

Package: cspec (via r-universe)

May 14, 2026

Type Package

Title Complete Discrete Fourier Transform (DFT) and Periodogram

Version 0.1.2

Date 2020-06-29

Author Junho Yang

Maintainer Junho Yang <junhoyang@stat.tamu.edu>

Description Calculate the predictive discrete Fourier transform, complete discrete Fourier transform, complete periodogram, and tapered complete periodogram. This algorithm is based on the preprint ``Spectral methods for small sample time series: A complete periodogram approach" (2020) by Sourav Das, Suhasini Subba Rao, and Junho Yang.

License GPL-2

NeedsCompilation no

Repository <https://cranhaven.r-universe.dev>

Date/Publication 2026-05-14 13:02:00 UTC

RemoteUrl <https://github.com/cranhaven/cranhaven.r-universe.dev>

RemoteRef package/cspec

RemoteSha fc8b49c6f967333dcc7c8b37bfc629bd678ef7bf

RemoteSubdir cspec

Contents

complete.pgram	2
completeDFT	3
fft2	4
predictiveDFT	5
taperDFT	6
tapered.complete.pgram	7

Index	9
--------------	----------

complete.pgram *Complete Periodogram*

Description

Function to calculate the complete periodogram.

Usage

```
complete.pgram(x, freq = 2 * pi * (1:length(x))/length(x), thres=NULL, ...)
```

Arguments

x	time series vector.
freq	frequency vector.
thres	(optional) positive threshold value.
...	Arguments used in the predictiveDFT function.

Details

The default frequencies are $2\pi(1:n)/n$, where n is a length of a time series.

Value

Real part of the complete periodogram vector.

Author(s)

Junho Yang

References

S. Das, S. Subba Rao, and J. Yang. *Spectral methods for small sample time series: A complete periodogram approach*. Submitted, 2020.

See Also

[predictiveDFT](#)

Examples

```
set.seed(123)
x <- arima.sim(model=list(ar=0.7), n=100)
v <- complete.pgram(x)
```

completeDFT	<i>Complete DFT</i>
-------------	---------------------

Description

Function to calculate the complete DFT.

Usage

```
completeDFT(x, freq = 2 * pi * (1:length(x))/length(x), ...)
```

Arguments

x	time series vector.
freq	frequency vector.
...	Arguments used in the predictiveDFT function.

Details

The default frequencies are $2\pi(1:n)/n$, where n is a length of a time series.

Value

Complex valued complete DFT vector.

Author(s)

Junho Yang

References

S. Das, S. Subba Rao, and J. Yang. *Spectral methods for small sample time series: A complete periodogram approach*. Submitted, 2020.

See Also

[predictiveDFT](#)

Examples

```
set.seed(123)
x <- arima.sim(model=list(ar=0.7), n=100)
v <- completeDFT(x)
```

`fft2`*A new DFT function*

Description

Function to calculate the DFT of time series vector as in the reference.

Usage

```
fft2(x, freq = 2 * pi * (1:length(x))/length(x))
```

Arguments

<code>x</code>	time series vector.
<code>freq</code>	frequency vector.

Details

\sqrt{n} standardized. The default frequencies are $2\pi(1:n)/n$, where n is a length of a time series.

Value

DFT vector.

Author(s)

Junho Yang

References

S. Das, S. Subba Rao, and J. Yang. *Spectral methods for small sample time series: A complete periodogram approach*. Submitted, 2020.

See Also

[taperDFT](#)

Examples

```
set.seed(123)
x <- arima.sim(model=list(ar=0.7), n=100)
v <- fft2(x)
```

predictiveDFT *Predictive DFT.*

Description

Function to calculate the predictive DFT.

Usage

```
predictiveDFT(x, freq = 2 * pi * (1:length(x))/length(x), taper = FALSE, ar = NULL, ...)
```

Arguments

x	time series vector.
freq	frequency vector.
taper	logical vector. If TRUE, use the tapered time series to estimate an AR coefficients. Default is FALSE.
ar	predetermined AR coefficients to evaluate the predictive DFT. Default is NULL.
...	Arguments passing ar() function.

Details

If ar vector is predetermined, then we evaluate the predictive DFT using the this AR coefficient vector (see eq (2.2) of the reference). If ar is not specified (default), then we fit the best fitting AR(p) model using AIC and Yule-Walker estimator.

Value

The complex valued predictive DFT vector.

Author(s)

Junho Yang

References

S. Das, S. Subba Rao, and J. Yang. *Spectral methods for small sample time series: A complete periodogram approach*. Submitted, 2020.

See Also

[complete.pgram](#), [tapered.complete.pgram](#)

Examples

```

set.seed(123)
x <- arima.sim(model=list(ar=0.7), n=100)

v1 <- predictiveDFT(x) #default
v2 <- predictiveDFT(x,ar=c(0.7)) #predictive DFT using AR(1) model with coefficient: 0.7.
v3 <- predictiveDFT(x,taper=TRUE) #Using tapered time series to fit the best AR model.
v4 <- predictiveDFT(x, method="ols") #Using ols method to fit the best AR model.

```

taperDFT

Tapered DFT

Description

Function to calculate the tapered DFT.

Usage

```
taperDFT(x, freq = 2 * pi * (1:length(x))/length(x), regularization.type = "1", p = 0.1)
```

Arguments

x	time series vector.
freq	frequency vector.
regularization.type	character. if "1": sum of the taper equals to n, if "2": square sum of the taper equals to n. Default is "1".
p	proportion of taper.

Details

Tapered DFT. The default taper is a Tukey's (or Cosine-bell) taper.

Value

Complex valued tapered DFT vector.

Author(s)

Junho Yang.

See Also

[fft2](#)

Examples

```
set.seed(123)
x <- arima.sim(model=list(ar=0.7), n=100)
v <- taperDFT(x)
```

tapered.complete.pgram

Tapered Complete Periodogram

Description

Function to calculate the tapered complete periodogram.

Usage

```
tapered.complete.pgram(x, freq=2*pi*(1:length(x))/length(x), taperx=NULL, thres=NULL, ...)
```

Arguments

x	time series vector.
freq	frequency vector.
taperx	predetermined tapered DFT. If NULL, we use default settings of the taperedDFT function to calculate the tapered DFT.
thres	positive threshold value.
...	Arguments used in the predictiveDFT function.

Details

The default frequencies are $2\pi(1:n)/n$, where n is a length of a time series.

Value

Real part of the tapered complete periodogram vector.

Author(s)

Junho Yang

References

S. Das, S. Subba Rao, and J. Yang. *Spectral methods for small sample time series: A complete periodogram approach*. Submitted, 2020.

See Also

[predictiveDFT](#), [complete.pgram](#)

Examples

```
set.seed(123)
x <- arima.sim(model=list(ar=0.7), n=100)
v <- tapered.complete.pgram(x)
```

Index

complete.pgram, [2](#), [5](#), [7](#)
completeDFT, [3](#)

fft2, [4](#), [6](#)

predictiveDFT, [2](#), [3](#), [5](#), [7](#)

taperDFT, [4](#), [6](#)

tapered.complete.pgram, [5](#), [7](#)