Interpreting Mass Spectra Differing from Their Peptide Models by Several Modifications

Albane Lysiak
Université de Nantes, CNRS, LS2N
Guillaume Fertin (guillaume.fertin@ls2n.fr)
Université de Nantes, CNRS, LS2N
Géraldine Jean
Université de Nantes, CNRS, LS2N
Dominique Tessier
INRAE, UR BIA, F-44316, Nantes

Research Article

Keywords: Proteomics, Peptide identification, Open Modification Search, Blind search, Post Translational Modification, Dynamic programming

Posted Date: December 8th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-1133992/v1

RESEARCH

Interpreting mass spectra differing from their peptide models by several modifications

Albane Lysiak[1,3†], Guillaume Fertin[1*†], Géraldine Jean[1†] and Dominique Tessier[2,3†]

*Correspondence:
guillaume.fertin@ls2n.fr
[1] Université de Nantes, CNRS, LS2N, F-44000, Nantes, France
Full list of author information is available at the end of the article
†Equal contributor

Abstract

Background: In proteomics, mass spectra representing peptides carrying multiple unknown modifications are particularly difficult to interpret. This issue results in a large number of unidentified spectra.

Methods: We developed SpecGlob, a dynamic programming algorithm that aligns pairs of spectra – each pair given by a Peptide-Spectrum Match (PSM) – provided by any Open Modification Search (OMS) method. For each PSM, SpecGlob computes the best alignment according to a given score system, interpreting the mass delta within the PSM as one or several unspecified modification(s). All the alignments are provided in a file, using a specific syntax. These alignments are then post-processed by an additional algorithm, which aims at interpreting the detected modifications.

Results: Using a large collection of theoretical spectra generated from the human proteome, we demonstrate that running SpecGlob as a post-analysis of an OMS method can significantly increase the number of correctly interpreted spectra, since SpecGlob is able to infer several, and possibly many, modifications. The post-processing algorithm is able to interpret unambiguously most of the modifications detected by SpecGlob in PSMs. In addition, we performed an extensive analysis to provide insight into the potential reasons for incomplete or erroneous interpretations that may remain after alignments of PSMs.

Conclusion: SpecGlob is able to correctly align spectra that differ by one or more modification(s) without any *a priori*. Since SpecGlob explores all possible alignments that may explain the mass delta within a PSM, it reduces interpretation errors generated by incorrect assumptions about the modifications present in the sample or the number and the specificity of modifications carried by peptides. Our results demonstrate that SpecGlob should be relevant to align experimental spectra, even if this consists in a more challenging task.

Keywords: Proteomics; Peptide identification; Open Modification Search; Blind search; Post Translational Modification; Dynamic programming

Background

Identifying peptides carrying modifications by mass spectrometry

Most of the spectra generated by mass spectrometry proteomic experiments remain unidentified after their analysis by an identification software. One of the main reasons for this poor identification rate is that most of these spectra correspond to the fragmentation of peptides carrying modifications. Conventional methods try to pair imperfect experimental spectra to ideal reference spectra – called *theoretical spectra* – generated from a peptide database. However, in order to avoid excessive computations, these methods limit the mass difference Δm between any experimental

spectrum and the theoretical models that will be tentatively paired to it – usually, Δm is limited to the mass spectrometer tolerance. As a result, many potential Peptide Spectrum Matches (PSMs) involving modifications cannot be detected, since these modifications may imply a larger Δm.

In order to overcome these limitations, Open Modification Search (OMS) methods, developed since the early 2000s [1, 2, 3, 4], allow matches between similar spectra that represent distinct chemical compounds with unequal masses whose difference may be large.

However, OMS methods were rarely used because of their excessive execution time, explained by the need to compare a very large number of theoretical models per experimental spectrum. The study of Gygi and al. in 2015 [5] reboosted the interest in OMS methods, stimulating the development of several approaches that resolved the execution time bottleneck [6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. Hence, most OMS methods have no constraint on the value of Δm between an experimental spectrum and its most similar theoretical spectrum paired in a PSM; others allow Δm to belong to a large range.

OMS methods can produce a sorted list of possible PSMs for each experimental spectrum but, usually, only the best candidate among theoretical spectra is provided.

Motivation for this study

Detecting modifications within a PSM is equivalent to finding an alignment between the paired spectra derived from the PSM, accompanied with one or several mass offset(s) allowing to better align the peaks of the theoretical spectrum to the peaks of the experimental spectrum.

When only *one modification* differentiates both spectra in a PSM (i.e., a subset of peaks is shifted by Δm), or when the types of possible modifications are known in advance, the alignment problem is already addressed: several efficient algorithms are available, with a scoring evaluation of the site modification's reliability (see [16] for a review and [17, 18] for new approaches).

However, when the two spectra of a PSM are separated by *several modifications*, Δm must be splitted into several parts, something which appears to be complex if no *a priori* on the nature of the modifications the experimental spectrum can carry is known. Some methods evaluate the most frequent modifications in the sample, and try to interpret Δm by a combination of these frequent modifications [19, 20]. Unfortunately, this strategy, by nature, limits the number of modifications that are considered, and interprets mainly artefacts. Consequently, they are prone to miss important biological modifications that are known to be present in low abundance in samples.

We have therefore developed a new method, called SpecGlob, that generates alignments of PSMs, even when *several unknown modifications* differentiate the paired spectra. In a nutshell, SpecGlob iteratively considers each mass difference between consecutive peaks in the theoretical spectrum – which correlates to the mass of an amino acid of the corresponding peptide – and tries to find two peaks in the experimental spectrum with the same mass difference, possibly including a mass offset to align this pair of peaks. For each PSM, SpecGlob returns its best alignment (based

on a predetermined scoring function, set by the user) under the form of a string we called *hitModified*, that gives several indications on how to align the peptide on the spectrum (see Example 5).

In order to evaluate the relevance of SpecGlob, we use theoretical spectra generated from the human proteome, similarly as we did in a previous study [21]. Theoretical spectra, whose sequences are known, are alternately playing the role of theoretical and experimental spectra. When a theoretical spectrum plays the role of an experimental spectrum, it is called a *bait*; given a *bait*, the most similar theoretical spectrum selected to constitute the PSM (excluding self-identification) is called a *hit*. Next, in a similar spirit as in [21], we introduce a classification of all *hitModified* sequences in three categories (green, orange and red), representing the level of difficulty of exactly retrieving the original amino acids sequence corresponding to a *bait*, starting from its *hitModified* string. To go further, we also developed a method that processes *hitModified* in order to build an amino acids sequence. This allows us to show that *hitModified* strings carry enough information to interpret most modifications, i.e. to unambiguously transform them into amino acids sequences, providing all or part of the *bait* sequences. Then, we analyse in depth the strengths and weaknesses of SpecGlob, and conclude that SpecGlob behaves very well on the tested dataset. Finally, we discuss its use in an experimental context.

Results

Description of SpecGlob's main features

SpecGlob takes a list of PSMs (*bait*, *hit*) as input, where *bait* is the spectrum to be interpreted, and *hit* is a peptide sequence (that can also be seen as a (theoretical) spectrum). Note that, at this point, such list of PSMs can be provided by any OMS method.

