

# Package ‘FTICRMS’

April 17, 2009

**Type** Package

**Title** Programs for Analyzing Fourier Transform-Ion Cyclotron Resonance Mass Spectrometry Data

**Version** 0.7

**Date** 2009-02-13

**Author** Don Barkauskas

**Maintainer** Don Barkauskas <barkda@wald.ucdavis.edu>

**Depends** Matrix,lattice,splines

**Description** This package was developed partially with funding from the NIH Training Program in Biomolecular Technology (2-T32-GM08799).

**License** GPL-2

**Repository** CRAN

**Date/Publication** 2009-02-15 09:25:18

## R topics documented:

FTICRMS-package . . . . .	2
baseline . . . . .	2
display.tests . . . . .	5
extract.pars . . . . .	6
locate.peaks . . . . .	8
make.par.file . . . . .	10
run.all . . . . .	12
run.analysis . . . . .	13
run.baselines . . . . .	15
run.cluster.matrix . . . . .	17
run.lrg.peaks . . . . .	19
run.peaks . . . . .	21
run.strong.peaks . . . . .	23

<b>Index</b>	<b>26</b>
--------------	-----------

---

FTICRMS-package      *Fourier Transform-Ion Cyclotron Resonance Mass Spectrometry (FT-ICR MS) Analysis*

---

### Description

Contains programs for identifying baseline curves and peaks and for statistical analysis of FT-ICR MS data.

### Details

Package: FTICRMS  
Type: Package  
Version: 0.7  
Date: 2009-02-13  
License: GPL-2

This package was developed partially with funding from the NIH Training Program in Biomolecular Technology (2-T32-GM08799).

### Author(s)

Don Barkauskas

Maintainer: Don Barkauskas ((barkda@wald.ucdavis.edu))

---

baseline      *Calculate Baselines for FT-ICR MS Spectra*

---

### Description

Computes an estimated baseline curve for a spectrum by a method of Rocke and Xi generalized by Barkauskas and Rocke.

### Usage

```
baseline(spect, init.bd, sm.par = 1e-11, sm.ord = 2, max.iter = 40, tol = 5e-8,  
sm.div = NA, sm.norm.by = c("baseline", "overestimate", "constant"),  
neg.div = NA, neg.norm.by = c("baseline", "overestimate", "constant"),  
rel.conv.crit = TRUE, zero.rm = TRUE, halve.search = FALSE)
```

**Arguments**

<code>spect</code>	vector containing the intensities of the spectrum
<code>init.bd</code>	initial value for baseline; default is flat baseline at median height
<code>sm.par</code>	smoothing parameter for baseline calculation
<code>sm.ord</code>	order of derivative to kill in baseline analysis
<code>max.iter</code>	convergence criterion in baseline calculation
<code>tol</code>	convergence criterion; see below
<code>sm.div</code>	smoothness divisor in baseline calculation
<code>sm.norm.by</code>	method for smoothness penalty in baseline analysis
<code>neg.div</code>	negativity divisor in baseline calculation
<code>neg.norm.by</code>	method for negativity penalty in baseline analysis
<code>rel.conv.crit</code>	logical; whether convergence criterion should be relative to current baseline estimate
<code>zero.rm</code>	logical; whether to replace zeros with average of surrounding values
<code>halve.search</code>	logical; whether to use a halving-line search if step leads to smaller value of function

**Details**

If the spectrum is given by  $y_i$ , then the algorithm works by maximizing the objective function

$$F(\{b_i\}) = \sum_{i=1}^n b_i - \sum_{i=2}^{n-1} A_{1,i}(b_{i-1} - 2b_i + b_{i+1})^2 - \sum_{i=1}^n A_{2,i}[\max\{b_i - y_i, 0\}]^2$$

using Newton's method with embedded halving line search using starting value  $b[i] = \text{median}(\text{spect})$  for all  $i$ . The middle term controls the smoothness of the baseline and the last term applies a "negativity penalty" when the baseline is above the spectrum.

The smoothing factor `sm.par` corresponds to  $A_1^*$  in Barkauskas (2009) and controls how large the estimated  $n$ th derivative of the baseline is allowed to be (for `sm.ord` =  $n$ ).

From a practical standpoint, values of `sm.ord` larger than two do not seem to adequately smooth the baseline because the Hessian becomes computationally singular for any reasonable value of `sm.par`.

The parameters `sm.div`, `sm.norm.by`, `neg.div`, and `neg.norm.by` determine the methods used to normalize the smoothness and negativity terms. The general forms are  $A_{1,i} = n^4 A_1^* / M_i / p$  and  $A_{2,i} = 1 / M_i / p$ . Here,  $n = \text{length}(\text{spect})$ ;  $p$  is `sm.div` or `neg.div`, as appropriate; and  $M_i$  is determined by `sm.norm.by` or `neg.norm.by`, as appropriate. Values of "baseline" make  $M_i = b'_i$ , where  $b'_i$  is the currently estimated value of the baseline; values of "overestimate" make  $M_i = b'_i - y_i$ ; and values of "constant" make  $M_i = \sigma$ , where  $\sigma$  is an estimate of the noise standard deviation.

The values of `sm.norm.by` and `neg.norm.by` can be abbreviated by their first letters and both have default value "baseline". The default values of NA for `sm.div` and `neg.div` are translated by default to `sm.div` = 0.5223145 and `neg.div` = 0.4210109, which are the appropriate parameters for the mass spectrometry machine that generated the spectra which were used

to develop this package. It is distinctly possible that other machines will require different parameters; see Barkauskas (2009) for a description for how these parameters were obtained.

If `zero.rm = TRUE` and  $y_a, \dots, y_b = 0$ , then these values of the spectrum are set to be  $(y_a + y_b)/2$ . (For typical MALDI FT-ICR spectra, a value of zero indicates an erased harmonic and should not be considered a real data point.)

### Value

A list containing the following items:

<code>baseline</code>	The computed baseline
<code>iter</code>	The number of iterations for convergence
<code>changed</code>	Numeric vector of length <code>iter</code> containing the number of indicator variables that switched value on each iteration

### Note

The original algorithm was developed by Yuanxin Xi and David Rocke. The code was originally adapted from a Matlab program by Yuanxin Xi, then modified to account for the new methodology in Barkauskas (2009).

