Package 'corr2D'

October 29, 2018

Type Package

Title Implementation of 2D Correlation Analysis in R

Version 0.3.0

Date 2018-10-29

Description Implementation of two-dimensional (2D) correlation analysis based on the Fourier-transformation approach described by Isao Noda (I. Noda (1993) <DOI:10.1366/0003702934067694>). Additionally there are two plot functions for the resulting correlation matrix: The first one creates colored 2D plots, while the second one generates 3D plots.

Imports doParallel (>= 1.0.8), foreach (>= 1.4.3), parallel (>= 3.0.2), fields (>= 8.2-1), mmand (>= 1.3.0), rgl (>= 0.93.996-1), stats (>= 3.0.2), grDevices (>= 3.0.2), graphics (>= 3.0.2), utils (>= 3.0.2), colorspace (>= 1.3-0), profr, xtable

License GPL-3

RoxygenNote 6.1.0

Encoding UTF-8

Suggests knitr, rmarkdown, R.rsp

VignetteBuilder R.rsp

NeedsCompilation no

Author Robert Geitner [cre, aut],
Robby Fritzsch [aut],
Thomas Bocklitz [aut],
Juergen Popp [ctb, cph]

Maintainer Robert Geitner <r.geitner@uu.nl>

Repository CRAN

Date/Publication 2018-10-29 16:50:17 UTC

R topics documented:

`codis2d` *Two-dimensional codistribution spectroscopy.*

Description

`codis2d` calculates the synchronous and asynchronous codistribution spectra.

Usage

```
codis2d(Mat, Ref = NULL, Wave = NULL, Time = NULL,
  Int = stats::splinefun, N = 2^ceiling(log2(NROW(Mat))),
  Norm = 1/(NROW(Mat) - 1), scaling = 0,
  corenumber = parallel::detectCores(), preview = FALSE)
```

Arguments

`Mat`	Numeric matrix containing the data which will be correlated; '*spectral variable*' by columns and '*perturbation variables*' by rows.
`Ref`	Numeric vector containing a single spectrum, which will be subtracted from `Mat` to generate dynamic spectra for 2D correlation analysis. Default is `NULL` in which case the `colMeans()` of `Mat` is used as reference. The length of `Ref` needs to be equal to the number of columns in `Mat`. 2D codistribution spectroscopy is only strictly defined using the perturbation-mean spectrum as refernece spectrum. Thus, any deviation from this definition can lead to unexpected results.
`Wave`	Numeric vector containing the spectral variable. Needs to be specified if column names of `Mat` are undefined.
`Time`	Numeric vector containing the perturbation variables. If specified, `Mat` will be interpolated to `N` equally spaced perturbation varibales using `Int`.
`Int`	Function specifying how the dataset will be interpolated to give `N` equally spaced perturbation variables. `splinefun` (default) or `approxfun` can for example be used.
`N`	Positive, non-zero integer specifying how many equally spaced perturbation variables should be interpolated using `Int`. `N` should be higher than 4. `corr2d` is fastest if `N` is a power of 2.

Norm	A number specifying how the correlation matrix should be normalized.
scaling	Positive real number used as exponent when scaling the dataset with its standard deviation. Defaults to 0 meaning no scaling. 0.5 (*Pareto scaling*) and 1 (*Pearson scaling*) are commonly used to enhance weak correlations relative to strong correlations. 2D codistribution spectroscopy is only strictly defined without the usage of any scaling techniques. Thus, any deviation from this definition can lead to unexpected results.
corenumber	Positive, non-zero integer specifying how many CPU cores should be used for parallel fft computation.
preview	Logical: Should a 3D preview of the asynchronous codistribution spectrum be drawn at the end? Uses persp3d from **rgl** package.

Details

codis2d calculates the the synchronous 2D correlation spectrum and uses the 2D spectrum to calculate the synchronous and asynchronous codistribution spectra. For parallelization the parCapply function is used. Large input matrices (> 4000 columns) can lead to long calculation times depending on the number of cores used. Also note that the resulting matrix can become very large, adjust the RAM limit with memory.limit accordingly. For a detailed description of the underlying math see references.

