therefore, at least one protein from each extension is selected in $\mathrm{P_r}'$, so $|\mathrm{P_r}'| \geq |\mathcal{S}|$. This implies that the score of instance I_2 is at least $\Psi_2 \cdot (|\mathcal{U}| + 3 \cdot |\mathcal{S}|) + \Psi_3 \cdot |\mathcal{S}|$. Since we assumed that it is at most that quantity, then it must be *equal* to $\Psi_2 \cdot (|\mathcal{U}| + 3 \cdot |\mathcal{S}|) + \Psi_3 \cdot |\mathcal{S}|$, which in turn means that all Δ_js are equal to

GSI:

Objective:

$$\textit{minimize} \quad \Psi_1 \cdot \Big(\sum_{p_j \in \mathrm{P_e}} (\overline{q_j} - \underline{q_j})\Big) + \Psi_2 \cdot \Big(\sum_{(p_j,s_k)\in \mathrm{E_2}} w((p_j,s_k)) \cdot x_{(p_j,s_k)}\Big) + \Psi_3 \cdot \Big(\sum_{P_i \in \mathrm{P_r}} X_i\Big)$$

Constraints:

$$\texttt{C.1} \quad \sum_{(p_j,s_k)\in \mathrm{E_2}} x_{(p_j,s_k)} = 1 \qquad \forall s_k \in \mathrm{S_p}$$

$$\texttt{C.2} \quad Q_i \leq M \cdot X_i \qquad \forall P_i \in \mathrm{P_r}$$

$$\texttt{C.3} \quad \underbrace{\sum_{(P_i,p_j)\in \mathrm{E_1}} w((P_i,p_j)) \cdot Q_i - \sum_{(p_j,s_k)\in \mathrm{E_2}} x_{(p_j,s_k)}}_{\overline{q_j} - \underline{q_j} (=\Delta_j)} \geq 0 \qquad \forall p_j \in \mathrm{P_e}$$

Boundaries:

$$\texttt{B.1} \quad Q_i \geq 0 \qquad \forall P_i \in \mathrm{P_r}$$

$$\texttt{B.2} \quad X_i \in \{0,1\} \qquad \forall P_i \in \mathrm{P_r}$$

$$\texttt{B.3} \quad x_{(p_j,s_k)} \in \{0,1\} \qquad \forall (p_j,s_k) \in \mathrm{E_2}$$

Figure 4 Mixed Integer Linear Program for the GSI problem.

zero. If we look at each extension of the graph (blue square in Figure 3), we see that for Δ_j^2 to be zero, the edges selected in the extension in question must necessarily be either the three edges going from vertices s_j^1, s_j^2, and s_j^3 to vertex p_j^1, or the three edges going from vertices s_j^1, s_j^2, and s_j^3 to, respectively, vertices p_j^2, p_j^3, and p_j^4. In the first case, then the quantity Q_j^2 is necessarily 0, and in the second case, the quantity Q_j^2 is necessarily 1. In both cases, the quantity q_j^1 is either equal to 3 or equal to 0. For Δ_j^1 to be equal to 0, it is therefore necessary that $\overline{Q_j^1} = 3$ (when $\underline{q_j^1} = 3$), or 0 (when $\underline{q_j^1} = 0$). Finally, if Q_j^1 is equal to 3 (resp. 0), then the three edges of $\mathrm{E_2}$ incident to p_j are necessarily selected (resp. not selected) in E_2'; otherwise Δ_j would be different from 0. This implies that we can partition the vertices s_k into triplets, such that, in the solution, each vertex s_k of the same triplet is adjacent to the same vertex p_j, i.e., connected to p_j by an edge of $\mathrm{E_2'}$. By construction, such a triplet contains the vertices s_k corresponding to elements e_k of $\mathcal{U}$, which themselves belong to the same set S_j. It then suffices to select, in $\mathcal{S}'$, all sets S_j that contain the elements of the same triplet to obtain a solution to instance I_1. Therefore, I_1 is indeed a YES-Instance, which completes our proof. ◀

