Olivier Caron-Lizotte

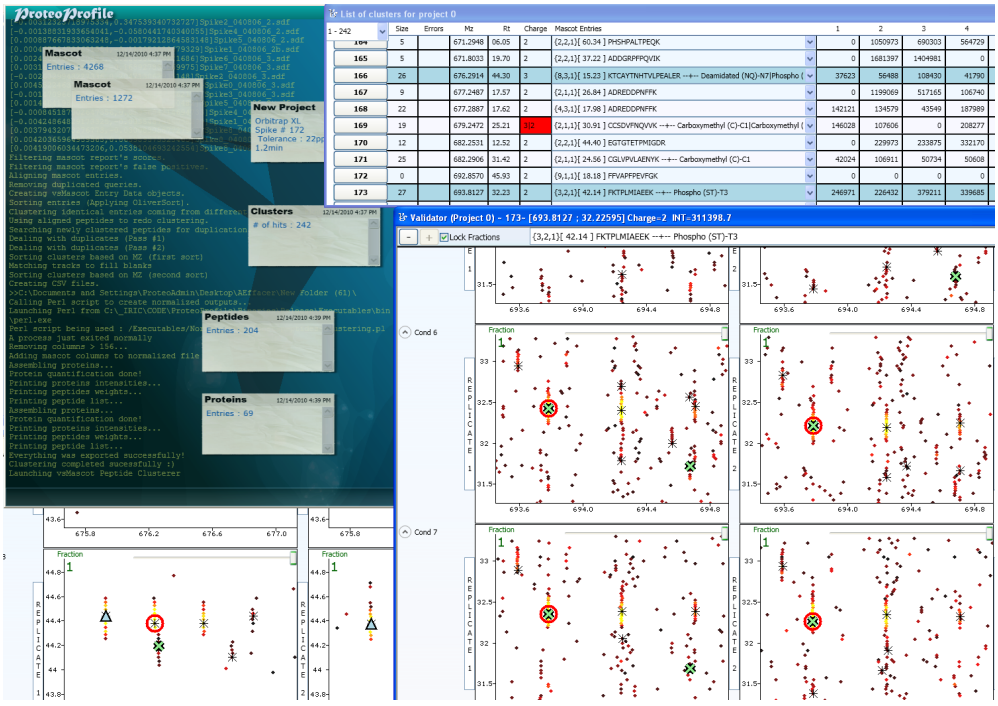
November 2012

IRIC, Proteomic Platform

ProteoProfile: Technical Documentation

What is ProteoProfile

ProteoProfile is a quantitative proteomics tool that integrates a suite of software and algorithms created to analyze information from high resolution liquid chromatography tandem mass spectrometry data (LC-MS/MS). Its primary use is to profile ion abundances across different conditions, fractions and replicates by clustering corresponding ions with peptide identifications found by search engine. LC-MS/MS files from proteolytic digests are analyzed and consolidated into exportable files reporting valuable information such as protein identification, changes in protein abundance, peptide identification and their corresponding coordinates (i.e. fraction, mass, retention time, abundance, etc…). All results can be viewed within ProteoProfile or exported as CSV (comma separated values) file reports that can be opened by Microsoft Excel or OpenOffice Calc. ProteoProfile currently process LC-MS/MS data from Thermo Fisher (.RAW), Waters (.raw), Agilent (.d) and other open formats (such as mzMLand mzXML).



Technologies behind ProteoProfile

Main technologies behind ProteoProfile

ProteoProfile is built using different technologies. The use and importance of these technologies is described throughout this documentation. Here is the list of technologies used by ProteoProfile (Follow the links after the technologies name to go directly to the related documentation):

- Main programming language : C#
- Graphical user interface : WPF
- Parrallel computing library : .Net 4.0 *
- Mass spectrometer files reading : ProteoWizard
- Database : MySQL *

*Prerequisites: MySQL database and a mapable network drive (server side); .Net 4.0 and MSFileReader (client side).

Overview of the data analysis workflow

Figure 1 shows a schematic representation of the different data processing steps, from the generation of peptide maps to the protein profiling. Four modules, highlighted in Figure 1, are integrated into ProteoProfile. The first module enables the conversion of LC-MS raw data into peptide maps (peptide detection). A second module (Mascot parser) converts the output of Mascot searches into a list of non redundant peptide identification. The third module (data clustering) correlates peptide identification with ions from all the peptide maps and determines changes in peptide and protein abundances. The data clustering module can be subdivided in four other modules: clustering, normalizing, peptide profiling and protein profiling. Finally, a fourth module (data validation) enables the user to inspect individual cluster to validate assignments and make manual changes when required. This last step will not be discussed in this document, since the process and result is solely dependent on users manipulations. Note that most steps produce a text file of the CSV format. This is helpful if you plan to integrate other tools with ProteoProfile.

