Package 'ptest'

October 14, 2022

Type Package

Title Periodicity Tests in Short Time Series

Version 1.0-8

Date 2016-11-12

Author Yuanhao Lai and A.I. McLeod

Maintainer A.I. McLeod <aimcleod@uwo.ca>

Depends R (>= 3.0),

Description Implements p-value computations using an approximation to the cumulative distribution function for a variety of tests for periodicity. These tests include harmonic regression tests with normal and double exponential errors as well as modifications of Fisher's g test. An accompanying vignette illustrates the application of these tests.

License GPL (>= 2)

LazyData true

NeedsCompilation yes

Imports quantreg (>= 5.0)

RoxygenNote 5.0.1

Suggests knitr, boot, lattice, rmarkdown, GeneCycle,

VignetteBuilder knitr

Repository CRAN

Date/Publication 2016-11-12 21:41:37

R topics documented:

---

alpha   *Microarray time series experiment for yeast cell cycle from alpha ex-*
                                periment

Description

6,178 yeast genes expression measures (log-ratios) with series length 18 from the alpha factor experiment.

Usage

```
data(alpha)
```

Format

Matrix with 6178 rows and 18 columns. Some missing data. Rows and columns are labelled. -attr(*, "dimnames")=List of 2 ..$: chr [1:6178] "YAL001C" "YAL002W" "YAL003W" "YAL004W" ... ..$: chr [1:18] "alpha0" "alpha7" "alpha14" "alpha21" ...

Source

The data is extracted from the ExpressionSet of the R package yeastCC.

References

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., ... & Futcher, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. Molecular biology of the cell, 9(12), 3273-3297.

Dudoit S (2016). yeastCC: Spellman et al. (1998) and Pramila/Breeden (2006) yeast cell cycle microarray data. R package version 1.12.0.

Examples

```
data(alpha)
qqnorm(colMeans(alpha, na.rm=TRUE))
qqnorm(rowMeans(alpha, na.rm=TRUE))
```

B1 *Benchmark set B1*

Description

List for yeast genes which are **most** likely to be periodic (the benchmark set 1 in de Lichtenberg et al. (2005)).

Usage

```
data(B1)
```

Format

A vector containg 113 genes' names.

Details

A total of 113 genes previously identified as periodically expressed in small-scale experiments. The set encompasses the 104 genes used by Spellman et al. (1998) and nine genes added by Johansson et al. (2003).

Source

The raw data can be downloaded from http://www.cbs.dtu.dk/cellcycle/yeast_benchmark/benchmark.php.

References

De Lichtenberg, U., Jensen, L. J., Fausboll, A., Jensen, T. S., Bork, P.,& Brunak, S. (2005). Comparison of computational methods for the identification of cell cycle-regulated genes. Bioinformatics, 21(7), 1164-1171.

Examples

```
data(alpha)
data(B1)
alphaB1 <- alpha[rownames(alpha) \%in\% B1, ]
```

B2 *Benchmark set B2*

Description

List for yeast genes which are **most** likely to be periodic (the benchmark set 2 in de Lichtenberg et al. (2005)).

Usage

```
data(B2)
```

Format

A vector containg 352 genes' names.

Details

Genes whose promoters were bound (P-value below 0.01) by at least one of nine known cell cycle transcription factors in both of the Chromatin IP studies by Simon et al. (2001) and Lee et al. (2002). To obtain a benchmark set that is independent of B1, we removed all genes included in B1 (50). The resulting benchmark set, B2, consists of 352 genes of which many should be expected to be cell cycle regulated, since their promoters are associated with known stage-specific cell cycle transcription factors.

Source

The raw data can be downloaded from http://www.cbs.dtu.dk/cellcycle/yeast_benchmark/benchmark.php.

References

De Lichtenberg, U., Jensen, L. J., Fausboll, A., Jensen, T. S., Bork, P.,& Brunak, S. (2005). Comparison of computational methods for the identification of cell cycle-regulated genes. Bioinformatics, 21(7), 1164-1171.

Examples

```
data(alpha)
data(B2)
alphaB2 <- alpha[rownames(alpha)  \%in\% B2,]
```

B3   *Benchmark set B3*

Description

List for yeast genes which are **less** likely to be periodic (the benchmark set 3 in de Lichtenberg et al. (2005)).