We now propose an MILP model to solve the GSI problem (Figure 4). In this model, variables Q_i (see `B.1`) represent the quantities assigned to proteins. Boolean variables X_i (see `B.2`) represent the fact that protein P_i may be chosen ($X_i = 1$) or discarded ($X_i = 0$). In the same spirit, boolean variables $x_{(p_j,s_k)}$ (see `B.3`) represent the fact that edge (p_j, s_k) in $\mathrm{E_2}$ may be selected ($x_{(p_j,s_k)} = 1$) or discarded ($x_{(p_j,s_k)} = 0$). Constraint `C.1` forces the selection of exactly one edge for each spectrum. Constraint `C.2`, through the classic "big M" method, enforces quantity Q_i to be equal to 0 when protein P_i is not chosen (i.e., $X_i = 0$), while imposing no condition otherwise. The last constraint (`C.3`) corresponds to the constraint on Δ_j in the problem formulation.

3 Application of GSI to real data

In this section, we focus on the practical implementation of GSI. Then, we benchmark GSI against two state-of-the-art protein inference methods, X!TandemPipeline [12] (XTP) and EPIFANY [18]. Finally, we discuss the obtained results and the specificities of the model.

Table 1 Summary of tests performed for benchmarking with GSI. From left to right, the columns indicate: the name of the dataset together with its number of spectra; the protein inference method used to compare GSI with; the peptide identification method used on the dataset together with the number of obtained PSMs; the name of the protein database as well as the total number of considered proteins; the type of evaluation method used to compare the results.

Dataset	Inference method	Peptide identification	Protein database	Evaluation method
iPRG2016 (27,363 spectra)	EPIFANY	Comet + Percolator (13,457 PSMs)	iPRG2016 PrESTs + E.Coli (5,592 proteins)	Number of identifications at 3% entrapment FDR
HeLa (10,155 spectra)	XTP	X!Tandem (4,859 PSMs)	Human (20,422 proteins)	Number of identifications
HeLa (10,155 spectra)	XTP	X!Tandem (4,712 PSMs)	Swiss-Prot (568,557 proteins)	Number of human identifications

3.1 GSI implementation and parameters

To allow further experimental studies about GSI, the proposed MILP was implemented in C++ using the CPLEX Optimizer (`https://www.ibm.com/fr-fr/products/ilog-cplex-optimization-studio`). Our implementation can be found on the following link: `https://github.com/BenoistEmile/GSI`.

All the following experiments have been done using a standard laptop (Intel Core Ultra 7 165H, 16 cores, 64GB RAM, Windows 11 25H2).

Detectabilities have been computed either with AP3 [7] or with DbyDeep [21], depending on the dataset under study.

Parameters Ψ_1, Ψ_2, Ψ_3 do not have default values. Instead, they have been set differently depending on the dataset at hand, after a series of tests to determine the ones that are favorable for GSI. This is also discussed, concerning dataset iPRG2016, in Section 3.3.

3.2 GSI benchmarking

In this section, we present a benchmark comparison between GSI and two well-established protein inference tools, which both rely on a statistical framework: X!TandemPipeline was selected because it is a convenient tool that integrates both peptide identification (using the widely-used X!Tandem algorithm) and protein inference, while EPIFANY is a more recent protein inference model that demonstrated strong performance on the iPRG2016 benchmark dataset [23]. Table 1 provides a comprehensive overview of the comparisons presented and detailed in the following paragraphs. While several protein inference models were presented in the Introduction, it was not possible to compare all of them. We therefore selected two reference models (X!TandemPipeline and EPIFANY) for our benchmark, but it should be noted that GraphPI achieved results close to EPIFANY on the iPRG2016 dataset that we used [14].

Table 2 Number of proteins identified by XTP and GSI on the human and Swiss-Prot databases. The ∩ column corresponds to the number of proteins that were identified by both tools. The "Groups" row indicates the number of protein groups identified by XTP. The "Human proteins" row indicates the number of human proteins among the identified proteins on the multi-species database.