SpecGlob relies on an algorithmic technique called dynamic programming [22] (Methods). Dynamic programming is a very common paradigm in algorithmics, that has proved to be useful in many contexts; it has already been used as a way to obtain PSMs for experimental spectra carrying modifications [1, 2, 3, 23]. However, although SpecGlob is also based on dynamic programming, its objective greatly differs from the above mentioned prior works. Indeed, its goal is not to obtain PSMs; instead, for each given PSM (*bait*, *hit*), SpecGlob aims at detecting, without any *a priori* on their nature, the modifications that explain the differences between the spectra representing *bait* and *hit*.

Since PSMs are produced by an OMS method, each pair of spectra is expected to share some similarities. Thus, at least some of the mass differences between consecutive *hit* masses – each such mass difference corresponding to the mass of an amino acid of the *hit* – are expected to be found between masses of *bait* (which are peaks in its corresponding spectrum). SpecGlob considers each PSM individually. For each PSM, its objective is to best align the *hit* masses to the *bait* masses, possibly splitting Δm into several mass offsets on the way. For this purpose, SpecGlob looks for an alignment that maximizes a certain (user defined) score, which globally takes (positively) into account the number of aligned *hit* amino acids, and (negatively) both the number of non-aligned *hit* amino acids and the number of mass offsets inserted to fit the alignment. It is important to note that the number of mass

offsets required for an optimal alignment is not defined or limited in advance, thus any number of modifications is allowed.

For each PSM, the best alignment found by SpecGlob is provided as a string, that we call *hitModified*, which uses a specific syntax that summarizes the detected modifications. An overview of the SpecGlob workflow is provided Figure 5.

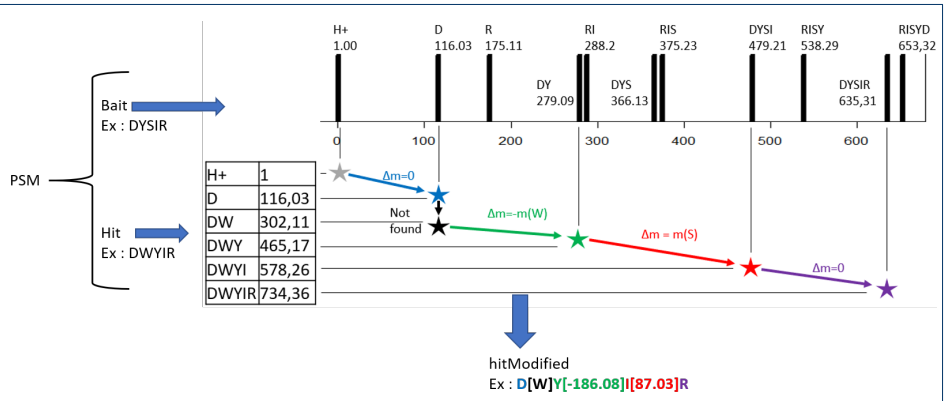

Figure 1: Overview of SpecGlob. As an example, the PSM (DYSIR, DWYIR) is taken as input in SpecGlob. Peptide DYSIR plays the role of the experimental spectrum, called *bait*, while peptide DWYIR plays the role of its best identification, called *hit*. The *hit* spectrum contains the masses of the *b*-ions after fragmentation of the *hit* peptide. As *bait* plays the role of a spectrum obtained from an unknown peptide, it contains both the masses of the *b*-ions and *y*-ions after fragmentation of the *bait* peptide (displayed in black vertical lines). To perform the best alignment of masses (according to a user defined score system), SpecGlob fills a dynamic programming table according to rules described in Equation (1) (Methods). The backtrack step returns the *hitModified* string, which can then be used to retrieve the original *bait* peptide. In our example, *hitModified* is interpreted as follows: D is aligned to the *bait* without any mass offset, W is not found in the *bait*, Y is aligned but the alignment requires a mass offset of -186.08 Da (which is the mass of W), I is aligned with a mass offset of 87.03 Da (which is the mass of S), and R is aligned without any mass offset. Thus, starting from *hitModified*, it is possible to retrieve the *bait* sequence DYSIR using a post-processing algorithm which will infer both the deletion of W and the insertion of S after Y.

For each PSM, the output string *hitModified* successively specifies the alignment of each amino acid from *hit*: (1) if two consecutive *hit* masses corresponding to the mass of an amino acid are aligned with two masses of the *bait* without any insertion of mass offset, then this amino acid is considered as found and is reported as such in *hitModified*; (2) if the mass difference between two consecutive *hit* masses corresponding to an amino acid is found between masses in *bait*, but the alignment of these masses requires a mass offset, then this amino acid is written in *hitModified* (since it has been encountered), followed by the value of this mass offset between brackets; (3) finally, if the mass difference between two consecutive *hit* masses is not used in the alignment (which means that the corresponding amino acid is absent), the amino acid is written between brackets in *hitModified*. Note that, by design, for each PSM the sum of all the mass offsets inserted in the alignment is equal to Δm. For an illustration, several alignments, yielding between one and three modifications, are given Table 1.

Evaluation of SpecGlob efficiency

In order to test SpecGlob, we ran SpecOMS [9] on a set of 1,148,608 theoretical spectra generated from the target-decoy human proteome database (see Methods). All of

Table 1: Examples of *hitModified* strings provided by SpecGlob, and their interpretations

bait	*hit*	*hitModified*	#Modification(s)	From *hit* to *bait*	Interpretable with SpecGlob?
GVTACCITK	GITACCITK	G[I]T[-14,02]ACCITK	1	mass(I) − 14.02 Da = mass(V) ⟶ Substitution(I,V)	Yes
EGASDEWIR	EASDEWIR	EA[57,02]SDEWIR	1	57.02 Da = mass(G) ⟶ Insertion(G)	Yes
DYSIR	DWYIR	D[W]Y[-186,08]I[87,03]R	2	186.08 Da = mass(W) ⟶ Deletion(W) 87.03 Da = mass(S) ⟶ Insertion(S)	Yes
VCASIYQK	VSFVIFVVIPIHASIYGAK	[V][S][F][V][I][F][V]V[-791,46] [I][P][I][H]A[-300,25]SIY[G][A]K	3	791.46 Da = mass(VSFVIFV) ⟶ Deletion (VSFVIFV) mass(IPIH) − 300.25 Da = mass(C) ⟶ Substitution(IPIH, C) mass(GA) = mass(Q) ⟶ Substitution (GA, Q)	Yes
QVSVIQWSSIVHGEQCCSVWNAK	QVSVIAK	QVSVIA[1957,82]K	1	1957.82 Da = mass(?)	No

Each row corresponds to a PSM (*bait, hit*) provided by SpecOMS, as well as its output *hitModified* by SpecGlob and its interpretation. For readability, spectra (*bait* and *hit*) are represented by their corresponding peptides: although *bait* plays the role of an unknown spectrum, its peptide sequence can be subsequently used to validate our results. The *hitModified* column shows the strings returned by SpecGlob, which provides an optimal alignment of *hit* to *bait*. Then, information contained in *hitModified* is transformed by a series of editing operations (deletions, insertions or substitutions) described in the fifth column, whenever this is possible. The rightmost column indicates whether the correct *bait* sequence can be retrieved after the editing operations.

these theoretical spectra were compared to themselves, excluding self-identifications, which ended up in 455,404 PSMs returned by SpecOMS. These 455,404 PSMs were then analysed by SpecGlob in about 3 minutes on a standard laptop (Supplementary File 1). In order to evaluate to which extent alignments provided by SpecGlob need the introduction of modifications, we have computed the number of modifications in each *hitModified* (see Figure 6). While about a quarter of the PSMs can be optimally aligned with only one modification, a large majority of PSMs yields an optimal alignment carrying *strictly more* than one modification.

