Proper alignment of MS/MS spectra from unsequenced species

F. Cliquet[1,2], **G. Fertin**[1], **I. Rusu**[1], **and D. Tessier**[2]
[1]LINA, UMR CNRS 6241 Université de Nantes, 2 rue de la Houssinière, 44322 Nantes, Cedex 03, France
[2]UR1268 BIA, INRA, Rue de la Géraudière, BP 71627, 44316 Nantes, France

Abstract— *Correct interpretation of tandem mass spectrometry (MS/MS) data is a critical step in the protein identification process. Comparing experimental spectra against a library of simulated spectra generated from a database is the most common strategy for this interpretation. Unfortunately, problems arise when treating* unsequenced *species since, in this case, the proteins to be identified are absent from the databanks and experimental spectra can only be compared to theoretical spectra from close and already sequenced organisms. In this context, spectra comparisons become a notoriously difficult problem.*

In this paper, we deal with this problem by considerably improving PacketSpectralAlignment (`PSA`), a method we presented in [1]. First, we explain how to take full advantage of `PSA` by carefully selecting the most promising alignment positions during the algorithm, and how to precisely fix the parameters of `PSA`. Second, we present a new method, referred to as `PSAwEL`, which allows a better localisation of modifications. We then propose a new peptide identification framework that integrates these improvements. Finally, we propose a comparison between `PSA` and the reference, SpectralAlignment [2], which shows that `PSA` behaves better in terms of: (i) quality of the results; and (ii) execution time. Our tests were conducted on the ISB dataset [3]. We then validate our new framework on Brachypodium data.

Keywords: Proteomics, MS/MS, Spectra Alignment, Peptide Identification, Unsequenced species

1. Introduction

In proteomics, tandem mass spectrometry (MS/MS) is a common technique used to analyse proteins. By selecting, isolating and fragmenting the peptides, the tandem mass spectrometer produces a large number of MS/MS spectra. The protein identification process relies on the correct interpretation of these spectra. Each spectrum must be associated with the peptide it best represents, and once the peptides are correctly identified, they must be clustered to identify proteins.

The correct interpretation of a spectrum is a crucial step toward protein identification. The most widely used method relies on spectra comparison. It compares the experimental spectra produced by the mass spectrometer with theoretical spectra inferred from the different peptides generated by an *in silico* digestion of all the proteins contained in a databank. After a filtering process that aims at eliminating noise and incorrect candidates [4], comparisons between experimental and theoretical spectra are evaluated and ordered according to a given scoring function. The theoretical spectrum with the best score is associated with the corresponding experimental spectrum. The most intuitive scoring function is known as Shared Peaks Count (or SPC), and consists in counting the number of common peaks between both spectra. Many commercial or academic spectra comparison software systems, such as MASCOT [5] or SEQUEST [6] are based on SPC. Other more recent tools such as X!Tandem [7] or InsPecT [8] have led to improvements in terms of speed by adding efficient filtering methods.

Unfortunately, even with their latest improvements, these methods have limits, notably when the species under study are *unsequenced*. Since no protein databank exists for such species, spectra comparison can only be undertaken with theoretical spectra obtained from close and already sequenced organisms. Consequently, this means that multiple and various *modifications* (i.e., insertion, deletion or substitution of one or several amino acid(s)) may appear inside most of the peptides, which implies that many MS/MS spectra will not have an exact match inside the databank.

There are several solutions available at this time to address this difficulty. The first one consists in extending the databank by applying all the possible modifications to each peptide from the databank. However, this solution, which leads to an exponential number of possibilities, is obviously too time-consuming [9] when applied to unsequenced organisms. The other one, SpectralAlignment (`SA`) [2], [10], [11], is a dynamic programming algorithm that has been designed to identify peptides in the presence of modifications and Post Translational Modifications (PTM). However, as we will see later, it cannot deal with too many modifications (at most 2 [10]), and also presents some difficulties in terms of execution time on large datasets.

The main objective of this paper is to show that it is possible to improve spectra comparison in order to allow a better identification of peptides in presence of unexpected modifications. By unexpected, we mean that the user does not need to define them prior to identification.