	Human Database			Swiss-Prot database		
	XTP	∩	GSI	XTP	∩	GSI
Identifications	690	642	676	1965	366	868
Groups (XTP)	654			653		
Human proteins				625 (32%)	283 (77%)	330 (38%)

Comparison with X!TandemPipeline

To compare GSI and XTP, we used a dataset composed of spectra obtained from a lysate of HeLa cells. We conducted two experiments on this dataset's spectra (lines 2 and 3, Table 1). In both experiments, the well-established database search engine X!Tandem [3], which is embedded in XTP, is used to provide PSMs, scored with X!Tandem's hyperscore. What distinguishes the experiments is the protein database used to query spectra. The first database is composed of the 20,422 human proteins present in the Uniprot/Swiss-Prot database (`https://www.uniprot.org`). Since the spectra were derived from human cells, all the proteins in this database were potentially present in this sample, although it was not known exactly which ones. Hence, XTP and GSI are evaluated based on the number of proteins they are capable of identifying. The second database is composed of the 568,557 proteins present in Swiss-Prot, all species combined. Using this database allowed us to compare the proportion of identifications of non-human proteins (which are necessarily wrong identifications), and therefore the ability of the two engines to produce reliable identifications. In GSI, peptide detectabilities were computed using AP3 [7] and parameters Ψ_1, Ψ_2 and Ψ_3 were all set to 1.

GSI and XTP were then run on the obtained PSMs. GSI achieved a mean runtime of 56 seconds (over 5 runs) on the Swiss-Prot database, using a standard laptop, which is comparable to what was achieved by XTP. The detailed numbers of identifications from the two engines are provided in Table 2. On the human database, XTP identifies 654 protein groups regrouping 690 proteins in total. A group identification is produced when XTP is unable to distinguish between several proteins (for example, because they share the same subset of identified peptides). Therefore, the number of proteins whose presence has been proven by XTP is the number of group identifications. GSI identifies 676 proteins, of which 642 are identified by both models. This shows that GSI can produce reliable identifications and even outperforms XTP by identifying 3% more proteins.

On the multi-species database, XTP identifies 1,965 proteins in 653 groups, while GSI identifies 868 proteins, of which 366 are also identified by XTP. Among the 1,965 proteins identified by XTP, 625 are human proteins, yielding 32% of correct identifications. GSI identifies 330 of human proteins, yielding 38% of correct identifications. XTP identifies almost three times more individual proteins than in the first experiment, but the number of groups remains very close. On the other hand, GSI identifies more proteins than in the first experiment, but the increase is better controlled than for XTP (increase of only 28.4%). This suggests that GSI is able to better distinguish between similar proteins, which one is truly present in the sample. It can be explained by the fact that XTP uses a grouping

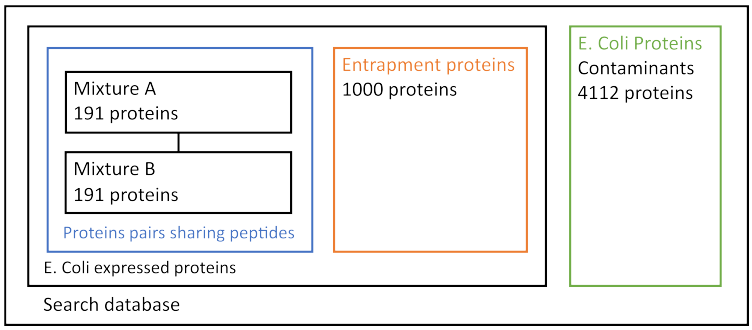

Figure 5 The proteins contained in the iPRG2016 search database belong to one of the following categories: proteins from mixtures A and B ; entrapment proteins, which are close to mixtures A and B proteins but are absent from the sample; contaminant proteins, which are likely to be present in the sample given its preparation protocol, but are not of interest. As we study a sample only containing mixture A proteins, we compute the entrapment FDR using mixture A proteins as targets and mixture B and entrapment proteins as decoys.

strategy without additional information, and hence is unable to distinguish which protein of each protein group is truly present in the sample. By contrast, GSI does not use this approach and instead integrates peptide detectability, which may explain (at least in part) its improved results. Furthermore, GSI identifies proportionally 18.8% more human proteins than XTP, which corroborates that GSI better distinguishes correct (in this case, human proteins) from incorrect (non-human proteins) identifications.

