LIBRARY PEPTIDE SEQUENCING SOFTWARE

SEQUENCE VERSION 1.5

James E. Redman
Keith M. Wilcoxen
M. Reza Ghadiri
The Scripps Research Institute

1. PURPOSE OF THE SOFTWARE ____ 3
2. SUMMARY OF SOFTWARE OPERATION ____ 4
3. SOFTWARE ORGANIZATION & USE ____ 5
3.1 System requirements ____ *5*
3.2 Tutorial on the Excel User Interface ____ *5*
3.3 Excel Sequence Control Worksheet Parameters ____ *7*
3.4 The Excel Sequence Menu ____ *9*
3.5 Error Messages ____ *12*
3.6 Assumptions ____ *13*
4. PROGRAMMING INFORMATION ____ 14
4.1 Properties ____ *14*
4.2 Methods ____ *17*
5. THE SOURCE CODE ____ 19

If you use this software please cite the following paper - Redman, J. E.; Wilcoxen, K. M.; Ghadiri, M. R. Automated Mass Spectrometric Sequence Determination of Cyclic Peptide Library Members. *J. Comb. Chem.* **2003**, *5*, 33.

1. Purpose of the Software

The software is intended for the mass spectral sequence determination of short synthetic peptides from combinatorial libraries prepared by the solid phase split-and-pool approach. The peptides may be linear or cyclic.

The user specifies the potential amino acids at each sequence position in the peptide, and supplies parent and fragment ion mass spectra. An Excel spreadsheet based user interface is provided, although an application programming interface enables the development of customized peptide sequencing applications.

The program was not designed for *de novo* sequencing of peptides from enzyme digests.

2. Summary of Software Operation

The program requires the user to specify the potential amino acid residues that could occur at each sequence position of the peptide of interest. All possible sequences are then generated and stored on a list. For peptides with a cyclic topology, sequences related by circular permutation symmetry are automatically excluded to avoid duplicates (although retaining sequences which are indistinguishable).

The user must also supply the filenames of mass spectra of the peptide, in Hitachi .stx format or as tab/space delimited text. The type of spectrum (parent or fragment ion) must be specified, and for MS^3 or higher details of which mass was fragmented must also be given. For MS^2 it is assumed that the parent ion was fragmented.

A score is calculated for each sequence on the list by comparison of an experimental spectrum with a corresponding spectrum calculated for that sequence. The scoring is performed by summing over all peaks the product of a scoring weight of the calculated peak and the intensity of a peak found at the same mass, within the specified tolerance, in the experimental spectrum. The scores are normalized by the sum of the weights in the calculated spectrum, to avoid biasing the scoring towards sequences that yield a greater number of peaks. The user specifies the weighting given to different fragmentations. A higher score indicates a closer match between the experimental and calculated spectra. Sequences with a score below a certain fraction of the highest scoring sequence can be ejected from the list. The process can then be repeated for another spectrum and the modified list. Normally the parent ion spectrum would be considered first, followed by a fragment ion spectrum. The highest scoring sequence on the list should then be the true sequence.

An optional step is to directly compare calculated spectra for a pair of sequences (call them 1 & 2) with the experimental spectrum. The unique peaks in the calculated spectrum are determined for each of the two sequences, and scores are calculated using only these peaks. A peak in the spectrum of peptide 1 would be unique if there was not a peak at the same (within a certain tolerance) mass in the spectrum of peptide 2. We have named this process “Critical Analysis”. This approach appears useful when applied to a few top scoring sequences on the list as it can sometimes pick out the correct sequence even when this is not the highest scoring when all peaks are considered.
Other routines are provided, for tasks such as calculating parent ion masses for all the peptides in a library, and overlaying plots of calculated and experimental spectra.

