Modules developed for PTM nnotation workflow:

FDRFilterer

FDRFilterer filters the input file by peak, global and local FDR according to the conditions set by the user, in the configuration.

FDRFilterer needs the following input files:

- .tsv file
- Configuration file (.ini):
    - FDRFilterer parameters:
        - *GlobalThres*: Global FDR threshold.
        - *PeakThres*: Peak FDR threshold.
        - *LocalThres:* Local FDR threshold.
        - *GlobalFDR_column_name:* name of the column containing Global FDR values.
        - *PeakFDR_column_name:* name of the column containing Peak FDR values.
        - *LocalFDR_column_name:* name of the column containing Peak Local values.
        - *Label_column_name:* name of the column that indicates label.
        - *decoys_naming:* parameter that indicates how decoys are named.

And delivers two output files:

- FDRFilterer output (default suffix: “FDRFiltered”)
- A log file (default suffix: "FDRFiltered_logFile")

DM0Solver

DM0Solver is a module that detects if a modified peptide has a Δmass belonging to a list provided by the user (Table 1), for that purpose absolute error is calculated. In such a case, the Δmass is appended at the end of the clean sequence (DM0Sequence output column) and the corresponding label is added in an additional column named DM0Label. If Δmass does not belong to the list, the module passes the modified sequence without any modification to the output columns.

DM0Solver needs needs the following input files:

- .tsv file
- Apex_file: is a file, generated by a SHIFS (https://github.com/CNIC-Proteomics/SHIFTS-4/releases/tag/v0.3.1) module that contains the peak Δmasses.
- Configuration file:
    - DM0Solver parameters:
        - *Absolute_Error:* absolute error calculated between each of Δmasses of the PSMs and the Δmass of the list provided by the user.
        - *PeakAssignation_Column_name:* column that contain the peak/orphan labels.
        - *DM_column_name:* name of the column that contains Δmass.
        - *Sequence_column_name:* peptide sequence with Δmass column name.
        - *DM0Sequence_output_column_name:* column name of the output where the new sequence is annotated.
        - *DM0Label_output_column_name:* column name of the output where the chosen label is annotated.
        - *DM0Label_ppm_output_column_name:* column name of the output where the calculated error is annotated.
        - *output_file_suffix:* chosen suffix for output file.

    - DM0Solver list (list of masses with its corresponding label):

An example of input list for DM0Solver configuration file can be observed in Table 1. The first column shows examples of labels (DM0 for zero Δmass, or DM0;C13 for carbon 13 Δmass), while the second column contains de mass in Da of each label. DM0Solver uses this list to assign a label, as long as it meets the error threshold.

Table 1. Example of input list for DM0Solver configuration file:

Label	Mass (Da)
Δmass	0
Δmass;C13	1.003355
Na adduct	21.981943
Ammonium adduct	17.026549
H20 loss	-18.010565

It delivers two output files:

- DM0Solver output (default suffix: “DM0S”)
    - New columns:
        - *DM0Sequence_output_column_name:* sequence corrected by DM0Solver.
        - *DM0Label_output_column_name:* selected label of the list provided by the user.
        - DM0Label_error_output_column_name: absolute error resulting from the selection of the label that appears in DM0Label_output_columnn_name.
- A log file (default suffix: "DS_logFile")

TrunkSolver

TrunkSolver is a module developed with the aim of detecting whether the Δmass, in a modified peptide, may be explained by a truncation or tryptic cut of a non-modified peptide inside the sequence of the corresponding protein, and its possible combination with the presence of a Δmass belonging to a list provided by the user. In such a case, the Δmass is appended at the end of the clean sequence (TrunkSequence output column), the corresponding label is added in an additional column named TrunkLabel, and recalculated Δmass (TrunkDM output column). If TrunkSolver is unable to explain the Δmass by a truncation, then it passes the modified sequence and its original Deltamass without any modification to the output columns. In both cases five extra columns (New_DM, New_Theo_mh, Trunk_stats_mods) will be created for subsequent PeakAssignatior (assigns every PSM to the closest Δmass peak found in the provided list and identify it as either belonging to that peak, or as an orphan) execution, if it is desired. To do so, the relative error will be calculated:

$$Relative\ error(ppm) = \text{abs}(\frac{(Theoretical\_mh+Label\_mass)-Exp\_mh)*1000000}{Theoretical\_mh+\ Label\_mass})$$

TrunkSolver needs one input file:

- .tsv file
- .fasta file
- MassMod configuration file
- Configuration file:

    - TrunkSolver parameters:

        - *Relative_Error_ppm:* relative error (ppm) allowed.