### Author(s)

Don Barkauskas ([barkda@wald.ucdavis.edu](mailto:barkda@wald.ucdavis.edu))

### References

Barkauskas, D.A. (2009) “Statistical Analysis of Matrix-Assisted Laser Desorption/Ionization Fourier Transform Ion Cyclotron Resonance Mass Spectrometry Data with Applications to Cancer Biomarker Detection”. Ph.D. dissertation, University of California at Davis.

Barkauskas, D.A. *et al.* (2009) “Detecting glycan cancer biomarkers in serum samples using MALDI FT-ICR mass spectrometry data”. *Bioinformatics*, **25**:2, 251–257.

Xi, Y. and Rocke, D.M. (2008) “Baseline Correction for NMR Spectroscopic Metabolomics Data Analysis”. *BMC Bioinformatics*, **9**:324.

### See Also

[run.baselines](#)

---

display.tests	<i>Display Full Test Information for Peaks</i>
---------------	--

---

### Description

Displays full test information (not just  $p$ -values) for peaks generated by `run.analysis`.

### Usage

```
display.tests(sig.rows = "all", summ = anova, tests,
             form = parameter.list$form)
```

### Arguments

<code>sig.rows</code>	numeric or character vector used to select rows of <code>sigs</code> ; default value returns all significant tests
<code>summ</code>	either a function which can be applied to the output of <code>lm</code> or "none"
<code>tests</code>	numeric or character vector used to select rows of <code>clust.mat</code> ; default value returns the rows in <code>clust.mat</code> corresponding to the rows in <code>sigs[sig.rows, ]</code>
<code>form</code>	formula for use in <code>lm</code> ; default is the one that was used to generate the significant peaks

### Details

If `use.t.test = FALSE` in `run.analysis`, then the only thing reported from the test on each peak is the  $p$ -value. This program takes a subset of the significant peaks and returns a list consisting of the linear models generated by `lm` (if `summ = "none"`) or `summ` applied to those models. Typical values for `summ` include `anova` and `summary`.

Although the program is designed to be used on significant peaks, by defining `tests` directly in the function call, you can access any of the peaks in `clust.mat`. If `tests` is defined in the function call, its value overrides anything specified by `sig.rows`.

### Value

A list with components equal to the requested linear models.

### Note

`clust.mat` and `sig.mat` must be defined in the workspace for this program to work—for example, in the results file output by `run.analysis`.

### Author(s)

Don Barkauskas ((`barkda@wald.ucdavis.edu`))

### See Also

`run.analysis`, `anova`, `lm`, `t.test`

---

extract.pars                      *Extract Parameters from File*

---

## Description

Extracts the parameters in the file specified by `par.file` and returns them in list form.

## Usage

```
extract.pars(par.file = "parameters.RData", root.dir = ".")
```

## Arguments

<code>par.file</code>	string containing name of parameters file
<code>root.dir</code>	string containing location of parameters file to be extracted from

## Details

Used by [run.analysis](#) to record all the parameter choices in an analysis for future reference.

## Value

A list with the following components:

<code>add.norm</code>	logical; whether to normalize additively or multiplicatively on the log scale
<code>add.par</code>	additive parameter for "shiftedlog" or "glog" options for <code>trans.method</code>
<code>align.method</code>	alignment algorithm for peaks
<code>base.dir</code>	directory for baseline-corrected files
<code>calc.all.peaks</code>	whether to calculate all possible peaks or only sufficiently large ones
<code>cluster.constant</code>	parameter used in running <code>cluster.method</code>
<code>cluster.method</code>	method for determining when two peaks from different spectra are the same
<code>cor.thresh</code>	threshold correlation for declaring isotopes
<code>covariates</code>	data frame containing covariates used in analysis
<code>FDR</code>	False Discovery Rate in Benjamini-Hochberg test
<code>form</code>	formula used in <code>t.test</code> or <code>lm</code>
<code>gengamma.quantiles</code>	whether to use generalized gamma quantiles when calculating large peaks
<code>halve.search</code>	whether to use a halving-line search if step leads to smaller value of function
<code>isotope.dist</code>	maximum distance for declaring isotopes
<code>lrg.dir</code>	directory for significant peaks file

lrg.file	name of file for storing large peaks
lrg.only	whether to consider only peaks that have at least one peak “large”; i.e., identified by <code>run.lrg.peaks</code>
masses	specific masses to test
max.iter	convergence criterion in baseline calculation
neg.div	negativity divisor in baseline calculation
neg.norm.by	method for negativity penalty in baseline analysis
normalization	type of normalization to use on spectra before statistical analysis
num.pts	number of points needed for peak fitting
oneside.min	minimum number of points on each side of local maximum for peak fitting
overwrite	whether to replace existing files with new ones
par.file	string containing name of parameters file
peak.dir	directory for peak location files
peak.method	method for locating peaks
peak.thresh	threshold for declaring large peak
pre.align	shifts to apply before running <code>run.strong.peaks</code>
pval.fcn	function to calculate $p$ -values if <code>use.t.test = FALSE</code>
R2.thresh	$R^2$ value needed for peak fitting
raw.dir	directory for raw data files
rel.conv.crit	whether convergence criterion should be relative to current baseline estimate
repl.method	how to deal with replicates
res.dir	directory for result file
res.file	name for results file
root.dir	directory for parameters file and raw data directory
sm.div	smoothness divisor in baseline calculation
sm.norm.by	method for smoothness penalty in baseline analysis
sm.ord	order of derivative to kill in baseline analysis
sm.par	smoothing parameter for baseline calculation
subtract.base	whether to subtract calculated baseline from spectrum
tol	convergence criterion in baseline calculation
trans.method	data transformation method
use.t.test	whether to use a $t$ -test to calculate $p$ -values
zero.rm	whether to replace zeros in spectra with average of surrounding values

**Note**

`do.call(make.par.file, extract.pars())` recreates the original parameter file

**Author(s)**

Don Barkauskas ((barkda@wald.ucdavis.edu))

**References**

Barkauskas, D.A. (2009) "Statistical Analysis of Matrix-Assisted Laser Desorption/Ionization Fourier Transform Ion Cyclotron Resonance Mass Spectrometry Data with Applications to Cancer Biomarker Detection". Ph.D. dissertation, University of California at Davis.