3. Software Organization & Use

3.1 System requirements

A PC running Windows 95 or later (tested with Me, NT 4 and XP). Although the program itself is small, it may use memory heavily during calculations. Over 128 Mb is recommended.
For the spreadsheet interface, Excel 2000 is required. Excel97 or XP might work too, although they have not been tested. Excel95 will not work.

To install, double click the Setup icon, and follow the directions. The program, input files, example spectra and documents will be copied to your hard disk. Using Windows NT, 2000 and XP you will need to be logged in as an administrator.

To uninstall, use the Add and Remove programs feature on the Windows control panel.

3.2 Tutorial on the Excel User Interface

Open the "SequenceV1_5.xls" workbook. Macros must be enabled. Some antivirus software will give a warning about ActiveX controls. If this happens then the ActiveX control must be allowed to run.

A new menu, **Sequence**, is added to the Excel menu bar. **Insert Filename**, **Process Spectrum**, **N Terminus** and **C Terminus** are added to the right mouse button shortcut menu.

All of the required parameters are entered in the worksheet, and the processing commands are selected from the **Sequence** menu.

1) Open the example parameter file "in.txt" by clicking **Sequence->Open Job**.
The parameters are read directly from the file, and the worksheet is automatically updated.

2) In the "Parent Spectrum" and "Fragment Spectrum" cells, change the filenames to the names of the parent and fragment spectrum files that are to be processed. To avoid typing, right click in the cell, and select **Insert Filename**, then choose the file from the dialog box.

For the example files KWLWKEparent.stx and KWLWKEfrag.stx the remaining parameters need not be changed. These files are for the cyclic peptide c[K<u>W</u>L<u>W</u>K<u>E</u>] (underscored residues are the *D* isomers).

The cyclicDL Topology indicates a cyclic peptide of residues of alternating *D* and *L* chirality.

No. of Residues is the length of the peptide (6).

Number of Spectra is the total number of experimental spectra (one parent + one fragment spectrum = 2)

The Residues section is where the amino acids are entered. For each sequence position in the peptide the possible residues are listed using the standard one letter code. For the example the first residue is fixed as lysine, the second as tryptophan, and the remaining residues can be any of EFHKLNRSW.

For details on the other parameters see section 3.3.

3) Click **Sequence->Open Masses**
Open the residue mass file "monomstab.txt". This is an ASCII file containing the monoisotopic mass of each residue. This file may be edited if non-standard residues are required.

4) Click **Sequence->Process->Process All**
This generates all of the sequences, processes the parent ion spectrum first, followed by the fragment ion spectrum. The "critical analysis" is then carried out for all possible pairs of the 30 highest scoring sequences. The experimental spectrum overlaid with the calculated peaks for the "best" sequence is displayed.

The results are displayed in a new workbook. The upper part of the worksheet shows the parameters that were used in the processing. The next section shows the commands that have been run, and their outcome (whether an error occurred or not, 0 = no error, other number = error code).

The number of sequences on the list is reported, together with the sequence that the program considers most likely to be correct, based on the critical analysis.

The critical analysis table is at the bottom of the worksheet. The values in the table are comparative scores for the pair of sequences given at the left and top of the table. A blue font color indicates the sequence at the top is in better agreement, whereas a black color indicates the sequence to the left is superior. The sequences are sorted according to their scores calculated using all peaks, with the highest scores at the top and left.
The fill colors indicate sequences that have no unique peaks (green = top sequence, red = left sequence, yellow = both sequences).

For the example, the correct sequence, KWLWKE, is at the top of the list, although the critical analysis is warning that this sequence is not expected to give any peaks which enable it to be distinguished from KWLWEK or KWWLKE. The latter two sequences *are* expected to yield peaks which would distinguish them from KWLWKE.

Below the critical analysis table, a picture of the spectrum will be pasted into the worksheet, overlaid with the calculated peaks (in red) for the sequence selected by the critical analysis. The mass labels refer to the calculated peak positions. It may be necessary to scroll the worksheet to see the spectrum.

