rTOFsPRO-package for Signal Processing of TOF MS

By Dariya Malyarenko, Maureen Tracy and William Cooke
The College of William and Mary

1 Overview of the rTOFsPRO signal processing capabilities for TOF MS

1.1 Introduction

rTOFsPRO library is a collection of computational tools for low-level signal processing of linear TOF mass spectra over the full m/z range of the record. The purpose of the tools is to enhance sensitivity of signal detection by model-based filtering and denoising. The parameters are optimized to intelligently compress data and produce aligned peak intensity matrix (and peak positions) that can be used for

further statistical analysis, e.g. in clinical or comparative proteomics studies. The input for the processing is in the form of R-structures (tofList and tofListMetaData) that are produced by (and described in) e.g., “WM Bruker Parser” R-package [1].

1.2 Background

Protein profiling using Time-Of-Flight (TOF) Mass spectrometry (MS) is an important technique in proteomics, as it can direct the identification of biologically significant species (biomarkers). Linear TOF MALDI Mass Spectrometers provide fast scanning over broad mass ranges which is useful for the analysis of multiple biological samples [2,3]. The unique ability of TOF-MS to analyze hundreds of patient samples in reasonable time makes it applicable to population studies in clinical proteomics. This distinguishes TOF-MS among other MS-based approaches typically capable of analysis of only dozens of samples over reasonable time. While peak broadening and low signal-to-noise are issues for high mass ranges, signal processing tools (model-based background subtraction, integrative down-sampling, deconvolution filtering, pedestal removal, peak detection and alignment) have been developed [4, 5] to address these challenges, and are provided in this R-package. The availability of broad-mass data allows the application of new signal processing techniques such as the detection of ionization satellites (distinct from molecular ions) and reconstruction of molecular protein signatures that further improves selectivity and sensitivity for molecular ion detection from MALDI-TOF [6, 7].

A typical large-scale study employing MALDI-TOF includes sample purification, using different affinity capture agents (e.g., C3 or IMAC magnetic beads [4]) and spotting several replicates of the purified protein mixture with matrix on a MALDI plate (e.g., 384 spots, numbered (1-24) x (A-P) by Bruker Ultraflex) for profiling and identification. The chemical preparation steps are preferably done with a programmable robotic bioprocessor for better control and reproducibility. A single clinProt (Bruker) bioprocessor run allows spotting of 96 samples (limited by number of bioprocessor wells) in 4 replicates (row or column pattern). As an example, a clinical study for 200 samples with three replicates would require at least 7 bioprocessor runs. The MALDI spectra may then be obtained for different mass ranges (e.g., 1-20, 15-100, and 20 -150 kDa) to ensure optimal data acquisition through balancing resolution and sensitivity [4]. The maximum allowed buffer size for a single TOF spectrum, e.g., aquired on a Bruker Ultraflex instrument, is 0.5 Mb (with 8-bit ADC). Therefore, for a typical experiment (including pooled QC samples), as in our example above, one may expect an initial (unprocessed) data set size of 0.3-0.5 Gb per mass range. The intelligent data compression provided by down-sampling [4,6] allows proportionally faster data processing of such data sets to produce **aligned peak intensities**. The experimental meta-data (spectrometer and acquisition settings [1]) associated with such studies and saved along with the corresponding MS spectra are used to optimize signal processing models and parameters and allow alignment of the data generated from different experiments or at different time [5].

1.3 Methods

The **rTOFsPRO package** is an enhanced equivalent of TOFsPRO Matlab toolbox (see matlabcentral/fileexchange/24469), with added capability to perform preliminary global alignment or initial delay correction in TOF. The "main" function links libraries to perform signal processing of the input tofList structure (using tofListMetaData parameters) to produce alignedPeakList output. "alignedPeakList$peaks" contains aligned peak positions, while

alignedPeakList$data[[i]] includes intensity and uncertainty information for each input spectrum "i" from the tofList. (See WMBrukerParser CRAN R-package [1] for detailed description of tofList structure and meta-data). Down-sampled (compressed) and filtered spectra are saved in tofResampled structure (similar to tofList). tofList, tofListMetaData, peakList, peakListMetaData, alignedPeakList and alignedPeakListMetaData are "reserved" names for data structures processed by this package. (See Chapter 3 of this "Overview" for the detailed description of the input and output data structures.) When new/modfied input is provided, the appropriate structure name should be used when saving it (as .Rdat) for further processing, e.g.: save(tofList, file="newInput_tofList.Rdat")

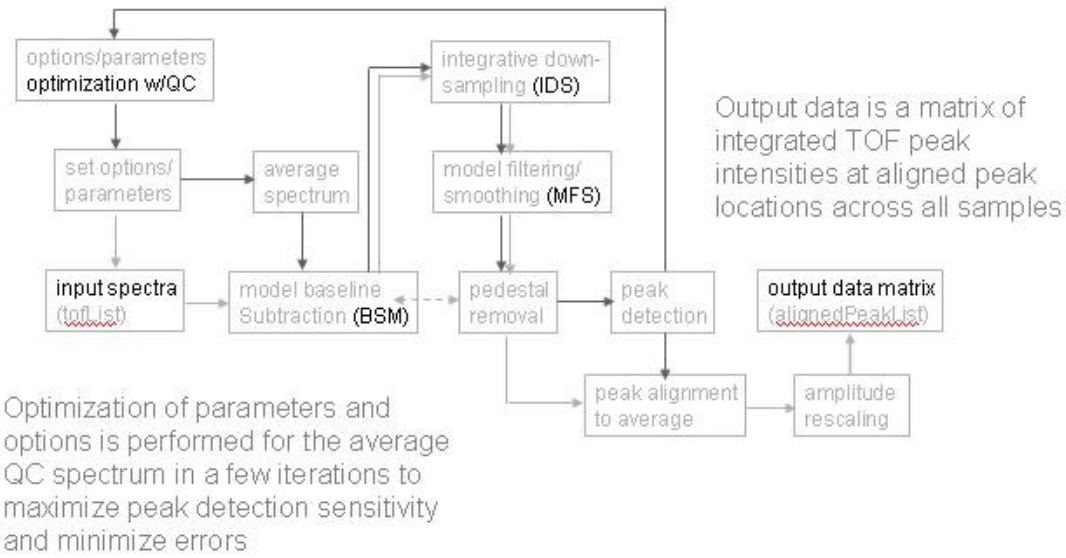

Figure 1: rTOFsPRO signal processing workflow.