- *Exp_mh_column_name:* calibrated experimental mh column name.
- *Theo_mh_column_name:* theoretical mh column name.
- *Sequence_column_name:* sequence with Δmass colum name.
- *Calibrated_Delta_MH_column_name:* calibrated Δmass mh column name.
- *MasterProtein_column_name:* Master Protein accession code column name.
- *static_modifications_column_name:* static modifications column name.
- *Decnum:* decimals points required in TrunkSequence column.
- *X:* parameter that indicates the extension (left and right from the original residue) that TrunkSolver is allowed to extend from the original Δmass position residue.
- *New_Deltamass_output_column_name:* new Δmass column name.
- *New_Theo_mh_output_column_name:* new theoretical mh column name.
- *TrunkSequence_output_column_name:* column name of the output where the chosen sequence is annotated.
- *TrunkDM_output_column_name:* column name of the output where the recalculated Δmass is annotated, considering the label.
- *TrunkLabel_output_column_name:* column name of the output where the chosen label is annotated.
- *TrunkLabel_ppm_output_column_name:* column name of the output where the calculated error in ppm is annotated.
- *Static_modifications_position_output_column_name:* column name of the output where the new fix modifications positions are annotated.
- *output_file_suffix:* chosen suffix for output file.
- Missing_cleavages_output_column_name: output column name for the number of missing cleavages.
- *Truncation_output_column_name*: output column name for truncations to be annotated.
- *TrunkPlainPeptide_output_column_name*: column name of the output where the chosen peptide without Δmass the is annotated.

- TrunkSolver list (list of masses with its corresponding label)

An example of input list for TrunkSolver configuration file can be observed in Table 2. The first column shows examples of labels, TMT: Δmass of tandem mass tag, 2TMT; two Δmasses of tandem mass tags; -TMT: minus Δmass of tandem mass tag ,-2TMT: minus two Δmasses of tandem mass tags. The second column contains de mass in Da of each label. TrunkSolver uses this list to assign a label, as long as it meets the error threshold.

Table 2. Example of input list for TrunkSolver configuration file

Label	Mass (Da)
TMT	229.162932
2TMT	458.325864
-TMT	-229.162932
-2TMT	-458.325864

And delivers two output files:

- TrunkSolver output (default suffix: "_TS")

- New columns:
    - *TrunkSequence_output_column_name:* output column name in which reassigned sequence is annotated.
    - *TrunkDM_output_column_name:* output column name that contains recalculated Δmass considering the labels.
    - *TrunkLabel_output_column_name:* output column name in which the selected label is saved. If the Δmass corresponds to a combination between one of the labels in the configuration file and a cut, the type of cut also will be noted (Table 3). The label will be TrypticCut if the choice of label involves a tryptic cut of the sequence, or a Truncation if it involves a non-tryptic cut.
    - *TrunkLabel_ppm_output_column_name:* output column in which the error, that is obtained selecting the label is annotated.
    - *New_Theo_mh_output_column_name:* output column name in which the recalculated theoretical mass is annotated.
    - *New_Deltamass_output_column_name:* output column name in which the recalculated Δmass is saved.
    - *Static_modifications_position_output_column_name:* output column name in which the new static modifications positions are saved, since with the changes in the sequence could vary.
    - *TrunkPlainPeptide_output_column_name*: column in which peptide without the Δmass is annotated
    - Missing_cleavages_output_column_name: number of missing cleavages wil be annotated.
    - *Truncation_output_column_name*: column in which truncations will be annotated as "1" (0;No-truncation , 1; Truncation)

- A log file (default suffix: "TS_logFile")

Table 3. Example of TrunSolver ouputfile columns, considering input file.