3.3 Excel Sequence Control Worksheet Parameters

This section describes the parameters in the Excel worksheet user interface.

Topology

The topology of the peptide. Options are cyclicDL, cyclic or linear. This determines whether/how cyclic permutation symmetry needs to be considered when generating the initial list of sequences from the amino acids.

No. of Residues

The length of the peptide in residues.

Number of Spectra

The total number of experimental spectra.

No. of sequences for critical analysis

The number of sequences to be included in the critical analysis. The sequences are picked from the top of the list, so provided that the sequences are scored and sorted, this will select the highest scoring sequences. Do not make this number too large as processing time increases with the square of the number of sequences.

Parent Spectrum parameters

Parent Spectrum

Path and filename of the experimental parent ion spectrum. Files should be tab/space delimited text or Hitachi .stx format (must consist of evenly sampled data). Text files should consist of pairs of mass and intensity values. The program was not designed to use peak lists, although if it encounters one in a text file it will attempt to use it by padding it out with zeroes.

Mass Tolerance

The peak matching tolerance for the parent ion spectrum. The program will search over ± this mass range for a peak. A peak is the most intense signal found in the mass range.

Threshold

This sets the score below which sequences are removed from the list. It is expressed as a fraction of the highest scoring sequence. A value of 0 removes everything except the top scoring sequence, 1 retains all sequences on the list.

Weight [M+H]+, Weight [M+2H]2+, and Weight [M+3H]3+

These parameters are the scoring weights for singly and multiply charged ions. Setting one of these parameters to zero will cause that ion to be ignored in the calculated parent ion spectrum. e.g. if triply charged parent ions are unlikely to be observed then set Weight [M+3H]3+ to zero.

Fragment spectrum parameters

More than one fragment spectrum may be specified in this section. Simply list the parameters in the columns.

Fragment Spectrum
Path and filename of the experimental fragment ion spectrum. Files should be tab delimited text or Hitachi .stx format. Files should contain raw data evenly sampled, although for text files if a peak list is encountered the program will attempt to use it.

MS-MS
The number of mass select/fragmentation steps. 1 = MS^2, 2 = MS^3

Multiple loss (y/n)
Determines whether multiple side chain losses will be considered. Value should be y or n.

Mass Tolerance
The peak matching tolerance for the fragment ion spectrum. The program will search over ± this mass range for a peak. A peak is the most intense signal found in the mass range.

Threshold
Equivalent to the threshold parameter for the parent ion spectrum.

Weight no loss, Weight -NH3, Weight -CO, and Weight -H2O
Scoring weights for different fragmentations. Weight no loss refers to fragments that arise from only cleavage of the peptide bond ('b' and 'y' fragments for linear peptides). Weight –NH3 refers to loss of ammonia from K, R and Q residues. Weight –H2O refers to loss of water from S, T, and E residues. Weight -CO is for fragments with CO loss from the C terminus ('a' fragment for linear peptides). If multiple side chain loss is being used then it is best to set Weight –NH3 and Weight –H20 to 1, so that the weights are handled entirely by the loss probabilities.

Multiplicity factor and Degeneracy tolerance
These parameters determine how the scoring weights combine for degenerate peaks. Calculated peaks which differ in *m/z* of less than Degeneracy tolerance are merged, and their scoring weights combined according to Multiplicity factor (0 = weights do not add up, 1 = weights add up linearly). (n.b. Currently this calculation does not quite work consistently for accidentally degenerate peaks with different scoring weights, although this should not cause any problems.)

Weight N terminal, Weight C terminal and Weight Internal
For linear peptides the peak weights for N, C terminal and internal fragments are multiplied by these factors. Set to zero to exclude fragments.

NH3 loss probability and H2O loss probability
Probability of loss of ammonia from R, K, Q and water from S, T, E. For peptides with several of these residues, every possible combination of side chain loss is considered and probabilities are calculated according to a binomial distribution. The scoring weight of

the peak is multiplied by this probability. These parameters are ignored if Multiple loss is n.

