Table of Contents

CompoundCalculator Suite

CompoundCalulator

Background

This notebook generates target lists of potential masses that are used to help interpret complex mass spectra. In particular, the notebook is used with Multi-Layered Analysis (MLA) which iteratively matches experimental peak lists to target lists and then matches the residual (unmatched) spectrum to a modified list.

Target calculation uses one or more base compounds and lists of potential adducts. The compound list can be extended by appending modified forms (e.g. metabolites), neutral losses, multimers (dimers, trimers, etc.) and heterodimers (dimers formed between different compound forms). Combinations of adducts, each with its own maximum value, are generated to a given maximum number. The final target list is formed by combining the compound and adduct lists and, for complex adduct patterns, can become very large.

Use cases include:

- Finding related compounds such as metabolites or degradation products from LCMS data. The target list has modified compounds and adducts and the peak list can include retention times.
- Analysis of complex spectra of single compounds that contain adducts and multimers, but potentially ions corresponding to reactions between analytes or analytes and background compounds (see J Chrom A 2019(1600)-174 for example.) Here the tool can be used to iteratively generate lists with different compounds and modifications that explain the observed spectrum.
- Analysis of the mixed spectra of different compounds with adducts.

A companion Notebook, **Match**, compares the target lists to experimental peak lists and produces files for matched and unmatched peaks.

Approach

CompoundCalculator is based on 'compositions' and 'limits'.

- **Compositions** provide a name, a mass and a 'Root' (note these are not elemental compositions so the chemistry cannot be checked). They are used to define the base compound and possible modifications (metabolites and adducts), e.g. ('Na-H', 21.981944) specifies an adduct formed by sodium replacing a proton. The root field is used to track the base or modified compounds so that complex lists can be sorted and filtered more easily.
- **Limits** define the maximum number for a particular modification, e.g (OH, 2) indicates there can be 0, 1 or 2 hydroxy metabolites

New entities are formed from others by adding the masses and appending the names. There is no attempt to validate the chemistry nor to remove redundant species (forms with the same mass).

Entities are treated as neutral and the charge is provided by adding or deleting the mass of a proton. This works because we specify the adducts as their canonical forms, for example (Na-H), which also generates the 'regular' forms, since:

$$M + (Na - H) + H+ \Rightarrow [M + Na]+$$

The quantity (Na-H) is known as the 'Effective Adduct Masses', M_A, and is equal to (M - vH) where M is the molecular weight and v the valency. This allows adduct combinations, including neutral species,

to be easily calculated and also means that the same list of compounds and adducts can be used for both positive and negative modes, the only difference being whether the proton mass is added or subtracted.

The core modification names and masses are stored in the Composition class as a dictionary, e.g. {....;'Na-H':21.981944;....} and initialized with a large number of compounds in the cell that defines the class:

```
Mods = {...}.
```

The dictionary can be viewed with the function get_mods_as_strings() and modified at any time with Composition.Mods = {}

The calculator generates two lists that are later combined:

- **adduct forms** are calculated by taking combinations of the allowed adduct forms up to a maximum number of modifications so it can calculate mixed forms such as [M + Na + K - H]+
- **compounds** are calculated by successively adding new forms generated by applying modifications to all of the compounds already in the list. This is performed in different steps as described below

Compound generation

Base compounds are provided as a list of (name, mass) tuples which specifies the main compound(s) but allows other components, specific known modifications, such as the loss of C2H4 in Vinpocetin, and unknowns to be considered. Unknown peaks can be included by assuming the observed mass, *m*, is an MH^+ ion, subtracting the mass of a proton and specifying a text label, for example 'x116'.

Additional compounds are generated in steps, each of which extends the list by adding a different form of modification to those compounds already in the list. Modifications are specified as lists of (name, limit) tuples where limit is the maximum number of this modification allowed (the lower limit is always zero). The root name can be left as the original compound or updated to allow different forms to be tracked more easily. The stages, corresponding to metabolism, are:

1. **Phase 1 mods** Generally these are oxidative products such as -OH, -(OH)2,... etc. and are applied to the main compounds. The default list of modifications includes 'COOH' which corresponds to the conversion of -CH3 to -COOH that occurs in ibuprofen. Multiple modifications of the same kind are calculated but different kinds are not combined.
2. **Phase 2 mods** These are conjugates such as glucuronides and sulphates and are applied to the base compounds and their phase 1 metabolites. As with phase 1 metabolites, multiple modifications of the same kind are calculated but different kinds are not combined.
3. **Multimers and heterodimers** Multimers for each compound in the extended list are calculated up to a user-specified limit, e.g. 2 or 3. If desired, the program can also calculate 'heterodimers', i.e. dimers formed by combining two different compounds rather than two of the same kind, i.e. A + B cf. 2A.

Ion generation

The final list of masses is generated by combining each of the compounds with each of the adducts and either adding or subtracting a proton. In some cases clusters of the adducts themselves can be observed, for example, clusters of sodium and potassium formates with or without the main analyte (JChromA 2019(1600)-174). These forms can be incorporated, in this example, by adding formic acid as a base compound and allowing, say, up to 10 monomers to be combined.

As shown in the Notebook, it is convenient to specify the polarity and use it to determine whether to add or subtract the proton mass, but also to specify the metabolites and limits. This is useful since some metabolites, e.g. glucuronides and sulphates, are often more intense in negative mode.