Options and parameters for processing are set in "OptionsAndParameters.txt" (Ch 2.3). The comments next to the parameter fields in this file provide instructions on how to optimize and change options and parameters for signal processing of a particular data set. The current (default) settings are optimized for an example "qc11" data set included with the package. The example contains eleven linear MALDI-TOF broad mass-range pooled-serum QC spectra (C3 beads), whose acquisition is described in [6]. The parameter optimization for a new data set is best performed using the average spectrum of QC replicates associated with the data. The user is provided with the option to run the package in "average_only" mode that automatically calculates the average spectrum for input data. This spectrum can then be used for parameter optimization, using "mainRTOFsPROcModular" function (Ch 2.1). The **"modular" main function** allows several debugging options useful for monitoring the code performance and easy tracking of processing function calls. However, due to its "modular" nature, it is using more memory, compared to its **"production"** mode counterpart "mainRTOFsPROc" (Ch 2.2), since it creates multiple copies of the input data set (for each processing step). When the performance is optimized and debugged for a small data example (~ dozen of spectra), the user may run larger data sets (hundreds of spectra) in the production mode using non-modular "main" function with the same "OptionsAndParameters".

The complete signal processing workflow, highlighting available methods is illustrated in Figures 1 and 2. The first block is a **setup** to load "ParametersAndOptions", processing libraries (Ch 2.4-2.6), and

.Rdata files containing tofList and tofListMetaData R-structures. When a single data set is spread over several input files (due to memory limitations [1]), the data can be loaded from multiple files and concatenated into a single data set after compression by down-sampling. The "setup" block is followed by a first processing block that enables **global pre-alignment** of the spectra in TOF domain, either using automated detection of required shifts or applying pre-assigned shift parameters as determined from the meta-data. These shifts are usually observed between TOF records due to triggering errors or slightly different settings of time-delay for acquisition [3, 5]. The automated global shift detection is performed using correlation between pivot peaks common for the spectra, and may be time-consuming, especially for long and noise records. To save time, the shift detection can be performed before/outside the "production" run. The measured shift can be manually added to meta-data ("offset") to, then, directly "apply" to the spectra, by setting appropriate "global alignment" parameter.

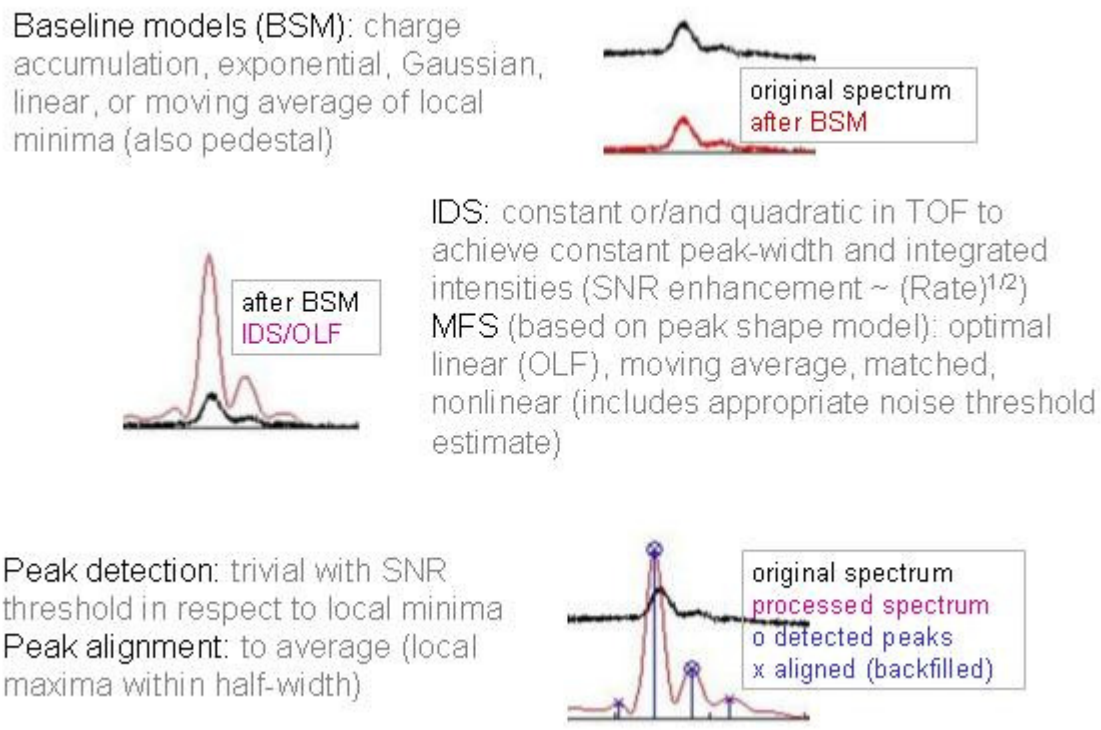

Figure 2: Signal processing methods.

The first signal processing block (Ch 2.4), following global pre-alignment, is **background correction**. It includes several baseline model (BSM) options: RC-based charge accumulation model [3], analytical baseline model (exponential, Gaussian or linear), or constant (ADC offset) correction for slowly varying trend underlying MS peaks. Baseline correction is followed by **integrative down-sampling** (IDS) block (Ch 2.5), whose effect is equivalent to peak-width dependent signal integrator. The down-sampling has a constant rate component in the early TOF range where the peak-width does not change, and quadratic rate for the later TOF [4,6]. IDS enhances signal-to-noise per spectrum point, efficiently compresses the data and restores the constant point density per peak, required for down-stream filtering. The down-sampling is followed by **model filtering/smoothing** (MFS) for further noise suppression, using different signal wavelet target shapes or moving average (MAV). Optimal linear filter (OLF) option allows most efficient noise suppression without broadening. Nonlinear filter (NLF) preserves the noise and doubles

the resolution, while matched filter (MF) most efficiently suppresses noise at the expense of broadening the signal [3, 4]. Following de-noising, **pedestal removal** block is offered to correct for residual pedestals in the down-sampled spectra due to, e.g., peak overlap or presence of the meta-stable ion background. Pedestal trend is estimated by MAV of local minima over several peak-widths. This block completes signal processing of the spectra before peak detection.

Trivial **peak detection** using the average spectrum is currently provided as the only "active" option in this package (Ch 2.6). This algorithm is based on the first difference method of finding local maxima (see e.g., Cromwell toolbox: http://bioinformatics.mdanderson.org/cromwell.html) with added noise threshold in respect to local minima around a tentative peak. The **noise amplitude** estimate is done automatically prior to peak detection by finding standard deviation of noise in down-sampled spectrum and adjusting for the expected effect of filtering [4]. Poisson dependence on baseline amplitude option is also included to account for dark current amplification observed for early TOF (e.g., for Bruker Ultraflex). The uncertainties in peak position and amplitude are recorded along with the corresponding values in the "peakList" structure. A place-holder is also provided for **high-precision** peak detection and peak alignment block using W&M likelihood method [5: U.S. Patent No. 7,219,038]. To "activate" this option, the users would need to contact the authors for the corresponding libraries and source them, as described in the example (see peak detection Note in "OptionsAndParameters").