Mass Select and Window

For MS^3 or greater it is necessary to specify the *m/z* range of the peak which is selected and fragmented. The mass selected range is Mass Select ± Window. If there is more than one mass selection step, then list each from left to right.

N Terminus and C Terminus

These are the masses of the terminal groups of linear peptides. Several common masses are on the right mouse button shortcut menu. These groups remain with the terminal residue during the fragmentation calculations. If you need terminal groups which fragment off then it is necessary to add custom residues in the residue mass file.

Residues

The potential residues at each position in the sequence are listed in this column. They should be written with the single letter code, uppercase characters, with no spaces. All uppercase letters are valid, provided that an appropriate mass is entered in the mass table file. The residues can be specified in any order although the program will rearrange them into alphabetic order.
Sequences longer than 8 residues are accepted – just extend the table downwards.

3.4 The Excel Sequence Menu

Sequence->Open Job

Opens a stored parameter file. Parameter files store all the parameters except for No. of sequences for critical analysis and any filenames for batch processing which have placed in the lower part of the worksheet. The parameter file is an ASCII text file, not an Excel workbook.

Sequence->Save Job

Saves the current parameters as an ASCII file that can be read with **Open Job**.

Sequence->Open Masses

Open a mass table file. This is an ASCII file containing the residue masses (i.e. amino acid mass – H_2O). It should contain an entry for all uppercase letters. The letter is immediately followed by the mass, with no space between them. Each entry should begin on a new line, and the file should end with a new line after the final entry.

Sequence->Calculate->Calculate Spectrum

Calculates an assigned fragment ion spectrum for a specified sequence. A dialog box will prompt for a sequence. If there is only one spectrum listed in the Fragment Spectrum part of the worksheet, then it is this that will be calculated. If there are several spectra listed, then choose one by clicking it before running this command.
The result appears in a new sheet. The mass, scoring weight and assignment of each peak are given. Monoisotopic masses are listed.

Sequence->Calculate->Copy Spectrum

This works similarly to **Calculate Spectrum** except a metafile graphic of the experimental spectrum overlaid with the calculated masses (monoisotopic) shown as red vertical lines is generated. The predicted peaks are aligned with experimental peaks within the limits imposed by Mass Tolerance. If the user does not supply a sequence then no predicted peaks are shown. The graphic can be pasted into a worksheet or other document using **Edit->Paste** or Ctrl-V.

Sequence->Calculate->Unique Peaks

Determine the unique (monoisotopic) peaks in a fragment ion spectrum for two sequences. If there is only one spectrum listed in the Fragment Spectrum part of the worksheet, then it is this that will be calculated. If there are several spectra listed, then choose one by clicking it before running this command. The program will prompt for the two sequences, then place the results in a new worksheet.

Sequence->List->Generate Parents

Generates the list of sequences according to the residues specified in the Residues section. The list is stored internally and no results are displayed. Each sequence is given a score of zero.

Sequence->List->Calculate Masses

Calculates the monoisotopic masses of the sequences on the list. A residue mass file must have been read using **Open Masses** before this command will work, and there must be some sequences on the list. The calculated masses are the sum of the residue masses, and do not include an additional proton. The masses are stored instead of scores, and will overwrite any scores that have already been calculated.
This command does not display any results.

Sequence->List->Sort List

Sorts the sequences on the list in descending order of score, or mass if **Calculate Masses** was previously used.
No results are displayed.

Sequence->List->Purge List

Eject sequences from the list if their score is below a certain fraction of the highest score. A dialog box prompts for a threshold parameter between 0 and 1. 0 ejects all sequences except the highest scoring, 1 keeps everything. This works identically to the Threshold parameter in the worksheet. No results are displayed.