Input file columns		
DM0Sequence	**Thoeretical_mh**	**Modifications**
LGEHNLDVLEGNEQFLNA[-499.331973]AK	2669.42	1_S_0.0000_N
GTFASLS[229.141559]ELHCDK	1923.00	1_S_0.0000_N, 11_S_57.021464, 13_S_229.162932
TLER[-499.281621]EACLLNANK	1990.115	1_S_0.0000_N,7_S_ 57.021464, 13_S_229.162932

Output file colums						
TrunkSequence	**TrunkDM**	**TrunkLabel**	**TrunkLabel_ppm**	**New_theoretical_mh**	**New_DM**	**Static_modifications_position**
LGEHNLDVLEGNEQFLN_-0.00015	-0.00015	Truncation; DM0	0.06	2170.09	-0.00015	1_L_229.162932_N
GTFASLSELHCDK_229.141	229.141	+TMT	9.93	2152.167	-0.02	1_G_229.162932_N,11_S_57.021464, 13_S_229.162932
EACLLNANK_-0.006	-0.006	Trypticcut; DM0	4.14	1490.84012	-0.006	1_E_229.162932_N,3_S_57.021464, 9_S_229.162932

SiteListMaker

This module was developed with the objective of acquiring a comprehensive overview of identified modifications within the proteome under analysis. This includes the frequency of each modification, as well as the precise residues where these modifications are localized. Three tables are generated as a result. The first table presents the raw frequencies of each modification (Δmass), detailing how frequently Δmass X occurs on amino acid Y at positions [-5,5], with zero denoting the position where the modification is specifically localized. The second table mirrors the first but incorporates a correction to eliminate background noise, according to the following equation:

$$\text{Expected} = \text{Observed} - (\Sigma \text{Position}_i * \Sigma \text{Residue}_j)/\text{Total of } DM_y$$

From each observed point is subtracted the division of the sum of the frequencies for that position (i) multiply by the sum of the frequency of that residue (j) across all positions, divided by the total sum of frequencies for that Δmass (y). Finally, it generates a final table containing only the information of the Δmass, its frequency, and the clean frequency of each residue but only at position zero, that is, when they are modified.

The third table exclusively focuses on the frequencies of modifications at residues in position zero, signifying instances where modifications are situated on that particular residue. The final table provides the user with a comprehensive insight into their proteome, elucidating the specific residues associated with each modification. This information proves invaluable for another module within this study, namely SiteSolver.

SiteListMaker needs as input files:

- .tsv file
- Configuration file:
    - SiteListMaker parameters:
        - Relative_Error_ppm: relative error (ppm) allowed.
        - *Theo_mh_column_name:* theoretical mh column name.
        - *Sequence_column_name:* sequence with Δmass colum name.
        - *Calibrated_Delta_MH_column_name:* calibrated Δmass mh column name.
        - *PeakAssignation_column_name:* output column that will contain the peak/orphan labels.
        - *PeakNaming*: parameter that indicates how decoys are named.
        - *Frequency_Table*: name for the output file of frequency table.
        - *Clean_Frequency_Table*: name for the output file of clean frequency table.
        - *Clean_P0_Frequency_Table*: name for the output file of clean position-zero frequency table.

And delivers four output files:

- Three tables:
    - *Frequency_Table*: contains the unprocessed occurrences of each modification, delineating the frequency with which Δmass X manifests on amino acid Y at positions [-5,5], with zero indicating the specific localization of the modification.
    - *Clean_Frequency_Table*: encompasses the unprocessed occurrences of each modification, delineating how frequently Δmass X manifests on amino acid Y at positions

[-5,5], with zero indicating the specific localization of the modification. It also integrates a correction mechanism to eliminate extraneous background noise.

- *Clean_P0_Frequency_Table*: contains the frequencies of modifications at residues in position zero, indicating instances where modifications are located on that particular residue.