Following peak detection, **peak alignment** is currently performed in respect to the peak list of average spectrum within the peak position uncertainty. Note that in case of alignment to average, the detection of the "peakList" for the individual records is optional, and may be omitted to save time in the "production" mode. The option is also provided to perform peak alignment to the peak list detected for the **training** set (read from file). After alignment, the peak amplitudes are scaled (to adjust for down-sampling) to represent integrated intensities rather than peak maxima. The aligned peak amplitudes, positions, corresponding uncertainties, and alignment shifts, are saved in "alignedPeakList" structure of the output. The functions of "MetaMatUtils" library (Ch 2.7) create the corresponding meta-data and Matlab-formatted output, if desired.

1.4 Examples

Two examples are included in this package. The subdirectory "Examples" contains input files "tofListPCAqc11.Rdat" and "tofListMetaDataPCAqc11.Rdat", and the two OptionsAndParameters text files. These latter files contain parameters required by the routines in the package and are intended to be templates edited by the user for their purposes/inputs. To change the working directory to point to new data, the user would need to edit the options and parameter file (which should be placed in the same directory) to point to the appropriate directory for IO, e.g.:

```
TOFfileDIR  <-"C:/ProgramFiles/R/rw2011/UserIOdataDIR"
```

The examples may also be run using library-based rTOFsPRO (included with this distribution in "inst/Doc/origLibs/"):
(1) move the data example files ("tofListPCAqc11.Rdat" & "tofListMetaDataPCAqc11.Rdat") to your working directorty (one level above rTOFsPRO);
(2) execute mainRTOFsPROc or mainTOFsPROc modular functions in R-environment:

```
> source(paste("rTOFsPRO/mainRTOFsPROC.txt",sep="/")) # or
> source(paste("rTOFsPRO/mainRTOFsPROCModular.txt",sep="/"))
```

1.4.1 Example 1: Single input-file processing (in "modular" mode)

The parameters contained in "OptionsAndParametersPCAqc11.txt" are used for the first example. They are optimized for the broad-range linear Ultraflex II MALDI-TOF spectra [6] for pooled serum QC collected with clinical PCA data. The input data example is included for 11 QC spectra in "tofListPCAqc11.Rdat": "tofList", and the corresponding meta-data is in "tofListMetaDataPCAqc11.Rdat". To set the parameters (and read the input structures), execute:

```
> source(paste("directoryPATH/OptionsAndParametersPCAqc11.txt",sep="/"))
```

Follow the instructions of "OptionsAndParamters" text to try different options and interpret the output. To debug, monitor memory usage and save all the binary output files (.Rdat and .mat), you may change the corresponding global settings to TRUE:

```
monitorMemory <- TRUE # monitor memory usage in compression mode
debug <- TRUE # do debugging for provided data-example. NOTE: only used by "main...Modular"
saveOut <- TRUE # save aux-output in "...Test.Rdat" files and remove from workspace
save4Mlb <- TRUE # save output for Matlab in .mat
```

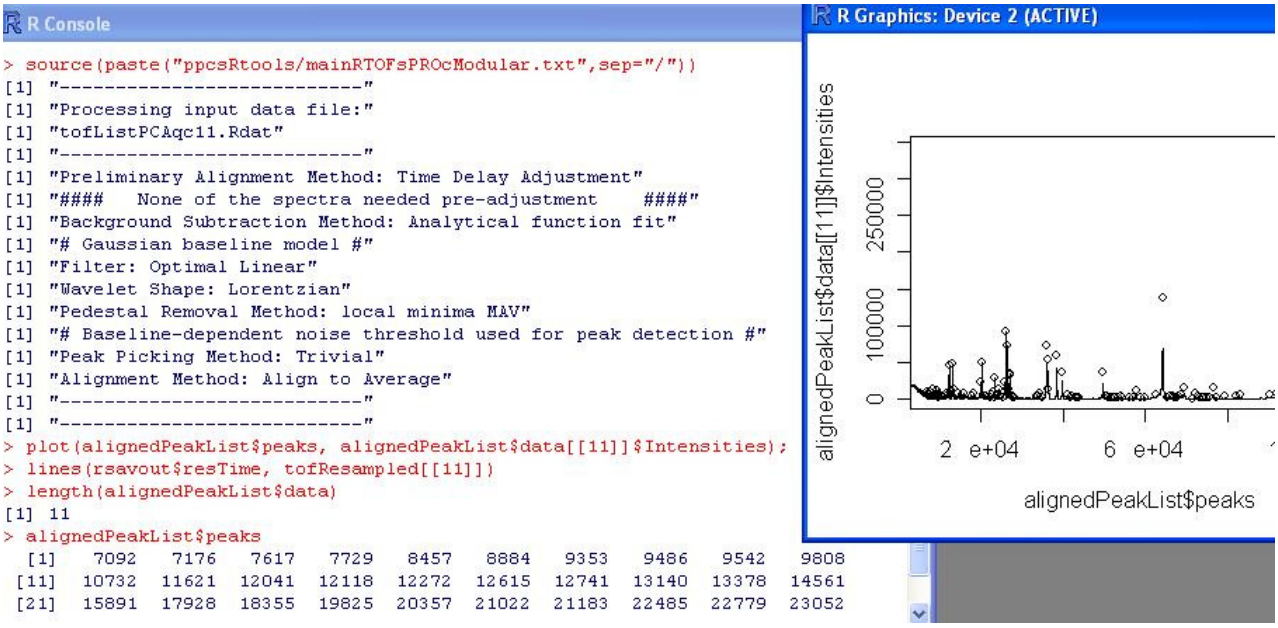

Figure 3: R-console output and a spectrum plot for Example 1.

To execute the Example 1, included with the package, type:

```
> directoryPATH <- system.file("Examples", package = "rTOFsPRO")
> outList <- mainRTOFsPROcModular(paste(directoryPATH,"/OptionsAndParametersPCAqc11.txt",sep=""))
> print(outList$alignedPeakList); print(outList$alignedPeakListMetaData);
```

The example of the output that reports on the successful completion of the signal processing blocks is provided in Figure 3. The resulting alignedPeakList structure (Ch 3.8) is available within the workspace and from "alignedPeakList_Test.Rdat" (e.g., load("alignedPeakList_Test.Rdat")). To access detected aligned peak positions, use "alignedPeakList$peaks". For aligned intensities of spectrum tofList[[i]], type alignedPeakList$data[[i]]$Intenisities. The other accessible "data" fields include "PositionsShifts", "PositionUncertainties" and "IntenistyUncertainties". Similar information can be output for the peakList structure (before alignment).