Comparison with EPIFANY

To benchmark GSI against EPIFANY, we chose to use the iPRG2016 dataset, which is designed for benchmarking protein inference, and in particular, to evaluate their ability to distinguish proteins whose sequences are close (see Table 1, line 1). EPIFANY was already benchmarked on this dataset, making it particularly convenient to compare the two models. We followed the same protocol as described in [18]. This protocol uses a search database composed of 4 categories of proteins (Figure 5): 191 *Mixture A* proteins, that are present in the analysed sample and that should be identified; 191 *Mixture B* proteins, that each share one peptide with mixture A proteins and are absent from the sample; 1,000 *entrapment* proteins, that are similar to mixture A and B proteins and are absent from the sample; 4,112 contaminant proteins, added to the database to avoid identifying proteins resulting from sample contamination. We downloaded the PSM identifications, scored by q-values, a statistical estimate of the FDR, as well as the protein identifications from EPIFANY. We computed peptide detectabilities for each identified peptide using DbyDeep [21] and set parameters $\Psi_1 = 1$, $\Psi_2 = 1$, and $\Psi_3 = 100$. We then applied GSI to the obtained PSMs, filtered at a 1% FDR (which represents 13,457 PSMs) as was done by EPIFANY, and obtained raw protein identifications. GSI achieved a mean runtime of 37 seconds (over 5 runs) on this dataset, using a standard laptop. For comparison, the reported runtime, in the paper, for EPIFANY on a similar number of peptides and proteins, and with a similar computer, is around 100 seconds. We used the same entrapment strategy as described in [18] (and used to validate EPIFANY's identifications) to compute an FDR and validate GSI's identifications. To compute the FDR used to validate the identifications, these proteins have to be labelled as *target* or *decoy*. The mixture A proteins, which we wish to identify, are labelled as targets. The mixture B and entrapment proteins are labelled as decoys.

Table 3 Number of proteins identified by EPIFANY and GSI on the iPRG2016 dataset, after filtering at a 3% entrapment FDR threshold. The ∩ column corresponds to the number of proteins that were identified by both tools.

	EPIFANY	∩	GSI
Identifications (at 3% FDR)	176	171	176

The identifications were filtered at a 3% entrapment FDR threshold for both models. The detailed numbers of identifications from the two engines are provided in Table 3. Both models identified 176 proteins, with an overlap of 171 proteins, thus showing that the two models have very close results. The benchmarking results on the iPRG2016 dataset therefore show that GSI can identify as many proteins as EPIFANY, with a runtime reduced by roughly two-thirds. Moreover, among the 1,681 spectra that were associated with several PSMs, 568 had PSMs with different scores. The model assigned 185 of these spectra (33%) to a peptide that was not the best-scoring one.

3.3 Discussion

The results of the benchmarks on both datasets show that GSI identifies as many proteins as a recent protein inference model (EPIFANY), with a considerably shorter runtime, and even outperforms an established tool (XTP). As seen earlier, most of GSI's comparative advantage likely derives from its use of peptide detectability, which allows it to disambiguate the identifications of similar proteins rather than regrouping them in common group identifications. Other approaches, such as the use of multiple digestion enzymes [16], can also perform such disambiguation but at a higher sample processing cost.

The analysis of GSI's performance is, however, hindered by the datasets used for the benchmarks. Indeed, the HeLa dataset provided, for most spectra, only one PSM per spectrum. With this dataset, it is therefore impossible to understand to what extent GSI's flexibility in assigning spectra to peptides is impacting the protein identification results. The iPRG2016 dataset, used for benchmarking GSI against EPIFANY, provides several PSMs for a higher number of spectra, and the benchmark on this dataset shows that GSI is able to provide a convincing solution while relying on less confidently identified peptides. However, only 5% of the identified spectra had multiple PSMs with discriminative scores (i.e., not all equal), which limits our understanding of the impact of this aspect of the modeling.

Moreover, in GSI, the non-probabilistic model associated with the constraints we set ensures that every protein identification can be explained by specific PSMs, thus providing a sounder interpretation than probabilistic models, which, although they are currently the most used, lack explainability.

Additionally, as shown in Table 4, we performed a benchmark concerning parameters Ψ_1, Ψ_2 and Ψ_3, on the iPRG2016 dataset.

