Package: Rlibeemd (via r-universe)

October 28, 2026

Type Package

Title Ensemble Empirical Mode Decomposition (EEMD) and Its Complete Variant (CEEMDAN)

Version 1.4.4

Description An R interface for libeemd (Luukko, Helske, Räsänen, 2016) <doi:10.1007/s00180-015-0603-9>, a C library of highly efficient parallelizable functions for performing the ensemble empirical mode decomposition (EEMD), its complete variant (CEEMDAN), the regular empirical mode decomposition (EMD), and bivariate EMD (BEMD). Due to the possible portability issues CRAN version no longer supports OpenMP, but you can install OpenMP-supported version from GitHub: <https://github.com/helske/Rlibeemd/>.

License GPL-3

NeedsCompilation yes

SystemRequirements GNU GSL

Imports stats, Rcpp (>= 0.11.0)

Suggests testthat

LinkingTo Rcpp

Encoding UTF-8

BugReports https://github.com/helske/Rlibeemd/issues

RoxygenNote 7.3.2

Author Jouni Helske [aut, cre] (R interface, <https://orcid.org/0000-0001-7130-793X>), Perttu Luukko [aut] (Original libeemd C library, <https://orcid.org/0000-0003-3786-9685>)

Maintainer Jouni Helske <jouni.helske@iki.fi>

Repository https://cran.r-universe.dev

Date/Publication 2025-04-09 11:40:05 UTC

RemoteUrl https://github.com/cran/Rlibeemd

RemoteRef HEAD

RemoteSha 279a8b052178fb4e455225a03365d8c7a5a78070

Contents

bemd 2
ceemdan 3
ECG 5
eemd 6
emd 7
extrema 8
float 9
nIMFs 10
Rlibeemd 11

Index **12**

---

bemd *Bivariate EMD decomposition*

Description

Function bemd implements the Bivariate EMD (Scheme 2 in the cited article).

Usage

```
bemd(input, directions = 64L, num_imfs = 0L, num_siftings = 50L)
```

Arguments

input	Complex vector of length N. The input signal to decompose.
directions	Vector of directional angles (in radians) to use for the decomposition, or an integer defining the number of equally spaced angles to use.
num_imfs	Number of Intrinsic Mode Functions (IMFs) to compute. If num_imfs is set to zero, a value of num_imfs = emd_num_imfs(N) will be used, which corresponds to a maximal number of IMFs. Note that the final residual is also counted as an IMF in this respect, so you most likely want at least num_imfs=2.
num_siftings	Use a maximum number of siftings as a stopping criterion. If num_siftings is zero, this stopping criterion is ignored. Default is 50.

Value

Time series object of class "mts" where series corresponds to IMFs of the input signal, with the last series being the final residual. @references

1. G. Rilling, P. Flandrin, P. Goncalves and J. M. Lilly, "Bivariate Empirical Mode Decomposition", IEEE Signal Processing Letters, Vol. 14 (2007) 936–939

Examples

```
N <- 512
t <- 2 * pi * (0:(N-1))/N
input <- cos(0.3 * t) * exp(2i * t) + 0.3 * abs(sin(2.3 * t)) * exp(17i * t)

# Use evenly spaced angles as directions
num_directions <- 64
directions <- 2 * pi * 1:num_directions / num_directions
imfs <- bemd(input, directions, num_imfs = 4, num_siftings = 10)

# plot the data
plot(Re(input), Im(input), xlim = c(-1, 2))
# plot signal and the imfs
for(i in 1:4)
  points(Re(imfs[,i]), Im(imfs[,i]), col = 1 + i)
legend("bottomright", col = 1:5, legend = c("signal", paste0("IMF ",1:4)), pch = 1)

data("float")
plot(float, type = "l")
signal <- float[, 1] + float[, 2] * 1i
imfs <- bemd(signal, num_siftings = 10, num_imfs = 4)

# plot the data and the imfs
oldpar <- par()
par(mfrow = c(5, 1), mar = c(0.5, 4.5, 0.5, 0.5), oma = c(4, 0, 2, 0))
ts.plot(float, col = 1:2, lty = 1:2, ylab = "signal", gpars = list(xaxt = "n"))
for(i in 1:4) {
  ts.plot(Re(imfs[, i]), Im(imfs[, i]), col = 1:2, lty = 1:2,
    ylab = if(i < 4) paste("IMF", i) else "residual", gpars = list(xaxt = "n"))
 }
axis(1)
title(xlab = "Time (days)", main = "Bivariate EMD decomposition", outer = TRUE)
par(oldpar)
```

---

ceemdan *CEEMDAN decomposition*

Description

Decompose input data to Intrinsic Mode Functions (IMFs) with the Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) algorithm [1], a variant of EEMD.