Barkauskas, D.A. *et al.* (2009) "Detecting glycan cancer biomarkers in serum samples using MALDI FT-ICR mass spectrometry data". *Bioinformatics*, **25**:2, 251–257.

Xi, Y. and Rocke, D.M. (2008) "Baseline Correction for NMR Spectroscopic Metabolomics Data Analysis". *BMC Bioinformatics*, **9**:324.

**See Also**

[make.par.file](#), [run.analysis](#)

---

locate.peaks

*Locate Peaks in a FT-ICR MS Spectrum*

---

**Description**

Locates peaks in FT-ICR MS spectra assuming that the peaks are roughly parabolic on the log scale.

**Usage**

```
locate.peaks(peak.base, num.pts = 5, R2.thresh = 0.98,  
             onside.min = 1, peak.method = "parabola",  
             thresh = -Inf)
```

**Arguments**

peak.base	numeric matrix with two columns containing the masses and the pre-transformed spectrum intensities
num.pts	minimum number of points needed to have a peak
R2.thresh	minimum $R^2$ needed to have a peak
onside.min	minimum number of points needed on each side of the local maximum
peak.method	how to locate peaks; currently the only options are "parabola" and "locmax"
thresh	only local maxes that are larger than this will be checked to see if they are peaks

## Details

If `peak.method = "parabola"`, the algorithm works by locating local maxima in the spectrum, then seeing if any `num.pts` consecutive points with at least `oneside.min` point(s) on each side of the local maximum have a coefficient of determination ( $R^2$ ) of at least `R2.thresh` when fitted with a quadratic. If, in addition, the coefficient of the squared term is negative, then this is declared a peak and the vertex of the corresponding parabola is located. The coordinates of the vertex give the components `Center_hat` and `Max_hat` in the return value. The `Width_hat` component is the negative reciprocal of the coefficient of the squared term.

If `peak.method = "locmax"`, then the algorithm merely returns the set of local maxima larger than `thresh`, and the `Width_hat` component of the return value is NA.

## Value

A data frame with columns

<code>Center_hat</code>	estimated mass of peak
<code>Max_hat</code>	estimated intensity of peak
<code>Width_hat</code>	estimated width of peak

## Note

An extremely large value for `Width_hat` most likely indicates a bad fit.

Using `peak.method = "locmax"` will vastly speed up the runtime, but may affect the quality of the analysis.

As noted in both papers by Barkauskas, a typical FT-ICR MS spectrum has far more peaks than can be accounted for by actual compounds. Thus, defining a good value of `thresh` will vastly speed up the computation without materially affecting the analysis.

## Author(s)

Don Barkauskas ([barkda@wald.ucdavis.edu](mailto:barkda@wald.ucdavis.edu))

## References

Barkauskas, D.A. (2009) "Statistical Analysis of Matrix-Assisted Laser Desorption/Ionization Fourier Transform Ion Cyclotron Resonance Mass Spectrometry Data with Applications to Cancer Biomarker Detection". Ph.D. dissertation, University of California at Davis.

Barkauskas, D.A. *et al.* (2009) "Detecting glycan cancer biomarkers in serum samples using MALDI FT-ICR mass spectrometry data". *Bioinformatics*, **25**:2, 251–257.

## See Also

[run.peaks](#)

---

make.par.file      *Create Parameter File for FT-ICR MS Analysis*

---

## Description

Creates a file of parameters that can be read by the functions in the FTICRMS package

## Usage

```
make.par.file(covariates, form, par.file = "parameters.RData", root.dir = ".", ...)
```

## Arguments

covariates	data frame with rownames given by raw data files
form	object of class “formula” to be used for testing using covariates
par.file	string containing name of file
root.dir	string containing location for file
...	parameters whose default values are to be overwritten (see below)

## Details

Creates a file with name given by `par.file` in directory given by `root.dir` which contains values for all of the parameters used in the programs in the FTICRMS package. The possible parameters that can be included in `...`, their default values, their descriptions, and the program(s) in which they are used are as follows:

<code>add.norm = TRUE</code>	logical; whether to normalize additively
<code>add.par = 0</code>	additive parameter for "shiftedlog"
<code>align.method = "spline"</code>	alignment algorithm for peaks
<code>base.dir = paste(root.dir, "/Baselines", sep = "")</code>	directory for baseline files
<code>calc.all.peaks = FALSE</code>	logical; whether to calculate all possible peaks
<code>cluster.constant = 10</code>	NA
<code>cluster.method = "ppm"</code>	NA
<code>cor.thresh = 0.8</code>	threshold correlation for declaring isotopes
<code>FDR = 0.1</code>	False Discovery Rate in Benjamini-Hochberg
<code>gengamma.quantiles = TRUE</code>	logical; whether to use generalized gamma
<code>halve.search = FALSE</code>	logical; whether to use a halving-line search
<code>isotope.dist = 7</code>	maximum distance for declaring isotopes
<code>lrg.dir = paste(root.dir, "/Large_Peaks", sep = "")</code>	directory for large peaks file
<code>lrg.file = "lrg_peaks.RData"</code>	name of file for storing large peaks
<code>lrg.only = TRUE</code>	logical; whether to consider only peaks that
<code>masses = NULL</code>	specific masses to test
<code>max.iter = 40</code>	convergence criterion in baseline calculation
<code>neg.div = NA</code>	negativity divisor in baseline calculation
<code>neg.norm.by = c("baseline", "overestimate", "constant")</code>	method for negativity penalty in baseline

```

normalization = "common"
num.pts = 5
oneside.min = 1
overwrite = FALSE
par.file = "parameters.RData"
peak.dir = paste(root.dir, "/All_Peaks", sep = "")
peak.method = "parabola"
peak.thresh = 3.798194
pre.align = FALSE
pval.fcn = "default"
R2.thresh = 0.98
raw.dir = paste(root.dir, "/Raw_Data", sep = "")
rel.conv.crit = TRUE
repl.method = max
res.dir = paste(root.dir, "/Results", sep = "")
res.file = "analyzed.RData"
root.dir = "."
sm.div = NA
sm.norm.by = c("baseline", "overestimate", "constant")
sm.ord = 2
sm.par = 1e-11
subtract.base = FALSE
tol = 5e-8
trans.method = "shiftedlog"
use.t.test = FALSE
zero.rm = TRUE