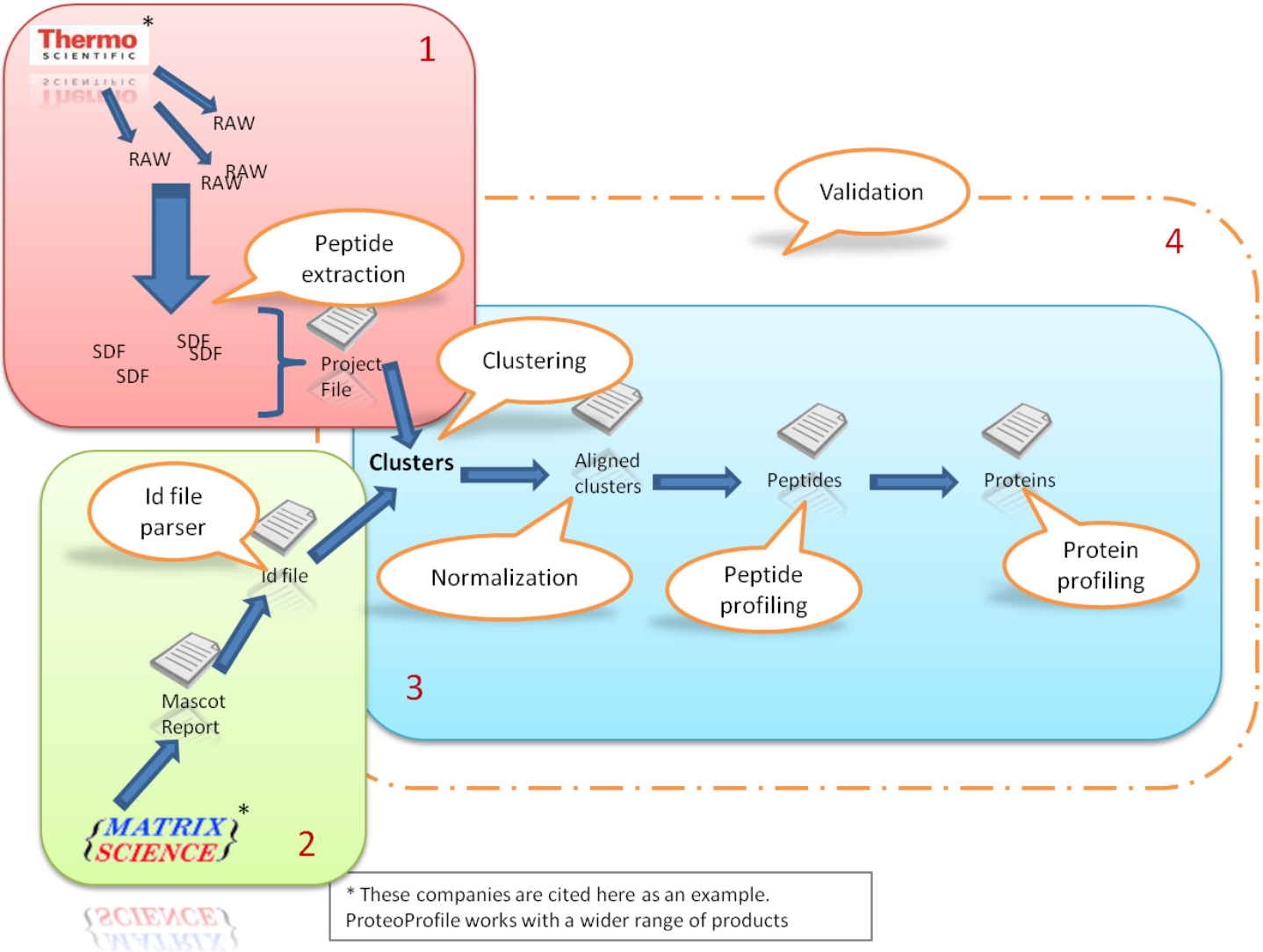


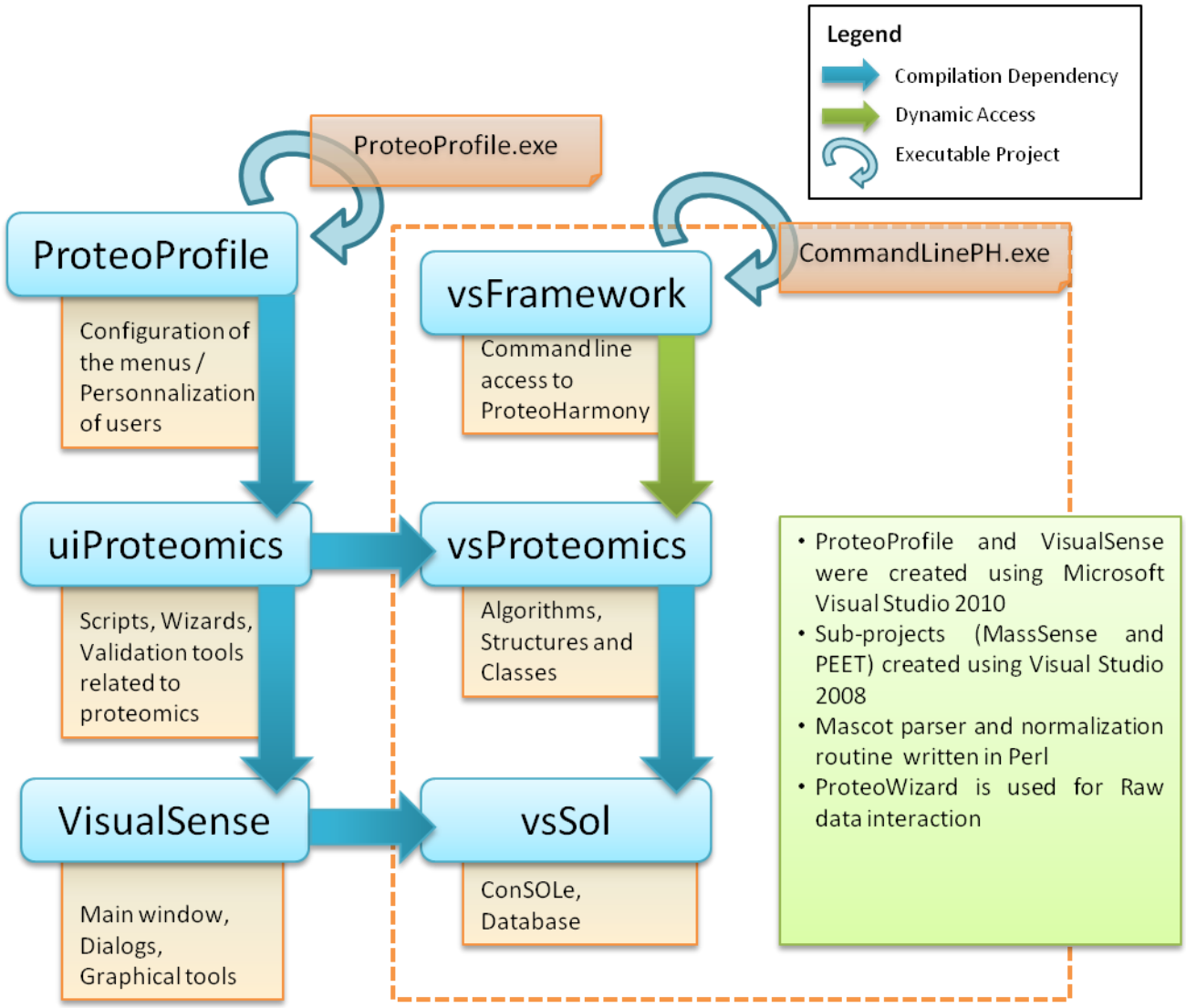
Figure 1 : Workflow depicting the data analysis steps for ProteoProfile : (1) Peptide extraction; (2) Identification file parser; (3) Clustering; (4) Validation

Software structure

ProteoProfile can be accessed with and without a graphical user interface. In the later case, a command line interface gives a direct access to the classes and functions, making it possible to cluster files or use other ProteoProfile tools from a supercomputer. ProteoProfile is constructed on VisualSense, a framework that was built specifically for this project. VisualSense is a WPF application framework that can be personnalized. It displays objects as visual nodes and adds menus to these objects based on their class type. VisualSense is ideal for work-in-progress software such as ProteoProfile. VisualSense also comes with a command line tool (Sol), used to access all objects from the wrapped software (in this case vsProteomics)

The command line interface of ProteoProfile is a C# compiler that executes the given commands within the ProteoProfile environement. Therefore, any publicly defined class or function is accessible from the command line. Note that this side of ProteoProfile has no written documentation. If you are interested in using it and are unsure about how to call a certain function, refer to the source code in the <vs...> projects. The <public static> classes are the most easily accessed functions.

Note that VisualSense, the graphical user interface of ProteoProfile, also has access to the command line version through the ConSOLe (Console Script Operated Language) located at the bottom of the main window.



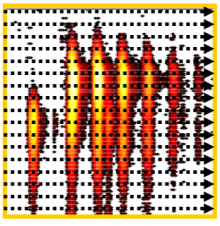
Module 1 : Peptide extraction

Overview

The main objective of the peptide extraction step is the generation of a list of peptides ions (ions with an associated charge state). To accomplish this, a 4 step algorithm was developed:

1. Determining two-dimensional peak tops (defining tracks)
2. Identifying monoisotopic components from peak top clusters and learned isotope distribution
3. Determining score for peptide likelihood as a relative measure of confidence assignment
4. Generate a sorted and filtered list of all peptide ions

1. Defining tracks

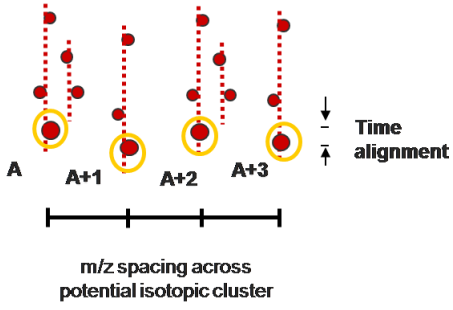


Raw files are searched from low to high m/z for each scan. The detection of peak for individual scan is done through a Butterworth routine. Tracks are then constructed for each of the detected peaks.

The tracks are built according to user settings, specifying the Mz tolerance, the minimal track length (number of scans), etc.

2. Identifying monoisotopic components

The charge state is determined by comparing the Mz spacing of tracks with a list of precomputed isotope profile distribution values. The isotopes are assigned to their corresponding tracks if they are within Mz and Rt tolerances.