Summary

The compounds, parameters used and results are summarized in a cell that also save this information in a single line that can be written to the output file.

Output

Masses less than an upper limit are saved to a text file named according to the compound names, the adducts used, the polarity and, optionally, the date and time the list was generated as a string YYMMDD_HHMMSS; existing files will be overwritten. The cell that implements the user-specified path code uses the python 'os' library to generate the output directory path based on a list of directory and sub-directory names in a platform-independent manner.

An example file name is:

DiMeSA_m3_5-Na3Ca2-H2O pos.txt

Which contains the results for the compound defined as DiMeSA (dimethyl succinic acid), with up to 3 monomers (m3), a maximum of 5 adducts chosen from 3(Na-H) and 2(Ca-2H) and the loss of up to 1 water molecule. If heterodimer calculation is enabled '3m' would be shown as '3mh', and the replacement of hydrogen atoms is ignored for simplicity.

The output can be in two forms:

1. mass, name: a simple tab-separated list for use with Match
2. mass, XIC width, name: a tab-delimited file that is designed for use with the PeakView 'Extract Ion Chromatograms' command which can accept external lists in this format (via 'Import'). The file name also indicates that this format was used. Chromatograms extracted in PeakView with this format are labelled with the target mass label.

The first line of the output file contains a conditions string prefixed with a '#' character and containing items separated by ';'. The first item is the compound adduct summary also used as part of the file name, i.e. DiMeSA_m3_5-Na3Ca2-H2O in the above example, and the others are 'name:value' pairs separated by ';'. The summary is used by the Match notebook to label its output files.

Instructions

All of the user-defined parameters, including output file location, are set in the "Setup" section. When the parameters have been set the code can be executed with "Run selected cell and all below" or run individually. Each cell reports its results.

The first cell allows a shared storage area to be defined. The path is defined in a platform-independent way although Windows Users must add the drive letter (see cell). **These settings should be adjusted for the User's environment.**

The next cell ('Compounds and Adducts') defines the base compound(s) as a list of (name, molecular weight) tuples, e.g. [('DiMeSA', 146.057909)], and the modifications to be used. The example code also shows a way to include unknown peaks by assuming they are MH+ ions to generate a molecular weight by subtracting the mass of a proton (1.00727) and providing any label.

As mentioned above it is convenient to use a parameter to define the polarity and switch between modification sets. As shown in the cell, modifications can be excluded by removing them from the lists or by setting the limit to zero. Compound modifications are specified in phase 1 and phase 2 lists. Executing the cell produces a summary of the adducts and losses used for review before proceeding further.

Calculation and output parameters are specified separately. Calculation params (with examples) are:

```
multimer_limit = 3          # maximum multimer count
```

```
max_adduct_count = 5          # total number of adducts allowed
include_hetero_dimers = True     # if True, calculate dimers of *different* compounds
```

Output parameters are:

```
output_mass_limit = 1000  # masses greater than this are not written to the file
xic_width = 0.0          # if 0 the normal output form is used...alternative, e.g. 0.01, for the PeakView XIC file
save_ion_list = True      # write the results a file (or print thm here)
include_date_in_file_name = False   #include the date_time in the file name
# Generate the outpt_path; optional - add a subfolder to the shared path
data_path = os.path.join(shared_path,'Test')
```

Comments

Since compound and adduct generation are combinatorial, the final list of compounds can easily be in the thousands.

The program does not calculate isotope masses since this would dramatically increase the final number of forms. Code included in the 'Match' Notebook looks in the peak list so see if there are 13C forms corresponding to the matched masses.

Since the program has no chemical intelligence, there is no attempt to choose between forms that have the same mass or the same elemental composition. When matching, however, peaks are shown sorted by absolute error.

The program does not yet calculate multiply charged forms nor does it look for actual isotope patterns.

Match

Background

The 'Match' notebook compares a list of masses and labels generated with CompoundCalculator, or from another source, to an input peak list. It is convenient to open the Calculator and Match notebooks in side-by-side windows (Jupyter Lab allows this) so it is easy to update the target ion list and repeat the matching.

Approach

The program matches the target ion and peak lists using a tolerance that can be specified in amu (fixed) or ppm (increases with mass). If both are non-zero the ppm is calculated and the larger window used; this allows the window to increase with mass but never be less that a certain value. The program also allows for the possibility that complex peak and target ion lists can result in multiple matches for each peak. The function that prints the matches has a simplify mode which shows only one match for each peak but appends a string showing the number of matches; the match shown is the one with the smallest absolute error or shortest label (often the simplest). Matched peaks can be grouped according to the 'root' field, which is part of the target ion list, and there is a cell that shows peaks with redundant matches to emphasize their existence and allow adjustment of the matching or ion generation parameters if necessary.

Unmatched peaks greater than a given intensity threshold (percent base peak intensity) are also displayed. The idea is that this is part of interactive spectrum interpretation, i.e. once the origin of unmatched peaks is understood the ion generation parameters can be adjusted and applied to other peaks.

In order to ensure that isotope peaks stay with the monoisotopic peaks, the program only searches for 13C peaks for matched peaks. Identified isotope peaks can also match entries in the target ion list so

other possibilities are shown. In the output lists each peak has an index as well as the index of a related monoisotopic peak; these are the same for actual monoisotopic peaks.