```

type of normalization to use on spectra before peak detection  
number of consecutive points needed for peak detection  
minimum number of points on each side of peak  
whether to replace existing files with new files  
string containing name of parameters file  
directory for peak location files  
method for locating peaks  
threshold for declaring large peak  
shifts to apply before running `run.struc`  
function to calculate  $p$ -values if `use.t.test`  
 $R^2$  value needed for peak fitting  
directory for raw data files  
whether convergence criterion should be used  
how to deal with replicates  
directory for result file  
name for results file  
directory for parameters file and raw data files  
smoothness divisor in baseline calculation  
method for smoothness penalty in baseline calculation  
order of derivative to kill in baseline analysis  
smoothing parameter for baseline calculation  
logical; whether to subtract calculated baseline from data  
convergence criterion in baseline calculation  
data transformation method  
whether to use a  $t$ -test to calculate  $p$ -values  
whether to replace zeros in spectra with a small value

## Value

No value returned; the file `par.file` is simply created in `root.dir`.

## Note

`do.call(make.par.file, extract.pars())` recreates the original parameter file.

See the individual function help pages for each function for more detailed descriptions of the above parameters.

## Author(s)

Don Barkauskas (([barkda@wald.ucdavis.edu](mailto:barkda@wald.ucdavis.edu)))

## References

- Barkauskas, D.A. (2009) "Statistical Analysis of Matrix-Assisted Laser Desorption/Ionization Fourier Transform Ion Cyclotron Resonance Mass Spectrometry Data with Applications to Cancer Biomarker Detection". Ph.D. dissertation, University of California at Davis.
- Barkauskas, D.A. *et al.* (2009) "Detecting glycan cancer biomarkers in serum samples using MALDI FT-ICR mass spectrometry data". *Bioinformatics*, **25**:2, 251–257.

Xi, Y. and Rocke, D.M. (2008) "Baseline Correction for NMR Spectroscopic Metabolomics Data Analysis". *BMC Bioinformatics*, **9**:324.

### See Also

[extract.pars](#)

---

run.all

*Complete Analysis of FT-ICR MS Data*

---

### Description

A wrapper that calls all six functions needed for a full analysis.

### Usage

```
run.all(par.file = "parameters.RData", root.dir = ".")
```

### Arguments

`par.file`        string containing the name of the parameters file  
`root.dir`        string containing location of raw data directory and parameters file

### Details

Requires `par.file` to be in place before starting—for example by creating it with [make.par.file](#).

Calls (in order) [run.baselines](#), [run.peaks](#), [run.lrg.peaks](#), [run.strong.peaks](#), [run.cluster.matrix](#), and [run.analysis](#).

### Note

The analysis described in Barkauskas *et al.* (2008) can be reproduced using the following parameter values instead of the defaults:

```
add.par = 10  
calc.all.peaks = TRUE  
gengamma.quantiles = FALSE  
neg.norm.by = "constant"  
peak.thresh = 4  
sm.norm.by = "constant"  
subtract.base = TRUE
```

### Author(s)

Don Barkauskas (([barkda@wald.ucdavis.edu](mailto:barkda@wald.ucdavis.edu)))

## References

Barkauskas, D.A. (2009) “Statistical Analysis of Matrix-Assisted Laser Desorption/Ionization Fourier Transform Ion Cyclotron Resonance Mass Spectrometry Data with Applications to Cancer Biomarker Detection”. Ph.D. dissertation, University of California at Davis.

Barkauskas, D.A. *et al.* (2009) “Detecting glycan cancer biomarkers in serum samples using MALDI FT-ICR mass spectrometry data”. *Bioinformatics*, **25**:2, 251–257.

Benjamini, Y. and Hochberg, Y. (1995) “Controlling the false discovery rate: a practical and powerful approach to multiple testing.” *J. Roy. Statist. Soc. Ser. B*, **57**:1, 289–300.

Xi, Y. and Rocke, D.M. (2008) “Baseline Correction for NMR Spectroscopic Metabolomics Data Analysis”. *BMC Bioinformatics*, **9**:324.

## See Also

[make.par.file](#), [run.baselines](#), [run.peaks](#), [run.lrg.peaks](#), [run.strong.peaks](#), [run.cluster.matrix](#), [run.analysis](#)

---

run.analysis	<i>Test for Significant Peaks in FT-ICR MS by Controlling FDR</i>
--------------	---

---

## Description

Takes the file generated by [run.cluster.matrix](#) and tests the peaks using Benjamini-Hochberg to control the False Discovery Rate.

## Usage

```
run.analysis(form, covariates, FDR = 0.1, normalization = "common",
             add.norm = TRUE, repl.method = max, use.t.test = FALSE,
             pval.fcn = "default", lrg.only = TRUE, masses = NULL,
             isotope.dist = 7, root.dir = ".", lrg.dir,
             lrg.file = lrg_peaks.RData, res.dir,
             res.file = "analyzed.RData", overwrite = FALSE,
             use.par.file = FALSE, par.file = "parameters.RData",
             ...)
```

## Arguments

form	formula used in <a href="#">t.test</a> or <a href="#">lm</a>
covariates	data frame containing covariates used in analysis
FDR	False Discovery Rate in Benjamini-Hochberg test
normalization	type of normalization to use on spectra before statistical analysis; currently, only "common", "postbase", "postrepl", and "none" are supported
add.norm	logical; whether to normalize additively or multiplicatively on the log scale