Usage

```
data(B3)
```

Format

A vector containg 518 genes' names.

Details

Genes annotated in MIPS (Mewes et al., 2002) as 'cell cycle and DNA processing'. From these, we removed genes annotated specifically as 'meiosis' and genes included in B1 (67), leaving 518 genes. As a large number of genes involved in the cell cycle are not subject to transcriptional regulation (not periodic), and because B1 was explicitly removed, a relatively small fraction of these genes should be expected to be periodically expressed.

Source

The raw data can be downloaded from http://www.cbs.dtu.dk/cellcycle/yeast_benchmark/benchmark.php.

References

De Lichtenberg, U., Jensen, L. J., Fausboll, A., Jensen, T. S., Bork, P.,& Brunak, S. (2005). Comparison of computational methods for the identification of cell cycle-regulated genes. Bioinformatics, 21(7), 1164-1171.

Examples

```
data(alpha)
data(B3)
alphaB3 <- alpha[rownames(alpha)  \%in\% B3,]
```

Cc	*Microarray time series experiment for Caulobacter crescentus bacterial cell cycle*

Description

In this microarray experiment there are 3062 genes measured every 1 hour. There are 19 is missing gene labels and these have been given labels ORFna1,...,ORFna19. There 310 with duplicate labels. Of these duplicate labels, 295 are duplicated twice, 12 are duplicated 3 times and 3 are duplicated 4 times. Duplicate labels are renamed ORF... to ORF...a and ORF...b etc.

Usage

```
data(Cc)
```

Format

Matrix with 3062 rows and 11 columns. Some missing data. Rows and columns are labelled. - attr(*, "dimnames")=List of 2 ..$: chr [1:3062] "ORF06244a" "ORF03152a" "ORF03156a" "ORF03161a" ... ..$: chr [1:11] "1" "2" "3" "4" ...

Details

Gene expression from synchronized cultures of the bacterium Caulobacter crescentus (Laub et al., 2000). (Laub et al., 2000) identified 553 genes whose messenger RNA levels varied as a function of the cell cycle but their statistical analysis was not very sophisticated and they probably identified too many genes. Wichert et al. (2004) found that 44 genes were found which displayed periodicity based on the Fisher's g-test using a FDR with q=0.05.

References

Laub, M.T., McAdams,H.H., Feldblyum,T., Fraser,C.M. and Shapiro,L. (2000) Global analysis of the genetic network controlling a bacterial cell cycle Science, 290, 2144-2148.

Wichert,S., Fokianos K. and Strimmer K. (2004) Identifying periodically expressed transcrips in microarray time series data. Bioinformatics, 18, 5-20.

Examples

```
data(Cc)
qqnorm(colMeans(Cc, na.rm=TRUE))
qqnorm(rowMeans(Cc, na.rm=TRUE))
```

---

cdc15   *Microarray time series experiment for yeast cell cycle from cdc15 experiment*

Description

6,178 yeast genes expression measures (log-ratios) with series length 24 from the cdc15 experiment.

Usage

```
data(cdc15)
```

Format

Matrix with 6178 rows and 24 columns. Some missing data. Rows and columns are labelled. - attr(*, "dimnames")=List of 2 ..$: chr [1:6178] "YAL001C" "YAL002W" "YAL003W" "YAL004W" ... ..$: chr [1:24] "cdc15.10" "cdc15.30" "cdc15.50" "cdc15.70" ...

Source

The data is extracted from the ExpressionSet of the R package yeastCC.

References

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., ... & Futcher, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. Molecular biology of the cell, 9(12), 3273-3297.

Dudoit S (2016). yeastCC: Spellman et al. (1998) and Pramila/Breeden (2006) yeast cell cycle microarray data. R package version 1.12.0.

Examples

```
data(cdc15)
qqnorm(colMeans(cdc15, na.rm=TRUE))
qqnorm(rowMeans(cdc15, na.rm=TRUE))
```

---

cdc28   *Microarray time series experiment for yeast cell cycle from cdc28 experiment*

Description

6,178 yeast genes expression measures (log-ratios) with series length 17 from the cdc28 experiment.