We first tried to understand the relative weights of the three objectives. Indeed, the orders of magnitude of the three objectives are very different: in the case of the iPRG2016 dataset, the first objective is computed from the Δ values on up to 13,457 peptides, the second one is computed from the scores that are q-values (thus all < 0.01) of around 500 spectra (all the other spectra have only one PSM, or PSMs with equal values, and thus have no impact on this objective value), and the third one can go up to 5,592 if all proteins are identified. We therefore tried several configurations of the Ψ parameters, which show that the best results are achieved when $\Psi_1 = 1$ and $\Psi_3 = 100$. This can be expected because, on

this dataset, the main difficulty is to identify as few entrapment proteins (False Positives) as possible. Increasing the weight of the parsimony constraint encourages the model to identify fewer proteins, thus, thanks to the other constraints, focusing its identifications on correct proteins, and consequently lowering the number of false positives and improving the results. Nevertheless, the results are identical whether $\Psi_2 = 1$ or $\Psi_2 = 100$. We scrutinized the impact of the corresponding objective by deactivating it ($\Psi_2 = 0$) or setting an intermediate value ($\Psi_2 = 10$) (see Table 4, Ψ_2 impact). In both cases, the results are still the same. However, as explained above, this objective's value has an amplitude of 5, while the third objective will increase by 100 for each additional inferred protein. This is why it is logical that the value of Ψ_2 does not influence the results in that case. However, the flexibility offered by GSI on the assignment of spectra to peptides not only impacts the Ψ_2 objective, but also the computation of the Δ values, and thus the Ψ_1 objective, which has been shown to be important in this case.

Finally, we studied the impact of deactivating one or two objectives by setting the corresponding Ψ parameters to 0 (see Table 4, deactivation of one or two objectives). None of the tested configurations obtained an F-Score and a number of identified proteins as high as the $\Psi_1 = 1$, $\Psi_3 = 100$ configuration. The only case where an objective could be deactivated while maximizing the F-Score and the number of identifications is $\Psi_1 = 1$, $\Psi_2 = 0$, $\Psi_3 = 100$, for the reason aforementioned.

Using a multi-objective optimization approach could also be interesting in the future, to better understand our model by circumventing the use of such parameters, and directly identifying the best configuration.

4 Conclusion

In this paper, we presented and discussed an optimization problem motivated by proteomics, and more precisely, protein inference in mass spectrometry, that simultaneously optimizes two previously separated steps, peptide identification and protein inference. We showed that GSI is NP-hard even in very restricted conditions, and provided an MILP formulation for it. The application of this MILP to real datasets appears to be fast, even on large ones. Moreover, our model showed promising results, as presented by their comparison to the results obtained by XTP and EPIFANY, and has the advantage of obtaining results that are better explicable than the probabilistic models we compared it with.

Nevertheless, there is still room for improvement. Our GSI model could indeed be further developed to account for additional biological features, such as chimeric spectra, that represent up to 50% of the spectra in large proteomics datasets [9]. For this type of spectrum, which is produced by two different peptides, it would be interesting to modify GSI so that it assigns one spectrum to up to two peptides. Another way to improve GSI is to focus on its objective function, which is currently a weighted sum of three separate objectives. Converting GSI into a multi-objective problem using these three objective functions could also prevent us from using parameters Ψ_1, Ψ_2, and Ψ_3, whose impact is not yet fully understood, as previously discussed. On this specific point, we are planning a new implementation of GSI using a multiobjective solver, such as benpy (`https://pypi.org/project/benpy/`), to study this approach. This new implementation also aims to be easier to use and integrate into pipelines than the current one.

Table 4 Number of identifications using different configurations of the Ψ parameters on the iPRG2016 dataset. The "TP", "FP", "FN", and "TN" columns respectively contain the number of true positives, false positives, false negatives, and true negatives for each configuration. The F-Score, defined by $F_1 = \frac{2TN}{2TN+FP+FN}$, is a measure of the predictions' quality. The last column indicates the number of proteins identified after filtering the results at 3% entrapment FDR. The best F-score and the highest number of identifications achieved are shown in bold.