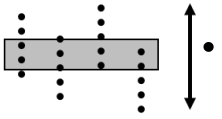


3. Determining score

$$S = \sum_{i=1}^{5} \left[\log \frac{\left| (A+i)/A)_{obs.} - (A+i)/A)_{theo.} \right|}{((A+i)/A)_{theo.}} \cdot \rho_i \right]$$

The score S is computed according to (where p is a weighting factor for 1^{st} (10^{-2}), .. 5^{th} (10^{-4}) isotopic ratios):

- The correlation of isotopic ratios (A+1/A, A+2/A,…,A+5/A) with theoretical values (averagine)



- An overlap function (overlap region must be greater than 0.3 of total length)

4. Generating peptide list

The last step is the generation of the peptide list. To achieve this, the peptides are sorted and filtered :

- Sorting of score based on charge state (charge state with highest score selected first)
- Removal of tracks (peptides without a charge are not part of this list)
- Selection of peptides score greater than -8 (predefined value)

Module 2 : Identification file parser

Overview

ProteoProfile was first developed and optimized using Mascot entries. Its file input format for the identification file still shows that. Although it is possible to use sources of peptide identification other than Mascot, the Id file must remain consistent and requires the creation of a personalized parser. ProteoProfile makes use of the Mascot score (column Pep Score) and it might be necessary to modify the scoring values (during parsing) to obtain a compatible scoring system. The score must be a value above 0 where bigger numbers are given to better identifications.

Obviously, the goal of the parser was to make a list of human readable peptide identifications. The result file, called the Id file, is in the CSV format, and the values it contains is imported in ProteoProfile to initiate the clustering process.

Header

Each peptide identification has a certain amount of information it can contain:

```
Search Log Num, FileName,Comment,UniProt ID, UniProt URL, PIR URL, EntrezID, Entrez URL, Protein
Description, Species, Mass, Num of Peptides, Peptide QueryNum, Peptide Sequence, Pep Modification, Protein
Assignments, Peptide Start, Peptide End, Pep Score, Pep Rank, Pep Observed Mz, Pep Calc Mass, Peptide
Observed Mass, Peptide Charge, Pep Elution Time, Pep Sample File, Peptide URL, Protein Score, Prot. PI,
Prot. Seq Length, Seq Coverage, PubMedID, MedLineID
```

All of these columns are not essential, but the number and position of columns must remain consistent from file to file. Leaving empty values is perfectly acceptable, as long as these values are always present:

```
FileName,UniProt ID, Protein Description, Peptide QueryNum, Peptide Sequence, Pep Modification, Pep Score,
Pep Observed Mz, Pep Calc Mass, Peptide Charge, Pep Elution Time and Prot. Seq Length
```

Content

Here is an example of a line of an Id file, with minimal information present:

```
,EXP3_040806_1.RAW,Spike J774,IPI00114209,,,,,Tax_Id=10090 Gene_Symbol=Glud1 Glutamate dehydrogenase 1;
mitochondrial,,61298.17,240,885,LVEDLK,,,85,90,32.69,1,716.36164,715.411591,715.354364,1,9.41,,,
1732.72033613445,,558,27,,,,0,
```

Module 3 : Data clustering

Objectives

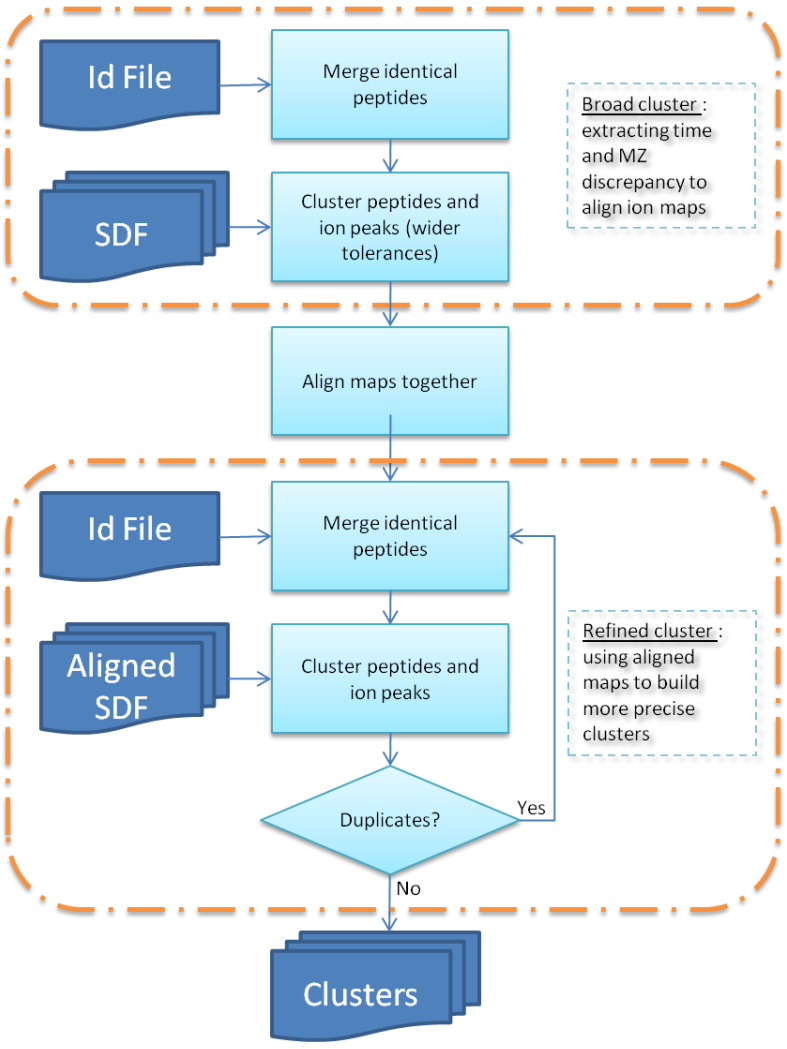
Data clustering is used to find common peptides across experimental conditions, replicates and fractions. The inputs for the clustering are:

- The peptide maps comprising a list of ion coordinates (m/z, charge, retention time, intensity) generated by MassSense (module 1). Each peptide map corresponds to an individual LC-MS run for a given condition/replicate/fraction.
- A list of identified peptides (Mascot entries, module 2). If unspecified, ions are clustered together, without identification (peptide and protein lists will be empty).

Overview

A brief description of the different steps applied during the clustering analysis is provided below:

1. Merge identical peptide entries (mascot entries) and update average mass / charge and elution time
2. Set the tolerance window 120% bigger in mass/ charge and 300% bigger in retention time
3. For each identified peptide, cluster corresponding ion from all maps together (see next section)
4. Based on computed clusters discrepancies, align the maps together (see section about alignment)
5. Redo step 3, this time using the user specified tolerances and the aligned maps
6. Remove the duplicates found by comparing clusters (see section about duplicates)
7. Redo step 4, this time using the user specified tolerances and the aligned maps
8. Remove duplicated ions by merging similar peptide identifications



Removing duplicates

A duplicate is an ion included in more than one cluster. It happens if several different identifications were given to the same ion. To resolve duplicated ions, ProteoProfile cycles through clusters from the highest peptide score to the lowest. When a duplicated entry is found between two clusters, it is removed from the list of possibility of the low score cluster. For example, if a duplication is found:

1. Remove duplicate from list of possibility
2. If all peptides found were duplicates, merge clusters
3. If clusters have same sequence and charge, merge clusters
4. Compute it again, based on its modified list of possibilities

This method is called to resolve duplicates when the ions included have a known charge. Duplicated ions without an assigned charge might still be included.

Previously, the clustering routine prevented these duplicates. But in doing so, it introduced more than a few errors. Until such a time when the clustering will not produce duplicates, users have to remove them manually using the Validator tool..

Definition of a cluster

A peptide cluster is defined as an identified Mascot entry that is correlated in all peptide maps. The algorithm chosen for clustering mimics a recursive test. But instead of going through all possibilities (which could take up to a month per cluster), it takes shortcuts to look only where the vast majority of results are found. At any moment, only the peptides within the limits of a tolerance window are tested. These limits include (but are not limited to) retention time and mass/charge ratio.