1.4.2 Example 2: Multiple input compression/concatenation (in "production" mode)

The settings in "OptionsAndParametersPCAqc33.txt" are for a "production" processing of three concatenated input "qc11" files. The input file name parameters that differ from those in "OptionsAndParametersPCAqc11" used in Example 1 follow:

```
---------------------------------------------------------------------------------------------
  rTofFileNames<-c("tofListPCAqc11.Rdat","tofListPCAqc11.Rdat","tofListPCAqc11.Rdat")
  rTofMetaFileNames<-
c("tofListMetaDataPCAqc11.Rdat","tofListMetaDataPCAqc11.Rdat","tofListMetaDataPCAqc11.Rdat")
```

```
> source(paste("ppcsRtools/mainRTOFsPROc.txt",sep="/"))
[1] "----------------------------"
[1] "Processing input data file:"
[1] "tofListPCAqc11.Rdat"
[1] "----------------------------"
[1] "Preliminary Alignment Method: none"
[1] "Background Subtraction Method: Analytical function fit"
[1] "# Gaussian baseline model #"
[1] "Filter: Optimal Linear"
[1] "Wavelet Shape: Lorentzian"
[1] "Pedestal Removal Method: local minima MAV"
[1] "----------------------------"
[1] "Processing input data file:"
[1] "tofListPCAqc11.Rdat"
[1] "----------------------------"
[1] "Preliminary Alignment Method: none"
[1] "Background Subtraction Method: Analytical function fit"
[1] "# Gaussian baseline model #"
[1] "Filter: Optimal Linear"
[1] "Wavelet Shape: Lorentzian"
[1] "Pedestal Removal Method: local minima MAV"
[1] "----------------------------"
[1] "Processing input data file:"
[1] "tofListPCAqc11.Rdat"
[1] "----------------------------"
[1] "Preliminary Alignment Method: none"
[1] "Background Subtraction Method: Analytical function fit"
[1] "# Gaussian baseline model #"
[1] "Filter: Optimal Linear"
[1] "Wavelet Shape: Lorentzian"
[1] "Pedestal Removal Method: local minima MAV"
[1] "# Baseline-dependent noise threshold used for peak detection #"
[1] "Peak Picking Method: Trivial"
[1] "Alignment Method: Align to Average"
[1] "----------------------------"
[1] "----------------------------"
> length(alignedPeakList$data)
[1] 33
```

Figure 4: R-console output for data concatenation and "production" mode processing in Example 2.

To execute the Example 2, included with the package, type:

```
> directoryPATH <- system.file("Examples", package = "rTOFsPRO")
> outList <- mainRTOFsPROc(paste(directoryPATH,"/OptionsAndParametersPCAqc11.txt",sep=""))
> print(outList$alignedPeakList); print(outList$alignedPeakListMetaData);
```

Figure 4 illustrates the output for Example 2, reporting on data concatenation and processing steps. The debugging option is not available for mainRTOFsPROc (non-modular). You may monitor the memory usage and compare it to "modular" mode by setting: monitorMemory <- TRUE. Check that the output from alignedPeakList structure is similar to Example 1, except the length(alignedPeakList$data) is 33.

1.5 Concerns

(1) Sometimes, **empty spectra** result from robot errors in spotting the material on MALDI plate. It is recommended that "empty" spectra are removed from input data before signal processing, e.g., by viewing the data in "movie mode". If left, empty spectra may corrupt baseline model optimization for average QC spectrum and cause errors for peak detection from the individual clinical spectra. They would also add unnecessary noise in downstream statistical analysis of aligned peak intensities.

(2) Since MS signal processing includes an alignment step, it would be reasonable to concatenate all spectra from all runs in a given study into a single tofList. However the data files are large and there is a limit to the amount of memory that R has available. Signal processing requires twice the memory compared to simply load the data (thee-times in the "Modular" mode. According to the R Documentation (on Package *utils* version 2.1.0) for *memory.size* and *memory.limit*, the Windows version of R is usually limited to 2GB of memory. To insure the efficient use of memory, the data can be loaded from separate files, **compressed by down-sampling** and concatenated into tofResampled structure of about 10-fold smaller than original tofList size. Even with this precaution, it is possible to exceed the memory available during loading (e.g., when down-sampling is less effective for low mass-range high-resolution data). As discussed above, an option has been included to print memory use during processing and save the concatenated tofResampled object.

(3) It is helpful to consider the size of the "tofList Rdata" files, and the memory required by R to load and process the data (as in Example 2). The "tofListMetaData" is always about thousand times smaller (~10 kB), and is unlikely to be a limiting factor for memory usage, compared to "tofList.Rdata". Note that *.Rdata* files maybe compressed (about 2-fold) on disk, while when loaded, they are automatically uncompressed to the full size before processing. Thus, the memory required for loading and processing the data may be greater by more than a factor of two times their size on disk. The concatenated "tofList" files are approximately as large as the sum of the individual files on disk and in R. In case of prohibitively large data (even after down-sampling), it is best to subdivide the input tofLists into manageable data files with somewhat overlapping spectra content, and produce and compare the **"training" aligned peak lists** from the "average_only" spectra. Then, the master "training" peak list can be used for alignment of separate tofList files, and the resulting alignedPeakList structures and meta-data can be concatenated outside the main processing workflow.

(4) Several **loop-based steps** that process tofList spectra are naturally slow for long records (before/without down-sampling): e.g., global alignment, RC-baseline model or pedestal removal. Some

work-arounds are possible to save time for global alignment, e.g.: the shift detection can be performed outside the "production" run. The measured shift can be manually added to meta-data ("offset") to, then, apply to the spectra in "production" mode. The best long-term solution (in future) would be to re-program the time-limiting functions in low-level language, like C.

1.6 Conclusions and Recommendations

rTOFsPRO package allows efficient signal processing of the TOF spectra over the broad mass range (2-150 kDa). The parameters and options can be optimized for piecewise as well as global processing in different mass-ranges. The output provides aligned peak positions and intensities, as well as, associated uncertainties, depending on the processing methods. This information is useful for downstream statistical analysis in comparative proteomics applications.

The signal processing functions of the package are combined into libraries ("Lib…txt") based on functionality. In addition to individual functions, the libraries are included with this package distribution as text files in "inst/Doc/" directory. In addition to regular CRAN installation, the users interested in developing package extension have an option to run and edit a library-based package version, which allows easier access to code. To run library-based version of the package, the users would need to copy the data examples into their working directory (one level above the directory containing library functions, e.g. "rTOFsPRO/"), and source one of the main functions, e.g.:

```
> source(paste("rTOFsPRO/mainRTOFsPROCModular.txt",sep="/"))
```

Brief description of the input/output and signal processing libraries (Ch 2.4-2.6) can be found in the header of "main.." functions, and in comments for individual processing blocks. The user is provided with an option to save "intermediate" output between processing blocks, like "tofResampled" or "peakList", in addition to alignedPeakList, in R-binary ("…_Test.Rdata") and Matlab-specific *(.mat)* files.

Note that with the down-sampling included in the signal processing steps the data is compressed (~10-fold) for each input file separately, which in many cases, allows concatenation of the full/large data set after down-sampling before alignment. This also speeds-up signal processing of individual (down-sampled) spectra proportional to compression efficiency. The memory usage has been improved ~3-fold in "mainRTOFsPROc", compared to "Modular" by eliminating original data copies and combining multiple loops through the spectra (in separate processing modules) into one main loop. The signal processing currently takes about 1 s per linear TOF spectrum (of 0.5 Mb, ~ 130K points) in R on 2 GHz processor (with 1 Gb of RAM). The algorithms and methods are applicable to high-resolution (reflectron) data as well, however with less efficient data compression and generally longer processing times. The time-limiting step remains the loop through the spectra and through each record for global alignment, pivot detection, charge-accumulation baseline model, and down-sampling. This performance could be improved (~10-fold) by reprogramming the time-limiting functions in low-level language.