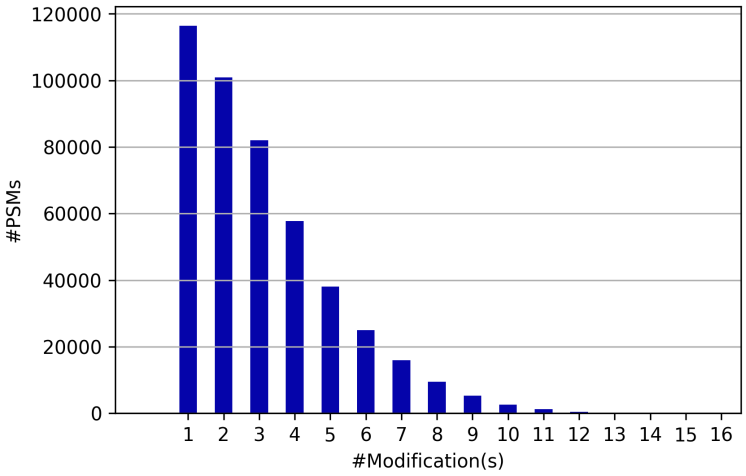

Figure 2: Distribution of the number of modifications per PSM predicted by SpecGlob. The 455,404 PSMs returned by SpecOMS were analysed by SpecGlob. In this dataset, the number of modifications predicted by SpecGlob varies between 1 to 16 in each PSM.

Once SpecGlob has aligned spectra and produced *hitModified* strings, it is possible to run a post-processing algorithm that aims at interpreting these results. In the

context of theoretical spectra, modifications between a *bait* and a *hit* are limited to insertions, deletions and substitutions of amino acids. Consequently, depending on the values of masses displayed in *hitModified*, modifications predicted by SpecGlob can be (more or less easily) interpreted as editing operations. In order to assess the degree of interpretability of PSMs processed by SpecGlob – including PSMs yielding several modifications –, we introduced a colour classification for modifications, in the same spirit as in [21]. The aim of this classification is to partition PSMs processed by SpecGlob into three categories (green, orange or red), each category reflecting a certain degree of complexity in retrieving the amino acids sequence of a *bait*, starting from the information provided by the *hitModified* string.

Firstly, a modification is considered *green* if it can be unambiguously converted into an editing operation. A modification whose mass value corresponds to the insertion of one amino acid or to the substitution of one amino acid by another illustrates this case. Deletion of one or several consecutive amino acids is also considered as a green modification. In this latter case, the corresponding mass offset displayed in *hitModified* is equal to the opposite of the total mass of the removed amino acids (displayed between brackets in *hitModified*). Secondly, a modification is considered *orange* if it can be explained as an edition operation of *bait*, but some ambiguity remains. This happens, for instance, in the case of a mass offset that corresponds to the insertion of two or three amino acids whose identities are known, but for which their order in the sequence remains uncertain. Note that because we need to limit execution time (and thus exploration space), in this implementation, we limit each insertion or substitution to a sequence of at most *three* amino acids. Finally, when a modification is neither green nor orange, it is *red*.

In our experiments on the human proteome, 52.23% of the modifications reported in the *hitModified* strings returned by SpecGlob are green, 20.22% are orange and 27.5% are red (Table 2).

We can also extend our colour classification from the *modification* level to the *PSM* level. Indeed, when all modifications in a given *hitModified* string are green, this means that we are able to unambiguously convert *hitModified* into a complete peptide sequence. In this case, the corresponding PSM is classified as *green*; a PSM is *orange* if the corresponding *hitModified* string returned by SpecGlob contains only green or orange modification(s), with at least one orange modification; a PSM is *red* otherwise (see Methods). In our experiments, we observe about 29% of green PSMs, while 25% are orange and 46% are red (Table 2).

Based on the results provided by SpecGlob and our PSM colour classification, and because we carried this study on theoretical spectra, we can evaluate *a posteriori* whether a green PSM is correct. Indeed, a model of *bait*, called *baitModel*, can be obtained from *hitModified* by converting each of its green modifications into its corresponding amino acid sequence. Hence, when *hitModified* is green, its corresponding *baitModel* is complete, and supposedly represents the *bait* sequence. Among the 132,137 green PSMs, an astonishing proportion of 97.3% (128,568 PSMs) has a *baitModel* that is strictly equal to the *bait* sequence. Among the few cases of disagreement between *baitModel* and *bait*, the following phenomenon occurs: an amino acid of *hit* has the same mass as the combination of two amino acids; in that case, SpecGlob considers that no modification occurs. For example, N has the

Table 2: Colour distribution of modifications and PSMs

	Green	Orange	Red	Total
#Modifications	740,458 232,925 in green PSMs 152,332 in orange PSMs 355,201 in red PSMs	286,616 141,132 is orange PSMs 145,484 in red PSMs	390,639 390,639 in red PSMs	1,417,713
#PSMs	132,137	114,454	208,813	455,404

The first line ,"#Modifications", shows the total number of green, orange and red modifications in the 455,404 PSMs, together with their distribution in each PSM colour (e.g., among the 740,458 green modifications, 355,201 are in red PSMs). The second line, "#PSMs" shows the number of PSMs in each colour category.

same mass as GG (resp. Q has the same mass as GA). In this case, N (resp. Q) is considered to be present in *baitModel* since its mass is encountered between two (non-consecutive) masses of *bait*, even though it is not present in *bait*.

Another phenomenon, although less frequent, may also occur when *bait* and *baitModel* disagree. This is when *y*-ions interfere in the alignment, and are wrongly considered as *b*-ions by SpecGlob (recall that only *b*-ions from *hit* are generated, see Figure 5); this leads SpecGlob to report modifications which are later considered as green, but are not.

Coming back to the colour classification of PSMs, we have identified two reasons for a PSM to be classified as red. The first one is that generating *baitModel* would have required a larger exploration of the combinatorics of masses than the one we allowed (recall that we limited a mass to correspond to a combination of at most 3 amino acids); therefore, if we allowed more computational efforts, these PSMs could be classified as orange. The other reason is that, at a certain point, the alignment computed by SpecGlob is simply wrong, due to the presence of *y*-ions in *bait* that were inadvertently aligned on *b*-ions in *hit*. In order to evaluate the impact of this latter case, we ran SpecGlob on the human proteome as we did before, but in this particular run, each spectrum generated from *bait* was represented by its *b*-ions only. Results are displayed in Table 3: in this case, even though the number of green PSMs has increased by 7.3%, the percentage of red PSMs still remains high (23.85%). We thus conclude that the red category mainly contains PSMs that require complex editing operations to convert *hit* into the *bait* sequence, rather than misaligned PSMs due to presence of noisy masses. However, in order to limit the number of misaligned masses (a phenomenon that could become an issue in an experimental, thus noisy, context), we modified SpecGlob in order to use it as a two-step algorithm. In this second implementation, we first use SpecGlob in the standard mode. Then, before the second step, *baits* are transformed as follows: since aligned masses that are identified are supposed to be mostly *b*-ions, their complementary ions (which are then supposed to be *y*-ions) are removed. This new representation of *baits* is then used to perform the second iteration of SpecGlob, and output the *hitModified* results. Using this two-step version of SpecGlob, a few thousands more PSMs are correctly interpreted, as shown in the last row of Table 3.