In a previous work, we presented a new spectra comparison algorithm, PacketSpectralAlignment (`PSA`), based on a dynamic programming approach described in detail in [1]. This algorithm uses two intrinsic characteristics of spectra that we referred to as “symmetry” and “packet”, two notions that had never been previously taken into account (they are briefly presented in Section 2). In this paper, we go one step further by showing how we can take better advantage of the notion of packet. In Section 3, we begin by fully exploiting its potential with the notion of *possible positions*. We then deal with the practical issues of setting the parameters of PSA in order to obtain a good trade-off between execution time and quality of the results. In Section 4, we propose an enhanced version of PSA, called *PacketSpectralAlignment with Exclusion List* (or PSAwEL), specifically designed to allow the precise localisation of modifications. In Section 5, we show how we can integrate PSA into a complete peptide identification framework. Section 6 includes experiments that: (i) validate the improvements we have made to PSA by comparing it to SpectralAlignment; and (ii) validate our new framework. Section 7 is the conclusion.

2. Preliminaries

An MS/MS analysis induces peptide fragmentations, and the mass spectrometer determines the masses for peptides and their fragments. Although many peptide bonds could possibly be fragmented, the most significant cuts appear along the peptide backbone.

An MS/MS spectrum is mainly composed of a set of peaks, where each significant peak represents the mass resulting from the dissociation of a given peptide into two fragmented ions: an N-terminal ion (a_i, b_i, c_i), and a C-terminal ion (x_i, y_i, z_i) (see Fig.1). Additionally, peaks corresponding to neutral loss (water, ammonia) are also frequently observed. It can be observed that a *symmetry* exists between N-terminal and C-terminal peaks for each given fragmentation: the N-terminal peak corresponding to the N-terminal ion and the C-terminal peak are linked by the relationship m(C-terminal) $=$ m(peptide) $-$ m(N-terminal) $-$ 20, where $m(X)$ represents the mass of the fragment X (note that the value of -20 is due to the fact that the peptide is not symmetric at its extremities (Fig.1) and to ionisation). This notion of symmetry is valid for all fragmentations.

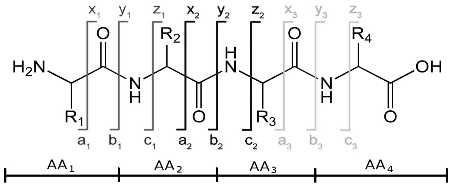


Fig. 1: Fragmentation points inside a peptide containing four amino acids (AA_i with $i \in [1;4]$). R_i ($i \in [1;4]$) are chemical compounds that determine the corresponding amino acid. The notation was introduced by Roepstorff et al. [12].

The presence of a modification in a peptide can be deduced by observing a difference in the peptide mass and a shift in all the masses of the fragments containing the modification. Nevertheless, in order to take a modification into account in a spectrum, it is not possible to simply shift all the peaks located to the right of the modification site because such shifts would destroy the link between N-terminal and C-terminal peaks, causing the inner symmetry to be broken. Given an experimental spectrum and a set of theoretical spectra obtained from a protein databank, our goal is to determine which theoretical spectrum best fits the experimental spectrum. In order to conserve the inner symmetry during the spectra alignment even if modification does exist, both theoretical and experimental spectra need to be pre-processed before alignment.

Construction of a theoretical spectrum using packets: Since the theoretical spectrum (S_t) is built *in silico*, we decided to replace the C-terminal peaks with their respective symmetric peaks for computational purposes. Let m_i be the mass of the i-th C-terminal peak. Its mass is then replaced by $M_{peptide} - m_i$, where $M_{peptide}$ is the mass of the peptide represented in S_t. After application of symmetry, the theoretical spectrum is referred to as the theoretical symmetric spectrum (or SS_t).

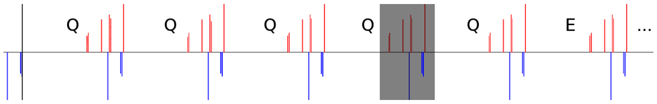


Fig. 2: A fragment of the theoretical spectrum corresponding to peptide QQQQQEGEEEGFIIR, after application of symmetry. The grey area represents a single packet.

For the construction of SS_t, we chose to keep, for each type of fragmentation, the nine most frequent peaks observed in experimental spectra when using a *Q-TOF* mass spectrometer [13], [14]. After the application of symmetry, these nine peaks may be clustered into a single *packet*. A packet represents all the fragmentations occurring between two consecutive amino acids of the peptide. Therefore, SS_t can now be represented by a list of packets rather than a list of peaks. An example of a packet is shown in the grey area of Fig.2.