After comparing different methods for clustering a theoric set of peptides, it was found that above 99% of the times, the best possible cluster was described by one of these 4 methods:

1. Aiming at the average of the merged mascot entries, pick the closest peptides
2. Aiming at the previous result (average of step 1), pick the closest peptides
3. Create a cluster aiming at the peptides with the highest retention time: aim at this new cluster and pick the closest peptides
4. Create a cluster aiming at the peptides with the lowest retention time: aim at this new cluster and pick the closest peptides

Among these, the cluster with the least disparity (smallest variations in retention times, mass/charge ratio, charge and intensity) is selected. Approximately 97% of the resulting clusters are found by the first method. The rest is found in the second (2.8%) while the last two methods account for less than 1% of clusters.

Picking the closest peptide

The closeness of peptide is defined by a score that takes into account the retention time, the mass / charge ratio, the charge and the intensity of peptides and peaks. Mass to charge ratio account for 50% of the score, while retention time and intensity account for 25% each. The function returns a number between 0 and 1 (from best to worst match). Here is the function :

```
float score = 1.0f;//If no match was possible, return 1
float modifier = 0.0f;//Score modifier

//Compare charges, worsen the score for give no mercy to different charges
if (charge1 != charge2)
    modifier += 0.2f;

//Compute difference in MZ and give twice as much importance to Mz
float tmpScore = 2 * Math.Abs(mz1 - mz2) / MzTol;

//If its within tolerances
if (tmpScore <= 2.0f)
{
    float rtScore = 1.0f;
    //Compute difference in RT but force results between 0 and 1 only if RT is defined
    if (rt1 != -1)
        rtScore = Math.Abs(rt1 - rt2) * (1 / RtTol);

    //If within RT tolerance
    if (rtScore <= 1.0f)
    {
        tmpScore += rtScore;

        //Compute difference in intensity and divide by the sum of intensities
        if (inten1 != -1)
           tmpScore += (Math.Abs(inten1 - inten2) / (inten1 + inten2));
        else
            tmpScore += 1.0f;//Assume the worst if no match is available
        score = modifier + tmpScore * 0.25f ;//sets the score between 0 and 1 by dividing by 4
    }
}
```

MzTol and RtTol are respectively the tolerance in mass/charge and the tolerance in retention time. The variables mz1, rt1, charge1 and inten1 are the mass/charge ratio, retention time, charge and intensity for the peptide while mz2, rt2, charge2 and inten2 are the aimed values. Note that mz2 is always the average observed mz values of the mascot entries part of this cluster. When used, inten2 represents the average intensity of the replicates.

If a value is outside the Mz and Rt tolerance window, it is given a 1, the worst possible score. A score of 0 means that the two compared objects are perfectly identical.

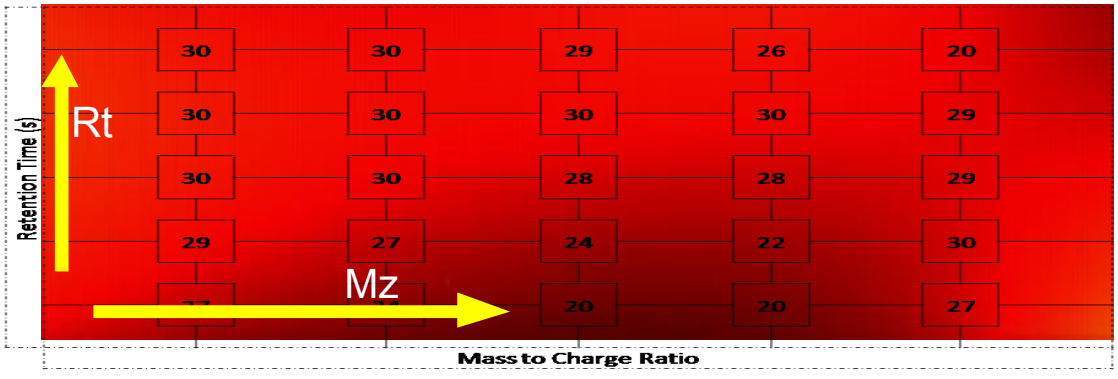
Alignment of ion maps

Alignment of the maps

Maps are aligned based on the discrepancy between clustered peptides computed in the first run (step 3 of the data clustering overview). To achieve this, the discrepancies must be collected both in Mz and retention time. For each map, the difference between the selected ion and its expected value (average value for the cluster) is computed locally in a grid (by default, it is a 2 by 2 grid, but in this example it is set to a 5 by 5 grid). This generates two maps: one representing the Mz discrepancy for the current Mz and retention time values, and a second one representing the retention time discrepancy. Although its impact on the mass/charge ratio is very small, it has been shown to improve precision in some cases.

Retention time (s)	Rt	ΔRt	ΔRt	ΔRt	ΔRt	ΔRt
	3300	30	30	29	26	20
	2700	30	30	30	30	29
	2100	30	30	28	28	29
	1500	29	27	24	22	30
	900	27	24	20	20	27
		600	800	1000	1200	1400
	Mass to charge ratio					

These numbers feed a bicubic spline interpolation* routine that computes corrections to be applied in regions of the maps. The Bicubic spline interpolation is an algorithm that computes a smooth surface from equally spaced data. It predicts values between presented data and is garenteed to remain coherent. In this case, the routine takes a 5 by 5 grid of averaged values and transforms it to obtain a smoothed map (values can be extrapolated for any [Mz;Rt] position.



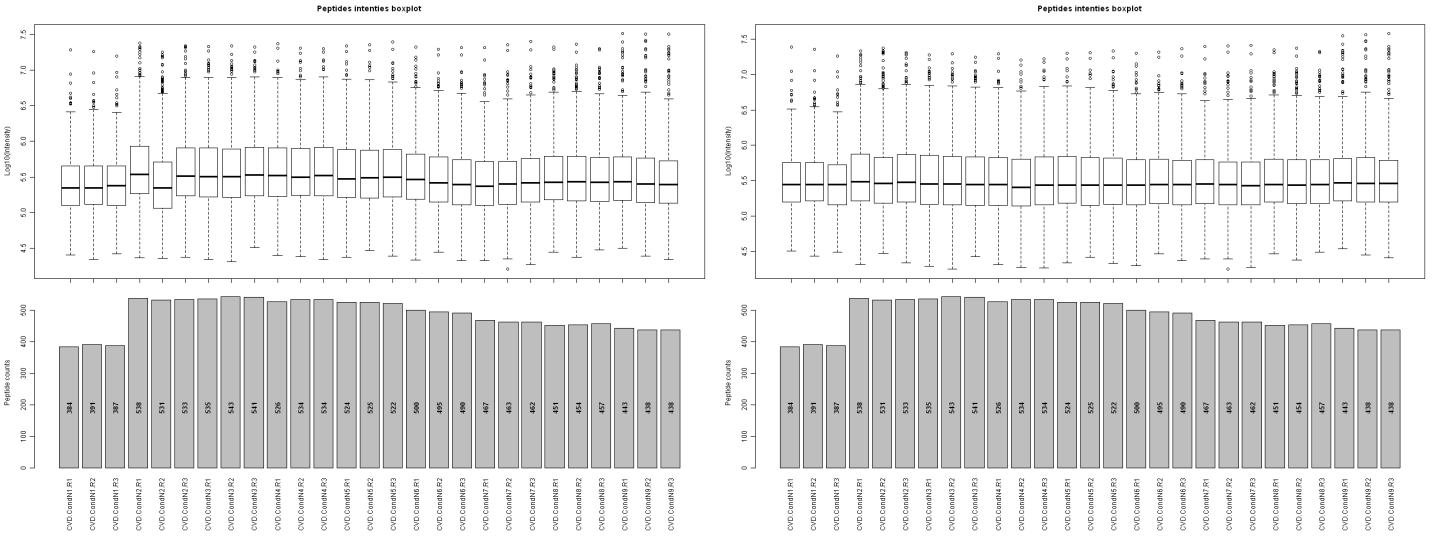
The SDFs are then processed one by one, and each ion is altered so that the Mz and retention time is moved based on its expected discrepancy (as defined by the bicubic spline). The previous picture represents the variations in alignment within a single map. Higher correction values are displayed in bright red while the darker color represents zones that needs less alignment.

