

# Package ‘ClamR’

October 12, 2022

**Type** Package

**Title** Time Series Modeling for Climate Change Proxies

**Version** 2.1-1

**Date** 2015-07-27

**Imports** stats

**Author** Jonathan M. Lees

**Maintainer** Jonathan M. Lees<jonathan.lees@unc.edu>

**Description** Implementation of the Wilkinson and Ivany (2002) approach to paleoclimate analysis, applied to isotope data extracted from clams.

**License** GPL

**LazyLoad** yes

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2015-07-29 06:21:30

## R topics documented:

ClamR-package . . . . .	2
CLAM1 . . . . .	3
climate . . . . .	4
elliott_yr1 . . . . .	5
elliott_yr2 . . . . .	6
error.bar . . . . .	7
NextPow2 . . . . .	8
otolith . . . . .	9
plotproxy.error . . . . .	10
proxyA . . . . .	12
proxyJK . . . . .	13
RESCALE . . . . .	15
rwp_limpet . . . . .	16
SinMod . . . . .	17
window.size . . . . .	18

---

ClamR-package	<i>Climate Change with Proxies</i>
---------------	------------------------------------

---

### Description

This program implements and improves upon the Wilkinson and Ivany approach to climate time series modeling. The jackknife is used to estimate the 95 percent confidence bounds for the modeled estimates. dx should be chosen to be approximately half a cycle or more.

### Details

Package:	ClamR
Type:	Package
Version:	1.0
Date:	2009-12-23
License:	GPL
LazyLoad:	yes

### Author(s)

Jonathan M. Lees

Maintainer: Jonathan M. Lees<jonathan.lees@unc.edu>

### References

Wilkinson, B. H. and Ivany, L. C., Paleoclimatic inference from stable isotope profiles of accretionary biogenic hardparts; a quantitative approach to the evaluation of incomplete data, *Palaeogeography, Palaeoclimatology, Palaeoecology*, vol. 185, no. 1-2, pp.95-114, 01 Sep 2002.

Wang, T., Surge, D., and Lees, J. M., (2015) ClamR: A Statistical Evaluation of Isotopic and Temperature Records in Sclerochronologic Studies. *Palaeogeography, Palaeoclimatology, Palaeoecology*, doi:10.1016/j.palaeo.2015.07.008.

### Examples

```
## Not run:  
data(CLAM1)  
x = CLAM1$x  
y = CLAM1$y  
  
dx = 3.392  
  
gout = proxyJK(x, y, dx)
```

```
plotproxy1(x, y, gout)

par(mfrow=c(2,1))

plotproxy.error(x, y, gout, type = 1)
plotproxy.error(x, y, gout, type = 2)

par(mfrow=c(2,1))
plotproxy.error(x, y, gout, type = 2)

plotproxy.all2(gout, YAXstyle=1 )

## End(Not run)
```

---

CLAM1

*Clam Proxy Data*

---

### Description

Proxy data from the Orkney Islands

### Usage

```
data(CLAM1)
```

### Format

The format is: List of 2 \$ x: num [1:91] 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 ... \$ y: num [1:91] 2.14 2.11 2.4 2.21 2.32 2.44 2.85 2.78 2.27 2.05 ...

### Details

Duplicate data has been removed.

### Source

One-year data interval (4.68-9.31 mm) of  $\delta-18O$  record of an archaeological limpet *Patella vulgata* (specimen QG2-1064-1) from Orkney, Scotland (Surge and Barrett, 2012).

### References

Wang, T., Surge, D., and Lees, J. M., (2015) ClamR: A Statistical Evaluation of Isotopic and Temperature Records in Sclerochronologic Studies. *Palaeogeography, Palaeoclimatology, Palaeoecology*, doi:10.1016/j.palaeo.2015.07.008.