2 Routines in the rTOFsPRO Package

2.1 mainRTOFsPROcModular

Description

This routine performs signal processing of tofList spectra from the input file according to the setting and parameters of “OptionsAndParameters.txt” to produce alignedPeakList and meta-data. The “modular” structure allows several debugging options useful for monitoring the code performance and easy tracking of processing function calls in “optimization” mode.

Usage

```
mainRTOFsPROcModular(OPfileName)
```

Arguments

OPfileName – options and parameters file name

Details

```
##================================================    ##
##                                                    ##
##        This script links and executes the functions of  rTOFsPRO     ##
##        package for signal processing of the linear TOF spectra       ##
##        to produce aligned peak intensity matrix (and peak positions) ##
##        that can be used for further statistical analysis             ##
##                                                    ##
##      >>>best used in "setup" or "optimization" mode<<<             ##
##        >>>  with small data sets < 50 spectra    <<<               ##
##                                                    ##
## OUTPUT:                                            ##
##                                                    ##
## alignedPeakList and alignedPeakListMetaData structures containing  ##
## on peak intensities and positions and their uncertainties, as well ##
## as clinical and instrumental meta-data.            ##
## "..._Test.RDat" files save auxhiliary info and data structures,    ##
## if desired (e.g. tofResampled and peakList) -- see documentation   ##
##                                                    ##
## INPUT:                                             ##
## tofList and tofListMetaData structures loaded from .Rdat files that ##
##  are specified in “OptionsAndParameters.txt”, also the rest of     ##
## signal processing parameters required by the following processing steps ##
```

```
##                                                                    ##
##      PROCESSING STEPS INCLUDE (in a loop over spectra)             ##
##                                                                    ##
##      Set up:                                                       ##
##            Load "OptionsAndParameters.txt" file                    ##
##            Load processing libraries                               ##
##                                                                    ##
##      Load Data:                                                    ##
##            Load Data and MetaDataSet parameters                    ##
##            "multiple files" Loop begins (See note 1)               ##
##                                                                    ##
##      Background Correction. Options include:                       ##
##            RC (charge accumulation model)                          ##
##            Analytical (Gaussian, Exponential, linear) model        ##
##            None (default)                                          ##
##                                                                    ##
##      Preliminary alignment. Options include:                       ##
##            Global alignment (using correlation between pivot peaks)##
##            TOF offset adjustment (from metaData)                   ##
##            None                                                    ##
##                                                                    ##
##      Downsampling and Filtering. Options include:                  ##
##            Integrative downsampling (IDS) only                     ##
##            IDS and Optimal Linear Filter (default)                 ##
##            IDS and Nonlinear Filter                                ##
##            IDS and Matched Filter                                  ##
##            IDS and MAV Filter                                      ##
##                                                                    ##
##      Pedastal removal. Options include:                            ##
##            MAV of local minima                                     ##
##            None (default)                                          ##
##                                                                    ##
##      Peak Picking. Options include:                                ##
##            Trivial (1st difference with local minim SNR, Note 2)   ##
##            W & M (Maximum Likelihood Method) (needs extra lib)     ##
##            (call commented off: lib available from wecook@wm.edu)  ##
##                                                                    ##
##      Alignment. Options include:                                   ##
##            Align to average spectra                                ##
##            W & M global+binning (call commented off:               ##
##            seprate lib available from wecook@wm.edu)               ##
##                                                                    ##
#====================================================================#
# Note 1. Due to the large size of data files, in "compression" mode  #
# raw tofList data can be loaded and partially processed by IDS from  #
# multiple files. The "multiple TOFs loop" includes blocks from "Load  #
```

```
# data" through "Downsampling and Filtering".At the end of the loop, the      #
# downsampled spectra and their meta-data are concatenated and the             #
# original tofList is removed.                                                 #
#==============================================================#
# Note 2. Basic algorithm adopted from Matlab-based Cromwell toolbox.          #
# (Copyright 2005.) Originally developed by MD Anderson and distributed       #
# under "BSD" Mathworks license: http://bioinformatics.mdanderson.org/        #
# cromwell.html. Local minima SNR threshold and baseline-dependent            #
# noise model for each preliminary filtering option was added here.           #
##============================================================##
```

Value

alignedPeakList - A list of time-of-flight mass spectrum vectors addressable by spectrumName.
alignedPeakListMetaData - A data.frame containing string values for experimental meta-data, with rows and columns addressable by spectrumName and attributeName.

Author

Dariya Malyarenko, College of William and Mary

Note

Main function for “optimization” mode

2.2 mainRTOFsPROc

Description

This routine performs signal processing of tofList spectra from the input file according to the setting and parameters of “OptionsAndParameters.txt” to produce alignedPeakList and meta-data. The processing is similar to “mainRTOFsPROcModular”, except it is more efficient for multiple input files in “production” mode (no “debugging” is enabled).