<code>repl.method</code>	function or string representing a function; how to deal with replicates
<code>use.t.test</code>	whether to use a <i>t</i> -test to calculate <i>p</i> -values
<code>pval.fcn</code>	function to calculate <i>p</i> -values if <code>use.t.test = FALSE</code> ; default is overall <i>p</i> -value of <i>F</i> -test using <code>lm</code>
<code>lrg.only</code>	logical; whether to consider only peaks that have at least one “large” peak; i.e., identified by <code>run.lrg.peaks</code>
<code>masses</code>	specific masses to test
<code>isotope.dist</code>	maximum distance for declaring isotopes
<code>root.dir</code>	directory for parameters file and raw data
<code>lrg.dir</code>	directory for large peaks file; default is <code>paste(root.dir, "/Large_Peaks", sep = "")</code>
<code>lrg.file</code>	name of file to store large peaks in
<code>res.dir</code>	directory for results file; default is <code>paste(root.dir, "/Results", sep = "")</code>
<code>res.file</code>	name for results file
<code>overwrite</code>	whether to replace existing files with new ones
<code>use.par.file</code>	logical; if TRUE, then parameters are read from <code>par.file</code> in directory <code>root.dir</code>
<code>par.file</code>	string containing name of parameters file
<code>...</code>	additional parameters to be passed to <code>t.test</code> or <code>pval.fcn</code>

### Details

Reads in information from file created by `run.strong.peaks` and creates a file named `res.file` in `res.dir` which contains variables

<code>amps</code>	matrix of transformed amplitudes of alignment peaks
<code>centers</code>	matrix of calculated masses of alignment peaks
<code>clust.mat</code>	matrix of transformed amplitudes of peaks used in statistical testing
<code>min.FDR</code>	FDR level required to get at least one significant test given the starting set of peaks
<code>sigs</code>	matrix containing all tests which are significant under at least one scenario
<code>which.sig</code>	matrix containing all peaks tested
<code>parameter.list</code>	if <code>use.par.file = TRUE</code> , a list generated by <code>extract.pars</code> ; otherwise not defined

### Value

No value returned; the file is simply created.

### Note

If `use.par.file = TRUE`, then the parameters read in from the file overwrite any arguments entered in the function call.

To analyze replicates as independent samples, use `repl.method = "none"`. This will also speed up the run time if there are no replicates in the data set.

The normalization schemes are as follows: "common" divides all peak heights in each spectrum by the average peak height of the alignment peaks from that spectrum in amps; "postbase" divides all peak heights in each spectrum by the average of all peak heights in that spectrum; and "postrepl" first combines replicates by applying `repl.method` to the peaks and then does "postbase".

If `masses` is not `NULL`, then the listed masses plus anything that could be in the first six isotope peaks of each mass are tested.

If something other than the  $p$ -value for the overall  $F$ -statistic is needed, then the user-defined function for `pval.fcn` should have the form `function(form, dat, ...)`, where `form` and `dat` are as in `lm`; and should have a return value of the desired  $p$ -value.

### Author(s)

Don Barkauskas ((`barkda@wald.ucdavis.edu`))

### References

Barkauskas, D.A. (2009) "Statistical Analysis of Matrix-Assisted Laser Desorption/Ionization Fourier Transform Ion Cyclotron Resonance Mass Spectrometry Data with Applications to Cancer Biomarker Detection". Ph.D. dissertation, University of California at Davis.

Barkauskas, D.A. *et al.* (2009) "Detecting glycan cancer biomarkers in serum samples using MALDI FT-ICR mass spectrometry data". *Bioinformatics*, **25**:2, 251–257.

Benjamini, Y. and Hochberg, Y. (1995) "Controlling the false discovery rate: a practical and powerful approach to multiple testing." *J. Roy. Statist. Soc. Ser. B*, **57**:1, 289–300.

### See Also

[run.strong.peaks](#)

---

run.baselines	<i>Calculate Baselines for FT-ICR spectra</i>
---------------	---

---

### Description

Takes the spectra from files in `raw.dir`, calculates the baselines from them, and writes the results in the directory `base.dir`.

### Usage

```
run.baselines(root.dir = ".", raw.dir, base.dir, overwrite = FALSE,
              use.par.file = FALSE, par.file = "parameters.RData",
              sm.par = 1e-11, sm.ord = 2, max.iter = 40, tol = 5e-8,
              sm.div = NA, sm.norm.by = c("baseline", "overestimate", "constant"),
              neg.div = NA, neg.norm.by = c("baseline", "overestimate", "constant"),
              rel.conv.crit = TRUE, zero.rm = TRUE, halve.search = FALSE)
```

**Arguments**

<code>root.dir</code>	directory for parameters file and raw data
<code>raw.dir</code>	directory for raw data files; default is <code>paste(root.dir, "/Raw_Data", sep = "")</code>
<code>base.dir</code>	directory for baseline files; default is <code>paste(root.dir, "/Baselines", sep = "")</code>
<code>overwrite</code>	whether to replace existing files with new ones
<code>use.par.file</code>	logical; if TRUE, then parameters are read from <code>par.file</code> in directory <code>root.dir</code>
<code>par.file</code>	string containing name of parameters file
<code>sm.par</code>	smoothing parameter for baseline calculation
<code>sm.ord</code>	order of derivative to kill in baseline analysis
<code>max.iter</code>	convergence criterion in baseline calculation
<code>tol</code>	convergence criterion; see below
<code>sm.div</code>	smoothness divisor in baseline calculation
<code>sm.norm.by</code>	method for smoothness penalty in baseline analysis
<code>neg.div</code>	negativity divisor in baseline calculation
<code>neg.norm.by</code>	method for negativity penalty in baseline analysis
<code>rel.conv.crit</code>	logical; whether convergence criterion should be relative to current baseline estimate
<code>zero.rm</code>	logical; whether to replace zeros with average of surrounding values
<code>halve.search</code>	logical; whether to use a halving-line search if step leads to smaller value of function

**Details**

Goes through the entire directory `raw.dir` file-by-file and computes each baseline using [baseline](#), then writes the spectrum and the baseline to a file in directory `base.dir`. The name of the new file is the same as the name of the old file with “.txt” replaced by “.RData”, and the new file is ready to be used by [run.peaks](#).

See [baseline](#) for descriptions of all the parameters after `par.file`.

**Value**

No value returned; the files are simply created.

**Note**

If `use.par.file = TRUE`, then the parameters read in from the file overwrite any arguments entered in the function call.

**Author(s)**

Don Barkauskas (([barkda@wald.ucdavis.edu](mailto:barkda@wald.ucdavis.edu)))

## References

Barkauskas, D.A. (2009) “Statistical Analysis of Matrix-Assisted Laser Desorption/Ionization Fourier Transform Ion Cyclotron Resonance Mass Spectrometry Data with Applications to Cancer Biomarker Detection”. Ph.D. dissertation, University of California at Davis.