- A log file (default suffix: "SLM_logFile")

SiteSolver

This module detects if a modified peptide has its Δmass in an incorrect position. In such a case, Δmass location within the sequence is corrected in "SiteSequence" column. If the module does not find any possible position, it passes the modified sequence without any modification to the output column. Foremost, it is ascertained whether the amino acid position, in which the Δmass is originally located, is prohibited. This is tested up using a list based on SiteListMaker. If the amino acid is allowed, for that Δmass, it passes the modified sequence without any modification. On condition that the amino acid is prohibited, SiteSolver will analyze the contiguous amino acids to that position. Provided that only one of them is prohibited the sequence will be corrected by assigning the Δmass to the amino acid that is allowed, according to SiteListMaker, based on the frequency. If both are allowed, for that Δmass, SiteSolver will take in to account the order of the appearance of the residues. The process is repeated until the number of positions to be analyzed, on each side, exceeds the maximum (X parameter). To do so, the relative error will be calculated:

$$Relative\ error(ppm) = \text{abs}(\frac{(Label_{mass} + \Delta mass\_user\_selection) - Exp\_mh) * 1000000}{Theoretical\_mh + \ Label\_mass})$$

SiteSolver needs three input files:

- .tsv file
- UserList.txt file (user)

  The user file will consist of two columns, as can be observed in Table 4. In the first column, will appear all the Δmass values that we want to relocate, and in the second column, all the permissible residues for that Δmass will be listed in descending order of occurrence of the Δmass-residue combination in the proteome under study. All this information will have been provided earlier through SiteListMaker.

Table 4. Example of input user list for SiteSolver

Δmass	Residue
15.99492	W, P, Y
14.01565	D, E
3.994915	W

- Configuration file:
  - SiteSolver parameters:
    - *Theo_mh_column_name:* theoretical mh column name.
    - *Relative_Error_ppm:* relative error (ppm).

- *Sequence_column_name:* peptide sequence with Δmass column name.
- *cal_Dm_mh_column_name:* calibrated Δmass name.
- *x:* parameter that indicates the extension (left and right from the original residue) of the amino acids wanted to be analyse.
- *SiteSequence_column_name:* column name of the output where the sequence with the correct Δmass position is annotated.
- *SiteCorrection_column_name:* column name of the output where correction site is annotated.
- *SiteDM_column_name:* column name of the output where selected Δmass is annotated.
- *SiteDMError_ppm_column_name:* column name of the output where the error of the selected Δmass is annotated.
- *Output_file_suffix: chosen suffix for output file*.

And delivers two output files:

- SiteSolver output (default suffix: "SS"). New columns:

  - *SiteSequence_column_name:* sequence with the Δmass positioned in the correct residue.
  - *SiteCorrection_column_name:* Amino acid change. It appears the previous residue in which Δmass was located and the residue in which the Δmass is now relocated.
  - *SiteDM_column_name:* column name of the output where selected Δmass from de user list is annotated.
  - *SiteDMError_ppm_column_name:* column name of the output where the error of the selected Δmass is annotated.

- A log file (default suffix: " SS_logFile")

PTM quantification modules:

PDMTableMaker

PDMTableMaker computes several indispensable parameters for the subsequent program execution and the ensuing implementation of newly developed quantification workflows crucial for the accurate interpretation of data.

PDMTableMaker input files:

- .tsv file
- .fasta file
- Configuration file:

  - PDMTableMaker parameters):

    - *Sequence_column_name:* sequence with Δmass colum name.
    - *DM_column_name:* Δmass column name.
    - *Theo_mh_column_name:* theoretical mh column name.

- *MasterProtein_column_name:* Master Protein accession code column name.
- *output_file_suffix:* chosen suffix for output file.
- *Missing_Cleavage_column_name:* missing cleavage number column name.
- *Truncated_column_name:* column name in which truncations are annotated column name.
- Score_parameter: score parameter to select best scan identifier. 1; if the best score is the highest and 0; if the best score is the lowest.
- Score_column_name: column in which scores for the best scan identifier are annotated.
- ScanID_column_name: column name in which best scan identifier is annotated.

- PDMTableMaker conditions:

  - *number_of_conditions*: Number of conditions
  - *Condition*$_i$: Column name of condition$_i$ ($_i$: condition numeration)
  - *Value*$_i$: Chosen value for condition$_i$ ($_i$: value numeration)

    In this section, as many conditions as desired can be specified. It is important to note that the equality will be established between the specified condition (condition$_i$) and its corresponding value (value$_i$). The condition should serve as a header in the input file, indicating the parameter to be associated with the desired value.