Applying symmetry to experimental spectra: In order to make the alignment consistent, modifications similar to those applied to the theoretical spectrum should be applied to the experimental spectrum. Unfortunately, the distinction between C-terminal and N-terminal peaks in an experimental spectrum (S_e) is a complex task. For this reason, we decided to add a new symmetric peak for each existing peak in S_e, generating a new experimental symmetric spectrum (SS_e). However, in that case, it is necessary to prohibit the alignment of certain pairs of peaks: for example, N-terminal peaks from SS_t should only be aligned with the "original" peaks from SS_e. To do this, we just need to keep track of whether or not it is original, for each peak.

Alignment of spectra: PacketSpectralAlignment searches for the best alignment between the two spectra, SS_e and SS_t, that is, the alignment that obtains the best possible score, based on the assumption that a higher score means a better similarity. In order to find this alignment, we rely on a dynamic programming approach, as was previously done by Pevzner et al. [2]. The main distinction is that PSA does not align the peaks from SS_t on the peaks of SS_e, but the *packets* from SS_t on peaks from SS_e. Moreover, PSA prohibits the overlapping of two packets (see [1] for more details about PSA).

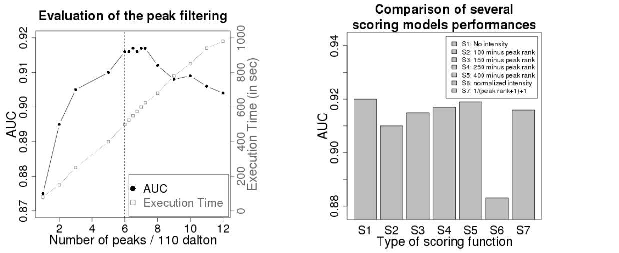
Datasets and evaluation of the results: In this paper, we often need to evaluate the quality of PSA, notably in order to tune correctly the different parameters. For this, we use a set of experimental data coming from the analysis of 18 well-known proteins: the ISB dataset [3]. By comparing the results provided by the ISB on these experimental data to the results we have obtained using PSA, we can compute the number of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). Using these values, it is possible to plot a Receiver Operating Characteristic (ROC) curve for each experiment, using ROCR [15]. A ROC curve represents the plotting of the true positive rate ($TPR = TP/(TP + FN)$) versus the false positive rate ($FPR = FP/(FP + TN)$). In order to decide the value to which a parameter should be set, we look at the area under the ROC curve (AUC). AUC is a good value to assess the quality of the results and is equivalent to the probability that a given method will rank a randomly chosen positive instance higher than it will rank a randomly chosen negative instance. Thus, the higher the AUC is, the better the results are [16].

In this paper, all parameters are evaluated on run number 2 of the 2^{nd} Mix on the *Q-TOF* 1 mass spectrometer from the ISB dataset [17]. This run contains 589 experimental spectra and is referred to as the $ISB\_dataset$ in this paper. Concerning databanks, two were used: (i) the $18mix$ that contains the 18 proteins from the above-mentioned analysed mix; (ii) the $18mix\_rice1700$ databank, obtained by adding 1700 randomly chosen rice proteins to the $18mix$ databank, to incorporate noise.

3. Improvements of PSA

3.1 Filtering the peaks from SS_e

Before any comparison between an experimental spectrum and theoretical spectra can be made, we need to filter out the background noise it contains. Based on results found in [4] and in our own experiments (see Fig.3(a)) , we chose to select only the six most intense peaks along a sliding window with a width of 110 Da, which corresponds to the average size of an amino acid. Fig.3(a) shows that these parameters guarantee the best trade-off between quality (AUC) and execution time on the $ISB\_dataset$.



(a) *(b)*

Fig. 3: *(a)* Behaviour of PSA in function of the number of peaks kept during the filtering process. *(b)* AUC for several packet scoring functions that take into account peak intensity. Peak rank refers to the rank of the peaks when they are ordered by intensity, 0 being the most intense. All the tests were conducted on the $ISB\_dataset$.