Barkauskas, D.A. *et al.* (2009) “Detecting glycan cancer biomarkers in serum samples using MALDI FT-ICR mass spectrometry data”. *Bioinformatics*, **25**:2, 251–257.

Xi, Y. and Rocke, D.M. (2008) “Baseline Correction for NMR Spectroscopic Metabolomics Data Analysis”. *BMC Bioinformatics*, **9**:324.

## See Also

[baseline](#), [run.peaks](#)

---

run.cluster.matrix *Identify Equivalent Peaks from Different Subjects*

---

## Description

Takes the file generated by [run.lrg.peaks](#), identifies equivalent peaks in each spectrum, and fills in missing values.

## Usage

```
run.cluster.matrix(pre.align = FALSE, align.method = "spline",
                  trans.method = "shiftedlog", add.par = 0,
                  subtract.base = FALSE, lrg.only = TRUE,
                  calc.all.peaks = FALSE, masses = NULL,
                  isotope.dist = 7, cluster.method = "ppm",
                  cluster.constant = 10, num.pts = 5,
                  R2.thresh = 0.98, onside.min = 1,
                  peak.method = "parabola", root.dir = ".",
                  base.dir, peak.dir, lrg.dir,
                  lrg.file = lrg_peaks.RData,
                  overwrite = FALSE, use.par.file = FALSE,
                  par.file = "parameters.RData")
```

## Arguments

pre.align	either FALSE, or a numeric vector of shifts to apply to spectra, or a two-component list (of the form described in the <code>Note</code> section below) to be used before identifying peaks from different spectra
align.method	alignment algorithm for peaks
trans.method	type of transformation to use on spectra before statistical analysis; currently, only "shiftedlog", "glog", and "none" are supported
add.par	additive parameter for "shiftedlog" or "glog" options for trans.method

subtract.base	logical; whether to subtract calculated baseline from spectrum
lrg.only	logical; whether to consider only peaks that have at least one “large” peak; i.e., identified by run.lrg.peaks
calc.all.peaks	logical; whether to calculate all possible peaks or only sufficiently large ones
masses	specific masses to test
isotope.dist	maximum distance for declaring isotopes
cluster.method	NA
cluster.constant	NA
num.pts	number of consecutive points needed for peak fitting
R2.thresh	$R^2$ value needed for peak fitting
oneside.min	minimum number of points on each side of local maximum for peak fitting
peak.method	method for locating peaks
root.dir	directory for parameters file and raw data
base.dir	directory for baseline files; default is <code>paste(root.dir, "/Baselines", sep = "")</code>
peak.dir	directory for peak location files; default is <code>paste(root.dir, "/All_Peaks", sep = "")</code>
lrg.dir	directory for large peaks file; default is <code>paste(root.dir, "/Large_Peaks", sep = "")</code>
lrg.file	name of file to store large peaks in
overwrite	whether to replace existing files with new ones
use.par.file	logical; if TRUE, then parameters are read from par.file in directory root.dir
par.file	string containing name of parameters file

### Details

Reads in information from file created by `run.strong.peaks`, calculates the cluster matrix, fills in missing values, and overwrites the file named `lrg.file` in `lrg.dir`. The resulting file contains variables

amps	data frame of amplitudes created by <code>run.strong.peaks</code>
centers	data frame of centers created by <code>run.strong.peaks</code>
clust.mat	data frame with columns given by samples and rows given by the distinct peaks in the samples
num.sig	vector of the number of peaks in each row of <code>clust.mat</code> which were not missing
lrg.peaks	the data frame of significant peaks created by <code>run.lrg.peaks</code>

and is ready to be used by `run.strong.peaks`.

**Value**

No value returned; the file is simply created.

**Note**

If `use.par.file = TRUE`, then the parameters read in from the file overwrite any arguments entered in the function call.

`pre.align` is used if the spectra have not already been aligned by the mass spectroscopists. If it is not `FALSE`, it can either be a vector of additive shifts to be applied to the spectra, or a list with components `targets` and `actual`. In the last case, `targets` is a vector of target masses, and `actual` is a matrix with `length(targets)` columns and a row for each spectrum, `actual[i, j]` being the mass in spectrum `i` that should be matched exactly to `target[j]`, with `NA` being a valid entry in `actual`. The matching is done (depending on the number of non-missing values in row `i`) either with a simple shift (one non-missing value), an affine transformation (two non-missing values), a piecewise affine transformation (three non-missing values), or an interpolation spline (four or more non-missing values).

Suppose `cluster.constant = K` and we have two peaks in different spectra with masses  $m_1$  and  $m_2$ . If `cluster.method = "constant"`, then the peaks are considered to be the same peak if we have  $m_2 - m_1 < K$ . If `cluster.method = "ppm"`, then the peaks are considered to be the same peak if we have  $m_2 - m_1 < Km_2/10^6$ . If `cluster.method = "usewidth"`, then the algorithm uses the observation that `log(Width_hat)` and `log(Center_hat)` appear to be linearly related. Tolerances are then computed using this relationship.

**Author(s)**

Don Barkauskas ((barkda@wald.ucdavis.edu))

**References**

Barkauskas, D.A. (2009) "Statistical Analysis of Matrix-Assisted Laser Desorption/Ionization Fourier Transform Ion Cyclotron Resonance Mass Spectrometry Data with Applications to Cancer Biomarker Detection". Ph.D. dissertation, University of California at Davis.

Barkauskas, D.A. *et al.* (2009) "Detecting glycan cancer biomarkers in serum samples using MALDI FT-ICR mass spectrometry data". *Bioinformatics*, **25**:2, 251–257.

**See Also**

[run.lrg.peaks](#), [run.strong.peaks](#), [interpSpline](#)

---

run.lrg.peaks

*Extract "Large" Peaks from Files*

---

**Description**

Takes the files output by [run.peaks](#), extracts "large" peaks, combines them into a single data frame, and writes the data frame to a file.