Value

codis2D returns a list of class "corr2d" containing the complex codistribution matrix ($FT), the synchronous correlation spectrum ($corr), the used reference spectrum $Ref1 and $Ref2, the spectral variables $Wave1 and $Wave2 as well as the (interpolated) perturbation variables ($Time).

References

I. Noda (2014) <DOI:10.1016/j.molstruc.2014.01.024>

See Also

For plotting of the resulting list containing the 2D codistribution spectra see plot_corr2d and plot_corr2din3d.

Examples

```
testdata <- sim2ddata(C = NULL, Camp = NULL)
codis <- codis2d(testdata, corenumber = 1)

plot_corr2d(codis, Im(codis$FT),
            xlab = expression(paste("Wavenumber" / cm^-1)),
            ylab = expression(paste("Wavenumber" / cm^-1)))
```

corr2d *Two-dimensional correlation analysis.*

Description

corr2d calculates the synchronous and asynchronous correlation spectra between Mat1 and Mat1 (homo correlation) or between Mat1 and Mat2 (hetero correlation).

Usage

```
corr2d(Mat1, Mat2 = NULL, Ref1 = NULL, Ref2 = NULL, Wave1 = NULL,
  Wave2 = NULL, Time = NULL, Int = stats::splinefun,
  N = 2^ceiling(log2(NROW(Mat1))), Norm = 1/(pi * (NROW(Mat1) - 1)),
  scaling = 0, corenumber = parallel::detectCores(), preview = FALSE)
```

Arguments

Mat1, Mat2	Numeric matrix containing the data which will be correlated; '*spectral variable*' by columns and '*perturbation variables*' by rows. For hetero correlations Mat1 and Mat2 must have the same number of rows.
Ref1, Ref2	Numeric vector containing a single spectrum, which will be subtracted from Mat1 (or Mat2, respectively) to generate dynamic spectra for 2D correlation analysis. Default is NULL in which case the colMeans() of Mat1 (or Mat2, respectively) is used as reference. The length of Ref1 (or Ref2) needs to be equal to the number of columns in Mat1 (or Mat2).
Wave1, Wave2	Numeric vector containing the spectral variable. Needs to be specified if column names of Mat1 (or Mat2) are undefined.
Time	Numeric vector containing the perturbation variables. If specified, Mat1 (and Mat2 if given) will be interpolated to N equally spaced perturbation varibales using Int to speed up the fft algorithm.
Int	Function specifying how the dataset will be interpolated to give N equally spaced perturbation variables. splinefun (default) or approxfun can for example be used.
N	Positive, non-zero integer specifying how many equally spaced perturbation variables should be interpolated using Int. N should be higher than 4.corr2d is fastest if N is a power of 2.
Norm	A number specifying how the correlation matrix should be normalized.
scaling	Positive real number used as exponent when scaling the dataset with its standard deviation. Defaults to 0 meaning no scaling. 0.5 (*Pareto scaling*) and 1 (*Pearson scaling*) are commonly used to enhance weak correlations relative to strong correlations.
corenumber	Positive, non-zero integer specifying how many CPU cores should be used for parallel fft computation.
preview	Logical: Should a 3D preview of the synchronous correlation spectrum be drawn at the end? Uses persp3d from **rgl** package.

Details

corr2d uses a parallel fast Fourier transformation (fft) to calculate the complex correlation matrix. For parallelization the foreach function is used. Large input matrices (> 4000 columns) can lead to long calculation times depending on the number of cores used. Also note that the resulting matrix can become very large, adjust the RAM limit with memory.limit accordingly. For a detailed description of the underlying math see references.

Value

corr2D returns a list of class "corr2d" containing the complex correlation matrix ($FT), the used reference spectra ($Ref1, $Ref2), the spectral variables ($Wave1, $Wave2), the (interpolated) perturbation variables ($Time) and logical variable ($Het) indicating if homo (FALSE) or hetero (TRUE) correlation was done.

References

I. Noda (1993) <DOI:10.1366/0003702934067694>
I. Noda (2012) <DOI:10.1016/j.vibspec.2012.01.006>

See Also

For plotting of the resulting list containing the 2D correlation spectra see plot_corr2d and plot_corr2din3d.