3.2 Possible Positions

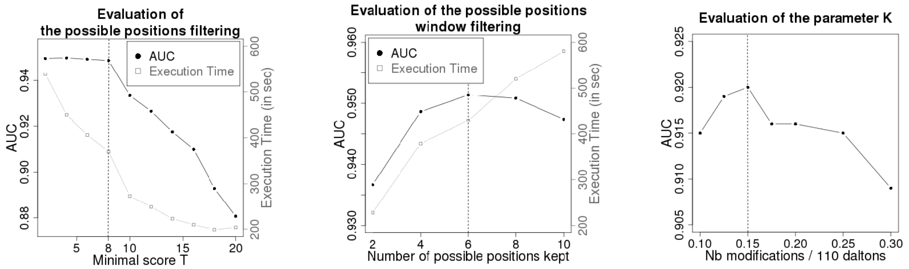
For any theoretical spectrum SS_t obtained from a peptide $\mathcal{P}$, it is important to note that the notion of packet does not depend on the precise identity of the amino acids that compose $\mathcal{P}$. Thus, PSA aims at aligning peaks from SS_e to peaks from SS_t by sliding a single packet P (that contains all the peaks from a single fragmentation) along SS_e. However, PSA can be easily improved, in terms of execution time, by a pre-process that consists in keeping only *possible positions* in memory, that is, positions in SS_e where the peaks from P are "sufficiently well" aligned with peaks from SS_e. More precisely, the possible positions will be positions in SS_e for which the alignment score between P and SS_e will be greater than or equal a given threshold T. It remains to determine the value of T, which can only be done once the peaks from SS_e are filtered and the "packet scoring function" is fixed.

Packet Scoring Function: As seen above, the packet scoring function is used to evaluate the quality of the alignment between packet P and SS_e, at each position m of SS_e. It is defined as follows:

$$Packet\_Score(P, m, SS_e) = \sum_{i=1}^{9} w(p_i) * s(p_i, m) \qquad (1)$$

where i is the index of a peak in the packet P, $w(p_i)$ is the weight associated with each peak p_i of P, and s is the "peak scoring function", that gives a value to each peak in SS_e that is aligned with p_i when P is at position m.

For each peak p_i of packet P, we decided to set $w(p_i)$ according to the probability of occurrence of the corresponding ion in an experimental spectrum (for a *Q-TOF* mass spectrometer, these probabilities can be found in [13],



(a) *(b)* *(c)*

Fig. 4: *(a)* Evolution of the quality and execution time of PSA in function of parameter T. *(b)* Evolution of the quality of the results depending on the number of possible positions kept for a 110 Da width window. *(c)* Evolution of the quality of the results depending on the number K of modifications searched by PSA. All the tests were conducted on the $ISB\_dataset$.

[14]). In SS_e, different factors could be used to define s. For example, if we simply consider the presence of a peak without any other information (such as peak intensity), then s is set to 1 (see S1 in Fig.3(b)). It is also possible to use the *peak rank* (i.e., the rank of the peak when they are ordered by decreasing intensity) to produce more elaborate values of s. Score S2 (resp. S3, S4, S5) in Fig.3(b) sets s to $C - peak\_rank$ with $C = 100$ ($C = 150$, $C = 250$ and $c = 400$, respectively). We also tested two other peak scoring functions, referred to as S6 and S7 in Fig.3(b): S6 uses a normalised intensity (i.e., s is set to $peak\_intensity/highest\_intensity$), while in S7, s is set to $1 + 1/(peak\_rank + 1)$.

The tests we conducted on the $ISB\_dataset$ are summarised in Fig.3(b). The comparison of these different values using the AUC reveals that taking intensity into account beyond the probabilistic weight $w(p_i)$ (see S2 to S7) does not improve the results compared to S1. We therefore decided to use S1 for our packet scoring function.

Filtering possible positions: Now that the packet scoring function is set, we need to define the threshold T over which a given alignment will be considered in the list of possible positions. Experiments run on the $ISB\_dataset$, and illustrated in Fig.4(a), show that opting for a value $T > 8.00$ weakens the results, while for any $T \leq 8.00$, the quality of the results remains roughly the same. In order to substantially decrease the execution time, we look for the greatest T that allows good results; we therefore decided to set $T = 8.00$. Moreover, this corresponds to the following intuitive idea: a position is kept if at least a b ion or a y ion (for which the respective scores are 8.3 and 8.7) is used in the alignment.

Another idea consists in filtering, this time by looking at how each possible position contributes to the packet score in comparison with other possible positions around it. More precisely, our idea is to use a sliding window and to keep only the best possible positions inside this window. Experiments on the $ISB\_dataset$ (see Fig.4(b)) showed that using a window with a width of 110 Da and keeping only six possible positions per window is a good trade-off. As an illustration, when we use this filtering on the $ISB\_dataset$, the average number of possible positions in a given experimental spectrum drops from 260 to 80.