The peak list must be tab-delimited and have mass values but can also contain columns for Retention Time (RT) and Intensity; the function that reads the peak list tries to determine which columns are present.

Results can be saved in several ways including a simple mass/intensity list and more detailed lists. Text lists can be imported into PeakView as spectra and overlaid on the original data to visualize the matches and highlight unmatched peaks.

Setup

Most values, flags and files are set in this area so 'Execute selected cell and all below' can be used after setting. Alternatively the cells can be executed in order and progress observed and reviewed.

Parameters related to main and 13C matching are in one cell; 13C matching includes a RT tolerance, which is ignored if the peak list does not include retention times, and a flag indicating that isotopes must have lower intensity than the preceding peak. (This is simplistic and will likely be changed in the future).

Output related parameters apply to information displayed in the notebook with 'print' statements and for output files. The program can write matched and unmatched peaks as text or MGF (Mascot Generic File format) files. The latter can include multiple spectra and are useful for input peak lists that include retention time; as this is assumed, the peaks are grouped by retention time before writing.

A cell defines a common storage area that is shared between the CompoundCalculator and Match so generated target ion lists are immediately available.

The experimental peak list is tab-delimited and must contain the mass but the other fields (intensity, retention time) are optional and will be stored internally as zero if absent. If the file has only one column it is assumed to contain masses otherwise the code assumes that the first column is Mass, the second is Inten and the RT is absent. If the file contains a header line it is used to define the order of the columns by looking for matches with common labels e.g. mz, m/z, Mass, etc. for masses. The RT column can only be used via a header line containing a column 'RT' or 'rt'.

The target ion list is read 'as is' and the conditions line printed so it can be checked. The compound-adduct summary line is extracted for later use.

Step 1 - Matching

Matching compares the target ion and peak lists looking for matches using a threshold defined in both amu and ppm (see above) and allows for 'many-to-many' matches.

In a second step, the program looks for 13C isotope peaks corresponding to matched peaks. A separate tolerance window is used for this, since it can usually be smaller, and potential isotope peaks must also be within an RT tolerance window (if RT is present) and, optionally, be less intense than the matched ion.

The number of peaks matched and the percent TIC explained are reported.

Step 2 - Print, review, save

Cells illustrate various ways to report the matches:

- print all or some of them inside the notebook; there is an option to show all matches, including redundant ones, or to simplify the output to only show the shortest (generally the simplest) or the one with the smallest error. This is achieved by specifying sort_order = 'label_len' or 'error' in the

call to ‘print_match_list. Isotopes are reported as the monoisotopic mass with the number of 13C atoms in brackets.

- print the matches grouped by the 'Root' field from the target ion list and optionally suppress isotopes. This is especially useful with multiple compounds, either provided by the calculator or as part of a general list (i.e. contaminants).
- count the number of peaks that have redundant matches and optionally print them. This can be a useful guide for adjusting the parameters or target ion list.
- print the unmatched peaks above a threshold (as a percentage of the base peak intensity)

Unmatched peaks are stored internally as mass/intensity while matched peaks have additional metadata such as the matched targets. The unmatched peak list is a good way to find peaks that still need to be explained and is often first step in further interpretation.

The detailed list of matches can be written to a file for use with the MatchAnalyzer module or elsewhere and a final cell summarizes the parameters and results.

A cell summarizes the results and generates a single string that is written to the output file.

Output

In addition to MGF Files, the Notebook can save the matches, unmatched (residual) peaks and a summary of the matching parameters and results. These have a common base name formed by appending the compound-adduct summary to the input peak file name. If the latter ends with ‘residual’ it is assumed to be part of MLA and ‘residual’ is removed first. The file name can optionally include the date and time of the match as a string YYMMSS_HHMMSS. In both the matched and unmatched files, the first column is mass and the second intensity so the file can be opened in spectrum display software, for example PeakView, and overlaid on the spectrum used to generate the initial peak list. This provides a nice way to visualize the matches and identify unmatched peaks for further consideration.

A final summary is printed in the Notebook to allow further review.

MatchAnalyzer

MatchAnalyzer is a preliminary Notebook that visualizes the matches which can help reveal incorrect or unlikely results.

The match file is read to a pandas DataFrame, cleaned to remove duplicate entries and isotope peaks, and visualized as a heatmap or pivot table. Heatmaps can show adduct patterns in complex sets while pivot tables allow more flexibility.

Processing

The match file is read to a DataFrame and ‘Pk_label’ and ‘Rel_inten’ columns added. Pk_label summarizes the mass and intensity in a single string and Rel_inten contains the peak intensity as a fraction of the base peak intensity (most intense).

Handling duplicate entries (peaks or matches) is an area that needs additional development. Currently, if duplicate peaks are found to the same match, the one with the smallest error is retained unless one is an isotope in which case all are removed. Essentially we favour isotope assignments, but remove them anyway; a subsequent step removes isotope peaks that were not duplicates. Duplicate matches to the same peak are reported but retained for visualization allowing us to chose.

The monoisotopic DataFrame is further prepared by dropping some of the unnecessary columns and processing the label field to generate a separate column for each adduct type where the value is the

count of that adduct. The base compound, base count (multimer count), loss (if present) and charge agent columns are retained. We differentiate the more common adducts (Na, K, Ca) since these often occur in combination with others and are best visualized last.