When a PSM is not green, this means that the *baitModel* is not a complete sequence of amino acids. However, even though this sequence is incomplete, it may contain long enough sequences of consecutive amino acids, which already are useful

Table 3: Colour classification of PSMs according to different representations of *bait* spectrum

Method	#Green PSMs	#Orange PSMs	#Red PSMs	Total
Classical	132,137	114,454	208,813	
Without y-ions	141,786	204,989	108,629	455,404
Without masses of complementary ions	135,362	148,308	171,734	

The first row, called *Classical*, recalls results obtained by the original version of SpecGlob applied on unmodified *baits*. On the second row, *bait* spectra were represented only by masses corresponding to their b-ions. The third row shows results obtained with the two-step version of SpecGlob. *Bait* spectra were transformed after the first step of SpecGlob: complementary ions to those that were aligned in the first step (which are then supposed to be b-ions) are thus supposed to be y-ions. In order to avoid misalignments between b-ions and y-ions, these complementary masses were removed before the second step of SpecGlob. When the output of SpecGlob differs between the first and the second step, the most favourable colour (green, orange or red) is chosen.

pieces of information. Moreover, as shown in Table 2, more than 50% of the modifications in PSMs are green. In order to more precisely evaluate the gain provided by green modifications in orange or red PSMs, we computed, for each PSM, the longest sequence of amino acids in both *hitModified* and its corresponding *baitModel*. The subsequent length distributions can be seen in Figure 7. On average, longest sequences in *hitModified* strings are of length 5.68 amino acids, while this average length increases to 7.46 in *baitModel* strings, thus is increased by almost 2.

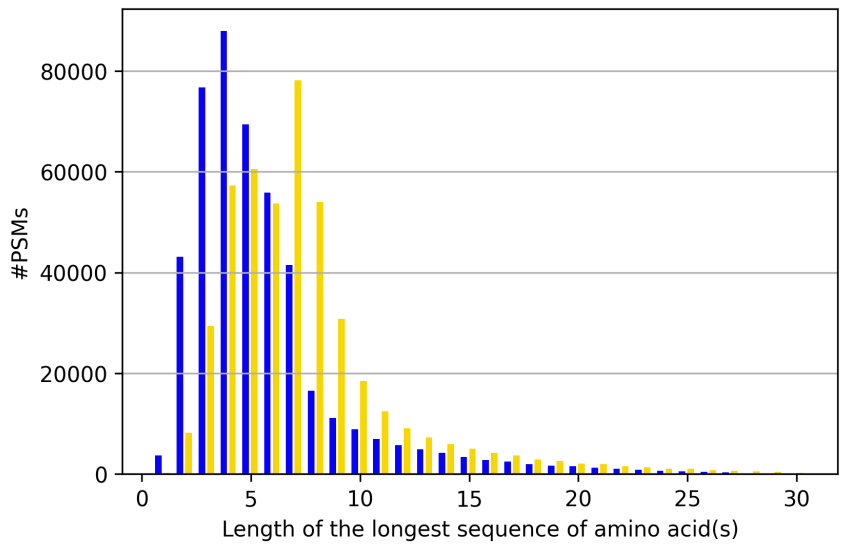

Figure 3: Distribution of longest sequences of amino acids in SpecGlob results. For each PSM, the length of the longest sequence of amino acid(s) was computed in *hitModified* strings (blue) and in *baitModel* strings (yellow).

Target-decoy validation

Validation of peptides obtained by interpretation software has been subject of much debate in the past [24, 25]. However, the target-decoy approach [26] has gradually

gained acceptance, both supported by convincing benchmark criteria in conventional searches and by its simplicity of implementation. The concept of separate False Discovery Rate (FDR) was later introduced, so as to better estimate the FDR associated to peptides carrying modifications [27]. Despite this substantial progress, it is still not clear whether the target-decoy approach is well adapted to evaluating the results returned by OMS methods.

In this study, by design, *bait* and *hit* always represent two different peptides from the human proteome, because we intentionally prohibit self-identification of peptides during the OMS method processing. Thus, at first glance, using the FDR measure in our context may seem surprising. The idea behind its use is that if a population of pairs of spectra were sufficiently similar to each other – a non-random similarity for the population of PSMs being determined by the FDR threshold –, then it would be possible to recover *baits* sequences from *hits*. Based on this assumption, we have reported in Figure 8 the PSMs colours, depending on whether *hit* is a target or a decoy peptide.

Among the 132,137 green PSMs, it is interesting to note that 38,350 of the *hits* (29%) are decoys. We thus conclude that a large number of PSMs are interpreted thanks to decoy *hits*, which can be considered as an issue, since the decoy database should reflect randomness. Moreover, we have shown that at 1% FDR (SPC = 21), only 57,784 PSMs (12.6% of the 455,404 PSMs obtained when SPC threshold is equal to 7) remain, thus are validated by the FDR. However, only 28,252 (48.9%) of these validated PSMs are green. This means that, in addition to the loss of many PSMs, about half of the remaining PSMs validated through the FDR cannot be interpreted without an additional computational effort, even though SpecGlob is able to align spectra with more than one modification.

Discussion

SpecGlob aligns pairs of spectra from collections of PSMs returned by OMS methods, possibly introducing several mass offsets, in order to increase the quality of the initial alignments and to obtain 'high quality alignments'. Since we compared theoretical spectra, we could define a 'high quality alignment' of a PSM as one that allows to restore the exact peptide sequence of *bait* (representing the experimental spectrum of the PSM) starting from *hit* (representing the peptide of the PSM). In that context, our main result is that SpecGlob is able to interpret a large proportion of PSMs returned by SpecOMS, even though several editing operations differentiate *bait* from *hit*.

Expanding the search space during the Peptide-Spectrum Matching process is already considered as crucial to avoid incorrect PSMs, due to the absence of comparison with the true peptide arising from closed searches [28]. By means of dynamic programming, SpecGlob goes one step further, since it compares, for each PSM, *all* possible alignments of the paired spectra derived from that PSM, and retains only the best one, regardless of the number of mass offsets required for that alignment. When the best alignment found by SpecGlob is a 'high quality alignment' for a given PSM, it is classified as green, and almost always perfectly interpreted. Despite this valuable progress, many PSMs remain classified as orange or red, and thus cannot be completely interpreted. However, since we chose to work in a theoretical context,

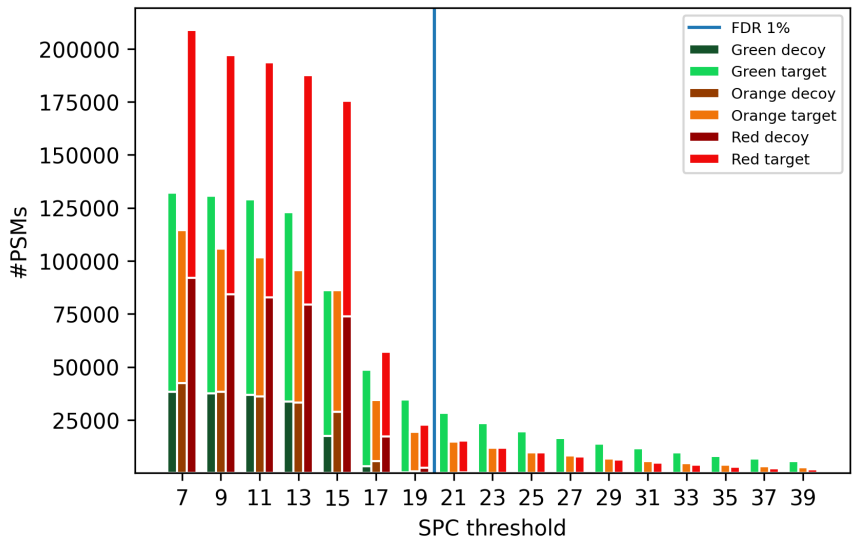

Figure 4: Distribution of target and decoy PSMs into the colour categories according to a SPC threshold. For SPC ranging from 7 to 39, all PSMs having at least this score are classified into the three colour categories; for each colour category, PSMs are classified as target or decoy according to the origin of *hit*. For example, for a threshold of 7, there are 132,137 green PSMs (93,787 targets and 38,350 decoys), 114,454 orange PSMs (71,952 targets and 42,502 decoys) and 208,813 red PSMs (116,748 target and 92,065 decoy). The blue vertical line shows the SPC for which we reach an FDR $<$ 1% (in our case, 21).