And delivers two output files:

- PDMTableMaker output (default sffix: "PDM"),that contains the following columns:

  - *p:* peptide
  - *pdm*: peptidoform defined by peptide sequence, Δmass and position. Ex: ABCD[xxx]EFGHK.
  - *pd:* ABCEDEFGHK:XXX (includes a set of pdm elements, do not confound with ABCEDEFGHK_XXX, which is a unmodified pdm, not a pd)
  - *d:* modification (Δmass)
  - *m:* position in peptide (C-Terminus=-1, N-Terminus=0)
  - *l:* position in peptide, from right to left (C-Terminus=0, N-Terminu=-1)
  - *n:* position in protein. This column will contain all n, if the peptide appears several times in the sequence.
  - *first_n*: position in protein (number) (C-Terminus =length peptide plus 1, N-Terminus = 0). This column contains just the first n, first apparition of the peptide in protein sequence.
  - *b:* position in protein of the first residue of the peptide. This column will contain all b.
  - *First_b*: position in protein of the first residue of the peptide. This column contains the first b.
  - *e*: position in protein of the last residue of the peptide. This column will contain all e.
  - *a:* modified residue.
  - *q:* protein.
  - *M:* (razor m, property of a pd) Is the m corresponding to the pdm with highest original PSM frequency in a pdm table.
  - *L:* (razor l, property of a pd) is the l which corresponds with the razor m.

- *N:* (razor n, property of a pd) Is the n which corresponds with the razor m (NT=0).
- *A:* (razor a, property of a pd) Is the a which corresponds with the razor m.
- *qdna:* information of q,d,n,a (ex: HPT: Δmass:300:M).
- *qdNA:* razor qdna, property of a pd.
- *Theo_mh:* theoretical mh.
- *ScanFreq:* count of PSMs that each pdm has.
- *k*: minimum cluster of qna elements that are contained in overlapping peptides.
- *c*: minimum cluster of peptide elements that are contained in overlapping peptides.
- *qFreq*: count of PSMs that each protein has.
- *pFreq*: count of PSMs that each peptide has.
- *qK*: q and k concatenation.
- *qc*: q and c concatenation.
- *qKFreq*: count of PSMs that each qk has.
- *qcFreq*: count of PSMs that each qc has.
- *qdnaFreq*: count of PSMs that each qdna has.
- *qnaFreq*: count of PSMs that each qna has.
- *BesScanID*: column that contains the identifier of the best scored scan
- *MissingCleavages*: column that contains the number of missing cleavages preciously calculates by trunkSolver of each pdm
- *Truncation_output_column_name*: That will contain 0, if the is no truncation and 1 if the is a truncation in the pdm. This information was previously obtained by TrunkSolver.

- A log file (default suffix: " PDM_logFile")

GroupMaker

GroupMaker is a module developed with the purpose of grouping information to enhance the interpretation of results in subsequent quantification. In this manner, GroupMaker forms groups as long as they meet all the specified criteria outlined in an input table created by the user and based on SiteListMaker output table. If one of the conditions is numerical, a relative error will be computed. If it surpasses the user-defined error threshold, the condition will be deemed valid. To do so, the relative error will be calculated:

$$Relative\ error(ppm) = \text{abs}(\frac{(\Delta\text{mass\_user\_selection} - \Delta\text{mass\_of\_pdm}) * 1000000}{Theoretical\_mh + \ \Delta\text{mass\_user\_selection}})$$

GroupMaker needs three input files:

- .tsv file
- Configuration file:
    - GroupMaker parameters:
        - *Relative_Error: elative error.*
        - *Output_file_suffix*: chosen suffix for output file.
        - *Theo_mh_column_name*: theoretical mh column name.
        - *Decnum:* decimals points required if the group is numerical.

- GroupMaker user input table. This table will be composed of as many columns as there are desired conditions. The column headers of this table must match those of the input .tsv file. If the information in each column matches, a group will be created; otherwise, the group column will remain empty. In the case where a condition is numerical, it will be checked against the user-defined threshold as absolute error. The last column appearing in this table will be the group column. In the output file, a new column with the same name as the last column in this input table will be created, and it will record the different groups.

It delivers two output files:

- Groupmaker output (default suffix: “GM”). New columns:
  - *Group_output column:* this column contains the group.
- A log file (default suffix: "GM_logFile")

Joiner

This module joins the labels of the columns indicated by the user without repetitions.