A cell generates a summary containing the intensity explained by each adduct form and each multimer. In MLA, adducts are added at each layer and any intensity assigned to previous adducts is due to combinations. The example below shows the output following Ca-2H matched to the residual from matching with Na-H. The total matched intensity is 29233 of which 1451 comes from Na-Ca combinations (all matches include Ca). The greatest intensity is for 3M (18575) although the greatest number of forms is for 2M (7).

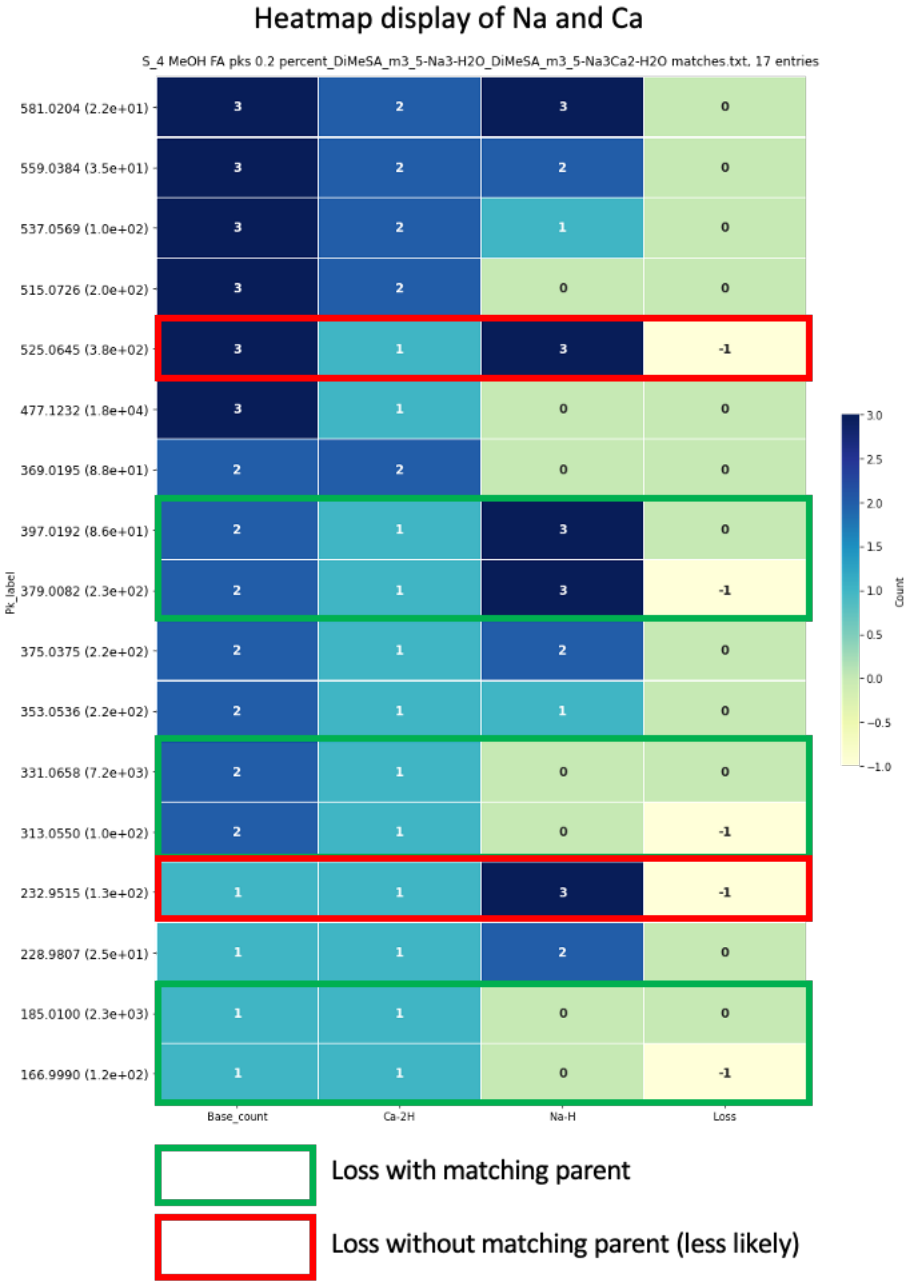
```
Intensities by adduct
Ca-2H:  17 items        TIC    29233.0
Na-H:   10 items        TIC     1451.0

Intensities by base count
1: 4 items      TIC     2558.0
2: 7 items      TIC     8100.0
3: 6 items      TIC    18575.0
```

Visualize as heatmap

A new DataFrame is built using the label as index and containing columns for each adduct; a loss is converted to 0 or -1, the latter indicating the row that has the loss. The columns are order so that the commonest are last and the loss, if any, is the last of all so that it is next to the form without the loss (if present).

Heatmaps offer many parameters for customization, including colour and labelling, and it is usually necessary to adjust the figure size and font size to get readable annotation, especially for large spectra. The example below readily shows that (Ca-2H)2 is reasonably common for 3M although one entry corresponds to a loss of H2O without the precursor which is less likely; entries (marked in green) that have the loss and the precursor seem more reasonable.