Examples

```
data(FuranMale, package = "corr2D")
twod <- corr2d(FuranMale, Ref1 = FuranMale[1, ], corenumber = 1)

plot_corr2d(twod, xlab = expression(paste("relative Wavenumber" / cm^-1)),
                  ylab = expression(paste("relative Wavenumber" / cm^-1)))
```

---

corr2t2d *Two-trace two-dimensional (2T2D) correlation spectroscopy*

Description

corr2t2d compares a pair of spectra in the form of a cross correlation analysis.

Usage

```
corr2t2d(Sam, Ref, Wave = NULL, preview = FALSE)
```

Arguments

Sam	Numeric vector containing the sample spectrum to be correlated. Can contain the spectral variable of the sample and reference spectrum as names.
Ref	Numeric vector containing the sample spectrum to be correlated. Can contain the spectral variable of the sample and reference spectrum as names.
Wave	Numeric vector containing the spectral variable. Needs to be specified if names of Sam and Ref are undefined.
preview	Logical: Should a 3D preview of the asynchronous codistribution spectrum be drawn at the end? Uses persp3d from **rgl** package.

Details

corr2t2d implements the Two-trace two-dimensional (2T2D) approach as described by I. Noda (2018) <DOI:10.1016/j.molstruc.2018.01.091>. The idea is to compare two spectra in a 2D correlation-like approach which was previously not possible as 2D correlation analysis usually needs at least three spectra.

Value

corr2t2d returns a list of class "corr2d" containing the complex correlation matrix ($FT), the correlation and disrelation coefficient as a complex matrix ($coef), the sample $Ref1 and reference spectrum $Ref2 as well as the spectral variable $Wave1 and $Wave2.

References

I. Noda (2018) <DOI:10.1016/j.molstruc.2018.01.091>

See Also

For plotting of the resulting list containing the 2D correlation spectra or correaltion coefficient see plot_corr2d and plot_corr2din3d.

Examples

```
testdata <- sim2ddata()

twodtest <- corr2t2d(testdata[4, ], testdata[5, ])

plot_corr2d(twodtest, Im(twodtest$FT))
```

FuranMale *FT-Raman spectra of furan maleimide based self-healing polymer*

Description

Six preprocessed FT-Raman spectra of a self-healing polymer. The wavenumber region shows the C=C vibrations of furan, maleimide and their respective Diels-Alder adduct. The row names show the measurement temperature in degree Celsius, while the column names show the relative wavenumber.

Format

A matrix containing 6 spectra by rows with 145 wavenumbers by columns.

Source

R. Geitner, J. Koetteritzsch, M. Siegmann, T. Bocklitz, M. Hager, U. S. Schubert, S. Graefe, B. Dietzek, M. Schmitt and J. Popp (2015) <DOI:10.1039/C5CP02151K>

is.corr2d *Check for object class "corr2d"*

Description

The function checks if an object is of class "corr2d".

Usage

```
is.corr2d(x)
```

Arguments

x	An object which should be check if it is of class "corr2d".

Details

The function uses the inherits function.

Value

A logical scalar

Examples

```
data(FuranMale, package = "corr2D")
twod <- corr2d(FuranMale, Ref1 = FuranMale[1, ], corenumber = 1)

# TRUE
is.corr2d(twod)
# FALSE
is.corr2d(2)
```

---

plot_corr2d *Plot two-dimensional correlation spectra.*

Description

plot_corr2d plots two-dimensional correlation spectra either as an image or a contour plot. Red color indicates positive correlations, while blue color shows negative ones.

Usage

```
plot_corr2d(Obj, what = Re(Obj$FT), specx = Obj$Ref1,
  specy = Obj$Ref2, xlim = NULL, ylim = NULL,
  xlab = expression(nu[1]), ylab = expression(nu[2]), Contour = TRUE,
  axes = 3, Legend = TRUE, N = 20, zlim = NULL, Cutout = NULL,
  col = par("col"), lwd = par("lwd"), lwd.axis = NULL,
  lwd.spec = NULL, cex.leg = NULL, at.xaxs = NULL,
  label.xaxs = TRUE, at.yaxs = NULL, label.yaxs = TRUE,
  line.xlab = 3.5, line.ylab = 3.5, ...)
```