Usage

```
data(cdc28)
```

Format

Matrix with 6178 rows and 17 columns. Some missing data. Rows and columns are labelled. - attr(*, "dimnames")=List of 2 ..$: chr [1:6178] "YAL001C" "YAL002W" "YAL003W" "YAL004W" ... ..$: chr [1:17] "cdc28.0" "cdc28.10" "cdc28.20" "cdc28.30" ...

Source

The data is extracted from the ExpressionSet of the R package yeastCC.

References

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., ... & Futcher, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. Molecular biology of the cell, 9(12), 3273-3297.

Dudoit S (2016). yeastCC: Spellman et al. (1998) and Pramila/Breeden (2006) yeast cell cycle microarray data. R package version 1.12.0.

Examples

```
data(cdc28)
qqnorm(colMeans(cdc28, na.rm=TRUE))
qqnorm(rowMeans(cdc28, na.rm=TRUE))
```

---

fitHReg | *Fits Three Parameter Harmonic Regression*

Description

Estimates A, B and f in the harmonic regression, $y(t) = mu + A*cos(2*pi*f*t) + B*sin(2*pi*f*t) + e(t)$. The default algorithm is enumerative but an exact non-linear LS option is also provided.

Usage

```
fitHReg(y, t = 1:length(y), algorithm = c("enumerative", "exact"))
```

Arguments

y	series.
t	Time points.
algorithm	method for the optimization

Details

Program is interfaced to C for efficient computation.

Value

Object of class "HReg" produced.

Author(s)

A.I. McLeod and Yuanhao Lai

Examples

```
set.seed(193)
z <- simHReg(10,f=2.5/10,1,1)
ans <- fitHReg(z)
ans$freq #optimal frequency = 0.2376238
#
#ORF06806 in Cc dataset.
z<-c(0.42, 0.89, 1.44, 1.98, 2.21, 2.04, 0.82, 0.62, 0.56, 0.8, 1.33)
ans2 <- fitHReg(z, algorithm="exact")
sum(resid(ans2)^2) #0.2037463
ans1 <- fitHReg(z)
sum(resid(ans1)^2) #0.242072
#compare with nls()
t <- 1:length(z)
ans <- nls(z ~ mu+alpha*cos(2*pi*lambda*t+phi),
           start=list(mu=1, alpha=1, lambda=0.1, phi=0.0))
coefficients(ans)
sum(resid(ans)^2) #0.2037
```

---

pgram	*Periodogram computation*

Description

The periodogram is computed.

Usage

```
pgram(z, fr = "default", method = c("periodogram", "regression"))
```

Arguments

z	time series vector of length n, say.
fr	use "default" for usual Fourier frequencies, $1/n, ..., floor(n/2)/n$. Set fr = N, to evaluate the periodogram at the Fourier frequencies corresponding to a time series of length N. Finally set fr to any desired set of frequencies. Note frequencies are in cycles per unit time sometimes called temoral frequency to distinguish from angular frequency. Both are widely used in time series.
method	either periodogram or regression

Details

Uses FFT. So if the length of z is a highly composite number, the computation is very efficient. Otherwise the usual DFT is used.

Value

Periodogram evaluated at the Fourier frequencies or R-square.

Author(s)

A.I. McLeod and Yuanhao Lai

Examples

```
z<-sunspot.year
n<-length(z)
I<-pgram(z)
f<-I[,1]
I <- I[,2]
plot(f, I, xlab="f", ylab="f", type="l")
title(main="Periodogram for Annual Sunpots, 1700-1988")
#
z<-c(0.42, 0.89, 1.44, 1.98, 2.21, 2.04, 0.82, 0.62, 0.56, 0.8, 1.33)
fr <- (1:50)/101
pgram(z)
pgram(z, fr=101)
pgram(z, fr=fr)
pgram(z, method="regression")
pgram(z, method="regression", fr=101)
pgram(z, method="regression", fr=fr)
```

---

ptestg   *Test short time series for periodicity based on periodograms*

Description

This function is used to test the existence of the periodicity for a short time series (length<=100). Several methods based on periodograms are provided with the response surface method implemented for efficiently obtaining accurate p-values.