we are able to understand the reasons for which a PSM is classified as orange or red.

A PSM is classified as orange if the post-processing of *hitModified* is not directly able to select one among several peptide sequences to interpret *bait*. An additional computational effort is then necessary in order to achieve this selection, and hence to transform all orange PSMs into green PSMs. For this, it suffices to test the alignment of all possible alternatives corresponding to the mass of the modification, and finally select the one that aligns the maximum number of masses. We decided not to implement this method, as our objective was obviously not to improve the alignment of theoretical spectra, but rather to evaluate our method, in order to be able to transpose it to real-life spectra. In this latter context, it could be difficult, and sometimes impossible, to explore the combinatorics implied by some of the modification masses of the orange category, without knowing *a priori* the types of modifications carried by peptides, and thus their masses. Interpreting red PSMs is even more complex, for two reasons: first, post-processing each *hitModified* would rely on an even greater mass combinatorial search space than for orange PSMs; second, the presence of y-ions can perturb alignments, and the best alignment provided by SpecGlob may lead to a wrong result. However, we have showed that this latter situation is rare.

The number of PSMs classified in each colour category strongly depends on the composition of the PSMs provided as input to our algorithm. SpecGlob is most effective when *hit* contains most of the information needed to determine *bait*, i.e. when *hit* contains a large proportion of *bait*'s amino acids, in a relatively similar

order – even though SpecGlob is able to detect simple permutations. To support that idea, we have noticed that many green PSMs have a negative Δm (i.e., *hit* contains more amino acids than *bait*); on the contrary, many orange PSMs have a positive Δm (data not shown). Hence, one reason for which our method is not able to unambiguously interpret *bait* could be the absence of too many *bait* amino acids in *hit*. This problem could be intrinsic to the protein database, which in some cases may not be representative of certain peptides.

The score system we defined for SpecGlob to determine the best alignment can also influence the classification of PSMs. In the tests we carried out, a small variation in the scores had little effect on the results (data not shown). We favoured the insertion of mass offsets to align the maximum number of *hit* amino acids, which could lead to the insertion of a large number of mass offsets. Moreover, we have shown that each insertion of a mass offset presents the risk of introducing a mass alignment error (namely, a *b*-ion mass aligned to a *y*-ion mass). While in a theoretical context, this has little impact on the results, in an experimental context where many noisy masses can be present in a spectrum, the scoring system should probably be chosen so as to limit, to a certain extent, the introduction of mass offsets.

An even more important factor influencing the colour classification is the filtering of PSMs upstream to SpecGlob by the chosen OMS method. It is far from easy to select promising PSMs as input to SpecGlob before computing any alignment. Unfortunately, given the computational load of SpecGlob, considering alignments of all *baits* to all *hits* is unrealistic. However, aligning a limited number (e.g., several hundreds) of selected *baits* to all *hits* is perfectly conceivable, as it will not severely deteriorate SpecGlob performances in terms of execution time.

In our study, some promising PSMs were left aside by the chosen OMS algorithm. For instance, the fictitious PSM (DEFGHTQR, **W**DEFGHTQ**A**R) could be perfectly aligned by SpecGlob and would therefore be classified as "green", because only two modifications (one at the N-terminal extremity and the other before the last amino acid) are required to transform one peptide into the other. Unfortunately, as the two corresponding spectra have only one mass in common – since all *b*-ions masses are shifted, and only the mass of one *y*-ion is shared by both spectra –, this PSM will not be returned by SpecOMS which, like most methods, relies on the SPC to match spectra. Instead of WDEFGHTQAR, another *hit* could have been associated by SpecOMS, which in turn may not have given rise to a green PSM. In our study, SpecOMS has been set up so as to provide only one PSM per *bait* as input to SpecGlob. Delivering more than one PSM per *bait* and then selecting the 'best' *hit* returned by SpecGlob (considering e.g. the score of the obtained alignment and/or the colour of the corresponding PSM) would probably increase the number of green PSMs in the final results.

The FDR filtering of PSMs before any alignment of paired spectra, as it is usually done in association with OMS methods, is also questionable. For example, based on the FDR, PSMs that can be fully explained by SpecGlob, such as the actual PSM (GVPTEVK, GDPITVK), are discarded because of a low SPC (8 here). As shown in Figure 8, a large number of green PSMs come from the decoy database. This means that a *bait* from the target database can have a *hit* coming from the decoy database, and still end up as a perfect interpretation by SpecGlob, despite several

modifications and a relatively low SPC. Thus, although increasing the SPC reduces the FDR, many valuable PSMs are discarded during the FDR filtering.

Consequently, based on our results, we argue that PSM filtering based on the FDR, achieved before applying SpecGlob, does not seem appropriate. We suggest that a statistical estimation of the reliability of the final alignment alone could be more appropriate than a PSM filtering based on an FDR criteria evaluated on a score before any realignment. Estimations such as those provided in tools like PTMProphet [17] or PTMiner [20] could be very useful in that case. However, this question is beyond this study since we used here theoretical spectra only, and thus were able to measure the exact accuracy of our results.

In summary, even though SpecGlob performs globally well, it is not able to interpret all PSMs yet. We just argued, however, that there is still room for improvement, notably if progress is made on the filtering of promising PSMs. Moreover, even in the case where the *bait* sequence is not completely restored, it is important to note that SpecGlob, followed by its post-processing step, can partly restore it. More precisely, SpecGlob can produce subsequences of the amino acids that are present in *bait*, thus giving clues to peptide identification – and in turn to protein identification – with an efficiency that depends on their length and composition.

The way we express *hitModified* is already useful in itself to summarise incomplete alignments, because the information it contains goes beyond the localisation of the modifications, by highlighting stretches of amino acids detected in the spectrum. An alternative is to graphically visualise the spectra, together with annotated masses, which gives access to more details but is very time consuming for the user. Besides, *hitModified* strings are easy to handle by various scripts, which allows to gather information for further investigations, as we did for instance when generating the *baitModel* strings.

Conclusion

Our study shows that SpecGlob is a promising tool for analysis of mass spectra through post-analysis of PSMs. The results we obtained on the interpretation of theoretical spectra should now be used for testing and using SpecGlob on experimental spectra, possibly adapting it on the way. Its expected relevance on experimental spectra is supported by its very good behavior, demonstrated in this paper. Indeed, SpecGlob revisits the results returned by an OMS method and runs a fast competition between several possible interpretations of a PSM, which avoids interpretation errors. Besides, SpecGlob supplements OMS methods by providing interpretation of spectra under the form of an handy alignment that is easy to interpret.