Since heatmaps are scaled and coloured according to the cell values, including the mass or intensity would dominate so these are encoded in the peak label (also the data frame index) which makes it harder to determine the most important matches.



Visualize as Pivot Tables

Pivot tables are an alternative visualization that allow peak intensity or mass to be displayed and work well with more complex adduct patterns.

In the example below, the marked row shows how the intensity varies for the base count and Ca-2H count with no Na-H and contains the three most intense peaks, the largest being 3M.Ca-2H. However, the mass display (right) shows a possible pitfall since the masses of the column showing the Na-H variation for 2M.Ca-2H do not all differ by the expected value of 21.984. The reason is that the pivot table will aggregate any unused columns, such as the loss in this case, so they are combined using a user specified function. In this case the 'min' function was used so the cell containing two entries (2M.Ca-2H and 2M.Ca-2H-H2O) shows the mass of the latter.

S_4 MeOH FA pks 0.2 percent_DiMeSA_m3_5-Na3-H2O_DiMeSA_m3_5-Na3Ca2-H2O matches.txt, inten > 0, 17/17 entries, bpi 17844.0

				Pk_inten	
Base_count	**1**	**2**		**3**	
Ca-2H	**1**	**1**	**2**	**1**	**2**
Na-H					
0	2279	7150	88	17844	195
1	-	216	-	-	103
2	25	224	-	-	35
3	132	232	-	376	22

```
Marked row (Na-H) = 0
  M.Ca-2H.H+   =  2279

 2M.Ca-2H.H+   =  7150
2M.(Ca-2H)2.H+ =  88

 3M.Ca-2H.H+   =  17844
3M.(Ca-2H)2.H+ =  195
```

S_4 MeOH FA pks 0.2 percent_DiMeSA_m3_5-Na3-H2O_DiMeSA_m3_5-Na3Ca2-H2O matches.txt, inten > 0, 17/17 entries, bpi 17844.0

				Pk_mass	
Base_count	**1**	**2**		**3**	
Ca-2H	**1**	**1**	**2**	**1**	**2**
Na-H					
0	166.999	313.055	369.019	477.123	515.073
1	-	353.054	-	-	537.057
2	228.981	375.038	-	-	559.038
3	232.952	379.008	-	525.064	581.020

Loss combined with min function, so M.Ca-2H-H2O is shown @ 313.055 not M.Ca-2H @ 331.066

Some solutions to this problem include:

- Aggregate with a different function , e.g. 'max' will show the mass of the precursor if both are present
- Include the Loss in the table (shown below)

S_4 MeOH FA pks 0.2 percent_DiMeSA_m3_5-Na3-H2O_DiMeSA_m3_5-Na3Ca2-H2O matches.txt, inten > 0, 17/17 entries, bpi 17844.0

		Pk_mass				
	Base_count	1	2		3	
	Ca-2H	1	1	2	1	2
Na-H	Loss					
0	0	185.010	331.066	369.019	477.123	515.073
	H2O	166.999	313.055	-	-	-
1	0	-	353.054	-	-	537.057
2	0	228.981	375.038	-	-	559.038
3	0	-	397.019	-	-	581.020
	H2O	232.952	379.008	-	525.064	-

Loss specifically shown so not combined

As shown below, it is also possible to use the peak label in the pivot table using 'first' or 'last' as the aggregation function.

S_4 MeOH FA pks 0.2 percent_DiMeSA_m3_5-Na3-H2O_DiMeSA_m3_5-Na3Ca2-H2O matches.txt, inten > 0, 17/17 entries, bpi 17844.0

		Pk_label				
	Base_count	1	2		3	
	Ca-2H	1	1	2	1	2
Na-H	Loss					
0	0	185.0100 (2.3e+03)	331.0658 (7.2e+03)	369.0195 (8.8e+01)	477.1232 (1.8e+04)	515.0726 (2.0e+02)
	H2O	166.9990 (1.2e+02)	313.0550 (1.0e+02)	-	-	-
1	0	-	353.0536 (2.2e+02)	-	-	537.0569 (1.0e+02)
2	0	228.9807 (2.5e+01)	375.0375 (2.2e+02)	-	-	559.0384 (3.5e+01)
3	0	-	397.0192 (8.6e+01)	-	-	581.0204 (2.2e+01)
	H2O	232.9515 (1.3e+02)	379.0082 (2.3e+02)	-	525.0645 (3.8e+02)	-

In this case the loss was included so the aggregation issue doesn't arise.

Prospector

Prospector is a preliminary notebook used to 'mine' peak lists for evidence of adducts of a given mass from a real or proposed compound mass.

An equation to calculate the Effective Adduct Mass, M_A, of a putative adduct is applied to each peak using limits for multimer size and charge range, generating a list of possible M_A values. This list is compared to a list of target values and those passing acceptance criteria are retained and summarized.

This provides a way to detect possible adducts in a spectrum but does not consider combinations and is best used to generate input for the CompoundCalculator, however the likelihood of a particular adduct being present can be assessed from the number or matches and their intensities and errors.

Setup

We read the peak list and define the target adduct list which, in the example here, includes ('H',0) to look for protonated forms and FeII in addition to FeIII.