Usage

```
ptestg(z, method = c("Fisher", "robust", "extended", "extendedRobust",
  "FisherRSR"), multiple = FALSE)
```

Arguments

z	A series or a matrix containg series as columns
method	The statistical test to be used. See details for more information.
multiple	Indicating whether z contains multiple series.

Details

The null hypothesis is set as no peridicities, H0: f=0. Discriptions of different test statistics (methods) are as follow:

`Fisher`: The Fisher's g test statistic. The p-value is computed directly from the exact distribution.

`robust`: The robust g test proprosed in Ahdesmaki et al. (2005), where the p-value is computed by the response surface regression method.

`extended`: The extended Fisher's g test statistic, which extend the Fisher's g test by enlarging the searching region of the frequency from the fourier frequencies to be $En = j/101|j = 1, \ldots, 50 and j/101 \geq 1/n$. The p-value is computed by the response surface regression method.

`extendedRobust`: Extend the frequency searching region of the robust $En = j/101|j = 1, \ldots, 50 and j/101 \geq 1/n$. The p-value is computed by the response surface regression method.

`FisherRSR`: Only for experimental purposes, the Fisher;s g test with p-value computed form the response surface regression method.

Value

Object of class "Htest" produced.

An object of class "Htest" is a list containing the following components:

obsStat	Vector containing the observed test statistics.
pvalue	Vector containing the p-values of the selected tests.
freq	Vector containing the estimated frequencies.

Author(s)

Yuanhao Lai and A.I. McLeod

References

Fisher, R.A. (1929). Tests of significance in harmonic analysis. Proc. Roy. Soc. A, 125, 54-59.

Ahdesmaki, M., Lahdesmaki, H., Pearson, R., Huttunen, H., and Yli-Harja O.(2005). *BMC Bioinformatics* **6**:117. http://www.biomedcentral.com/1471-2105/6/117.

MacKinnon, James (2001) : Computing numerical distribution functions in econometrics, Queen's Economics Department Working Paper, No. 1037.

See Also

ptestReg

Examples

```
# Simulate the harmonic regression model with standard Gaussian error terms
set.seed(193)
## Non-Fourier frequency
z <- simHReg(n = 14, f=2/10, A = 2, B = 1, model="Gaussian",sig=1)
ptestg(z,method="Fisher")
ptestg(z,method="robust")
ptestg(z,method="extended")
ptestg(z,method="extendedRobust")
ptestg(z,method="FisherRSR")

# Perform tests on the alpha factor experiment
data(alpha)
## Eliminate genes with missing observations
alpha.nonNA <- alpha[complete.cases(alpha),]
## Using the multiple option to do the test for all the genes
## Transpose the data set so that each column stands for a gene
alpha.nonNA <- t(alpha.nonNA)
result <- ptestg(alpha.nonNA, method = "extended",multiple=TRUE)
str(result)


# The movtivating example: gene ORF06806 in Cc
data(Cc)
x <- Cc[which(rownames(Cc)=="ORF06806"),]
plot(1:length(x),x,type="b", main="ORF06806",
     xlab="time",ylab="Gene expression")
ptestg(x,method="Fisher") #Fail to detect the periodicity
ptestg(x,method="robust")
ptestg(x,method="extended")
```

---

ptestReg	*Test short time series for periodicity with maximum likelihood ratio tests*

Description

This function is used to test the existence of the periodicity for a short time series (length<=100). Likelihood ratio tests under the Gaussian or the Laplace assumptions are provided with the response surface method implemented for efficiently obtaining accurate p-values.

Usage

```
ptestReg(z, method = c("LS", "L1"), multiple = FALSE)
```

Arguments

z   A series or a matrix containing series as columns

| method | The statistical test to be used. See details for more information. |
| multiple | Indicating whether z contains multiple series. |

Details

The null hypothesis is set as no peridicities, H0: f=0. Discriptions of different test statistics (methods) are as follow:

LS: The -2 loglikelihood ratio test statistic based on the likelihood ratio test with normal noises, where the p-values are efficiently computed by the response surface method.