Ψ_1	Ψ_2	Ψ_3	TP	FP	FN	TN	F-Score	Number of identifications at 3% FDR
Relative importance of the three objectives								
1	1	1	176	47	16	1077	0.85	141
100	1	1	176	56	16	1068	0.83	140
1	100	1	176	46	16	1078	0.85	141
1	1	100	176	6	16	1118	**0.94**	**176**
1	100	100	176	6	16	1118	**0.94**	**176**
100	1	100	176	47	16	1077	0.85	141
100	100	1	176	56	16	1068	0.83	140
Ψ_2 impact								
1	0	100	176	6	16	1118	**0.94**	**176**
1	10	100	176	6	16	1118	**0.94**	**176**
Deactivation of one objective								
0	1	1	172	9	20	1115	0.92	172
1	0	1	176	49	16	1075	0.84	141
1	1	0	176	56	16	1068	0.83	140
Deactivation of two objectives								
1	0	0	176	56	16	1068	0.83	140
0	1	0	176	21	16	1103	0.90	149
0	0	1	172	9	20	1115	0.92	172

References

1 Jesse Angelis, Eva Ayla Schröder, Zixuan Xiao, Wassim Gabriel, and Mathias Wilhelm. Peptide Property Prediction for Mass Spectrometry Using AI: An Introduction to State of the Art Models. *PROTEOMICS*, 25(9-10):e202400398, 2025. doi:10.1002/pmic.202400398.

2 Émile Benoist, Géraldine Jean, Hélène Rogniaux, Guillaume Fertin, and Dominique Tessier. SpecPeptidOMS Directly and Rapidly Aligns Mass Spectra on Whole Proteomes and Identifies Peptides That Are Not Necessarily Tryptic: Implications for Peptidomics. *Journal of Proteome Research*, March 2025. doi:10.1021/acs.jproteome.4c00870.

3 Robertson Craig and Ronald C. Beavis. TANDEM: Matching proteins with tandem mass spectra. *Bioinformatics*, 20(9):1466–1467, June 2004. doi:10.1093/bioinformatics/bth092.

4 Joshua E. Elias and Steven P. Gygi. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nature Methods*, 4(3):207–214, March 2007. doi:10.1038/nmeth1019.

5 Jimmy K. Eng, Tahmina A. Jahan, and Michael R. Hoopmann. Comet: An open-source MS/MS sequence database search tool. *PROTEOMICS*, 13(1):22–24, 2013. doi:10.1002/pmic.201200439.

6 David Fenyö and Ronald C. Beavis. A Method for Assessing the Statistical Significance of Mass Spectrometry-Based Protein Identifications Using General Scoring Schemes. *Analytical Chemistry*, 75(4):768–774, February 2003. doi:10.1021/ac0258709.

7 Zhiqiang Gao, Cheng Chang, Jinghan Yang, Yunping Zhu, and Yan Fu. AP3: An Advanced Proteotypic Peptide Predictor for Targeted Proteomics by Incorporating Peptide Digestibility. *Analytical Chemistry*, 91(13):8705–8711, July 2019. doi:10.1021/acs.analchem.9b02520.

8 Michael R Garey and David S Johnson. *Computers and intractability*, volume 174. Freeman San Francisco, 1979.

9 Stephane Houel, Robert Abernathy, Kutralanathan Renganathan, Karen Meyer-Arendt, Natalie G. Ahn, and William M. Old. Quantifying the Impact of Chimera MS/MS Spectra on Peptide Identification in Large-Scale Proteomics Studies. *Journal of Proteome Research*, 9(8):4152–4160, August 2010. doi:10.1021/pr1003856.

10 Ting Huang, Jingjing Wang, Weichuan Yu, and Zengyou He. Protein inference: A review. *Briefings in Bioinformatics*, 13(5):586–614, September 2012. doi:10.1093/bib/bbs004.

11 R.M. Karp. Reducibility Among Combinatorial Problems. *Complexity of Computer Computations*, pages 85–103, 1972.

12 Olivier Langella, Benoît Valot, Thierry Balliau, Mélisande Blein-Nicolas, Ludovic Bonhomme, and Michel Zivy. X!TandemPipeline: A Tool to Manage Sequence Redundancy for Protein Inference and Phosphosite Identification. *Journal of Proteome Research*, 16(2):494–503, February 2017. doi:10.1021/acs.jproteome.6b00632.

13 Bin Ma, Kaizhong Zhang, Christopher Hendrie, Chengzhi Liang, Ming Li, Amanda Doherty-Kirby, and Gilles Lajoie. PEAKS: Powerful software for peptide de novo sequencing by tandem mass spectrometry. *Rapid Communications in Mass Spectrometry*, 17(20):2337–2342, 2003. doi:10.1002/rcm.1196.