Arguments

Obj	List from corr2d containing the 2D correlation data.
what	Real numeric matrix containing the z-values that should be plotted.
specx, specy	Numeric vector containing the data that should be plotted on top (specx) and/or on the left (specy) of the 2D spectrum. Mat, specx and/or specy should have the same dimensions, respectively. If NULL nothing will be plotted.
xlim, ylim	Numeric vector with two values indicating the borders of the 2D plot. Also truncates specx and/or specy to match the new plot range.
xlab, ylab	Character or expression containing the text that will be plotted on the bottom (xlab) and/or to the right (ylab) of the 2D plot. Labels can be suppressed with NA.
Contour	Logical: Should a contour (TRUE) or image (FALSE) be drawn?
axes	Integer ranging from 0 to 3. Should the axis of the 2D plot be drawn? "0" means no axes, "1" only bottom axis, "2" only right axis and "3" both axes are drawn.
Legend	Logical: Should a color legend be plotted in the top right corner?

N	Positive, non-zero integer indicating how many contour or image levels should be plotted.
zlim	Numeric vector with two values defining the z-range of the 2D plot.
Cutout	Numeric vector with two values defining which z-values should not be plotted. Use with care, because this can generate misleading 2D plots.
col	A specification for the plotting color of the reference spectra (top and left), axes, axes ticks and the central plot surrounding box. See par and contour for additional information.
lwd	A numeric value which sets the line width in the contour plot. See par and contour for additional information.
lwd.axis	A numeric value which sets the line width for axes and the central plot surrounding box. See par and axis for additional information.
lwd.spec	A numeric value which sets the line width in the reference spectra on top and to the left. See par and plot.default for additional information.
cex.leg	A numerical value giving the amount by which numbers at the legend should be magnified. See par and image.plot for additional information.
at.xaxs, at.yaxs	The points at which tick-marks are to be drawn at the x- and y-axis, respectively. See axis for additional information.
label.xaxs, label.yaxs	This can either be a logical value specifying whether (numerical) annotations are to be made at the tickmarks of the x- and y-axis, or a character or expression vector of labels to be placed at the tickpoints of the x- and y-axis. See axis for additional information.
line.xlab, line.ylab	Numeric value on which MARgin line the x- and y-label is plotted, respectively, starting at 0 counting outwards. See mtext for additional information.
...	Additional arguments either passed to image or contour. Can include graphics parameters par which are in part also used by other functions. This includes cex.axis (influences axes and thier label magnification), cex.lab (influences label magnification), col.axis (influences axes label color), col.lab (influences label color), font.axis (influences axes label font), font.lab (influences label font) and lty (influences line type for contour plot).

Details

For the synchronous correlation spectrum the real component (Re) of the complex correlation matrix must be plotted. The asynchronous spectrum is the respective imaginary component (Im). Cutout can be used to leave out smaller (noise) contributions, but should be used with care as it can be used to create misleading 2D correlation plots. See references for interpretation rules (so called Noda rules).

References

For interpretation rules see: I. Noda (2006) <DOI:10.1016/j.molstruc.2005.12.060>

See Also

See plot_corr2din3d for 3D plots.

Examples

```
data(FuranMale, package = "corr2D")
twod <- corr2d(FuranMale, Ref1 = FuranMale[1, ], corenumber = 1)

plot_corr2d(twod, xlab = expression(paste("relative Wavenumber" / cm^-1)),
                  ylab = expression(paste("relative Wavenumber" / cm^-1)))

plot_corr2d(twod, at.xaxs = c(1560, 1585, 1610),
            label.xaxs = c(1560, 1585, 1610),
            col = 2, lwd = 3, col.axis = 3, col.lab = 4, Legend = FALSE,
            cex.lab = 3, xlab = "Large x label", ylab = "Large y label",
            line.xlab = 5, line.ylab = 5)
```

---

plot_corr2din3d *3D plot of two-dimensional correlation spectra.*

Description

plot_corr2din3d plots two-dimensional correlation spectra as an 3D surface.