Methods

Spectra dataset and protein database

We ran SpecGlob on the human proteome, downloaded from Ensembl99, release GrCh38 [29]. The protein database includes the *target* database (110,210 proteins with the annotation "protein coding" plus the 116 contaminants from the cRAP database) and the *decoy* database, generated by reversing all the protein sequences. After *in silico* digestion by trypsin, peptides whose lengths are strictly out of the range 7 to 30 amino acids were removed.

Theoretical spectra generated from the human proteome are then compared to themselves (self-identification excluded). All peptides from the database thus simultaneously play the role of experimental spectra (in which case they are called *baits*), and the role of peptides (in which case they are called *hits*).

PSMs generation with SpecOMS

SpecOMS [9] is a very fast OMS identification software used to identify experimental spectra, by comparison with theoretical spectra generated from a peptide database. This comparison is based on the number of shared peaks (Shared Peaks Count, or SPC) between pairs of spectra. Note that, in our work, only the masses of the peaks are considered, while intensities are not taken into account. Hence, we consider the terms *peak* and *mass* as totally equivalent.

First, given a *bait*, SpecOMS selects all *hits* that share at least t peaks with the *bait* (where t is set by the user). Next, in order to produce the PSM (*bait*, *hit*), the *bait* that shares the maximum number of peaks is chosen according to either the raw SPC (when parameter shift=`false`), or to the shifted SPC (when parameter shift=`true`) – see [21] for details.

In this study, we selected the best PSMs based on the shifted SPC score. More precisely, given a *bait*, for each *hit* having an SPC greater than or equal to 7, a new score was computed by SpecOMS, namely the maximum SPC (called shifted SPC) obtained after a realignment of *bait* and *hit* that successively assumes a single modification of mass Δm on each amino acid of the *hit*. For each *bait*, the *hit* having the maximum shifted SPC is retained, and SpecOMS produces the PSM (*bait*, *hit*).

SpecOMS was run with the following parameters: threshold=7, shift=`true` single_match=`true`, nbMissCleavage=0, minimumPeptideLength=7, maximumPeptideLength=30, maxMassesCount=60, minimumScore=60, decoyBase=`true`.

Generation of theoretical spectra in SpecGlob

Depending on whether a peptide plays the role of *bait* or *hit*, its representation through SpecGlob will differ. More precisely, when a peptide plays the role of *hit*, it is modeled by a spectrum only containing b-ions, which is enough to represent the amino acids (because y-ions are complementary to b-ions and can be deduced from the mass of the peptide). When a peptide plays the role of *bait*, it is considered as an experimental spectrum coming from an unknown peptide. Consequently, one cannot differentiate the b-ions from the y-ions, and for this reason both types of peaks are generated in the associated spectrum.

SpecGlob algorithm

SpecGlob relies on a dynamic programming approach to find the best alignment between the N masses of *hit* and the M masses of *bait*, which are respectively stored in arrays $hitMasses[]$ and $baitMasses[]$. A 2-dimensional matrix D of size $N \times M$ is filled according to a score system and a set of rules (see Equation (1)) which allow to compute, for any $0 \leq i \leq N-1$ and any $0 \leq j \leq M-1$, the value of $D[i][j]$.

$$D[i][j] = \begin{cases} \max\left(D[i-1][k] + s_A; \max\limits_{0 \le m < k} D[i-1][m] + s_R\right) & \text{If } aafound\text{=}\texttt{true} \\ D[i-1][j] + s_N & \text{If } aafound\text{=}\texttt{false} \end{cases} \quad (1)$$

The value of $D[i][j]$ depends on a boolean, $aafound$, which is set to `true` only if the mass of the i-th amino acid, say aa, of *hit* is found in *bait*. More precisely, we have $aafound$=`true` if there exists $k < j$ such that following holds:

$$baitMasses[j] - baitMasses[k] = hitMasses[i] - hitMasses[i-1]$$

Our scoring system uses three different score values, namely s_A, s_R and s_N (where A stands for Align, R for Realign and N for No-align). We use these three values to determine $D[i][j]$ as follows (see also Equation (1)):

- if $aafound$ is `true`, there are two possibilities, depending on whether the corresponding amino acid aa is found with a mass offset:
  - if aa is found without a mass offset, the alignment comes from cell $D[i-1][k]$, and the corresponding score is $D[i-1][k] + s_A$;
  - if aa is found with a mass offset, the corresponding score is $D[i-1][m] + s_R$, where m is the index (ranging from 0 to $k-1$) that maximizes the score. If several such indices m exist, the largest index is chosen.

  The maximum value between the two above hypotheses is selected.
- if $aafound$ is `false`, $D[i][j]$ takes the value $D[i-1][j] + s_N$.

For our experiments, we set the score system as follows: $s_A = 5$, $s_R = 2$ and $s_N = -4$.

The first line of D is set to 0 (thus $D[0][j] = 0$ for every $0 \le j \le M-1$), because $baitMasses[0]$ is always the mass of H^+ ion. For every $0 \le i \le N-1$, $D[i][0]$ is set to $i \cdot s_N$, which allows us to mark amino acids as not found when the alignment starts.

All cells $D[i][j]$ are then computed for i from 1 to $N-1$ and for j from 1 to $M-1$, i.e. from left to right and top to bottom, according to Equation (1). In the process, we store the origin (i.e. the coordinates) of the current cell's score, together with the corresponding type of alignment (Align, Realign or No-Align), in a matrix called $Origin$.

Once D is filled, a backtrack algorithm is performed, starting from the value $D[N-1][p]$, which corresponds to the maximum value present in the last row of D. If several maximum values exist, we choose the cell that adjusts Δm, where a cell of $D[N-1][j]$ is said to be *adjusting* Δm when $baitMasses(j) - hitMasses(size(hit) - 1) = \Delta m$, i.e. the sum of all mass offsets is equal to Δm; otherwise we choose the smallest value of p. During the backtrack procedure, the origin of each cell value is retrieved using $Origin$. According to the three cases described in Equation (1) together with the mass offset, the output string *hitModified* will contain the i-th *hit* amino acid as is, with a mass offset or marked as not found. We refer to Figure 5 for an illustration.

Execution time measurement

SpecGlob is implemented in Java, and has been executed on a laptop equipped with an Intel i7 (2.6 GHz) with 16 Gb of memory allocated to the JVM, running under Windows 10.

On the dataset we tested, it took approximately 3 minutes to run SpecGlob in the standard mode, while its execution time was approximately 5 minutes when the two-steps algorithm was used.

Post-processing algorithms

In addition to SpecGlob, we have also developed post-processing algorithms, which allow, based on *hitModified*, to (1) classify the modifications according to their degree of interpretability (green/orange/red modifications); (2) classify the PSMs in a similar fashion; and (3) transform green modifications into amino acid sequences in order to build the *baitModel*. The whole process is described hereafter – see also Table 4 for several illustrations.

Assigning a colour to each modification

Each *hitModified* string can be seen as a concatenation of substrings, which are of three possible types: a sequence of consecutive amino acids without brackets, a sequence of consecutive amino acids between brackets, or numerical values (which are mass offsets) between brackets – see Tables 1 and 4 for several examples of *hitModified* strings. Each of these substrings can be pairwise associated or treated alone, according to the type of modification in the PSM.