14 Zheng Ma, Jiazhen Chen, Lei Xin, and Ali Ghodsi. GraphPI: Efficient Protein Inference with Graph Neural Networks. *Journal of Proteome Research*, 23(11):4821–4834, November 2024. doi:10.1021/acs.jproteome.3c00845.

15 Brendan MacLean, Daniela M. Tomazela, Nicholas Shulman, Matthew Chambers, Gregory L. Finney, Barbara Frewen, Randall Kern, David L. Tabb, Daniel C. Liebler, and Michael J. MacCoss. Skyline: An open source document editor for creating and analyzing targeted proteomics experiments. *Bioinformatics*, 26(7):966–968, April 2010. doi:10.1093/bioinformatics/btq054.

16 Rachel M. Miller, Robert J. Millikin, Connor V. Hoffmann, Stefan K. Solntsev, Gloria M. Sheynkman, Michael R. Shortreed, and Lloyd M. Smith. Improved Protein Inference from Multiple Protease Bottom-Up Mass Spectrometry Data. *Journal of Proteome Research*, 18(9):3429–3438, September 2019. doi:10.1021/acs.jproteome.9b00330.

17 Alexey I. Nesvizhskii, Andrew Keller, Eugene Kolker, and Ruedi Aebersold. A Statistical Model for Identifying Proteins by Tandem Mass Spectrometry. *Analytical Chemistry*, 75(17):4646–4658, September 2003. doi:10.1021/ac0341261.

18 Julianus Pfeuffer, Timo Sachsenberg, Tjeerd M. H. Dijkstra, Oliver Serang, Knut Reinert, and Oliver Kohlbacher. EPIFANY: A Method for Efficient High-Confidence Protein Inference. *Journal of Proteome Research*, 19(3):1060–1072, March 2020. doi:10.1021/acs.jproteome.9b00566.

19 Changyu Shen, Zhiping Wang, Ganesh Shankar, Xiang Zhang, and Lang Li. A hierarchical statistical model to assess the confidence of peptides and proteins inferred from tandem mass spectrometry. *Bioinformatics*, 24(2):202–208, January 2008. doi:10.1093/bioinformatics/btm555.

20 Steven R. Shuken. An Introduction to Mass Spectrometry-Based Proteomics. *Journal of Proteome Research*, 22(7):2151–2171, July 2023. doi:10.1021/acs.jproteome.2c00838.

21 Juho Son, Seungjin Na, and Eunok Paek. DbyDeep: Exploration of MS-Detectable Peptides via Deep Learning. *Analytical Chemistry*, 95(30):11193–11200, August 2023. doi:10.1021/acs.analchem.3c00460.

22 David L. Tabb, W. Hayes McDonald, and John R. Yates. DTASelect and Contrast: Tools for Assembling and Comparing Protein Identifications from Shotgun Proteomics. *Journal of Proteome Research*, 1(1):21–26, February 2002. doi:10.1021/pr015504q.

23 Matthew The, Fredrik Edfors, Yasset Perez-Riverol, Samuel H. Payne, Michael R. Hoopmann, Magnus Palmblad, Björn Forsström, and Lukas Käll. A Protein Standard That Emulates Homology for the Characterization of Protein Inference Algorithms. *Journal of Proteome Research*, 17(5):1879–1886, May 2018. doi:10.1021/acs.jproteome.7b00899.

24 The UniProt Consortium. UniProt: The Universal Protein Knowledgebase in 2025. *Nucleic Acids Research*, 53(D1):D609–D617, January 2025. doi:10.1093/nar/gkae1010.

25 Stefka Tyanova, Tikira Temu, and Juergen Cox. The MaxQuant computational platform for mass spectrometry-based shotgun proteomics. *Nature Protocols*, 11(12):2301–2319, December 2016. doi:10.1038/nprot.2016.136.

26 Bo Wen, Jack Freestone, Michael Riffle, Michael J. MacCoss, William S. Noble, and Uri Keich. Assessment of false discovery rate control in tandem mass spectrometry analysis using entrapment. *Nature Methods*, 22(7):1454–1463, July 2025. doi:10.1038/s41592-025-02719-x.