*Thomas A. Foley, Weighted bicubic spline interpolation to rapidly varying data, ACM Transactions on Graphics (TOG), v.6 n.1, p.1-18,Jan.1987

Normalization

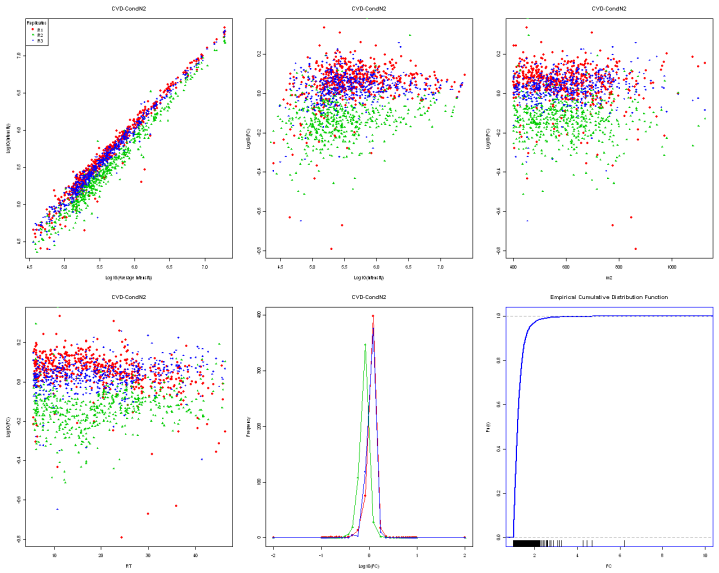
Normalization is used to compensate for hardware or manipulation errors by adjusting maps between replicates. This is another important step of ProteoProfile as it allows more precision in the definition of individual peptides.

In ProteoProfile, the ion maps themselves are not normalized: this process is only done for the clusters (and by definition for peptides and proteins). As a consequence, validation of clusters do not use normalized maps and the discrepancy between replicates can still be seen there. While very useful, normalization can be dangerous. In the rare cases where it could fail (for example, if there is an insufficient number of peptides in the map), non-normalized clusters entries are always listed. Graphs are also drawn to assess the quality of the normalization. Below are examples of some of the graphs. The left graph depicts the values for each of the 9 conditions before alignment while the top-right one displays the results. In the graph to the bottom-right, the green dots and lines show the lower intensities present in the second replicate of the second condition.



Normalization routine

To normalize, common peptides are chosen for their presence in multiple maps (at least 90% of them) and their overall stability (p value > 95%). One map at a time, the median intensity of these peptides is computed. By comparing each median to the average of the medians, a correction factor is affected to each map. This factor is finally used to bonify each corresponding peptide.



Peptide profiling

Clusters are a group of peptides with similar characteristics that where found at the same Mz/Rt region between maps of a project. To get a list all present peptides, one must merge together the clusters that share the same sequence and protein attachment, even when their charge and modifications differ. ProteoProfile uses the sum of intensities to merge clusters with the same peptide sequence that belong to the same protein.

If you are interested in observing certain modifications, this particular peptide profiling will not be useful. You should rather use the cluster intensities. But for protein profiling, this step is necessary.

Take as an example: This list of clusters ...

Mz	Rt	Sequence	Charge	Modification	Intensity	Protein
432.1	23	PPPPP	2	Phospho	10000	A
432.1	23	PPPPP	2		20000	A
864.2	25	PPPPP	4		30000	A
900.1	31	PPGGG	3		40000	A
1001.2	29	PGPGP	2	Phospho	50000	B

will be transformed into this list of peptides...

Peptide	Sequence	Intensity	Protein
1	PPPPP	60000	A
2	PPGGG	40000	A
3	PGPGP	50000	B

The Sum of Intensities

Described in this paper as one of the most precise methods, the Sum of Intensities is used to determine changes in protein abundance across conditions. It is a peptide to protein algorithm. This method was proven effective to compute the fold change of proteins by summing the intensities of all its peptides.

In ProteoProfile, it is used to **merge together peptides with the same sequence assigned to the same protein**. Peptides with different sequences are not merged at this stage, since the goal is to output the integral list of peptides. The relative intensity of the peptides is computed later, during protein profiling. In the literature, the sum of intensities is often associated with an outlier removal scheme. ProteoProfile does this step later, while regrouping peptides to compute protein fold changes.

Protein profiling is the final step, merging different peptides (belonging to the same protein) together. It is described in the following section.

Protein profiling

To compute the fold change of a protein, we use the variation of intensities between conditions. But giving an intensity count to proteins is irrelevant. Instead, we assign a number between 0 and 1 that describe the amount of representation of the protein within the different conditions. Called relative intensities, it is used to describe fold changes over multiple conditions (greater or equal to 2).

To compute protein's relative intensities, ProteoProfile assigns weights to the peptides composing the protein. Each peptide has a weight that is used to represent its potential to describe correctly the protein. Each peptide starts with a weight in proportion to its own intensity level (Log 10 of the average intensity of the peptide divided by 10). Based on the Weiszfeld's iteratively re-weighted least squares algorithm, this weight is multiplied by the closeness of the peptide to the protein's fold change, throught a series of iterations.

ProteoProfile removes outlying peptides by mesuring the negative impact of each peptide on the protein's fold change. Furthermore, if a peptide's fold change is less than 50% similar to the protein's fold change, it is not used in the description of the protein (its weight is set to zero). Note that this percentage changes over the iterations to ensure a sufficient number of peptides remain to describe the protein. (Each iteration, the threshold is set by multiplying the current threshold by 0.9, starting with 0.5)

Here is how ProteoProfile computes the profiles for protein *A*:

1. After clustering, normalize the intensities of the peptides
2. Merge peptides with the same sequence by doing the sum of their intensities
3. For each peptide *p* in protein *A*,
   - Compute *p*'s fold changes;
   - Compute *A*'s fold changes without *p* (see below)
   - Compute *d*: the difference (in %) between *b*'s and *A*'s fold changes
4. Amongst the peptides computed, pick the peptide *w* with the most different fold change compared to the protein
5. Set a score to peptide *w* to account for the difference in fold changes:
   - If more than 50% of conditions are totally different, *w*'s score = 0
   - Else, *w*'s score = (1 - *d*) * Log10(peptide intensity)
6. Repeat steps 3 to 5 until all peptide scores are computed, going from the worst peptide to the one that best describes the protein and always using the most up to date peptide scores

Protein relative intensities

Here is how to compute the intensities of protein *A*:

- Each peptide starts with a score corresponding to its total intensity (log10 of intensity, result divided by 10)
- Divide each peptide score by the sum of all the score (the sum of peptide scores must equal one)
- Sum the relative intensities of the peptides times their score to get the protein relative intensities

To get the intensity for a certain condition for a peptide, the intensities of each fractions are summed, while the replicates are averaged together. Thus, fold changes between conditions *a* and *b* are computed by this simple formula:

IF($a >= b$) fold change = a / b

IF ($a < b$) fold change = $-b / a$

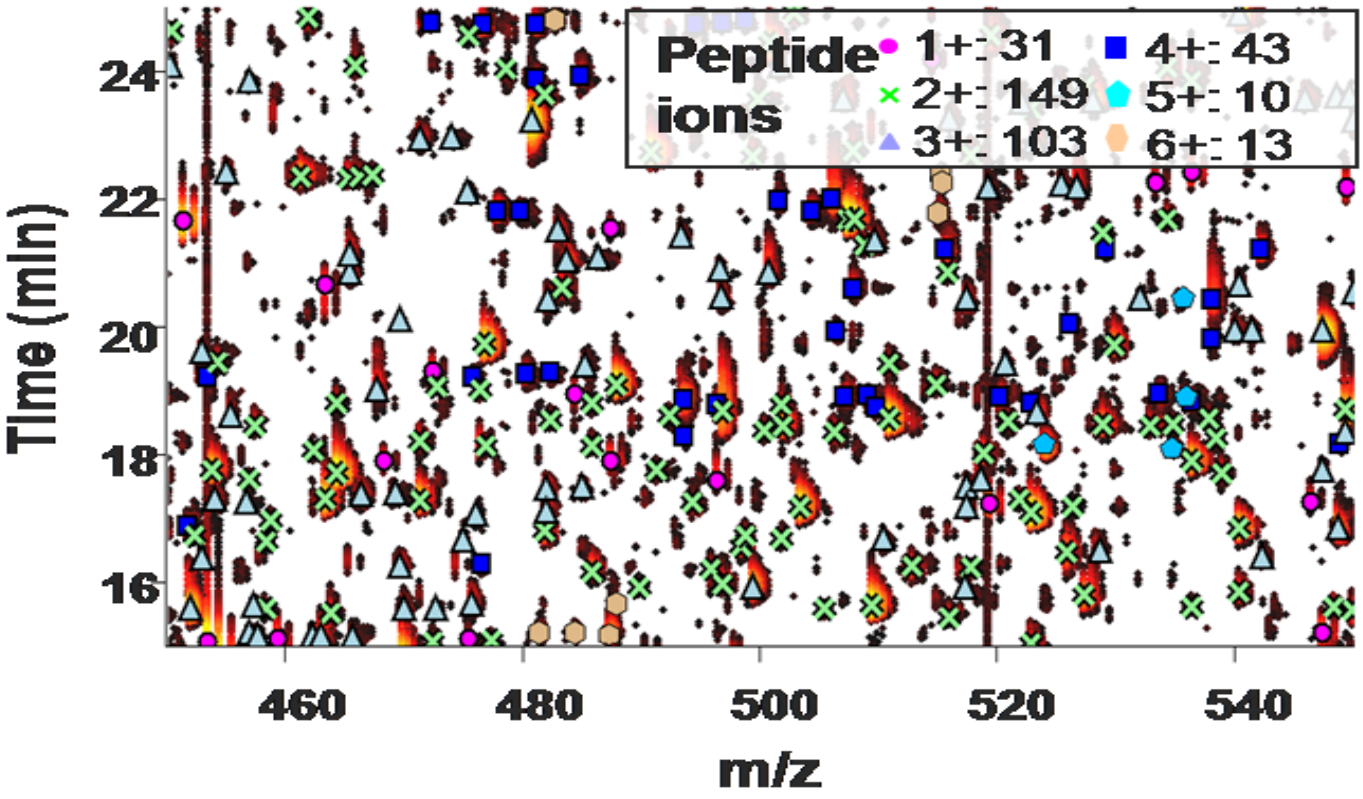
Results

The experiment

To validate the software's results, a special experiment was done on an Orbitrap XL mass spectrometer. Proteins were spiked in tryptic digest of J774 protein extract, scaling the spiked proteins on 9 conditions, 3 replicates. The "Results" section is an overview of the analysis done after the clustering and profiling by ProteoProfile, without manually validating the clusters.

Peptide extraction

The following image is a heat map (LC-MS file represented as an image) over which charge states for detected isotopes were added. The image is an actual snapshot of MassSense's interface, displaying the heat map from an LC-MS (Orbitrap) Tryptic digest of 8 proteins standards (200 fmols inj.).

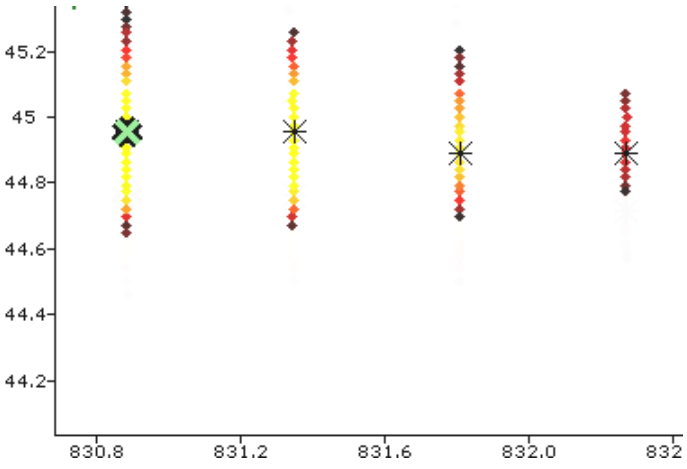


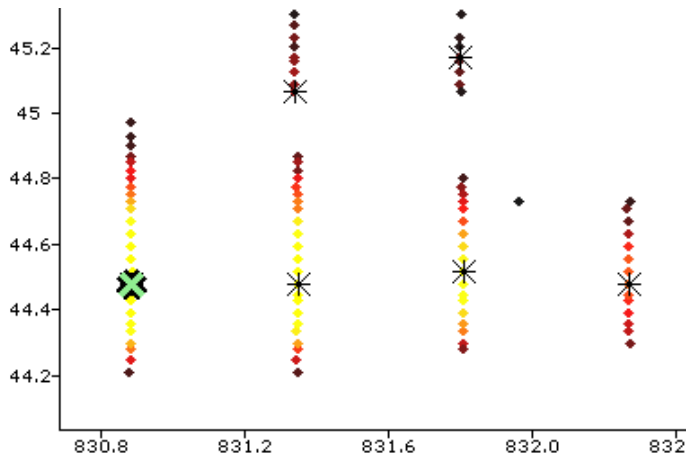
Results

Alignment

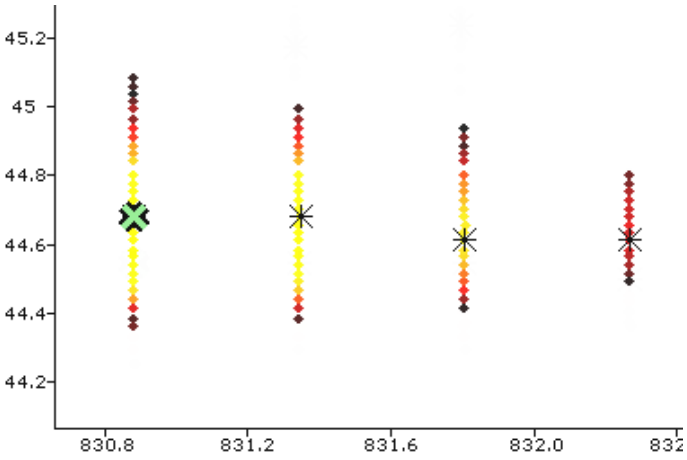
ProteoProfile aligns maps together during the clustering process. This step is essential to improve precision on non trivial clusters. The following images show the difference before and after alignement between the heat maps of two different conditions.