**Usage**

```
run.lrg.peaks(trans.method = "shiftedlog", add.par = 0, subtract.base = FALSE,
              root.dir = ".", peak.dir, base.dir, lrg.dir,
              lrg.file = lrg_peaks.RData, overwrite = FALSE,
              use.par.file = FALSE, par.file = "parameters.RData",
              calc.all.peaks = FALSE, gengamma.quantiles = TRUE,
              peak.thresh = 3.798194)
```

**Arguments**

<code>trans.method</code>	type of transformation to use on spectra before statistical analysis; currently, only "shiftedlog", "glog", and "none" are supported
<code>add.par</code>	additive parameter for "shiftedlog" or "glog" options for <code>trans.method</code>
<code>subtract.base</code>	logical; whether to subtract calculated baseline from spectrum
<code>root.dir</code>	directory for parameters file and raw data
<code>peak.dir</code>	directory for peak location files; default is <code>paste(root.dir, "/All_Peaks", sep = "")</code>
<code>base.dir</code>	directory for baseline files; default is <code>paste(root.dir, "/Baselines", sep = "")</code>
<code>lrg.dir</code>	directory for large peaks file; default is <code>paste(root.dir, "/Large_Peaks", sep = "")</code>
<code>lrg.file</code>	name of file to store large peaks in
<code>overwrite</code>	whether to replace existing files with new ones
<code>use.par.file</code>	logical; if TRUE, then parameters are read from <code>par.file</code> in directory <code>root.dir</code>
<code>par.file</code>	string containing name of parameters file
<code>calc.all.peaks</code>	logical; whether to calculate all possible peaks or only sufficiently large ones
<code>gengamma.quantiles</code>	logical; whether to use generalized gamma quantiles when calculating large peaks
<code>peak.thresh</code>	threshold for declaring large peak; see below

**Details**

Reads in information from each file created by `run.peaks`, extracts peaks which have zero weight in the spectrum they come from when using Tukey's biweight with parameter `k.biweight` to estimate center and scale, and creates the file `lrg.file` in `lrg.dir`. The resulting file contains the data frame `lrg.peaks`, which has columns

<code>Center_hat</code>	estimated mass of peak
<code>Max_hat</code>	estimated intensity of peak
<code>Width_hat</code>	estimated width of peak
<code>File</code>	name of file the peak was extracted from, with "_peaks.RData" deleted

and is ready to be used by `run.strong.peaks`.

### Value

No value returned; the file is simply created.

### Note

If `gengamma.quantiles = TRUE`, then a peak is “large” if it is at least `peak.thresh` times as large as the estimated baseline at that point.

If `gengamma.quantiles = FALSE`, then a peak is “large” if it has zero weight in the data generated by `run.peaks` for the spectrum it comes from when using Tukey’s biweight with parameter  $3/2 * peak.thresh$  to estimate center and scale.

If `use.par.file = TRUE`, then the parameters read in from the file overwrite any arguments entered in the function call.

### Author(s)

Don Barkauskas ((barkda@wald.ucdavis.edu))

### References

Barkauskas, D.A. (2009) “Statistical Analysis of Matrix-Assisted Laser Desorption/Ionization Fourier Transform Ion Cyclotron Resonance Mass Spectrometry Data with Applications to Cancer Biomarker Detection”. Ph.D. dissertation, University of California at Davis.

Barkauskas, D.A. *et al.* (2009) “Detecting glycan cancer biomarkers in serum samples using MALDI FT-ICR mass spectrometry data”. *Bioinformatics*, **25**:2, 251–257.

### See Also

`run.peaks`, `run.cluster.matrix`

---

run.peaks

*Locate Potential Peaks in FT-ICR MS Spectra*

---

### Description

Takes baseline-corrected data and locates potential peaks in the spectra.

### Usage

```
run.peaks(trans.method = "shiftedlog", add.par = 0, subtract.base = FALSE,  
          root.dir = ".", base.dir, peak.dir, overwrite = FALSE,  
          use.par.file = FALSE, par.file = "parameters.RData",  
          num.pts = 5, R2.thresh = 0.98, onside.min = 1,  
          peak.method = "parabola", calc.all.peaks = FALSE,  
          gengamma.quantiles = TRUE, peak.thresh = 3.798194)
```

## Arguments

<code>trans.method</code>	type of transformation to use on spectra before statistical analysis; currently, only "shiftedlog", "glog", and "none" are supported
<code>add.par</code>	additive parameter for "shiftedlog" or "glog" options for <code>trans.method</code>
<code>subtract.base</code>	logical; whether to subtract calculated baseline from spectrum
<code>root.dir</code>	directory for parameters file and raw data
<code>base.dir</code>	directory for baseline files; default is <code>paste(root.dir, "/Baselines", sep = "")</code>
<code>peak.dir</code>	directory for peak location files; default is <code>paste(root.dir, "/All_Peaks", sep = "")</code>
<code>overwrite</code>	whether to replace existing files with new ones
<code>use.par.file</code>	logical; if TRUE, then parameters are read from <code>par.file</code> in directory <code>root.dir</code>
<code>par.file</code>	string containing name of parameters file
<code>num.pts</code>	number of consecutive points needed for peak fitting
<code>R2.thresh</code>	$R^2$ value needed for peak fitting
<code>oneside.min</code>	minimum number of points on each side of local maximum for peak fitting
<code>peak.method</code>	method for locating peaks
<code>calc.all.peaks</code>	logical; whether to calculate all possible peaks or only sufficiently large ones
<code>gengamma.quantiles</code>	logical; whether to use generalized gamma quantiles when calculating large peaks
<code>peak.thresh</code>	threshold for declaring large peak; see below

## Details

Reads in information from each file created by `run.baselines`, calls `locate.peaks` to find potential peaks, and writes the output to a file in directory `peak.dir`. The name of each new file is the same as the name of the old file with ".RData" replaced by "\_peaks.RData". The resulting file contains the data frame `all.peaks`, which has columns

<code>Center_hat</code>	estimated mass of peak
<code>Max_hat</code>	estimated intensity of peak
<code>Width_hat</code>	estimated width of peak

and is ready to be used by `run.lrg.peaks`.