Usage

```
ceemdan(
  input,
  num_imfs = 0,
  ensemble_size = 250L,
  noise_strength = 0.2,
```

```
  S_number = 4L,
  num_siftings = 50L,
  rng_seed = 0L,
  threads = 0L
)
```

Arguments

`input`	Vector of length N. The input signal to decompose.
`num_imfs`	Number of Intrinsic Mode Functions (IMFs) to compute. If num_imfs is set to zero, a value of num_imfs = emd_num_imfs(N) will be used, which corresponds to a maximal number of IMFs. Note that the final residual is also counted as an IMF in this respect, so you most likely want at least num_imfs=2.
`ensemble_size`	Number of copies of the input signal to use as the ensemble.
`noise_strength`	Standard deviation of the Gaussian random numbers used as additional noise. **This value is relative** to the standard deviation of the input signal.
`S_number`	Integer. Use the S-number stopping criterion for the EMD procedure with the given values of S. That is, iterate until the number of extrema and zero crossings in the signal differ at most by one, and stay the same for S consecutive iterations. Typical values are in the range 3–8. If `S_number` is zero, this stopping criterion is ignored. Default is 4.
`num_siftings`	Use a maximum number of siftings as a stopping criterion. If `num_siftings` is zero, this stopping criterion is ignored. Default is 50.
`rng_seed`	A seed for the GSL's Mersenne twister random number generator. A value of zero (default) denotes an implementation-defined default value. For `ceemdan` this does not guarantee reproducible results if multiple threads are used.
`threads`	Non-negative integer defining the maximum number of parallel threads (via OpenMP's `omp_set_num_threads`. Default value 0 uses all available threads defined by OpenMP's `omp_get_max_threads`.

Details

The size of the ensemble and the relative magnitude of the added noise are given by parameters `ensemble_size` and `noise_strength`, respectively. The stopping criterion for the decomposition is given by either a S-number [2] or an absolute number of siftings. In the case that both are positive numbers, the sifting ends when either of the conditions is fulfilled.

Value

Time series object of class `"mts"` where series corresponds to IMFs of the input signal, with the last series being the final residual.

References

1. M. Torres et al, "A Complete Ensemble Empirical Mode Decomposition with Adaptive Noise" IEEE Int. Conf. on Acoust., Speech and Signal Proc. ICASSP-11, (2011) 4144–4147
2. N. E. Huang, Z. Shen and S. R. Long, "A new view of nonlinear water waves: The Hilbert spectrum", Annual Review of Fluid Mechanics, Vol. 31 (1999) 417–457

See Also

eemd

Examples

```
imfs <- ceemdan(UKgas, threads = 1)
# trend extraction
ts.plot(UKgas, imfs[, ncol(imfs)], col = 1:2,
        main = "Quarterly UK gas consumption", ylab = "Million therms")

# CEEMDAN for logarithmic demand, note that increasing ensemble size
# will produce smoother results
imfs <- ceemdan(log(UKgas), ensemble_size = 50, threads = 1)
plot(ts.union("log(obs)" = log(UKgas), Seasonal = imfs[, 1],
     Irregular = rowSums(imfs[, 2:5]), Trend = imfs[, 6]),
     main = "Quarterly UK gas consumption")
```

ECG *Electrocardiogram Data Example ECG data from MIT-BIH Normal Sinus Rhythm Database, ECG1 of record 16265, first 2049 observations (0 to 16 seconds with sampling interval of 0.0078125 seconds)*

Description

Electrocardiogram Data

Example ECG data from MIT-BIH Normal Sinus Rhythm Database, ECG1 of record 16265, first 2049 observations (0 to 16 seconds with sampling interval of 0.0078125 seconds)

Format

A time series object.

Source

MIT-BIH Normal Sinus Rhythm Database, PhysioBank ATM, https://archive.physionet.org/cgi-bin/atm/ATM

Examples

```
data("ECG")
plot(ECG)
```

eemd *EEMD Decomposition*

Description

Decompose input data to Intrinsic Mode Functions (IMFs) with the Ensemble Empirical Mode Decomposition algorithm [1].