3.3 Fine tuning of PSA parameters

As an input, PSA needs a list of (pre-computed) possible positions, a theoretical spectrum SS_t and the maximum number of modifications K that we allow. PSA then computes the score of the best alignment between the set of possible positions and the theoretical spectra, provided the alignment contains at most K modifications. PSA also outputs the list of modifications necessary to obtain this result. Moreover, for each modification, PSA outputs its position (in Daltons) and the corresponding mass difference (in Daltons as well).

PSA Scoring Function: To compare theoretical and experimental spectra, we need to define the scoring function used by PSA. The score S resulting from the alignment of several packets on a subset of possible positions of SS_e will be the sum of each corresponding packet score, minus a penalty of 10 for each modification. In fact, this modification penalty, which is slightly higher than the score of a b or a y ion (8.3 and 8.7 respectively), ensures that if PSA allows a modification, then at least *two more peaks* will be aligned. Otherwise, modifications that only align one more peak may occur too frequently.

Setting the value of K: As a parameter in our tests, we chose to use a value of K that was dependent on the size of the two compared spectra. In fact, the longer a spectrum is, the more its corresponding peptide could be subject to modifications. Fig.4(c) shows that 0.15 modifications per 100 Da (i.e., 1 modification per 660 Da) guarantees the best results.

4. Localisation of modifications

A first step toward the precise localisation of modifications is correct peptide identification. However, such a localisation raises new difficulties since it requires the perfect alignment of SS_t with SS_e. While alignment methods work well in most cases, they may sometimes detect unwanted modifications: this is mainly due to the fact that deciding whether a peak represents an N-terminal ion or a C-terminal ion is a difficult problem. In fact, in methods such as SA, a given peak could be considered N-terminal and C-terminal at the same time (of course, this may happen in experimental spectra where peaks are superimposed, but this phenomenon is not likely to frequently occur). However, allowing modifications during the alignment highly increases this risk because a modification could cause such a superposition and falsely increase the score.

In the case of PSA, the same problem can occur with possible positions: when a group of peaks implies a possible position p_1, their respective symmetry can create a **complementary** possible position p_2. The N-terminal peaks from p_1 (resp. p_2) are considered as C-terminal peaks in p_2 (resp. p_1).

In order to avoid this problem, we developed a new method that we called PacketSpectralAlignment with Exclusion List (or PSAwEL), in which we associate their complementary position inside an exclusion list with each possible position. Then, when PSAwEL tries to align a packet on p_2, for example, it checks whether p_1 is already aligned. If this is the case, then the contribution to the score for p_2 (referred to as S_2) will be changed to $S_2 - max(S_1, S_2)$, where S_1 is the contribution of p_1 to the score. This implies that p_1 and p_2 can both be aligned simultaneously, but in that case, the score will not be increased.

Adding the exclusion list notion inside PSA changes its complexity from $O(kmp)$ (where k is the number of modifications, m the number of possible positions in SS_e and p the number of packets in SS_t) to $O(kmp^2)$, considerably increasing its execution time. Hence, we cannot use it to search the entire databank. However, it can be used as a subroutine, in order to improve the results given by another method (such as PSA), by rescoring the n best results and reordering them. More precisely, for each SS_e, we keep the n peptides with the n highest scores, and we apply PSAwEL only to these peptides.

5. Identification Framework

In Fig.5, we illustrate our overall strategy to reduce the complexity and increase the performance of peptide-spectrum comparison in a complete peptide identification framework. This framework takes advantage of all the improvements presented in this paper, including the PSAwEL method, to allow the precise identification of modifications. Since the mass spectrometer can only evaluate the masses of peptides within a certain range, we filter the peptides from the databank by taking only the peptides that match this range into account (Fig.5 (b)). Experimental spectra (Fig.5 (f)) are filtered (Fig.5 (g)), as explained in Section 3.1. Next, symmetry is applied on both spectra (Fig.5 (d,h)), as explained in Section 2. Then, for experimental spectra, possible positions are created and filtered (Fig.5 (i)), as explained in Section 3.2. These steps are necessary to prepare the data to be compared using PSA (Fig.5 (k)). Using our PSA method allows a rapid identification of peptides, even in the presence of unexpected modifications. We then use PSAwEL only on the best scores given by PSA (Fig.5 (l)) making it possible to enhance the order of the results and, more importantly, to precisely localise the modifications. In this particular case (enhancing PSA using PSAwEL on a subset of the results), we chose to keep only the best $n = 100$ results for each spectrum.