The parameters required are the molecular weight of the monomer, the charge and multimer size limits, and parameters that determine which matches are reported:

```
min_inten_to_include = 20   # for each match
min_count_to_include = 2    # for one adduct
max_error_in_amu = 0.01     # applied to all matches
```

Processing and reporting

M_A is calculated for each peak in the list using the change and multimer size values. The resulting list is searched for each target mass in turn and matches within the specified mass tolerance and with the minimum intensity are retained. An adduct is reported if it has at least the minimum number of matches required.

As shown below, the report shows each adduct and each matched ion and includes the target and calculated M_A values and the associated errors. Note that the matches for the protonated forms include the same peak in different charge states and multimer sizes, indicating the need to check the isotope patterns, which in this case, are all singly charged (also the form with the smallest mass error). The need to check the isotope patterns is also shown by the Fe results. While there are matches to both forms, the trimer with FeII (493.0921) could be a 13C isotope of the FeIII ion at 492.0869 and, in fact, the possibility of mixture cannot be ruled out. This is also true for the dimer but the tetramer is only matched for FeII and appears to be supported by the isotope pattern.

A second cell prints a summary of the adduct matches, ranked in descending order of the percentage of the TIC they explain, This is useful for deciding the order of adducts to use in MLA.

The final cell summarizes the matches in mass order and reveals possible conflicts that need to be resolved from the isotope patterns. In the example shown below, the conflicts arise from the protonated forms in different charge states and the data shows that only the singly charged forms exist.

```
8459 possible Ma
H (0.00000), 5 pks, tic 4715.6, errors (mmu): range 5.95. median -5.95, st_dev 2.48
    147.0622, 3M 3+   1505.1 (m-vH)_calc: -0.00893 error: -8.93
    147.0622, 2M 2+   1505.1 (m-vH)_calc: -0.00595 error: -5.95
    147.0622, 1M 1+   1505.1 (m-vH)_calc: -0.00298 error: -2.98
    293.1200, 4M 2+    100.2 (m-vH)_calc: -0.00617 error: -6.17
    293.1200, 2M 1+    100.2 (m-vH)_calc: -0.00309 error: -3.09
Mg-2H (21.96939), 4 pks, tic 13443.5, errors (mmu): range 17.15. median -4.88, st_dev 7.74
    315.1017, 2M 1+  12866.1 (m-vH)_calc: 21.97859 error: 9.20
    377.1349, 5M 2+     56.4 (m-vH)_calc: 21.96563 error: -3.76
    461.1444, 3M 1+    486.9 (m-vH)_calc: 21.96339 error: -6.00
    607.2003, 4M 1+     34.2 (m-vH)_calc: 21.96143 error: -7.96
Na-H (21.98194), 2 pks, tic 120861.3, errors (mmu): range 1.77. median -2.47, st_dev 1.25
    169.0455, 1M 1+ 107995.3 (m-vH)_calc: 21.98036 error: -1.58
    315.1017, 2M 1+  12866.1 (m-vH)_calc: 21.97859 error: -3.35
Al-3H (23.95806), 3 pks, tic 45923.1, errors (mmu): range 1.19. median -3.86, st_dev 0.62
    171.0194, 1M 1+     41.9 (m-vH)_calc: 23.95420 error: -3.86
    317.0776, 2M 1+   3303.8 (m-vH)_calc: 23.95450 error: -3.56
    463.1343, 3M 1+  42577.4 (m-vH)_calc: 23.95331 error: -4.75
Ca-2H (37.94694), 6 pks, tic 29441.5, errors (mmu): range 5.42. median -5.67, st_dev 2.10
    166.0353, 2M 2+    170.8 (m-vH)_calc: 37.94031 error: -6.63
    185.0100, 1M 1+   2279.0 (m-vH)_calc: 37.94478 error: -2.16
    239.0638, 3M 2+    490.8 (m-vH)_calc: 37.93937 error: -7.57
    331.0658, 2M 1+   7150.0 (m-vH)_calc: 37.94274 error: -4.20
    477.1232, 3M 1+  17844.3 (m-vH)_calc: 37.94222 error: -4.72
    623.1786, 4M 1+   1506.7 (m-vH)_calc: 37.93970 error: -7.24
Ti-2H (45.93229), 2 pks, tic 1337.1, errors (mmu): range 0.01. median 1.71, st_dev 0.01
    339.0571, 2M 1+     67.2 (m-vH)_calc: 45.93400 error: 1.71
    485.1150, 3M 1+   1269.8 (m-vH)_calc: 45.93399 error: 1.70
V-2H (48.92831), 2 pks, tic 45.8, errors (mmu): range 10.41. median -0.85, st_dev 7.36
    171.5315, 2M 2+     25.4 (m-vH)_calc: 48.93266 error: 4.35
    212.0586, 4M 3+     20.4 (m-vH)_calc: 48.92225 error: -6.06
Fe-3H (52.91146), 2 pks, tic 6243.8, errors (mmu): range 1.18. median -4.93, st_dev 0.83
    346.0302, 2M 1+    598.6 (m-vH)_calc: 52.90712 error: -4.34
    492.0869, 3M 1+   5645.2 (m-vH)_calc: 52.90595 error: -5.51
Fe-2H (53.91928), 5 pks, tic 3530.5, errors (mmu): range 5.38. median -5.05, st_dev 2.30
    200.9808, 1M 1+     64.7 (m-vH)_calc: 53.91563 error: -3.66
    320.0813, 4M 2+     24.5 (m-vH)_calc: 53.91645 error: -2.84
    347.0373, 2M 1+    926.3 (m-vH)_calc: 53.91423 error: -5.05
    493.0921, 3M 1+   2468.8 (m-vH)_calc: 53.91107 error: -8.22
    639.1509, 4M 1+     46.2 (m-vH)_calc: 53.91203 error: -7.25
Ni-2H (55.91969), 2 pks, tic 491.4, errors (mmu): range 2.11. median -4.23, st_dev 1.49
    349.0396, 2M 1+    155.6 (m-vH)_calc: 55.91652 error: -3.17
    495.0954, 3M 1+    335.7 (m-vH)_calc: 55.91441 error: -5.28
Co-2H (56.91754), 3 pks, tic 286.9, errors (mmu): range 6.63. median -1.59, st_dev 3.54
    166.0353, 3M 3+    170.8 (m-vH)_calc: 56.91047 error: -7.07
    350.0402, 2M 1+     36.7 (m-vH)_calc: 56.91709 error: -0.45
    496.0969, 3M 1+     79.5 (m-vH)_calc: 56.91595 error: -1.59
Zr-4H (85.87339), 3 pks, tic 267.1, errors (mmu): range 1.40. median 8.18, st_dev 0.74
    190.0061, 2M 2+     42.2 (m-vH)_calc: 85.88184 error: 8.45
    263.0344, 3M 2+    132.7 (m-vH)_calc: 85.88044 error: 7.05
    671.1205, 4M 1+     92.2 (m-vH)_calc: 85.88157 error: 8.18
Sr-2H (85.88996), 6 pks, tic 1007.9, errors (mmu): range 5.88. median -7.29, st_dev 2.26
    190.0061, 2M 2+     42.2 (m-vH)_calc: 85.88184 error: -8.12
    232.9515, 1M 1+    132.0 (m-vH)_calc: 85.88632 error: -3.64
    263.0344, 3M 2+    132.7 (m-vH)_calc: 85.88044 error: -9.52
    379.0082, 2M 1+    232.4 (m-vH)_calc: 85.88512 error: -4.84
    525.0645, 3M 1+    376.4 (m-vH)_calc: 85.88349 error: -6.47
    671.1205, 4M 1+     92.2 (m-vH)_calc: 85.88157 error: -8.39
Cd-2H (111.88771), 3 pks, tic 239.3, errors (mmu): range 2.37. median -8.02, st_dev 1.20
    551.0599, 3M 1+     30.8 (m-vH)_calc: 111.87889 error: -8.82
    697.1186, 4M 1+    187.6 (m-vH)_calc: 111.87969 error: -8.02
    843.1781, 5M 1+     20.9 (m-vH)_calc: 111.88126 error: -6.45
Ba-2H (135.88960), 7 pks, tic 12765.3, errors (mmu): range 5.64. median -6.48, st_dev 2.00
    141.9781, 1M 2+    196.0 (m-vH)_calc: 135.88381 error: -5.79
    215.0067, 2M 2+    662.3 (m-vH)_calc: 135.88312 error: -6.48
    282.9513, 1M 1+   2018.9 (m-vH)_calc: 135.88612 error: -3.48
    288.0344, 3M 2+   1817.2 (m-vH)_calc: 135.88047 error: -9.13
    429.0079, 2M 1+   3812.1 (m-vH)_calc: 135.88481 error: -4.79
    575.0639, 3M 1+   3936.1 (m-vH)_calc: 135.88288 error: -6.72
    721.1198, 4M 1+    322.6 (m-vH)_calc: 135.88094 error: -8.66
```