The parameters `gengamma.quantiles` and `peak.thresh` are relevant only if `calc.all.peaks = FALSE`. In that case, if `gengamma.quantiles = TRUE`, then `peak.thresh` is interpreted as a multiplier for the baseline. Anything larger than `peak.thresh` times the estimated baseline is declared to be a real peak. If `gengamma.quantiles = TRUE`, then `peak.thresh` is interpreted as two-thirds of the value of  $K$  used in a Tukey's biweight estimation of center and

scale (so roughly equal to the number of standard deviations above the mean for iid normal data). Anything with weight zero in the calculation is then declared to be a real peak.

### Value

No value returned; the files are simply created.

### Note

If `use.par.file = TRUE`, then the parameters read in from the file overwrite any arguments entered in the function call.

Using `calc.all.peaks = FALSE` will speed up computation time immensely, but will affect the final result. It probably won't affect it much, but *caveat emptor*.

### Author(s)

Don Barkauskas ([barkda@wald.ucdavis.edu](mailto:barkda@wald.ucdavis.edu))

### References

Barkauskas, D.A. (2009) "Statistical Analysis of Matrix-Assisted Laser Desorption/Ionization Fourier Transform Ion Cyclotron Resonance Mass Spectrometry Data with Applications to Cancer Biomarker Detection". Ph.D. dissertation, University of California at Davis.

Barkauskas, D.A. *et al.* (2009) "Detecting glycan cancer biomarkers in serum samples using MALDI FT-ICR mass spectrometry data". *Bioinformatics*, **25**:2, 251–257.

### See Also

[run.baselines](#), [run.lrg.peaks](#), [locate.peaks](#)

---

`run.strong.peaks`    *Locate Peaks that are "Large" in All Samples*

---

### Description

Takes the file generated by [run.peaks](#), extracts all peaks that are "large" in all samples, and writes the results to a file.

### Usage

```
run.strong.peaks(cor.thresh = 0.8, isotope.dist = 7, pre.align = FALSE,  
                root.dir = ".", lrg.dir, lrg.file = "lrg_peaks.RData",  
                overwrite = FALSE, use.par.file = FALSE,  
                par.file = "parameters.RData")
```

**Arguments**

<code>cor.thresh</code>	threshold correlation for declaring isotopes
<code>isotope.dist</code>	maximum distance for declaring isotopes
<code>pre.align</code>	either <code>FALSE</code> , or a numeric vector of shifts to apply to spectra, or a two-component list (of the form described in the <code>Note</code> section below) to be used before identifying peaks from different spectra
<code>root.dir</code>	directory for parameters file and raw data
<code>lrg.dir</code>	directory for large peaks file; default is <code>paste(root.dir, "/Large_Peaks", sep = "")</code>
<code>lrg.file</code>	name of file to store large peaks in
<code>overwrite</code>	whether to replace existing files with new ones
<code>use.par.file</code>	logical; if <code>TRUE</code> , then parameters are read from <code>par.file</code> in directory <code>root.dir</code>
<code>par.file</code>	string containing name of parameters file

**Details**

Reads in information from file created by `run.lrg.peaks`, locates peaks which appear in all samples, and overwrites the file `lrg.file` in `lrg.dir`. The resulting file contains variables

<code>amps</code>	data frame of amplitudes of non-isotope peaks that occur in all samples
<code>centers</code>	data frame of centers of non-isotope peaks that occur in all samples
<code>lrg.peaks</code>	the data frame of significant peaks created by <code>run.lrg.peaks</code>

and is ready to be used by `run.cluster.matrix`.

**Value**

No value returned; the file is simply created.

**Note**

If `use.par.file = TRUE`, then the parameters read in from the file overwrite any arguments entered in the function call.

`pre.align = FALSE` is used if the spectra have already been aligned by the mass spectroscopists. Otherwise, it can either be a vector of additive shifts to be applied to the spectra, or a list with components `targets` and `actual`. In the last case, `targets` is a vector of target masses, and `actual` is a matrix with `length(targets)` columns and a row for each spectrum, `actual[i, j]` being the mass in spectrum `i` that should be matched exactly to `target[j]`, with `NA` being a valid entry in `actual`. The matching is done (depending on the number of non-missing values in row `i`) either with a simple shift (one non-missing value), an affine transformation (two non-missing values), a piecewise affine transformation (three non-missing values), or an interpolation spline (four or more non-missing values).

**Author(s)**

Don Barkauskas (([barkda@wald.ucdavis.edu](mailto:barkda@wald.ucdavis.edu)))

**References**

Barkauskas, D.A. (2009) “Statistical Analysis of Matrix-Assisted Laser Desorption/Ionization Fourier Transform Ion Cyclotron Resonance Mass Spectrometry Data with Applications to Cancer Biomarker Detection”. Ph.D. dissertation, University of California at Davis.

Barkauskas, D.A. *et al.* (2009) “Detecting glycan cancer biomarkers in serum samples using MALDI FT-ICR mass spectrometry data”. *Bioinformatics*, **25**:2, 251–257.

**See Also**

[run.lrg.peaks](#), [run.cluster.matrix](#), [interpSpline](#)

# Index

## \*Topic **package**

FTICRMS-package, [1](#)

`anova`, [5](#)

`baseline`, [2](#), [16](#)

`display.tests`, [4](#)

`extract.pars`, [5](#), [11](#), [14](#)

`formula`, [9](#)

FTICRMS (*FTICRMS-package*), [1](#)

FTICRMS-package, [1](#)

`interpSpline`, [19](#), [24](#)

`lm`, [4-6](#), [13](#), [14](#)

`locate.peaks`, [8](#), [22](#), [23](#)

`make.par.file`, [7](#), [9](#), [12](#)

`run.all`, [11](#)

`run.analysis`, [4](#), [5](#), [7](#), [10](#), [11](#), [12](#), [12](#)

`run.baselines`, [4](#), [10-12](#), [15](#), [22](#), [23](#)

`run.cluster.matrix`, [10-12](#), [16](#), [21](#), [24](#)

`run.lrg.peaks`, [6](#), [10-12](#), [16](#), [18](#), [19](#), [19](#),  
[22-24](#)

`run.peaks`, [9-12](#), [16](#), [19](#), [20](#), [21](#), [21](#), [23](#)

`run.strong.peaks`, [6](#), [10](#), [12-14](#), [18-20](#),  
[23](#)

`summary`, [5](#)

`t.test`, [5](#), [6](#), [13](#)