Starting from a set of experimental spectra and a databank, once all the steps from this framework have been executed, we obtain an associated peptide for each spectrum (corresponding to the highest score match), together with a list of modifications.

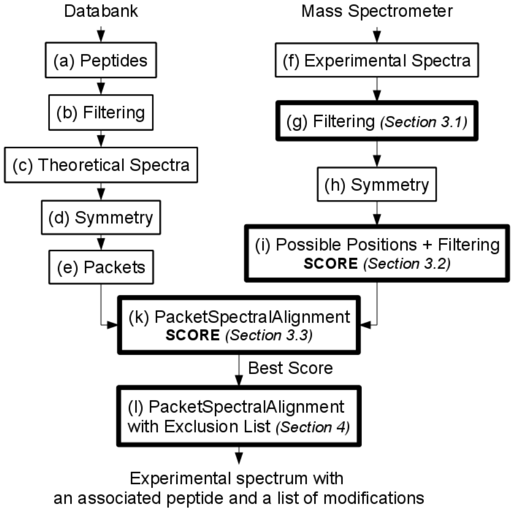

Databank
Mass Spectrometer
(a) Peptides
(b) Filtering
(c) Theoretical Spectra
(d) Symmetry
(e) Packets
(f) Experimental Spectra
(g) Filtering (Section 3.1)
(h) Symmetry
(i) Possible Positions + Filtering
SCORE (Section 3.2)
(k) PacketSpectralAlignment
SCORE (Section 3.3)
Best Score
(l) PacketSpectralAlignment
with Exclusion List (Section 4)
Experimental spectrum with
an associated peptide and a list of modifications

Fig. 5: The framework we propose to improve spectra identification in presence of modifications. Steps with a bold border are developed in this paper.

6. Results

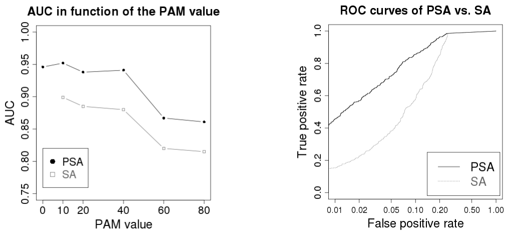
6.1 PSA versus SA

Modifications: In order to evaluate the behaviour of PSA and SA in the presence of modifications, we needed to introduce modifications inside the databank. To do so, we chose to use PAM matrices [18]. A PAM matrix contains the probability for each amino acid to mutate into another [18]. By definition, a PAM matrix induces 1 modification per 100 amino acids, but it is possible to apply it several times to increase its effect. For example, applying a PAM40 matrix is the same as applying a PAM1 matrix 40 times, and thus represents a higher average number of modifications. Consequently, it is possible to tune the distance we want to obtain between modified and unmodified sequences, simply by choosing a specific value x for the PAMx matrix. Table 1 shows the percentage of sequence identity obtained by comparing peptide sequences with their mutated version using different PAMx matrices, on the $18mix\_rice1700$ databank.

Table 1: Percentage of sequence identity between unmodified and modified peptides using different PAMx matrices on the $18mix\_rice1700$ databank.

Mutations	N/A	PAM10	PAM20	PAM40	PAM60	PAM80
Identity	100%	90.3%	82.3%	68.1%	57.4%	48.8%

We ran both methods, PSA and SA, on our experimental dataset using different PAMx matrices, in order to artificially mutate the $18mix\_rice1700$ databank. Even if SA was primarily designed to detect few post translational modifications, it is still adapted to detect other types of modifications in greater quantity. This test allows us to evaluate the behaviour of PSA as a function of the number of modifications. PSA and SA are both similarly parameterised, when possible: we search for the same number of modifications and the same filtering methods are applied to the databank. Using those parameters and filtering methods, we ran both PSA and SA on the $ISB\_dataset$ and on the $18mix\_rice1700$ databank, where both databanks were modified using different PAM matrices (i.e., PAM0, PAM10, PAM20, PAM40, PAM60 and PAM80).



(a) *(b)*

Fig. 6: *(a)* AUC values of PSA and SA for the search of the ISB dataset with different types of modifications on the databank $18mix\_rice1700$. *(b)* ROC curves for PSA and SA for the search of the $ISB\_dataset$ on the $18mix\_rice1700\_PAM10$ databank with a logarithmic scale for the FPR.