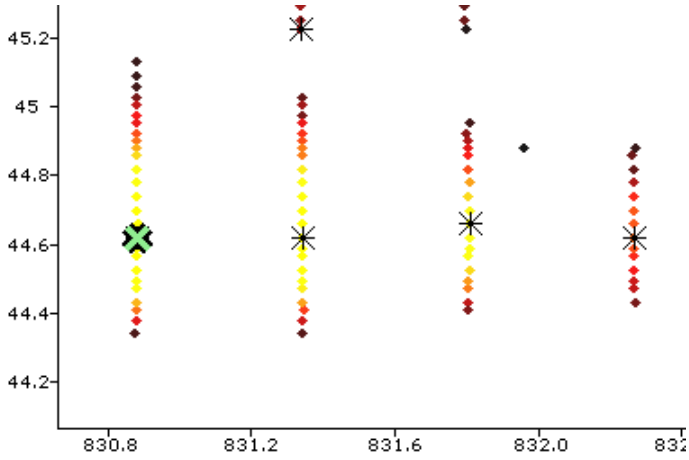
Before alignment:





After alignment :

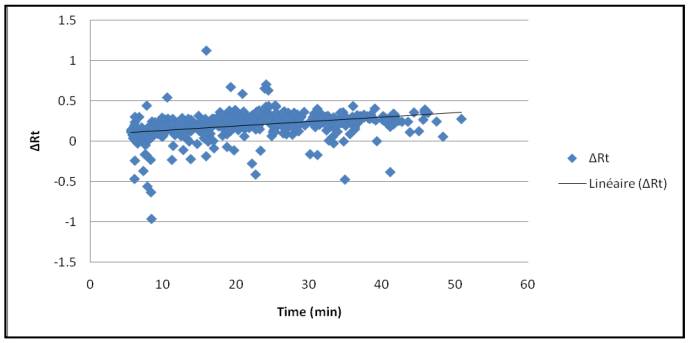




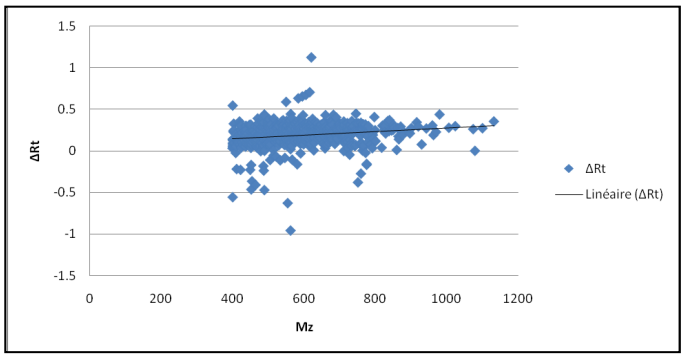
These two graphics show the effects of alignment on the retention time and Mz of two ion maps. The following section gives a broader view of the importance of the alignment.

Alignment example

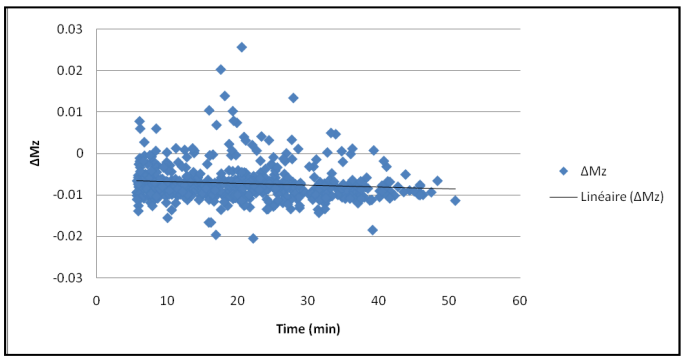
The following graphs show the difference between the average cluster values and condition 3, replicate 1 of the spiked experiment **before alignment**).



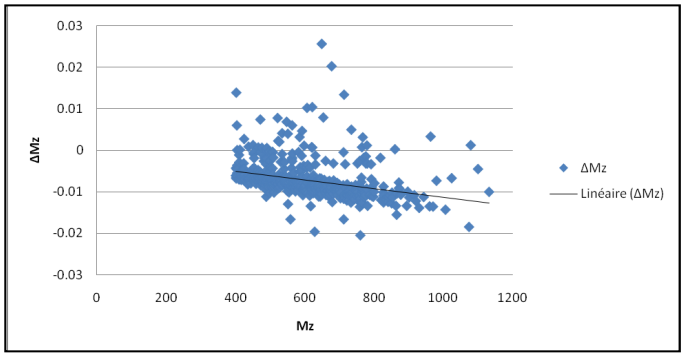
Variation over time between peptides and their corresponding cluster's retention time



Variation over Mz between peptide's and their corresponding cluster's retention time

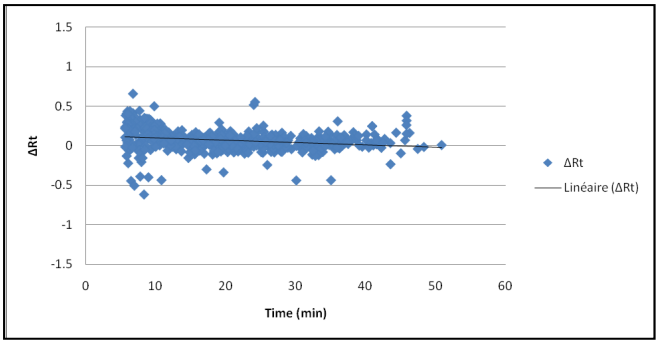


Variation over time between peptides and their corresponding clusters Mz values

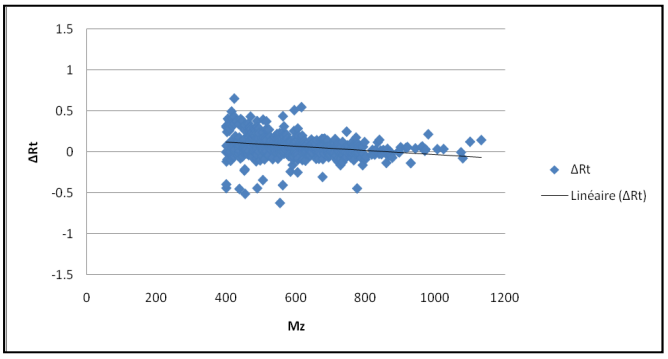


Variation over Mz between peptides and their corresponding clusters Mz values

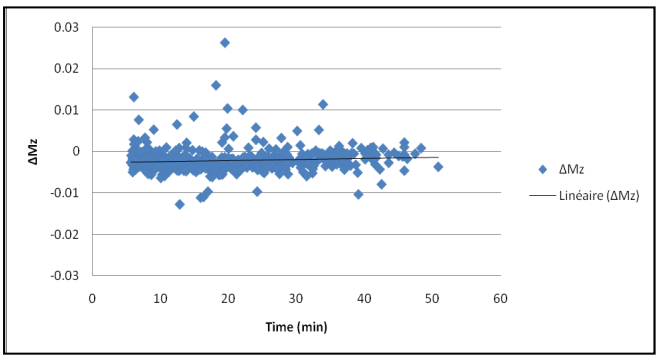
The graphs below show the effects of the alignment on the difference between the average cluster values and condition 3, replicate 1. Note that the content of the cluster changes when the maps are **aligned**.