Usage

```
eemd(
  input,
  num_imfs = 0,
  ensemble_size = 250L,
  noise_strength = 0.2,
  S_number = 4L,
  num_siftings = 50L,
  rng_seed = 0L,
  threads = 0L
)
```

Arguments

input	Vector of length N. The input signal to decompose.
num_imfs	Number of Intrinsic Mode Functions (IMFs) to compute. If num_imfs is set to zero, a value of num_imfs = emd_num_imfs(N) will be used, which corresponds to a maximal number of IMFs. Note that the final residual is also counted as an IMF in this respect, so you most likely want at least num_imfs=2.
ensemble_size	Number of copies of the input signal to use as the ensemble.
noise_strength	Standard deviation of the Gaussian random numbers used as additional noise. **This value is relative** to the standard deviation of the input signal.
S_number	Integer. Use the S-number stopping criterion for the EMD procedure with the given values of S. That is, iterate until the number of extrema and zero crossings in the signal differ at most by one, and stay the same for S consecutive iterations. Typical values are in the range 3–8. If S_number is zero, this stopping criterion is ignored. Default is 4.
num_siftings	Use a maximum number of siftings as a stopping criterion. If num_siftings is zero, this stopping criterion is ignored. Default is 50.
rng_seed	A seed for the GSL's Mersenne twister random number generator. A value of zero (default) denotes an implementation-defined default value.
threads	Non-negative integer defining the maximum number of parallel threads (via OpenMP's omp_set_num_threads. Default value 0 uses all available threads defined by OpenMP's omp_get_max_threads.

Details

The size of the ensemble and the relative magnitude of the added noise are given by parameters `ensemble_size` and `noise_strength`, respectively. The stopping criterion for the decomposition is given by either a S-number [2] or an absolute number of siftings. In the case that both are positive numbers, the sifting ends when either of the conditions is fulfilled.

Value

Time series object of class `"mts"` where series corresponds to IMFs of the input signal, with the last series being the final residual.

References

1. Z. Wu and N. Huang, "Ensemble Empirical Mode Decomposition: A Noise-Assisted Data Analysis Method", Advances in Adaptive Data Analysis, Vol. 1 (2009) 1–41
2. N. E. Huang, Z. Shen and S. R. Long, "A new view of nonlinear water waves: The Hilbert spectrum", Annual Review of Fluid Mechanics, Vol. 31 (1999) 417–457

See Also

ceemdan

Examples

```
x <- seq(0, 2*pi, length.out = 500)
signal <- sin(4*x)
intermittent <- 0.1 * sin(80 * x)
y <- signal * (1 + ifelse(signal > 0.7, intermittent, 0))

plot(x = x,y = y,type = "l")
# Decompose with EEMD
imfs <- eemd(y, num_siftings = 10, ensemble_size = 50, threads = 1)

plot(imfs)
# High frequencies
ts.plot(rowSums(imfs[, 1:3]))
# Low frequencies
ts.plot(rowSums(imfs[, 4:ncol(imfs)]))
```

---

emd *EMD decomposition*

Description

Decompose input data to Intrinsic Mode Functions (IMFs) with the Empirical Mode Decomposition algorithm.

Usage

```
emd(input, num_imfs = 0, S_number = 4L, num_siftings = 50L)
```

Arguments

input	Vector of length N. The input signal to decompose.
num_imfs	Number of Intrinsic Mode Functions (IMFs) to compute. If num_imfs is set to zero, a value of num_imfs = emd_num_imfs(N) will be used, which corresponds to a maximal number of IMFs. Note that the final residual is also counted as an IMF in this respect, so you most likely want at least num_imfs=2.
S_number	Integer. Use the S-number stopping criterion [1] for the EMD procedure with the given values of S. That is, iterate until the number of extrema and zero crossings in the signal differ at most by one, and stay the same for S consecutive iterations. Typical values are in the range 3–8. If S_number is zero, this stopping criterion is ignored. Default is 4.
num_siftings	Use a maximum number of siftings as a stopping criterion. If num_siftings is zero, this stopping criterion is ignored. Default is 50.

Details

This is a wrapper around eemd with ensemble_size = 1 and noise_strength = 0.

Value

Time series object of class "mts" where series corresponds to IMFs of the input signal, with the last series being the final residual. @references

1. N. E. Huang, Z. Shen and S. R. Long, "A new view of nonlinear water waves: The Hilbert spectrum", Annual Review of Fluid Mechanics, Vol. 31 (1999) 417–457

See Also

eemd, ceemdan

---

extrema *Local Extrema of Time Series*

Description

Find the local minima and maxima from input data. This includes the artificial extrema added to the ends of the data as specified in the original EEMD article [1]. In the case of flat regions at the extrema, the center point of the flat region will be considered the extremal point [2].