Usage

```
mainRTOFsPROc(OPfileName)
```

Arguments

OPfileName – options and parameters file name

Details

```
##================================================     ##
##                                                      ##
##        This script links and executes the functions of  rTOFsPRO      ##
##        package for signal processing of the linear TOF spectra        ##
##        to produce aligned peak intensity matrix (and peak positions)  ##
##        that can be used for further statistical analysis              ##
##                                                                       ##
##                >>> best used in PRODUCTION mode <<<                  ##
##         >>> for larger data sets (as memory permits) <<<              ##
##                   >>> no "debug" option <<<                           ##
##                                                                       ##
## OUTPUT:                                                              ##
##                                                                       ##
## alignedPeakList and alignedPeakListMetaData structures containing    ##
## on peak intensities and positions and their uncertainties, as well   ##
## as clinical and instrumental meta-data.                              ##
## "..._Test.RDat" files save auxhiliary info and data structures,      ##
## if desired (e.g. tofResampled and peakList) -- see documentation     ##
##                                                                      ##
## INPUT:                                                               ##
## tofList and tofListMetaData structures loaded from .Rdat files that  ##
##  are specified in “OptionsAndParameters.txt”, also the rest of       ##
## signal processing parameters required by the following processing steps ##
##                                                                      ##
##        PROCESSING STEPS INCLUDE (in a loop over spectra)             ##
##                                                                      ##
##        Set up:                                                       ##
##                Load "OptionsAndParameters.txt" file                  ##
##                Load processing libraries                             ##
##                                                                      ##
##        Load Data:                                                    ##
##                Load Data and MetaDataSet parameters                  ##
##                "multiple files" Loop begins (See note 1)             ##
##                                                                      ##
##        Background Correction. Options include:                       ##
##                RC (charge accumulation model)                        ##
##                Analytical (Gaussian, Exponential, linear) model      ##
##                None (default)                                        ##
##                                                                      ##
##        Preliminary alignment. Options include:                       ##
##                Global alignment (using correlation between pivot peaks) ##
##                TOF offset adjustment (from metaData)                 ##
##                None                                                  ##
##                                                                      ##
```

```
##        Downsampling and Filtering. Options include:                    ##
##                Integrative downsampling (IDS) only                     ##
##                IDS and Optimal Linear Filter (default)                 ##
##                IDS and Nonlinear Filter                                ##
##                IDS and Matched Filter                                  ##
##                IDS and MAV Filter                                      ##
##                                                                        ##
##        Pedastal removal. Options include:                              ##
##                MAV of local minima                                     ##
##                None (default)                                          ##
##                                                                        ##
##        Peak Picking. Options include:                                  ##
##                Trivial (1st difference with local minim SNR, Note 2)   ##
##                W & M (Maximum Likelihood Method) (needs extra lib)     ##
##                (call commented off: lib available from wecook@wm.edu)  ##
##                                                                        ##
##        Alignment. Options include:                                     ##
##                Align to average spectra                                ##
##                W & M global+binning (call commented off:               ##
##                seprate lib available from wecook@wm.edu)               ##
##                                                                        ##
#=========================================================#
# Note 1. Due to the large size of data files, in "compression" mode     #
# raw tofList data can be loaded and partially processed by IDS from     #
# multiple files. The "multiple TOFs loop" includes blocks from "Load    #
# data" through "Downsampling and Filtering".At the end of the loop, the #
# downsampled spectra and their meta-data are concatenated and the       #
# original tofList is removed.                                           #
#=========================================================#
# Note 2. Basic algorithm adopted from Matlab-based Cromwell toolbox.    #
# (Copyright 2005.) Originally developed by MD Anderson and distributed  #
# under "BSD" Mathworks license: http://bioinformatics.mdanderson.org/   #
# cromwell.html. Local minima SNR threshold and baseline-dependent       #
# noise model for each preliminary filtering option was added here.      #
##=========================================================##
```

Value

alignedPeakList - A list of time-of-flight mass spectrum vectors addressable by spectrumName.
alignedPeakListMetaData - A data.frame containing string values for experimental meta-data, with rows and columns addressable by spectrumName and attributeName.

Author

Dariya Malyarenko, College of William and Mary

Note

Main function for “production” mode

2.3 OptionsAndParameters

```
#########################################################################
## This script sets options and parameters for signal processing performed    ##
## by "mainRTOFsPROc" and "mainRTOFsPROcModular"                              ##
##                                                                            ##
## USAGE:                                                                     ##
## source(paste("directoryPATH/OptionsAndParametersPCAqc11.txt",sep="/"))     ##
## Dependency: none                                                           ##
## NOTE: Current default settings are optimized for the broad-range linear    ##
## Ultraflex II MALDI-TOF spectra [6],                                        ##
## e.g. included in "tofListPCAqc11.Rdat" example: "tofList".                 ##
##                                                                            ##
## For new TOF data, the parameters should be optimized using average         ##
## spectrum for the QC data set and following instructions for each field     ##
## below using corresponding auxiliary functions of the package or            ##
## built-ins. For input/output and usage description for auxiliary            ##
## functions, type: help(function_name);                                      ##
##                                                                            ##
## m/z calibration should be performed externally for an average spectrum, and ##
## corresponding calibration parameters can be used. Otherwise, calibration   ##
## constants are read from meta-data. Same m/z axis is assumed for all spectra. ##
#########################################################################
```

2.4 LibBSalgos functions

userError – auxiliary function to stop execution with the informative message when error occurs while executing a library function
chargeBkgr_call - calculates baseline due to capacitor- detector amplifier coupling (charge accumulation)
offsetBkgr_call -- calculates baseline from ADC offset
ExpBkgr_call - estimates slow baseline according to analytical model
findBsEnv - automatically finds baseline envelope for “ExpBkgr_call”
pedRmMAV_call - estimates baseline/pedestals using moving average of local minima

2.5 LibIDSfilts functions

pivpeak -- pivot peak detection by first difference
hwfmGausPeak- estimates left half-width of a peak, called by "peakWidth"
peakWidth- empirically measures peak half-width across the TOF record
msResample - integrative down-sampling of the data according to observed half-width dependence on TOF (to recover constant point density per peak)
lorLHalf\lorHalf - calculated left\right half of Lorentzian wavelet truncated

to instrumental precision
gausHalf\gausRHalf - calculates left\right half of Gaussian wavelet truncated to instrumental precision
myxcorr – Matlab-like calculation of cross-correlation
circShift – Matlab-like circular shift function for a vector
coefTrunc - truncates filter coefficients to avoid periodic artifacts
coefOptFiltGL\LL\GG - calculates optimal linear filter coefficients for half-Gaussian-half-Lorentzian (GL) \LL\GG signal wavelet [4]
coefNLFiltGL\LL\GG - calculates three non-linear filters assuming GL\LL\GG - wavelet
optFilt - applies the optimal smoothing (non-broadening) filter to input signal [4]
geomavFilt - applies non-linear geometric average deconvolution (half-narrowing) filter to input signal [4]
matchFiltCoefGG\LL\GL - calculates matched filter coefficients for GG\LL\GL signal wavelet
matchedFilter - applies matched filter (40% - broadening) filter to input signal.
trgFilt – solves Toeplitz matrix equation for target filter coefficients [4]
mavSmoothing – moving average smoothing

2.6 LibPPalignSNREst functions

trivPP_call - peak detection of local maxima above SNR threshold in respect to global baseline and local minima (see Cromwell by MDAnderson)
peakAlignAv - peak alignment within half-width in respect to the peak list of average spectrum
noiseEstmtr2 - estimates STD of high-frequency noise for TOF spectrum assuming constant peak width
pivPeak – pivot peak finder
locjitdet – detection of local jitter near a peak by correlation of the spectra range
globalAlign – global alignment call for records

2.7 LibMetaMatUtils functions

plMeta – peakList meta-data compiler
aplMeta – aligned peak list meta-data compiler
OutputForML – saves output for Matlab in ".mat"-format

3 Input and Output Data Stuctures

3.1 tofList (generated, e.g., by WMBrukerParser)

tofList - A list of time-of-flight mass spectrum vectors addressable by spectrumName.

To access a spectrum vector:

```
spectrum <- tofList[[ spectrumName ]];
```

For example:

```
spectrum <- tofList[[ "pool 2_8" ]];
```

Alternately, spectrum vectors can be addressed by spectrum index. This option should be used with caution since data can be reordered during processing.

To update a spectrum's vector:

```
 tofList[ spectrumName ] <- list(spectrum);
```

To update the tofList names (desirable if default assignments were used due to missing sampleInfo.sampleNames meta-data):

```
newSpecNames<-list();
numSpec<-length(tofList);
for (i in 1:numSpec) { newSpecNames[i]<-paste("spec", i , sep="") };
names(tofList)<-newSpecNames;
```

3.2 tofListMetaData

A data.frame containing string values for experimental meta-data, with rows and columns addressable by spectrumName and attributeName.