In order to evaluate the results provided by SA and PSA, we defined what we consider to be an expected identification. In the presence of modifications, the distance separating each databank peptide from the unmodified peptide found in the ISB results is computed. This distance is computed using the Needleman-Wunsch global alignment algorithm [19]. If this distance, which characterises the number of modifications between the two peptide sequences, is below a given value D, we consider the peptide from the databank as an expected result. We defined D using the same rule as for parameter K: $D = ((M_{peptide}/0.0015) + 1)$, i.e., using this value for D allows an average of one modification per 660 Da (1 modification every 5-6 amino acids). Note that the $+1$ is due to the fact that modifications at the beginning of a peptide are not counted within the K modifications tolerated. That way, for each comparison, we are able to detect whether the result is a positive identification.

We then computed each area under the ROC curve (AUC) using ROCR [15], as explained in Section 2. In Fig.6 *(a)*, we can see a plot of the different AUC values as a function of the PAMx matrix applied. We can easily see that PSA behaves better than SA in all the cases. We can also observe that difficulties arise in both methods when PAMx matrices are applied, with $x \geq 40$.

Using the AUC values, we can see that, globally, PSA gives better results than SA, but a direct look at the ROC curves (Fig.6 *(b)*) reveals another major difference. In this figure, we can see that the PSA ROC curve grows faster than the SA ROC curve, and that PSA gives good results for small FPR. For example, if we want a maximum FPR of 1% we obtain a TPR of 47.1% for PSA versus 13.5% for SA. Fig.6 *(b)* also shows a big difference in the results for an FPR of less than 10%.

Table 2: Average execution time of PSA versus SA on the $ISB\_dataset$ on a regular machine (Athlon X2 4800+ with 2GB of RAM).

Method	Avg. time (ISB dataset)	Avg. time (1 spectrum)
PSA	16,800 seconds	28.5 seconds
SA	84,000 seconds	142.5 seconds

The benefits of the packet notion from PSA are clearly visible in the execution time (see Table 2). In fact, the execution time of PSA is five times faster than the execution time of SA on the same input data.

6.2 Brachypodium

Experimental tests were undertaken to evaluate the capacity of our new framework to localise unexpected modifications. To do this, we used a set of experimental spectra coming from the analysis of a globulin from the Brachypodium organism compare to the databank containing the first chromosome of Brachypodium. Table 3 shows the number of analysed spectra, the number of identified spectra (peptides from the input protein that have obtained the higher score) for both PSA and our new framework (the only difference is that step Fig.5(l) is only applied for the framework case), the number of different identified peptides and, finally, the number of modified spectra detected (expected or not) using the framework.

Table 3: Number of identified peptides and of localized modifications on Brachypodium experimental data.

	Brachypodium dataset
# Spectra	846
# Id. Peptides (PSA)	92
# Id. Peptides (Framework)	135
# Diff. Id. Pep. (PSA)	10
# Diff. Id. Pep. (Framework)	17
# Localized Modif. (Framework)	32

By comparing the number of identified peptides obtained by PSA alone and by our framework, we can see the interest of rescoring using PSAwEL. We can note that a higher number of different peptides implies a better protein coverage, and a greater chance to find modifications.

7. Conclusion

In this paper, we proposed a solution to deal with the peptide identification problem in the case of unsequenced species.

We first showed how we strongly enhanced PSA, a method we developed in [1], by selecting the most interesting alignment positions and by fixing each parameter, such as the filtering of the peaks or the number of allowed modifications. All of these improvements were validated in terms of (i) quality and (ii) speed in comparison to the reference method, SA, on experimental data [3]. These results confirmed that PSA behaves better than SA and is therefore a good choice for our new framework.

We then presented a new method, PSAwEL, that makes it possible to increase first rank identifications in presence of modifications.

We have then proposed integrating all of these improvements, together with PSAwEL, within a new peptide identification framework. This new framework was evaluated using Brachypodium experimental data, showing that it led to the improvement of the number of first-ranked peptide identifications.

In the future, we plan to add another step to this framework, in order to be able to automatically interpret the detected modifications. Such an interpretation may be possible in most cases by using a databank that references all known modifications.

Acknowledgements

The authors thank Dr Colette Larré for providing Brachypodium data and Dr Hélène Rogniaux for fruitful discussions about MS/MS spectra interpretation. This research was supported by grant from the Région Pays de la Loire, France.

References

[1] F. Cliquet, G. Fertin, I. Rusu, and D. Tessier, "Comparison of spectra in unsequenced species." in *Proc. 4th Brazilian Symposium on Bioinformatics (BSB 2009)*, vol. 5676, 2009, pp. 24–35.