Sequence->List->Critical Analysis

Performs the critical analysis for the specified number of highest scoring sequences on the list. If there is more than one fragment ion spectrum then one can be selected prior to running the command, else the final listed is used. A comparative score is calculated for pairs of sequences, taken from the top of the list, using only the unique peaks in the calculated spectrum. These scores are displayed in a color-coded table in a new worksheet.

A blue font color indicates the sequence at the top is in better agreement, whereas a black color indicates the sequence to the left is superior. The fill colors indicate sequences that have no unique peaks (green = top sequence, red = left sequence, yellow = both sequences). The sequences are in the same order as they are on the list, with the top sequence at the top and left.

Sequence->List->Results Sheet

Places the sequences on the list and their scores (or masses) in a new worksheet. A warning is issued if there are many sequences, and the command will generate an error if there are too many sequences to fit into a worksheet.

Sequence->Process->Process Spectrum

A spectrum should be selected in the worksheet by clicking it before choosing this command. Parent or fragment spectra may be used. The sequences on the list are processed according to the parameters specified in the worksheet. After running the command the list will contain sorted scored sequences which are a subset (according to the Threshold parameter) of the sequences originally on the list.

No results are displayed. Use **Results Sheet** to view the sequences and scores.

This command is also accessible using the right mouse button menu.

Sequence->Process->Process All

This performs an aggregate of operations. The list of sequences is generated from the Residues, then the spectra are processed, starting with the parent ion, followed by the fragment spectra in the order in which they are listed in the worksheet. A critical analysis is carried out and the results displayed in a new sheet, along with the final fragment spectrum overlaid with the calculated peaks for the best sequence from the critical analysis.

Sequence->Process->Batch Process All

Performs Process All steps on spectra of a series of compounds and summarizes the results in a worksheet. Enter the spectrum filenames in the blank area below the Residues section. The file names should be inserted in the cells to form a rectangular block in the order:

Parent_Compound1	Fragment1_Compound1
Parent_Compound2	Fragment1_Compound2
Parent_Compound3	Fragment1_Compound3

Click somewhere in the block of filenames then select Batch Process All to run the command.

3.5 Error Messages

All error messages have an associated numerical code which is listed as the Outcome of the command in any worksheet which displays results. Errors 7 and 12 should not normally be encountered. Error 3 may be an indication that the computer is about to crash.

Code	Message	Meaning
0	Completed.	Command was successful. No error.
1	File not found.	Trying to open a non-existent file. Check the name and path.
2	Error in file.	There was a problem reading the file. Usually caused by an error in the file format.
3	Allocation error.	Allocation of memory failed – probably because there is insufficient.
4	Empty list.	There are no sequences on the list, and the selected command requires a list of sequences.
5	No job parameters.	The processing parameters are missing. Type some in or use **Open Job**.
6	Spectrum not exist.	Spectrum parameters do not exist.
7	Math error.	An error occurred during calculation.
8	No residue masses.	A residue mass table needs to be read. Use **Open Masses**.
9	Invalid parameter.	There is an invalid processing parameter.
10	Cannot write file.	The parameter file cannot be written.
11	Missing parameter.	A parameter is missing.
12	Index out of range.	An array index is out bounds.
13	Invalid selection.	The command requires a spectrum to be selected in the worksheet. Click on a cell in the row of parameters associated with a spectrum, then try the command again
14	List is too large.	There are too many sequences on the list to fit the results into a worksheet.

Table 1.

3.6 Assumptions

Some approximations/limitations have been made to increase processing speed:
1) The experimental data points must be evenly sampled (same mass interval between data points). This should be true for Hitachi .stx data – but it is a good idea to check first as with some methods this is not the case. The program attempts to pad out data from regular text files so that this assumption is met.