L1: The -2 loglikelihood ratio test statistic based on the likelihood ratio test with Laplace noises, where the p-values are efficiently computed by the response surface method.

Value

Object of class "Htest" produced.

An object of class "Htest" is a list containing the following components:

obsStat	Vector containing the observed test statistics.
pvalue	Vector containing the p-values of the selected tests.
freq	Vector containing the estimated frequencies.

Author(s)

Yuanhao Lai and A.I. McLeod

References

Islam, M.S. (2008). Peridocity, Change Detection and Prediction in Microarrays. Ph.D. Thesis, The University of Western Ontario.

Li, T. H. (2010). A nonlinear method for robust spectral analysis. Signal Processing, IEEE Transactions on, 58(5), 2466-2474.

MacKinnon, James (2001) : Computing numerical distribution functions in econometrics, Queen's Economics Department Working Paper, No. 1037.

See Also

fitHReg, ptestg

Examples

```
# Simulate the harmonic regression model with standard Gaussian error terms
set.seed(193)
# Non-Fourier frequency
z <- simHReg(n = 14, f=2/10, A = 2, B = 1, model="Gaussian",sig=1)
ptestReg(z,method = "LS") #Normal likelihood ratio test
ptestReg(z,method = "L1") #Laplace likelihood ratio test
fitHReg(z, algorithm="exact") #the nls fitted result
```

```
# Performe tests on the alpha factor experiment
data(alpha)
## Eliminate genes with missing observations
alpha.nonNA <- alpha[complete.cases(alpha),]
## Using the multiple option to do the test for all the genes
## Transpose the data set so that each column stands for a gene
alpha.nonNA <- t(alpha.nonNA)
result <- ptestReg(alpha.nonNA, method = "LS",multiple=TRUE)
str(result)


# The movtivating example: gene ORF06806 in Cc
data(Cc)
x <- Cc[which(rownames(Cc)=="ORF06806"),]
plot(1:length(x),x,type="b", main="ORF06806",
     xlab="time",ylab="Gene expression")
ptestg(x,method="Fisher") #Fail to detect the periodicity
ptestReg(x,method="LS") #The periodicity is significantly not zero
ptestReg(x,method="L1") #The periodicity is significantly not zero
```

| simHReg | *Simulate harmonic regression models* |

Description

Simulates a harmonic regression. Possible types of models are normal, t(5), Laplace, cubic and AR1.

Usage

```
simHReg(n, f, A, B, model = c("Gaussian", "t5", "Laplace", "cubic", "AR1"),
  phi = 0, sig = 1)
```

Arguments

n	Length of series.
f	Frequency.
A	Cosine amplitude.
B	Sine amplitude.
model	The model used for generating the error term. See details.
phi	Only used if AR1 error distribution is selected.
sig	The standard error of the series.

Details

Generate a harmonic series y with length n, where $y_t = A * cos(2 * pi * f * t) + B * sin(2 * pi * f * t) + sig * e_t$, $t = 1, ..., n$, and e comes from one of the following specified distributions with mean 0 and standard error 1:

Gaussian: A standard normal distribution (i.i.d.).

t5: A t distribution with 5 degrees of freedom (i.i.d., standardized to mean 0 and variance 1).

Laplace: A Laplace (double exponential) distribution (i.i.d., standardized to mean 0 and variance 1).

cubic: A standard normal distribution for e, but $y = y^3$ this time.

AR1: An AR(1) series with autocorrelation paramater phi (standardized to mean 0 and variance 1).

Value

Vector of length n, simulated harmonic series.

Author(s)

A.I. McLeod and Yuanhao Lai

References

McLeod, A.I., Yu, Hao and Krougly, Z. (2007), Algorithms for Linear Time Series Analysis: With R Package, Journal of Statistical Software 23, 5 1-26.

See Also

fitHReg, ptestReg

Examples

```
#Simulate the harmonic regression model with standard Gaussian error terms
z <- simHReg(10, f=2/10, 1, 2, model="Gaussian",sig=1) #Fourier Frequency
plot(1:10,z,type="b")

#Simulate the AR(1) errors
z <- simHReg(10, f=0/10, 0,0, model="AR1",phi=0.2,sig=1)
acf(z)
```

Index