When an insertion has to be made in *hit* in order to retrieve the *bait* sequence, it corresponds to a numerical value x alone in *hitModified*. This modification is considered to be green if x corresponds to the mass of one amino acid – and does not correspond to the mass of 2 to n amino acids (n being a parameter of our classification algorithm, which we set to 3 in our experiments) –, since in that case the insertion is non ambiguous. If x corresponds to the mass of a sequence S of 2 to n amino acids, the modification is orange, because several sequences (notably, permutations of S) can correspond to this mass. Otherwise, the modification is red.

In case of a deletion, a sequence of amino acids between brackets is associated with a numerical value, which corresponds to the opposite of its mass. This type of modification is green.

In case of a substitution, the mass corresponding to amino acids between brackets and a numerical value are summed, and the resulting mass (say y) corresponds to a sequence present in *bait*. Similarly as for insertions, the colour we assign to the substitution depends on y. A substitution can correspond to a sequence of consecutive amino acids between brackets alone, without any associated mass offset. In this case, a substitution with a sequence S of the same mass must be done, and its colour depends on the length of S: green (resp. orange, red) if S is of length 1 (resp. between 2 and n, strictly greater than n).

Assigning a colour to each PSM

For each *hitModified* string, and by extension for the PSM from which it is derived, we assign a colour (green, orange or red), based on the colours of the modifications

it contains. First, when *hitModified* contains at least one red modification, the PSM is assigned colour red; otherwise, when *hitModified* contains at least one orange modification, the PSM is assigned colour orange; otherwise, this means that all modifications in *hitModified* are green, and thus the PSM is assigned colour green.

Generating the baitModel

A final procedure scans each *hitModified* string, and rewrites it by transforming each modification into an insertion, a deletion or a substitution, whenever this can be done unambiguously. Because lack of ambiguity is required here, only green modifications can go through this process. Once this is done, we obtain a sequence that we call *baitModel* (see Table 4 for different illustrations). If the PSM is green, then the *baitModel* sequence is complete (i.e. it is only composed of amino acids), and in that case *baitModel* can then be compared to the known *bait* sequence, in order to determine whether this is an exact match.

Table 4: Examples of *baitModel* strings determined from *hitModified* strings, together with their coloured modifications.

bait / *hit* / *colour*	*hitModified*	*baitModel*
VCASIYQK VSFVIFVVIPIHASIYGAK Green	[V][S][F][V][I][F][V]V[-791.46][I][P][I][H]A[-300.25]SIY[G][A]K Deletion of VSFVIFV; Substitution IPIH→C; GA→Q	VCASIYQK
GATPPAPPR GAAPAPPR Orange	G[A]A[-71.04]P[198.10]APPR Deletion of A; PT/TP	GA[o]PAPPR
FQMPDQGMTSADDFFQGTK GMTTVDDFFQGTK Red	G[746.31]M[T]T[-101.05][V]A[59.00]DFFQGTK ?; Deletion of T; GT/TG/AS/SA	[r]GMT[o]DDFFQGTK

Modifications and PSMs are coloured according to their degree of ambiguity, and *baitModel* strings are built by applying the modifications described below *hitModified* strings – only green modifications can be applied to build *baitModel*, as they are non ambiguous. For instance, the third row contains a *hitModified* string with three modifications: the first one is red because the mass offset does not correspond to any known mass in our repertoire (combination of masses of up to $n = 3$ amino acids); the second one is green, because the first T is not aligned, but the mass offset required to align the second T corresponds to its deletion; the third one is orange, because the mass of V added to the mass offset of 59 Da can be explained by several possible combinations of amino acid sequences (GT, TG, AS or SA). Because of the red modification, the PSM is classified as red. On the contrary, the first row contains only green modifications, hence the PSM is classified as green. The second row yields an orange PSM, because it contains one green and one orange modification.

Abbreviations

FDR: False Discovery Rate; OMS: Open Modification Search; PSM: Peptide Spectrum Match; SPC: Shared Peaks Count.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author's contributions

AL, GF, GJ and DT designed SpecGlob; AL designed the post-treatment algorithms and conducted the experiments; AL, GF, GJ and DT conceived the study and analysed the results. AL, GF, GJ and DT wrote the final manuscript.

Availability of data and materials

The human proteome was downloaded from Ensembl 99, release GrCh38 on the Ensembl FTP server ftp://ftp.ensembl.org/pub/release-99/fasta/homosapiens/pep/. Proteins predicted with the annotation "protein coding" were added to the contaminant proteins downloaded from the cRAP contaminant database ftp://ftp.thegpm.org/fasta/cRAP. The SpecOMS software is available at https://github.com/dominique-tessier/SpecOMS.

Funding
Supported by the French National Research Agency (ANR-18-CE45-004), ANR DeepProt.

Acknowledgements
Not applicable.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Author details
[1] Université de Nantes, CNRS, LS2N, F-44000, Nantes, France. [2] INRAE, BIBS facility, F-44316, Nantes, France. [3] INRAE, UR BIA, F-44316, Nantes, France.