**Examples**

```

data(CLAM1)
## maybe str(CLAM1) ; plot(CLAM1) ...
plot(CLAM1$x, CLAM1$y, type="b", xlab="Distance", ylab="d180" )

## Not run:
##### this is an example from Wang et al.:
#### it takes too long to run on CRAN, but should work

shellx=CLAM1$x[38:70]
shelly=CLAM1$y[38:70]

window_shell=windowsize(shellx,shelly,1.8,9.4,0.2)

#the window size is 5mm, and make all the plots together
gout_shell = proxyJK(shellx, shelly, 5)

par(mfrow=c(3,2))
plot(shellx,shelly,type="b", xlab="Distance from Margin (mm)",
      ylab=expression(delta*"180(ppm VPDB)"),
      xlim=c(4,10), ylim = c(1.5,4))
plot((window_shell$win)/2,window_shell$error,xlab="Window Size (mm)",
      ylab="Error", xlim=c(1.6/2,9.4/2), ylim=c(0,0.5))
abline(v=4.63/2, lty="dotdash",col="black")
abline(v=5/2, col="black")
plotproxy1(shellx, shelly, gout_shell, xlim=c(4,10), ylim = c(1.5,4),
            xlab="Distance from Margin (mm)",
            ylab=expression(delta*"180(ppm VPDB)"), main="")
plotproxy.all(gout_shell,YAXstyle=1, xlim=c(4,10), ylim1=c(0,4),
              ylim2=c(-15,5))
plotproxy.error(shellx, shelly, gout_shell, type = 1, xlim=c(4,10),
                ylim = c(1.5,4), xlab="Distance from Margin (mm)",
                ylab=expression(delta*"180(ppm VPDB)"))
plotproxy.error(shellx, shelly, gout_shell, type = 2, xlim=c(4,10),
                ylim = c(1.5,4), xlab="Distance from Margin (mm)",
                ylab=expression(delta*"180(ppm VPDB)"))

## End(Not run)

```

---

climate

*Climate Record At Croig Cave*


---

**Description**

Application to modern climate record at Croig Cave and make comparison between reconstructed temperatures and instrumentally measured temperatures.

**Usage**

```
data("climate")
```

**Format**

A data frame with 360 observations on the following 3 variables.

Month a numeric vector

overall a numeric vector

Temperature a numeric vector

**Source**

Monthly sea surface temperature (SST) record for the years 1961-1990 derived from observations near Croig Cave, an archaeological site on the Isle of Mull in the Hebrides Islands west of mainland Scotland (Extended Reconstructed Sea Surface Temperature, Smith and Reynolds, 2004).

**References**

Wang, T., Surge, D., and Lees, J. M., (2015) ClamR: A Statistical Evaluation of Isotopic and Temperature Records in Sclerochronologic Studies. *Palaeogeography, Palaeoclimatology, Palaeoecology*, doi:10.1016/j.palaeo.2015.07.008.

**Examples**

```
data(climate)
climate_month <- climate$overall
climate_temp <- climate$Temperature

plot(climate_month,climate_temp,type="l",
      xlab="month",ylab=expression(paste("Temperature ("^"o", "C)")))
```

---

elliot\_yr1

*Elliot Data Summer*

---

**Description**

Two years of data from Elliot

**Usage**

```
data("elliot_yr1")
```

**Format**

The format is: List of 4 \$ date1 : num [1:28] 1995 1995 1995 1995 1995 ... \$ d18o1 : num [1:28] -0.036 0.244 0.525 0.332 0.148 -0.43 -0.583 -0.366 -0.641 -0.86 ... \$ date\_temp1: num [1:31] 1995 1995 1995 1995 ... \$ d18o\_pred1: num [1:31] -0.58 0 0.54 0.88 0.66 0.35 0.09 -0.09 -0.53 -1.1 ...

**Details**

Data consists of date,  $\delta 18O$ , temperature and predicted anomaly for two years of data.

**Source**

Data sets are selected from the  $\delta 18O$  record of a modern *Mercenaria mercenaria* shell collected live from Cedar Key in northern Florida and analyzed by Elliot et al. (2003). Series *elliott\_yr1* records one summer (including the most negative  $\delta 18O$ ). Series *elliott\_yr2* records one winter (including the most positive  $\delta 18O$ ).

Because the modern *Mercenaria mercenaria* shell by Elliot et al. (2003) is well dated and its in situ records of SST and salinity are available, the predicted  $\delta 18O$  are also derived from the local instrumental data. Temp is the predicted  $\delta 18O$  for the summer interval of Year1 and Temp2 is the predicted  $\delta 18O$  for the winter interval of Year2.

**References**

Wang, T., Surge, D., and Lees, J. M., (2015) ClamR: A Statistical Evaluation of Isotopic and Temperature Records in Sclerochronologic Studies. *Palaeogeography, Palaeoclimatology, Palaeoecology*, doi:10.1016/j.palaeo.2015.07.008.