[2] P. A. Pevzner, V. Dancik, and C. L. Tang, "Mutation-tolerant protein identification by mass spectrometry." *J Comput Biol*, vol. 7, no. 6, pp. 777–787, 2000.

[3] J. Klimek, J. Eddes, L. Hohmann, J. Jackson, A. Peterson, S. Letarte, P. Gafken, J. Katz, P. Mallick, H. Lee, *et al.*, "The standard protein mix database: a diverse data set to assist in the production of improved peptide and protein identification software tools." *J Proteome Res*, vol. 7, no. 1, pp. 96–103, 2008.

[4] Y. Chen, J. Zhang, G. Xing, and Y. Zhao, "Mascot-Derived False Positive Peptide Identifications Revealed by Manual Analysis of Tandem Mass Spectra," *J Proteome Res*, vol. 8, no. 6, pp. 3141–3147, 2009.

[5] D. N. Perkins, D. J. Pappin, D. M. Creasy, and J. S. Cottrell, "Probability-based protein identification by searching sequence databases using mass spectrometry data." *Electrophoresis*, vol. 20, no. 18, pp. 3551–3567, 1999.

[6] J. Eng, A. McCormack, and J. R. Yates, "An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database," *J Am Soc Mass Spectrom*, vol. 5, no. 11, pp. 976–989, 1994.

[7] R. Craig and R. Beavis, "TANDEM: matching proteins with tandem mass spectra," *Bioinformatics*, vol. 20, no. 9, pp. 1466–1467, 2004.

[8] S. Tanner, H. Shu, A. M. Frank, L.-C. Wang, E. Zandi, M. Mumby, P. A. Pevzner, and V. Bafna, "InsPecT: identification of posttranslationally modified peptides from tandem mass spectra." *Anal Chem*, vol. 77, no. 14, pp. 4626–4639, 2005.

[9] J. R. Yates, J. Eng, A. McCormack, and D. Schieltz, "Method to correlate tandem mass spectra of modified peptides to amino acid sequences in the protein database." *Anal Chem*, vol. 67, no. 8, pp. 1426–1436, 1995.

[10] P. A. Pevzner, Z. Mulyukov, V. Dancik, and C. L. Tang, "Efficiency of database search for identification of mutated and modified proteins via mass spectrometry." *Genome Res*, vol. 11, no. 2, pp. 290–299, 2001.

[11] D. Tsur, S. Tanner, E. Zandi, V. Bafna, and P. A. Pevzner, "Identification of post-translational modifications by blind search of mass spectra." *Nat Biotechnol*, vol. 23, no. 12, pp. 1562–1567, 2005.

[12] P. Roepstorff and J. Fohlman, "Proposal for a common nomenclature for sequence ions in mass spectra of peptides." *Biomedical mass spectrometry*, vol. 11, no. 11, 1984.

[13] V. Dancik, T. Addona, K. R. Clauser, J. Vath, and P. A. Pevzner, "De novo peptide sequencing via tandem mass spectrometry," *Journal of Computational Biology*, vol. 6, no. 3-4, pp. 327–342, 1999.

[14] A. M. Frank, M. M. Savitski, M. L. Nielsen, R. A. Zubarev, and P. A. Pevzner, "De novo peptide sequencing and identification with precision mass spectrometry." *J Proteome Res*, vol. 6, no. 1, pp. 114–123, 2007.

[15] T. Sing, O. Sander, N. Beerenwinkel, and T. Lengauer, "ROCR: visualizing classifier performance in R." *Bioinformatics*, vol. 21, no. 20, pp. 3940–3941, 2005.

[16] T. Fawcett, "An introduction to ROC analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[17] J. Klimek, J. Eddes, L. Hohmann, J. Jackson, A. Peterson, S. Letarte, P. Gafken, J. Katz, P. Mallick, H. Lee, *et al.*, "ISB dataset," http://regis-web.systemsbiology.net/PublicDatasets/18_Mix/Mix_2/QTOF1/RAW_Data/QT20060103_S_18mix_02.raw/, 2007.

[18] M. Dayhoff, R. Schwartz, and B. Orcutt, "A model of evolutionary change in proteins," *Atlas of protein sequence and structure*, vol. 5, no. Suppl 3, pp. 345–352, 1978.

[19] S. Needleman and C. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *J. Mol. Biol*, vol. 48, no. 3, pp. 443–453, 1970.