References

1. Pevzner PA, Dančík V, Tang CL. Mutation-Tolerant Protein Identification by Mass Spectrometry. Journal of Computational Biology. 2000 Dec;7(6):777–787. Publisher: Mary Ann Liebert, Inc., publishers.
2. Pevzner PA, Mulyukov Z, Dancik V, Tang CL. Efficiency of database search for identification of mutated and modified proteins via mass spectrometry. Genome Research. 2001 Feb;11(2):290–299.
3. Tsur D, Tanner S, Zandi E, Bafna V, Pevzner PA. Identification of post-translational modifications via blind search of mass-spectra. Proceedings IEEE Computational Systems Bioinformatics Conference. 2005;p. 157–166.
4. Searle BC, Dasari S, Wilmarth PA, Turner M, Reddy AP, David LL, et al. Identification of protein modifications using MS/MS de novo sequencing and the OpenSea alignment algorithm. Journal of Proteome Research. 2005 Apr;4(2):546–554.
5. Chick JM, Kolippakkam D, Nusinow DP, Zhai B, Rad R, Huttlin EL, et al. An ultra-tolerant database search reveals that a myriad of modified peptides contributes to unassigned spectra in shotgun proteomics. Nature biotechnology. 2015 Jul;33(7):743–749.
6. Horlacher O, Lisacek F, Müller M. Mining Large Scale Tandem Mass Spectrometry Data for Protein Modifications Using Spectral Libraries. Journal of Proteome Research. 2016 Mar;15(3):721–731.
7. Burke MC, Mirokhin YA, Tchekhovskoi DV, Markey SP, Heidbrink Thompson J, Larkin C, et al. The Hybrid Search: A Mass Spectral Library Search Method for Discovery of Modifications in Proteomics. Journal of Proteome Research. 2017;16(5):1924–1935.
8. Kong AT, Leprevost FV, Avtonomov DM, Mellacheruvu D, Nesvizhskii AI. MSFragger: ultrafast and comprehensive peptide identification in mass spectrometry-based proteomics. Nature Methods. 2017 May;14(5):513–520.
9. David M, Fertin G, Rogniaux H, Tessier D. SpecOMS: A Full Open Modification Search Method Performing All-to-All Spectra Comparisons within Minutes. Journal of Proteome Research. 2017 Aug;16(8):3030–3038. Publisher: American Chemical Society.
10. Chi H, Liu C, Yang H, Zeng WF, Wu L, Zhou WJ, et al. Comprehensive identification of peptides in tandem mass spectra using an efficient open search engine. Nature Biotechnology. 2018 Oct;.
11. Bittremieux W, Meysman P, Noble WS, Laukens K. Fast Open Modification Spectral Library Searching through Approximate Nearest Neighbor Indexing. Journal of Proteome Research. 2018;17(10):3463–3474.
12. Solntsev SK, Shortreed MR, Frey BL, Smith LM. Enhanced Global Post-translational Modification Discovery with MetaMorpheus. Journal of Proteome Research. 2018;17(5):1844–1851.
13. Na S, Kim J, Paek E. MODplus: Robust and Unrestrictive Identification of Post-Translational Modifications Using Mass Spectrometry. Analytical Chemistry. 2019;91(17):11324–11333.
14. Bittremieux W, Laukens K, Noble WS. Extremely Fast and Accurate Open Modification Spectral Library Searching of High-Resolution Mass Spectra Using Feature Hashing and Graphics Processing Units. Journal of Proteome Research. 2019;18(10):3792–3799.
15. Devabhaktuni A, Lin S, Zhang L, Swaminathan K, Gonzalez CG, Olsson N, et al. TagGraph reveals vast protein modification landscapes from large tandem mass spectrometry datasets. Nature Biotechnology. 2019;37(4):469–479.
16. Chalkley RJ, Clauser KR. Modification Site Localization Scoring: Strategies and Performance. Molecular & Cellular Proteomics : MCP. 2012 May;11(5):3–14.
17. Shteynberg DD, Deutsch EW, Campbell DS, Hoopmann MR, Kusebauch U, Lee D, et al. PTMProphet: Fast and Accurate Mass Modification Localization for the Trans-Proteomic Pipeline. Journal of proteome research. 2019 Dec;18(12):4262–4272.
18. Cifani P, Li Z, Luo D, Grivainis M, Intlekofer AM, Fenyö D, et al. Discovery of Protein Modifications Using Differential Tandem Mass Spectrometry Proteomics. Journal of Proteome Research. 2021 Apr;20(4):1835–1848. Publisher: American Chemical Society.
19. Geiszler DJ, Kong AT, Avtonomov DM, Yu F, Leprevost FdV, Nesvizhskii AI. PTM-Shepherd: Analysis and Summarization of Post-Translational and Chemical Modifications From Open Search Results. Molecular & Cellular Proteomics. 2021 Jan;20:100018.
20. An Z, Zhai L, Ying W, Qian X, Gong F, Tan M, et al. PTMiner: Localization and Quality Control of Protein Modifications Detected in an Open Search and Its Application to Comprehensive Post-translational Modification Characterization in Human Proteome. Molecular & Cellular Proteomics : MCP. 2019 Feb;18(2):391–405.
21. Lysiak A, Fertin G, Jean G, Tessier D. Evaluation of open search methods based on theoretical mass spectra comparison. BMC bioinformatics. 2021 Apr;22(Suppl 2):65.
22. Bellman R. On the Theory of Dynamic Programming. Proceedings of the National Academy of Sciences. 1952 Aug;38(8):716–719. Publisher: National Academy of Sciences Section: Mathematics.

23. Cliquet F, Fertin G, Rusu I, Tessier D. Comparison of Spectra in Unsequenced Species. In: Springer-Verlag, editor. 4th Brazilian Symposium on Bioinformatics (BSB 2009). vol. Lecture Notes in BioInformatics (LNBI) of Lecture Notes in BioInformatics (LNBI). Porto Alegre, Brazil: Springer-Verlag; 2009. p. 24–35. Issue: 5576.
24. Noble WS. Mass spectrometrists should search only for peptides they care about. Nature Methods. 2015 Jul;12(7):605–608.
25. Sticker A, Martens L, Clement L. Mass spectrometrists should search for all peptides, but assess only the ones they care about. Nature Methods. 2017;14(7):643–644.
26. Elias JE, Gygi SP. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. Nature Methods. 2007 Mar;4(3):207–214.
27. Fu Y, Qian X. Transferred Subgroup False Discovery Rate for Rare Post-translational Modifications Detected by Mass Spectrometry. Molecular & Cellular Proteomics : MCP. 2014 May;13(5):1359–1368.
28. Degroeve S, Gabriels R, Velghe K, Bouwmeester R, Tichshenko N, Martens L. ionbot: a novel, innovative and sensitive machine learning approach to LC-MS/MS peptide identification; 2021. Company: Cold Spring Harbor Laboratory Distributor: Cold Spring Harbor Laboratory Label: Cold Spring Harbor Laboratory Section: New Results Type: article.
29. Yates AD, Achuthan P, Akanni W, Allen J, Allen J, Alvarez-Jarreta J, et al. Ensembl 2020. Nucleic Acids Research. 2020 Jan;48(D1):D682–D688. Publisher: Oxford Academic.

Figure Legends

Figure 5: Overview of SpecGlob. As an example, the PSM (DYSIR, DWYIR) is taken as input in SpecGlob. Peptide DYSIR plays the role of the experimental spectrum, called *bait*, while peptide DWYIR plays the role of its best identification, called *hit*. The *hit* spectrum contains the masses of the *b*-ions after fragmentation of the *hit* peptide. As *bait* plays the role of a spectrum obtained from an unknown peptide, it contains both the masses of the *b*-ions and *y*-ions after fragmentation of the *bait* peptide (displayed in black vertical lines). To perform the best alignment of masses (according to a user defined score system), SpecGlob fills a dynamic programming table according to rules described in Equation (1) (Methods). The backtrack step returns the *hitModified* string, which can then be used to retrieve the original *bait* peptide. In our example, *hitModified* is interpreted as follows: D is aligned to the *bait* without any mass offset, W is not found in the *bait*, Y is aligned but the alignment requires a mass offset of -186.08 Da (which is the mass of W), I is aligned with a mass offset of 87.03 Da (which is the mass of S), and R is aligned without any mass offset. Thus, starting from *hitModified*, it is possible to retrieve the *bait* sequence DYSIR using a post-processing algorithm which will infer both the deletion of W and the insertion of S after Y.

Figure 6: Distribution of the number of modifications per PSM predicted by SpecGlob. The 455,404 PSMs returned by SpecOMS were analysed by SpecGlob. In this dataset, the number of modifications predicted by SpecGlob varies between 1 to 16 in each PSM.

Figure 7: Distribution of longest sequences of amino acids in SpecGlob results. For each PSM, the length of the longest sequence of amino acid(s) was computed in *hitModified* strings (blue) and in *baitModel* strings (yellow).

Figure 8: Distribution of target and decoy PSMs into the colour categories according to a SPC threshold. For SPC ranging from 7 to 39, all PSMs having at least this score are classified into the three colour categories; for each colour category, PSMs are classified as target or decoy according to the origin of *hit*. For example, for a threshold of 7, there are 132,137 green PSMs (93,787 targets and 38,350 decoys), 114,454 orange PSMs (71,952 targets and 42,502 decoys) and 208,813 red PSMs (116,748 target and 92,065 decoy). The blue vertical line shows the SPC for which we reach an FDR $<$ 1% (in our case, 21).

Supplementary information

Additional file 1 : An extraction of SpecGlob output file supporting the conclusions of this article. To comply with the file size limit of the journal, only PSMs with a SpecGlob score $\geq$ 12 were included in this file.