Individual match report

```
Na-H  (21.982)  2 matches, TIC 1.209e+05 (30.43%). Errors (mmu): range 1.77, med  -2.47, stdev 1.25
Al-3H (23.958)  3 matches, TIC 4.592e+04 (11.56%). Errors (mmu): range 1.19, med  -3.86, stdev 0.62
Ca-2H (37.947)  6 matches, TIC 2.944e+04 ( 7.41%). Errors (mmu): range 5.42, med  -5.67, stdev 2.10
Mg-2H (21.969)  4 matches, TIC 1.344e+04 ( 3.38%). Errors (mmu): range 17.15, med  -4.88, stdev 7.74
Ba-2H (135.890) 7 matches, TIC 1.277e+04 ( 3.21%). Errors (mmu): range 5.64, med  -6.48, stdev 2.00
Fe-3H (52.911)  2 matches, TIC 6.244e+03 ( 1.57%). Errors (mmu): range 1.18, med  -4.93, stdev 0.83
H     (0.000)   5 matches, TIC 4.716e+03 ( 1.19%). Errors (mmu): range 5.95, med  -5.95, stdev 2.48
Fe-2H (53.919)  5 matches, TIC 3.531e+03 ( 0.89%). Errors (mmu): range 5.38, med  -5.05, stdev 2.30
Ti-2H (45.932)  2 matches, TIC 1.337e+03 ( 0.34%). Errors (mmu): range 0.01, med   1.71, stdev 0.01
Sr-2H (85.890)  6 matches, TIC 1.008e+03 ( 0.25%). Errors (mmu): range 5.88, med  -7.29, stdev 2.26
Ni-2H (55.920)  2 matches, TIC 4.914e+02 ( 0.12%). Errors (mmu): range 2.11, med  -4.23, stdev 1.49
Co-2H (56.918)  3 matches, TIC 2.869e+02 ( 0.07%). Errors (mmu): range 6.63, med  -1.59, stdev 3.54
Zr-4H (85.873)  3 matches, TIC 2.671e+02 ( 0.07%). Errors (mmu): range 1.40, med   8.18, stdev 0.74
Cd-2H (111.888) 3 matches, TIC 2.393e+02 ( 0.06%). Errors (mmu): range 2.37, med  -8.02, stdev 1.20
V-2H  (48.928)  2 matches, TIC 4.579e+01 ( 0.01%). Errors (mmu): range 10.41, med  -0.85, stdev 7.36
```