2) Isotopes are simplistically treated.
Only one isotope peak is considered for each ion. The program attempts to guess the most abundant isotope peak, assuming a typical peptide composition (Biomed. Env. Mass. Spectrom. **1986**, *13*, 373). If M = monoisotopic mass,
$M \leq 1818$ most abundant isotope peak is M
$1818 < M \leq 3247$ most abundant isotope peak is M+1
$M > 3247$ then most abundant isotope peak is M+2

When the program reports calculated masses (e.g. with **Calculate Spectrum**), these are monoisotopic. The isotope distribution described above is only used in spectra generated for the purpose of calculating scores.

For MS^n, ensure that the Window parameter is large enough to enclose the isotopic envelope of the peak being fragmented.

3) Except for parent ions, multiply charged species are ignored.
In the absence of a good way of predicting which multiply charged species are likely to be observed it was decided to omit them entirely during the calculation of fragment spectra.

4) A limited number of fragmentations are considered:
Fragmentation of the peptide bond.
Loss of CO.
Loss of NH_3 from K,Q,R.
Loss of H_2O from S,T,E.

5) If a side chain fragmented peak is mass selected then the side chain fragmentation is ignored in the calculation of the fragment spectrum.

4. Programming Information

This section of the guide covers features which can be accessed without the need to change the C++ source code and recompile the program.
The dll is an automation server with a COM interface so it can be controlled from Visual Basic and other programming languages that support COM. There is only one object called AutoSeq with properties and methods which control all aspects of the program.
CreateObject should be used in VB to create an AutoSeq object, e.g.

```
Dim MassServer As Object
Set MassServer = CreateObject("sequencedll.AutoSeq")
```

Properties and methods of AutoSeq are listed below in C++ style. Part of the code has been rewritten for Builder 6 to provide a type library, but for some reason it executes much slower, and will be released when these problems are resolved. In VB the AnsiString type corresponds to an ordinary string, **int** and **float** are 32 bit integer and floating point numbers respectively. Array properties are shown with their indices in square brackets e.g. [**int** *Index1*][**int** *Index2*], in VB this would be written as (*Index1*, *Index 2*).
VB example code:

```
MassServer.Parms(1, 5) = 0.2
errcode = MassServer.CalculateSpectrum(1, “KRWLWL”)
```

4.1 Properties

AnsiString *About*
An ‘about’ string.

AnsiString *Sequence*[**int** *Index*]
The list of sequences. *Index* is in the range 0...*Count*-1.
Read only.

float *Score*[**int** *Index*]
The list of scores (or masses) associated with each sequence. *Index* is in the range 0...*Count*-1.
Read only.

int *Length*
The original number of array elements in the list, as set by GenerateParentList. After calling PurgeList the *Length* is unaltered, although *Count* will be updated to reflect the size of the list.
Read only.

int *Count*
The number of sequences on the list.
Read only

int *Nres*
The length of the peptide.
Read/write.

int *Nspec*
The total number of experimental mass spectra.
Read/write.

int *NPeaks*[**int** *sIndex*]
The number of peaks in calculated spectrum *sIndex*. Up to 2 separate spectra may be stored simultaneously and referenced by values of *sIndex* of 0 or 1.
Read only.

float *Mass*[**int** *sIndex*][**int** *pIndex*]
The *m/z* of peak *pIndex* in calculated spectrum *sIndex*.
sIndex must be 0 or 1.
pIndex is in the range 0...*Npeaks*[*sIndex*]-1.
Read only.

float *Weight*[**int** *sIndex*][**int** *pIndex*]
The scoring weight of peak *pIndex* in calculated spectrum *sIndex*.
sIndex must be 0 or 1.
pIndex is in the range 0...*Npeaks*[*sIndex*]-1.
Read only.

float *Ccap*
The mass of the C terminal group of a linear peptide.
Read/write.

float *Ncap*
The mass of the N terminal group of a linear peptide.
Read/write.

AnsiString *Annotation*[**int** *sIndex*][**int** *pIndex*]
The peak annotation of peak *pIndex* in calculated spectrum *sIndex*.
sIndex must be 0 or 1.
pIndex is in the range 0...*Npeaks*[*sIndex*]-1.
Read only.