To access a value:

```
    sampleName <- tofListMetaData[ spectrumName, attributeName ];
```

For example:

```
    sampleName <- tofListMetaData["pool 2_8", " sampleInfo.sampleName " ];
```

Alternately, meta-data can be addressed using spectrum and attribute indices. Again, this option should be used with caution since data can be reordered during processing.

To update a value (which is desirable if tofListMetaData includes NAs due to missing optional parameter files):

```
    tofListMetaData[ spectrumName, attributeName ] <- "Test";
```

To update the tofListMetaData rownames (desirable if default assignments were used due to missing sampleInfo.sampleNames)

```
newSpecNames<-list();
numSpec<-length(tofList);
for (i in 1:numSpec) { newSpecNames[i]<-paste("spec", i , sep="") };
row.names(tofListMetaData)<-newSpecNames;
```

A partial listing of a sample tofListMetaData structure is shown in the lower right of Figure 2.

Meta-data attributes include:

acquisitionInfo.arrayBarcode – target (Bruker plate) identifier string, includes target and target serial number - *Required*
acquisitionInfo.ionPolarity – "positive" or "negative" - *Required*
acquisitionInfo.refId – spectrum identifier, (same as experiment.id), used for spectrum vector names in tofList and rows names in tofListMetaData (generated by concatenating strings: sampleInfo.sampleName,"_", and replicateNumber. If sampleInfo.SampleName is unavailable, a default assignment of the directory name is used.) - *Optional*
acquisitionInfo.sourceFile – file name (with path) containing binary mass spectrum data, (from directory structure) - *Automatic*
acquisitionInfo.spotIndex – spot index on MALDI plate - *Automatic*
acquisitionInfo.spotName – spot name on MALDI plate - *Automatic*
adcGain.high – digitizer gain factor - *Optional*
adcGain.low – linear detector voltage - *Optional*
array.affinityType – capture agent used for purification of biofluid samples - *Optional*
deflector.mode – 1=linear, 2=quadratic - *Optional*
experiment.id – same as acquisitionInfo.refId - *Optional*
instrument.instrumentType – "Ultraflex 3" (hard coded) - *Automatic*
instrument.instrumentVendor – name of instrument vendor - *Required*
instrument.serial – instrument serial number - *Required*
instrumentSpecificSettings.adcBandwidth – digitizer bandwith limit - *Optional*
instrumentSpecificSettings.adcOffset – digitizer offset, depends on spectrum type and gain factor - *Optional*
instrumentSpecificSettings.adcScale – digitizer sensitivity, depends on gain factor - *Optional*
instrumentSpecificSettings.detectorBaseVoltage – detector base voltage, depends on spectrum type - *Optional*
instrumentSpecificSettings.laserIntensityBaseRange – minimum laser power - *Optional*
instrumentSpecificSettings.laserShots – number of laser shots per spot - *Required*
instrumentSpecificSettings.timeZero – intial time in counts (calculated) - *Required*
instrumentSpecificSettings.TOFmode – spectrometer mode: LINEAR or REFLTOR – *Optional*
instrumentSpecificSettings.warmingShots – number of warming laser shots - *Optional*
laserIntensity.units – % (hard coded) - *Automatic*
laserIntensity.value – maximum laser power, percent of full scale - *Optional*
laserIntensityWarming.units – % (hard coded) - *Automatic*
laserIntensityWarming.value – warming Laser power, percent of full scale - *Optional*
mass.units – "Da" (hard coded) - *Automatic*
mass.value – low mass cut off for matrix suppression - *Optional*
massCalibration.calibrationSpectrumFile – calibration folder name. (from folder name containing "calibration." If unavailable "?" is assigned as default)

- *Optional*
massCalibration.dateCalibrated – calibration date - *Required*
massCalibration.equation – "c1*((T0+(X-1)*Tdelta)/U)^2+c0*((T0+(X-1)*Tdelta)/U)+c2" (hard coded) - *Automatic*
massEnd.units – mass units, "Da" (hard coded) - *Automatic*
massEnd.value – highest mass in acquisition range – *Optional*
massStart.units – mass units, "Da" (hard coded) – *Automatic*
massStart.value – lowest mass in acquisition range - *Optional*
param.c0 – linear coefficient in calibration equation - *Required*
param.c1 – quadratic coefficient in calibration equation - *Required*
param.c2 – constant term in calibration equation - *Required*
param.Mode – calibration mode: 1=linear, 2= quadratic - *Optional*
param.T0 – digitizer delay in ns - *Required*
param.TDelta – dwell time in ns - *Required*
param.U – nanoseconds/milliseconds conversion factor, "1e6" (hard coded) - *Automatic*
replicateNumber – index of occurrence of sampleInfo.sampleName - *Automatic*
sampleInfo.groupName – clinical group that sample belongs to, used for classification analysis - *Optional*
sampleInfo.sampleDescription – additional sample description - *Optional*
sampleInfo.sampleName – sample identifier (read from sample.xml, If sample.xml, or the field is missing, the directory name is used as the default value) - *Optional*
sampleInfo.sampleSource – laboratory where data was acquired (user supplied input parameter: ParserParams.dataSource) - *Required*
software.version – version of XACQ software - *Required*
spottingInfo.spotProtocol – laser movement patern during data acquisition - *Optional*
timeOfFlightData.dateCreated – data file creation date (from date stamp on fid file) - *Automatic*
timeOfFlightData.domain – data domain, "time" (hard coded) - *Automatic*
timeOfFlightData.encoding – data encoding, "base64" (hard coded) - *Automatic*
timeOfFlightData.end – number of data points in spectra - *Required*
timeOfFlightData.offset – initial value for offset of spectra, will be updated during alignment procedure, "0.0" (hard coded) - *Automatic*
timeOfFlightData.pairOrder – Order of measurement pairs for TOF data, e.g., time – intensity, "t-int" (hard coded) - *Automatic*
timeOfFlightData.scale – initial value for linear scale coefficient for spectra, will be updated during alignment procedure, hard coded: "1.0" (hard coded) - *Automatic*
timeOfFlightData.start – index for onset of data acquisition, hard coded: "1" – *Automatic*

3.3 average spectrum structures

The results of processing for the average spectrum are saved in several structures for monitoring and optimization:

(1) “savl” and “savbs” are of “tofList” type (Ch 3.1), and contain a single entry corresponding to the original average spectrum and the baseline-subtracted spectrum. These are also saved in "sav_bs_Test.Rdat". “namavl” is a “tofList” containing noise amplitude thresholds for the average spectrum, used for peak detection (similar to “naml” list for noise amplitude of all spectra available from “workspace”).
(2) “avspr1” is the average spectrum after complete signal processing; “rsavout” is a list of “rsout”-type (Ch 3.5), containing down-sampled average spectrum and relevant information after IDS; “tms”, “mast” and “t0” are TOF axis in milisecond, m/z and time index domain, respectively; and “pks3d” is an array of three vectors, containing peaks detected in average spectrum in down-sampled time, original time and “m/z”-domain. These structures are saved in "avSpec_peaks3d_tms_mz_ Test.Rdat".