Joiner needs two input files:

- .tsv file
- Configuration file (.ini): There is a default .ini in the “config” folder:
  - Joiner parameters:
    - *Output_column_name: output column name in which all labels will be joined*
    - *Output_file_suffix:* chosen suffix for output fil
    - *Decnum:* decimals points required if Δmass is one of the parameters
    - *group_column_name:* column name that contains the group name
    - *Non_modified_name*: parameter that indicates how unmodified peptidoforms are named
  - Joiner columns. Use ";" to select the column Joiner must join if the first column is empty
- Example:

1 = p

2 = g;d

3 = m

It delivers two output files:

- Joiner output (default suffix: “Joined”). New columns:

        - *Output_column_name*: this column contains all the labels of the columns indicated by the user, without repetitions.

- A log file (default suffix: "Joined_logFile")

<u>Visualization program</u>: PTMap

PTMap is a tool developed with the aim of visualizing, interpreting, and comparing the proteins PTMs of s. This module represents, as many maps as proteins for which any integration meets the threshold established by the user. Each map illustrates the change between one condition and another based on the p-value of all calculated integrations, on the y-axis.

$$-LPS = -log_2(p\text{-}value) * sign\ (Condition_2\text{-}Condicion_1)$$

On the x-axis, the position of each residue of the protein is represented. Specific modifications and hypermodified zones are represented by circles, while partial and total digestion, and zonal changes are represented by rectangles. The size of these markers depends on the frequency of each parameter, in a relative scale depending on the maximum and minimum PSMs frequency of each type of modification. These graphs offer interactivity and additionally enable the visualization of parameter frequency, modified residue, and the specific group of each Δmass.

PTMap needs two input files:

- .tsv file with limma p-values and -LPS calculated for each integration
- Configuration file (.ini): There is a default .ini in the “config” folder:

    - PTMap parameters:

        - *pgm_column_name:* pgm column name
        - *g_column_nam:* group column name
        - *a_column_name:* modified residue column name
        - *n_column_name:* modified position within the protein column name
        - *e_column_name:* position of the last residue of the peptide within the protein column name
        - *p_column_name:* peptide column name
        - *q_column_name:* protein column name
        - *d_column_name:* Δmass column name
        - *qc_column_name:* cluster containing overlapping peptides column name
        - *pFreq_column_name:* peptide frequency column name
        - *qcFreq_column_name:* qc frequency column name
        - *pgmFreq_column_name:* pgm frequency column name
        - *first_b_column_name*: position in protein of the first residue of the peptide
        - *description_column_name:* description of the protein column name
        - *Missing_Cleavages_column_name:* number of missing cleavages column name
        - *LPS_p2qc_column_name:* - LPS of p2qc integration column name
        - *LPS_qc2q_column_name:* -LPS of qc2q integration column name
        - *LPS_pgm2p_column_name:* -LPS of pgm2p integration column name
        - *LPS_pgm2p_NM_column_name:* -LPS of pgm2p integration only with the unmodified pepetides columna name
        - *Filter_pgm2p_NM_column_name:* filtering column (FDR,pvalue) of pgm2p integration corrected by NMCompare

- *Filter_pgm2p_column_name:* filtering column (FDR,pvalue) of pgm2p integration calculated only with NM
- *Filter_p2qc_column_name:* filtering column (FDR,pvalue) of p2qc integration
- *Filter_qc2q_column_name:* filtering column (FDR,pvalue) of qc2q integration
- *threshold_pgm2p_NM:* threshold of filtering column (pgm2p integration only with unmodified peptides)
- *threshold_pgm2p:* threshold of filtering column (pgm2p integration)
- *threshold_p2qc:* threshold of filtering column (p2qc integration)
- *threshold_qc2q:* threshold of filtering column (qc2q integration)
- *NM:* non modified peptides group
- *path_plots_with_threshold:* folder in which filtered PTM Maps will be saved
- *path_plots_Without_threshold:* folder in which PTM maps, without filters, will be saved

PTMap delivers two output folders. Both contain only maps of the proteins for which some of their modifications meet the threshold established by the user. In one of the folders (*path_plots_with_threshold*), only the maps featuring modifications that meet the threshold set by the user are represented, while in the other, complete maps of the proteins are depicted (*path_plots_Without_threshold*)