Adduct summary report

```
141.97813:   135.88960: Ba-2H  1M 2+     196.0 -5.79
147.06220:     0.00000: H      3M 3+    1505.1 -8.93
               0.00000: H      2M 2+    1505.1 -5.95
               0.00000: H      1M 1+    1505.1 -2.98
166.03533:    37.94694: Ca-2H  2M 2+     170.8 -6.63
              56.91754: Co-2H  3M 3+     170.8 -7.07
169.04554:    21.98194: Na-H   1M 1+  107995.3 -1.58
171.01938:    23.95806: Al-3H  1M 1+      41.9 -3.86
171.53151:    48.92831: V-2H   2M 2+      25.4  4.35
185.00996:    37.94694: Ca-2H  1M 1+    2279.0 -2.16
190.00610:    85.87339: Zr-4H  2M 2+      42.2  8.45
              85.88996: Sr-2H  2M 2+      42.2 -8.12
200.98081:    53.91928: Fe-2H  1M 1+      64.7 -3.66
212.05857:    48.92831: V-2H   4M 3+      20.4 -6.06
215.00674:   135.88960: Ba-2H  2M 2+     662.3 -6.48
232.95150:    85.88996: Sr-2H  1M 1+     132.0 -3.64
239.06382:    37.94694: Ca-2H  3M 2+     490.8 -7.57
263.03435:    85.87339: Zr-4H  3M 2+     132.7  7.05
              85.88996: Sr-2H  3M 2+     132.7 -9.52
282.95130:   135.88960: Ba-2H  1M 1+    2018.9 -3.48
288.03437:   135.88960: Ba-2H  3M 2+    1817.2 -9.13
293.12000:     0.00000: H      4M 2+     100.2 -6.17
               0.00000: H      2M 1+     100.2 -3.09
315.10168:    21.96939: Mg-2H  2M 1+   12866.1  9.20
              21.98194: Na-H   2M 1+   12866.1 -3.35
317.07758:    23.95806: Al-3H  2M 1+    3303.8 -3.56
320.08131:    53.91928: Fe-2H  4M 2+      24.5 -2.84
331.06583:    37.94694: Ca-2H  2M 1+    7150.0 -4.20
339.05709:    45.93229: Ti-2H  2M 1+      67.2  1.71
346.03021:    52.91146: Fe-3H  2M 1+     598.6 -4.34
347.03732:    53.91928: Fe-2H  2M 1+     926.3 -5.05
349.03960:    55.91969: Ni-2H  2M 1+     155.6 -3.17
350.04018:    56.91754: Co-2H  2M 1+      36.7 -0.45
377.13486:    21.96939: Mg-2H  5M 2+      56.4 -3.76
379.00820:    85.88996: Sr-2H  2M 1+     232.4 -4.84
429.00789:   135.88960: Ba-2H  2M 1+    3812.1 -4.79
461.14438:    21.96939: Mg-2H  3M 1+     486.9 -6.00
463.13431:    23.95806: Al-3H  3M 1+   42577.4 -4.75
477.12322:    37.94694: Ca-2H  3M 1+   17844.3 -4.72
485.11499:    45.93229: Ti-2H  3M 1+    1269.8  1.70
492.08694:    52.91146: Fe-3H  3M 1+    5645.2 -5.51
493.09207:    53.91928: Fe-2H  3M 1+    2468.8 -8.22
495.09541:    55.91969: Ni-2H  3M 1+     335.7 -5.28
496.09695:    56.91754: Co-2H  3M 1+      79.5 -1.59
525.06449:    85.88996: Sr-2H  3M 1+     376.4 -6.47
551.05989:   111.88771: Cd-2H  3M 1+      30.8 -8.82
575.06388:   135.88960: Ba-2H  3M 1+    3936.1 -6.72
607.20034:    21.96939: Mg-2H  4M 1+      34.2 -7.96
623.17861:    37.94694: Ca-2H  4M 1+    1506.7 -7.24
639.15094:    53.91928: Fe-2H  4M 1+      46.2 -7.25
671.12048:    85.87339: Zr-4H  4M 1+      92.2  8.18
              85.88996: Sr-2H  4M 1+      92.2 -8.39
697.11860:   111.88771: Cd-2H  4M 1+     187.6 -8.02
721.11985:   135.88960: Ba-2H  4M 1+     322.6 -8.66
843.17807:   111.88771: Cd-2H  5M 1+      20.9 -6.45
```

Matches by mass