AnsiString *Topology*
The topology of the peptide. Permitted values are cyclic, cyclicDL, linear and unspecified.
Read/write.

AnsiString *FileName*[**int** *Index*]
The filename of the experimental spectrum *Index*.
Index is in the range 0...*Nspec*-1.
Read/write.

int *Fragmentations*[**int** *Index*]
The number of fragmentations for spectrum *Index*.
Read/write.

int *MultiLoss*[**int** *Index*]
Multiple side chain losses. 0 = no multiple side chain losses permitted, else permit multiple side chain loss.

float *Parms*[**int** *Index1*][**int** *Index2*]
The parameter array holding the scoring weights, and mass selection information for spectrum *Index1*.
Index1 is in the range 0...*Nspec*-1.
Index2 refers to the identity of the parameter.
For a parent spectrum, *Fragmentations*[*Index1*] = 0,

Index2 =	0	Mass tolerance
	1	Threshold
	2	Scoring weight $[M+H]^{+}$
	3	Scoring weight $[M+2H]^{2+}$
	4	Scoring weight $[M+3H]^{3+}$

For a fragment spectrum, *Fragmentations*[*Index1*] > 0,

Index2 =	0	Mass tolerance
	1	Threshold
	2	Scoring weight, no side chain or CO losses
	3	Scoring weight, fragment $-NH_3$
	4	Scoring weight, fragment -CO
	5	Scoring weight, fragment $-H_2O$
	6	Multiplicity factor
	7	Degeneracy tolerance
	8	N terminal fragment scoring weight
	9	C terminal scoring weight
	10	Internal fragment scoring weight
	11	NH_3 loss probability
	12	H_2O loss probability
(MS^3 only)	13,14	Mass selection window (center mass and window width)

For MS^4 etc. there are further pairs of Mass select and window parameters, corresponding to each mass selection step.
Read/write.

AnsiString *Residues*[**int** *Index*]
The possible residues at sequence position *Index*.
Index is in the range 0...*NRes*-1.
Read/write.

float *CriticalScore*[**int** *Index1*][**int** *Index2*]
The critical analysis score of sequence *Index1* compared to sequence *Index2*. CriticalAnalysis must have been called first.
This is not the same score as appears in the critical analysis table in the Excel worksheet. This score is the difference between the scores of sequences *Index1* and *Index2*.
Index1 and *Index2* should be in the range 0...number of critical analysis sequences-1
Read only.

AnsiString *CritBest*
The 'best' sequence from the critical analysis. This is the highest scoring sequence on the list with the fewest 'disagreements' in the critical analysis. A sequence has a 'disagreement' when both sequences under consideration have unique peaks, and the critical analysis score is higher for the alternative sequence.
Read only

4.2 Methods

All methods return an **int** error code. The error codes are listed in Table 1.

int ReadFile(AnsiString *FileName*);
Reads the parameter file *FileName*.

int WriteFile(AnsiString *FileName*);
Writes the parameters to file *FileName*.

int ReadMasses(AnsiString *FileName*);
Read the residue mass table file *FileName*.

int CalculateMasses();
Calculate the masses of the sequences on the list. This mass does not include an extra proton. The masses are stored in *Score*.

int WriteOutputList(AnsiString *Filename*);
Writes out the sequences and scores on the list to file *FileName*.

int GenerateParentList();
Generates a list of sequences from *Residues* and *Topology*. For topologies cyclic and cyclicDL cyclic permutation symmetry is accounted for so that there will not be duplicate sequences on the list. The sequences are held in *Sequence* and the scores are initialized to zero.

int ProcessSpectrum(**int** *SpecToProcess*);
Processes spectrum number *SpecToProcess*. The experimental spectrum file is *FileName*[*SpecToProcess*] and the processing parameters are in *Parms*[*SpecToProcess*][*Index2*]. Processing scores the sequences on the list against the experimental spectrum. The list is sorted in descending order of score and purged to remove the lower scoring sequences (determined by the threshold parameter).