Usage

```
extrema(input)
```

Arguments

input	Numeric vector or time series object.

Value

a list with matrices minima and maxima which give time points and values of local minima and maxima of input where time points are transformed to match the sampling times of input.

References

1. Z. Wu and N. Huang, "Ensemble Empirical Mode Decomposition: A Noise-Assisted Data Analysis Method", Advances in Adaptive Data Analysis, Vol. 1 (2009) 1–41.
2. P. Luukko, J. Helske and E. Räsänen, "Introducing libeemd: A program package for performing the ensemble empirical mode decomposition", Computational Statistics (2015).

Examples

```
ext <- extrema(UKgas)
plot(UKgas, ylim = range(ext$maxima[, 2], ext$minima[, 2]))
points(ext$maxima, col = 2, pch = 19)
points(ext$minima, col = 2, pch = 19)

# Artificial extremas obtained by extrapolating last two extrema
# Beginning of the series
lines(ext$minima[1:3, ], col = 4)
# This is discarded as it produces smaller extrema than the last observation:
b <- lm(c(ext$maxima[2:3, 2]) ~ ext$maxima[2:3, 1])$coef[2]
points(x = ext$maxima[1, 1], y = ext$maxima[2, 2] - b, col = 4,pch = 19)
lines(x = ext$maxima[1:3, 1], y = c(ext$maxima[2, 2] - b, ext$maxima[2:3, 2]), col = 4)
# End of the series
# These produce more extreme values than the last observation which is thus disregarded
lines(ext$minima[27:29, ],col = 4)
lines(ext$maxima[26:28, ],col = 4)
```

---

float — *Float Data The data are a position record from an acoustically tracked subsurface oceanographic float, used as an example data in Rilling et al (2007).*

Description

Float Data

The data are a position record from an acoustically tracked subsurface oceanographic float, used as an example data in Rilling et al (2007).

Format

A time series object.

Source

http://wfdac.whoi.edu

References

1. G. Rilling, P. Flandrin, P. Goncalves and J. M. Lilly, "Bivariate Empirical Mode Decomposition", IEEE Signal Processing Letters, Vol. 14 (2007) 936–939

Examples

```
data("float")
plot(float, type = "l")
```

`nIMFs` *Number of IMFs*

Description

Return the number of IMFs extracted from input data of length N, including the final residual. This is just [log_2(N)] for N>3.

Usage

```
emd_num_imfs(N)
```

Arguments

`N`	An integer defining the length of input data.

Value

The number of IMFs which would be extracted from input data of length N, including the final residual.

Rlibeemd *Rlibeemd: Ensemble empirical mode decomposition (EEMD) and its complete variant (CEEMDAN)*

Description

Package Rlibeemd contains functions for the ensemble empirical mode decomposition (EEMD), its complete variant (CEEMDAN) or the regular empirical mode decomposition (EMD).

Details

Package is based on the libeemd C library: https://bitbucket.org/luukko/libeemd

Author(s)

Maintainer: Jouni Helske <jouni.helske@iki.fi> (ORCID) (R interface)

Authors:

- Perttu Luukko (ORCID) (Original libeemd C library)

References

- P. Luukko, J. Helske and E. Räsänen, "Introducing libeemd: A program package for performing the ensemble empirical mode decomposition", Computational Statistics (2015).
- Z. Wu and N. Huang, "Ensemble Empirical Mode Decomposition: A Noise-Assisted Data Analysis Method", Advances in Adaptive Data Analysis, Vol. 1 (2009) 1–41.
- N. E. Huang, Z. Shen and S. R. Long, "A new view of nonlinear water waves: The Hilbert spectrum", Annual Review of Fluid Mechanics, Vol. 31 (1999) 417–457.
- Torres et al, A Complete Ensemble Empirical Mode Decomposition with Adaptive Noise IEEE Int. Conf. on Acoust., Speech and Signal Proc. ICASSP-11, (2011) 4144–4147.

See Also

Useful links:

- Report bugs at https://github.com/helske/Rlibeemd/issues

Index

∗ **datasets**
ECG, 5
float, 9

bemd, 2

ceemdan, 3, *7*, *8*

ECG, 5
eemd, *5*, 6, *8*
emd, 7
emd_num_imfs *(*nIMFs*)*, 10
extrema, 8

float, 9

nIMFs, 10

Rlibeemd, 11
Rlibeemd-package *(*Rlibeemd*)*, 11