3.4 tofResampled

“tofResampled” is a list of down-sampled spectra of the same structure as “tofList” (Ch 3.1), obtained after IDS compression. It, generally, includes shorter records of integrated intensities, obtained from the original “tofList” under the same “spectrumIDs”: rsSpectrum <- tofResampled[[spectrumName]]. This intermediate processing results may also be saved in a file "tofResampledAllProc_Test.Rdat".

3.5 rsout

“rsout” structure is a list of output from integrative down-sampling (IDS) routine. “rsout$resTime” contains the time indexes in down-sampled domain; “rsout$resSignal” is the spectrum after IDS; “rsout$resMass” is the list of m/z coordinates corresponding to IDS-spectrum; “rsout$resRate” is the vector of IDS-rates for the spectrum; “resPikw” and “resAsym” are the fields containing new peak-width and asymmetry parameters after IDS.

3.6 peakList

Reserved R-variable name: “peakList"
Structure is a list of 2 or 4 element lists of vectors

```
list(
  "<sampleName>_<replicateNumber>" =
    list(
      "Positions" =
        c( 5536.0, 5572.0, ..., 29236.0 ),
      "Intensities" =
        c( 8249.072, 4351.5454, ..., 683.9336 ),
      "PositionUncertainties" =
        c( 7.0, 6.0, ..., 23.0 ),
      "IntensityUncertainties" =
        c( 66.666046, 66.55046, ..., 66.55046 )
    ),
  ...
)
```

To access a spectrum's peak data:
peakData <- peakList[[spectrumId]];
To access the vectors from peak data (acquired using above code):
peakPositions <- peakData[["Positions"]]
peakIntensities <- peakData[["Intensities"]]
peakPositionUncertainties <- peakData[["PositionUncertainties"]]
peakIntensityUncertainties <- peakData[["IntensityUncertainties"]]
Note that since "PositionUncertainties" and "IntensityUncertainties" are optional, you will need to do a NULL check before trying to use these vectors:
is.null(peakPositionUncertainties)
To construct new peak data:

```
peakData <- list(
    "Positions" = peakTime[1:numberPeaks]+dataStart-1,
    "Intensities" = peakAmps[1:numberPeaks],
    "PositionUncertainties" = timeUncerts[1:numberPeaks],
    "IntensityUncertainties" = ampUncerts[1:numberPeaks] );
```

To update a spectrum's peak list data:
peakList[spectrumId] <- list(peakData);

3.7 peakListMetaData

PeakList metadata: The data format is the same as the tofList metadata file.
variable name: "tofListMetaData", "peakListMetaData", etc.
Stucture is a data.frame (list-based structure with rows and columns addressable by name)

NOTE: The values are returned as strings, and it is the responsibility of the script to convert the string into an appropriate type (e.g., integer, float, etc).

To access a value:
tofListMetaData[spectrumName, "instrument.name"];
To set a value:
tofListMetaData[spectrumName, "instrument.name"] <- "Test";

```
PLmetaDataFields <- c("acquisitionInfo.refId",
"acquisitionInfo.spotName", "peakData.domain", "peakData.offset",
"peakData.scale", "replicateNumber", "sampleInfo.groupName",
"sampleInfo.sampleDescription", "sampleInfo.sampleName", "sampleInfo.sampleSource"
)
```

3.8 alignedPeakList

Reserved R-variable name: "alignedPeakList"
Structure is 2 element list ("peaks", "data").

"peaks" is a vector of aligned peak positions. "data" is a list of 2 or 4 element lists of vectors

```
structure(
  list(
    peaks = c(5656.5, 5707.5, ..., 28221),
    data = structure(
      list(
        "<sampleName>_<replicateNumber>" = structure(
          list(Intensities = c(13348.342, 22649.508, ...,)),
          .Names = "Intensities"),
        ...
      ),
      .Names = c("<sampleName>_<replicateNumber>", ...)
    )
  ),
  .Names = c("peaks", "data")
)
```

To access peak vector:
peaks <- alignedPeakList$peaks;
To access a spectrum's aligned peak data:
alignedPeakData <- alignedPeakList$data[[spectrumId]];
To access the vectors from aligned peak data (acquired using above code):
intensities <- alignedPeakData[["Intensities"]]
positionShifts <- alignedPeakData[["PositionShifts"]]
positionUncertainties <- alignedPeakData[["PositionUncertainties"]]
intensityUncertainties <- alignedPeakData[["IntensityUncertainties"]]
Note that since "PositionShifts", "PositionUncertainties", and "IntensityUncertainties" are optional, you will need to do a NULL check before trying to use these vectors:
is.null(positionShifts)
To create a new aligned peak list with empty data:
alignedPeakList <- list();
alignedPeakList$peaks <- myPeaks;
alignedPeakList$data <- list();
To construct new aligned peak data:

```
alignedPeakData <- list(
    "Intensities" = as.vector( y[ , i ] ),
    "PositionShifts" = zeroVector,
    "PositionUncertainties" = zeroVector,
    "IntensityUncertainties" = zeroVector );
```

To update a spectrum's aligned peak data:
alignedPeakList$data[spectrumId] <- list(alignedPeakData);

3.9 alignedPeakListMetaData

AlignedPeakList metadata: The data format is the same as the tofList metadata structure, but with the shorter attribute list:
APLmetaDataFields = c("acquisitionInfo.refId",

"acquisitionInfo.spotName", "alignedPeakData.domain", "alignedPeakData.offset", "alignedPeakData.scale", "replicateNumber", "sampleInfo.groupName", "sampleInfo.sampleDescription", "sampleInfo.sampleName", "sampleInfo.sampleSource")

4 Acknowlegement

This research was supported by NIH National Cancer Institute R01 Grant CA126118 from the Advanced Proteomics Platforms and Computational Sciences Program within the Clinical Proteomics Initiative of the National Cancer Institute (PI: Malyarenko).

5 References

[1] Tracy, M.B. and Cooke, W.E. *WMBrukerParser R-package* (2009): http://cran.r-project.org/web/packages/WMBrukerParser/index.html

[2] Karbassi, I. D., et al., *J. Proteome Res*., (2009) **8**, p.4182-4189

[3] Cazares, L. H., et al., *Clin. Cancer Res.*, (2009) **15**, p. 554-562

[4] Malyarenko, D. I., et al., *Rapid Commun. Mass Spectrom* (2006) **20**, 1670–1678

[5] Tracy, M.B., et al., *Proteomics* (2008) **6**, 4517-4524

[6] Gatlin-Bunai, C. L., et al., *J Proteome Res* (2007) **6**, 4517-4524

[7] Malyarenko, D.I., et al., *Rapid Commun. Mass Spectrom* (2010) **24**, 138-145