int CalculateSpectrum(**int** *SpecToProcess*, AnsiString *sequence*);
Calculates spectrum *SpecToProcess* for sequence *sequence*. The calculated monoisotopic spectrum *m/z*, weights and annotations are stored in *Mass*[0][*pIndex*], *Weight*[0][*pIndex*] and *Annotation*[0][*pIndex*] respectively. Peaks are listed in order of ascending *m/z*.

int CriticalPeak(**int** *SpecToProcess*, AnsiString *Sequence1*, AnsiString *Sequence2*);
Determines the unique monoisotopic peaks in the spectrum *SpecToProcess* of *Sequence1* and *Sequence2*. The two resulting spectra are stored in *Mass*[*sIndex*][*pIndex*], *Weight*[*sIndex*][*pIndex*] and *Annotation*[*sIndex*][*pIndex*] where *sIndex* = 0 or 1. All non-unique peaks are given a weight of zero, the unique peaks retain their original weighting.

int CriticalAnalysis(**int** *SpecToProcess*, **int** *NumberToProcess*);
Performs the critical analysis for *NumberToProcess* sequences from the top of the list with spectrum *SpecToProcess*.
The results are held in *CriticalScore* and *CritBest*.

int ProcessAll();
This calls *GenerateParentList* then for each experimental spectrum a call to *ProcessSpectrum* is made.

int PurgeList(**float** *threshold*);
Removes sequences from the list with a score below a fraction of the highest score, as specified by *threshold* (0 = remove everything except top score, 1 = retain everything). The sequences are 'removed' by decreasing the value of *Count*.

int SortList();
Sorts the sequences on the list in order of descending *Score*. This is intended for sorting the list after running CalculateMasses, when Score represents a mass.

int PlotSpectrum(**int** *SpecToProcess*, AnsiString *sequence*, **float** *scale*);
Pastes to the Windows clipboard a metafile image of the experimental spectrum *SpecToProcess* overlaid with the calculated peaks for sequence *sequence*. The calculated peaks are shown as red vertical lines displaced above the experimental spectrum, annotated with their monoisotopic *m/z*. The program attempts to match the positions of the calculated peaks to the observed peaks, within the given tolerance. Scale controls the size of the graphic. A value of 1 or larger should produce a reasonable looking plot.

5. The Source Code

C++ Classes

The dll was written in C++ and compiled using Borland C++ Builder version 1. The COM interface described above is essentially a wrapper for a C++ class called MsProcess. An MsProcess object is a member of the AutoSeq class, which is derived from the Visual Component Library (VCL) TAutoObject.

The methods of AutoSeq all have a corresponding method in MsProcess, which generally performs exactly the same operation. The function MsProcess::ProcessAll behaves slightly differently to AutoSeq::ProcessAll in that the former only calls GenerateParentList if the list is empty, whereas the latter always calls GenerateParentList.

Data such as the sequence list, scores, the critical analysis scores, and the calculated spectra are held in protected class members but can be accessed *via* public getter functions which do error checking on array bounds.

The processing parameters are held in a public input object called job. All the members of the input class are public, to enable access from MsProcess without any overhead associated with calling getter functions. However it is important that the setter functions of input are used as these perform memory management. See comments in the code for details.

There are a number of other classes, which handle sequence fragmentations and spectra, although knowledge of their operation is not necessary to use MsProcess.

Except for MsProcess::PlotSpectrum(**int** SpecToProcess, **char** *sequence) the code requires only standard library functions and is intended to be ANSI C++ compliant. The type **bool** is required, and the ANSI scoping rules regarding **for** loops must be adhered to. Borland VCL and the C++ Builder compiler are required to compile MsProcess::PlotSpectrum and the unit defining the AutoSeq class.