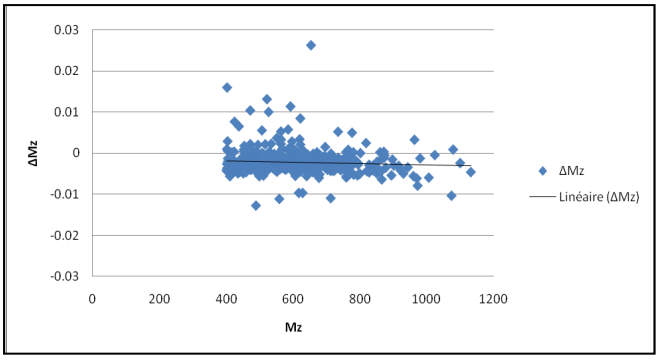
Variation over time between peptides and their corresponding cluster's retention time



Variation over Mz between peptide's and their corresponding cluster's retention time



Variation over time between peptides and their corresponding clusters Mz values



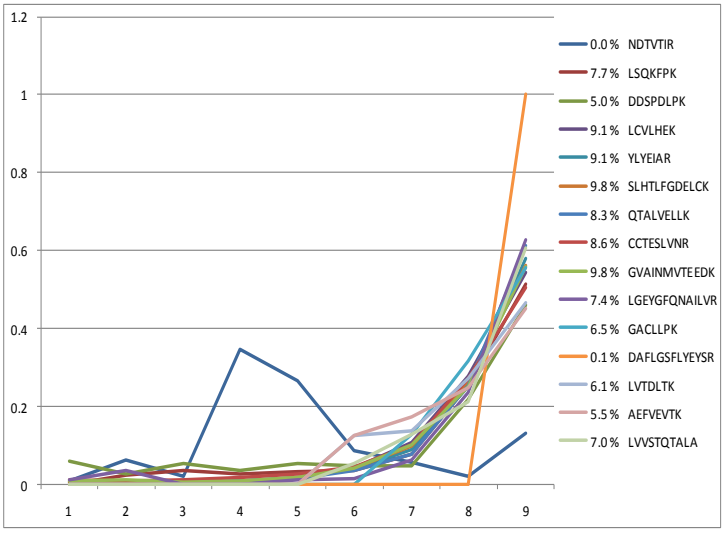
Variation over Mz between peptides and their corresponding clusters Mz values

Results

Peptides

To validate the software's results, a special experiment was done on an Orbitrap XL mass spectrometer. Proteins were spiked in tryptic digest of J774 protein extract, scaling the spiked proteins on 9 conditions, 3 replicates. The following is an overview of the analysis done after the clustering and profiling by ProteoProfile (without manually validating the clusters), for one of the spiked protein : Albumin [Bos taurus].

In the following graph and table, the peptides found and associated to the protein Albumin [Bos taurus] are listed. The horizontal axis of the graph lists the various conditions while the vertical axis denotes the relative abundance of each peptide. The percentage following the sequence of the peptide represents the weight of the peptide on the description of the protein. It is interesting to note that in this experiment, the first peptide (NDTVTIR) did not qualify as a descriptor for albumin's abundance.



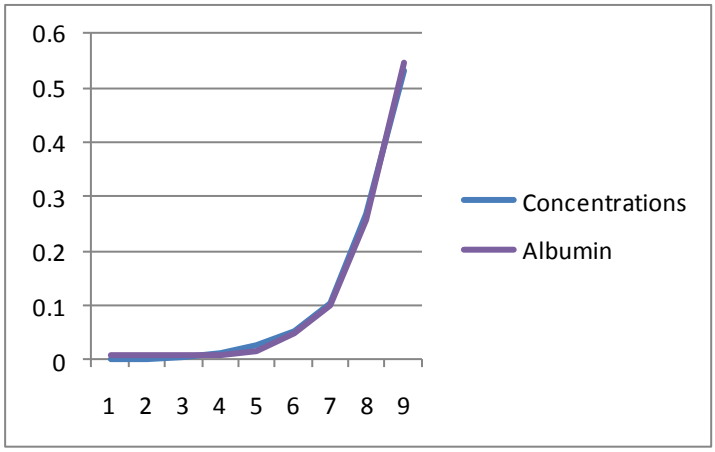
	Weight	Condition 1	Condition 2	Condition 3	Condition 4	Condition 5	Condition 6	Condition 7	Condition 8	Condition 9
Albumin Concentrations		**0.0%**	**0.0%**	**0.5%**	**1.3%**	**2.7%**	**5.3%**	**10.3%**	**26.7%**	**53.2%**
NDTVTIR	0.0 %	1.0%	6.2%	2.2%	34.6%	26.6%	8.7%	5.6%	2.1%	13.0%
LSQKFPK	7.8 %	0.0%	2.3%	3.7%	2.8%	3.2%	3.7%	7.8%	25.4%	51.3%
DDSPDLPK	5.3 %	5.9%	2.8%	5.3%	3.6%	5.2%	4.7%	4.7%	21.8%	46.0%
LCVLHEK	8.7 %	0.2%	0.2%	0.3%	0.8%	1.4%	4.1%	10.6%	27.9%	54.5%
YLYEIAR	8.8 %	0.6%	0.5%	0.5%	0.9%	1.7%	3.9%	8.9%	25.2%	57.9%
SLHTLFGDELCK	9.4 %	0.4%	0.8%	0.2%	0.7%	1.8%	4.0%	9.8%	26.3%	56.0%
QTALVELLK	7.8 %	0.0%	0.0%	0.0%	0.8%	1.3%	3.6%	7.6%	25.3%	61.4%
CCTESLVNR	8.6 %	1.1%	0.9%	1.1%	1.8%	2.7%	4.5%	10.2%	27.2%	50.5%
GVAINMVTEEDK	9.5 %	0.8%	1.0%	0.6%	0.9%	1.6%	4.1%	10.1%	25.0%	55.9%
LGEYGFQNAILVR	7.2 %	1.1%	3.7%	0.0%	0.0%	1.1%	1.5%	6.3%	23.7%	62.7%
GACLLPK	6.6 %	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	12.8%	31.5%	55.7%
DAFLGSFLYEYSR	1.4 %	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	100.0%
LVTDLTK	6.2 %	0.0%	0.0%	0.0%	0.0%	0.0%	12.6%	13.7%	27.2%	46.5%
AEFVEVTK	5.5 %	0.0%	0.0%	0.0%	0.0%	0.0%	12.6%	17.2%	24.9%	45.2%
LVVSTQTALA	7.1 %	0.0%	0.0%	0.0%	0.0%	0.0%	5.3%	12.8%	21.2%	60.6%

The weight affected to each peptide corresponds to the percentage of the protein this peptide is expected to define. The result of the profiling of the Albumin protein is presented in the next page.

Results

Proteins

The following graph and table are a comparison of the relative intensities of Albumin compared to the predicted values deduced from the detected peptides (see previous page). Predicted values are based on the known concentrations of Albumin in the 9 conditions. The two curves share a 97.0% similarity : the sum of the differences between relative intensity for each condition is exactly 6%.



	Condition 1	Condition 2	Condition 3	Condition 4	Condition 5	Condition 6	Condition 7	Condition 8	Condition 9
Albumin (Concentrations)	0.0%	0.0%	0.5%	1.3%	2.7%	5.3%	10.3%	26.7%	53.2%
Albumin (ProteoProfile)	0.5%	0.7%	0.4%	0.7%	1.2%	4.9%	10.2%	26.4%	55.0%
Difference	-0.5%	-0.7%	0.1%	0.6%	1.5%	0.4%	0.1%	0.3%	-1.8%

Comparing protein profiling technics

The following table compares the profiling of Albumin from ProteoProfile to other methods. The first row, dubbed “concentration” lists the expected relative intensities of the protein for each condition. The last column shows the proximity (or closeness) between the method and the expected values.

	Cond 1	Cond 2	Cond 3	Cond 4	Cond 5	Cond 6	Cond 7	Cond 8	Cond 9	Closeness
Concentrations	0.0%	0.0%	0.5%	1.3%	2.7%	5.3%	10.3%	26.7%	53.2%	100.0%
Sum of intensities (Unaligned)	0.9%	1.2%	1.0%	1.5%	2.6%	5.3%	11.1%	29.6%	46.8%	93.6%
SOF + Manual outlier removal (Unaligned)	1.0%	1.3%	1.0%	1.2%	1.9%	4.1%	9.3%	24.8%	55.3%	95.1%
ProteoProfile (Unaligned)	1.1%	1.7%	1.0%	1.2%	2.1%	5.6%	10.1%	23.5%	53.6%	95.9%
ProteoProfile	0.5%	0.7%	0.4%	0.7%	1.2%	4.9%	10.2%	26.4%	55.0%	97.0%