Usage

```
plot_corr2din3d(Mat, specx = NULL, specy = NULL, scalex = NULL,
  scaley = NULL, Col = colorspace::diverge_hcl(64, h = c(240, 0), c =
  100, l = c(20, 100), power = 0.4), reduce = NULL, zlim = NULL,
  projection = FALSE, ...)
```

Arguments

Mat	Real numeric matrix containing the z-values to plot.
specx, specy	Numeric vector containing the data, that will be plotted at the x and y axis. Can be any data and does not need to have the same dimensions as Mat.
scalex, scaley	A real number which describes how specx (or specy) get scaled. Positive numbers lead to a spectrum plotted inside the box, while negative numbers lead to a spectrum plotted outside the box.
Col	Vector containing colors used to plot the 3D plot and the respective projection.
reduce	Non-zero rational number describing how to resample the data values. Can reduce the computational demand and can be used for fast previews.
zlim	Numeric vector with two values indicating the z-range of the 3D plot.
projection	Logical: Should a 2D projection of the 3D surface be plotted a the bottom of the box?
...	Additional arguments passed to drape.plot.

Details

For the synchronous correlation spectrum the real component (`Re`) of the complex correlation matrix must be plotted. The asynchronous spectrum is the respective imaginary component (`Im`).

See Also

See `plot_corr2d` for 2D plots. See `drape.plot` for information on the plot function.

Examples

```
data(FuranMale, package = "corr2D")
twod <- corr2d(FuranMale, Ref1 = FuranMale[1, ], corenumber = 1)

plot_corr2din3d(Mat = Re(twod$FT), specx = twod$Ref1,
    specy = twod$Ref1, reduce = 2, scalex = -175, scaley = -175,
    zlim = c(-1.5, 2.2)*10^-3, projection = FALSE,
    border = gray(0.2), theta = 25, phi = 15, add.legend = FALSE,
    Col = fields::tim.colors(64))
```

`sim2ddata` *Simulate kinetic data from two-step sequential first-order reactions*

Description

`sim2ddata` simulates kinetic data for the sequential reaction A -> B -> C with the time constants k1 and k2.

Usage

```
sim2ddata(L = 400, t = 0:10, k1 = 0.2, k2 = 0.8, X = c(1000,
  1400), A = c(1080, 1320), Aamp = c(3, 8), B = c(1120, 1280),
  Bamp = c(5, 15), C = c(1160, 1240), Camp = c(4, 9))
```

Arguments

`L`	Positive, non-zero integer specifying how many spectral variables should be used to describe the kinetic dataset.
`t`	Numeric vector containing non-negative real numbers describing at which reaction times the kinetic data should be sampled.
`k1, k2`	Positive, non-zero real numbers describing the time constants used to simulate the reactions A -> B (`k1`) and B -> C (`k2`).
`X`	Numeric vector with two values specifying the range of the simulated spectral variables.

`A, B, C` Numeric vector with two real values specifying the two signal positions of species A, B and C, respectively. It's the `mean` used in `dnorm` to simulate the signal. C and Camp may be NULL in which case only the reaction A -> B is simulated and sampled.

`Aamp, Bamp, Camp` Numeric vector with two values specifying the signal width of species A, B and C, respectively. It's the standard deviation (`sd`) used in `dnorm` to simulate the signal. C and Camp may be NULL in which case only the reaction A -> B is simulated and sampled.

Details

The simulation assumes 2 spectral signals for each of the 3 species A, B and C. The sequential reaction is defined by 2 time constants k1 and k2. The spectral information can be sampled at every point during the reaction to get an arbitrary profile of the kinetic data. The signals of the three species are modeled by a normal distribution. In addition the spectral variable is assumed to be equidistant and the number of spectral variables can also be chosen arbitrary.

Value

`sim2ddata` returns a matrix containing the kinetic data. The matrix contains the sampled reaction times by rows and the spectral variables by columns. The reaction times are the row names while the spectral variables are saved as the column names. The matrix has the ideal format to be analyzed by `corr2d`.

References

The default values are inspired by: I. Noda (2014) <DOI:10.1016/j.molstruc.2014.01.024>

Examples

```
testdata <- sim2ddata()

twodtest <- corr2d(testdata, corenumber = 1)

plot_corr2d(twodtest)
```

Index