**Examples**

```
data(elliott_yr1)
plot(elliott_yr1$date1,elliott_yr1$d18o1,xlab="Age(years)",
      ylab=expression(delta*'180(ppm VPDB)'), xlim=c(1994.9,1996.2), ylim=c(-2.5,2))
```

---

elliott\_yr2

*Elliot Data Winter*

---

**Description**

Winter season of data from Elliot data.

**Usage**

```
data("elliott_yr2")
```

**Format**

The format is: List of 4 \$ date1 : num [1:28] 1995 1995 1995 1995 1995 ... \$ d18o1 : num [1:28] -0.036 0.244 0.525 0.332 0.148 -0.43 -0.583 -0.366 -0.641 -0.86 ... \$ date\_temp1: num [1:31] 1995 1995 1995 1995 ... \$ d18o\_pred1: num [1:31] -0.58 0 0.54 0.88 0.66 0.35 0.09 -0.09 -0.53 -1.1 ...

**Details**

Data consists of date,  $\delta O18$ , temperature and predicted data for winter season of data.

**Source**

See explanation in `elliot_yr1`.

**References**

Wang, T., Surge, D., and Lees, J. M., (2015) ClamR: A Statistical Evaluation of Isotopic and Temperature Records in Sclerochronologic Studies. *Palaeogeography, Palaeoclimatology, Palaeoecology*, doi:10.1016/j.palaeo.2015.07.008.

**Examples**

```
data(elliot_yr2)
plot(elliot_yr2$date2, elliot_yr2$d18o2, xlab="date(year)",
     ylab=expression(delta*'180(ppm VPDB)'), xlim=c(1995.2, 1996.85), ylim=c(-2.5, 1.8))
```

---

error.bar

*Error bar plot*

---

**Description**

Make an X-Y plot with error bars.

**Usage**

```
error.bar(x, y, lo, hi, pch = 1, col = 1, barw = 0.1, add = FALSE, ...)
```

**Arguments**

x	X-values
y	Y-values
lo	Lower limit of error bars
hi	Upper limit of error bars
pch	plotting character

col	color
barw	width of the bar
add	logical, add=FALSE starts a new plot
...	other plotting parameters

**Value**

graphical side effects

**Author(s)**

Jonathan M. Lees<jonathan.lees@unc.edu>

**Examples**

```
x = 1:10
y = 2*x+5
zup = rnorm(10)
zup = zup-min(zup)+.5
zdown = rnorm(10)
zdown = zdown-min(zdown)+.2
#### example with same error on either side:
error.bar(x, y, y-zup, y+zup, pch = 1, col = 'brown' , barw = 0.1, add =
FALSE)
#### example with different error on either side:
error.bar(x, y, y-zdown, y+zup, pch = 1, col = 'brown' , barw = 0.1, add
= FALSE)
```

---

NextPow2

*Next power of 2*

---

**Description**

Find the next integer power of 2

**Usage**

```
NextPow2(x)
```

**Arguments**

x	integer
---	---------

**Value**

integer that is a power of 2 higher than given integer



**Author(s)**

Jonathan M. Lees<jonathan.lees@unc.edu>

**Examples**

```
NextPow2(600)
NextPow2(1023)
NextPow2(1025)
```

---

otolith

*otolith Proxy Data*

---

**Description**

Early Oligocene otolith from the US Gulf Coast.

**Usage**

```
data(otolith)
```

**Format**

A data frame with 63 observations on the following 2 variables.

distance a numeric vector

d180 a numeric vector

**Details**

Duplicate data have been removed.

**Source**

$\delta^{18}O$  record of an aragonite otolith from the early Oligocene Rosefield Clay in the US Gulf Coast (Ivany, 2000).

**References**

Wang, T., Surge, D., and Lees, J. M., (2015) ClamR: A Statistical Evaluation of Isotopic and Temperature Records in Sclerochronologic Studies. *Palaeogeography, Palaeoclimatology, Palaeoecology*, doi:10.1016/j.palaeo.2015.07.008.

**Examples**

```
data(otolith)

plot(otolith$distance, otolith$d180)
```

---

plotproxy.error      *Plot Output Jack-Knife*

---

**Description**

Plot output of proxyJK, the jackknife estimate of the time series analysis fitting curves.

**Usage**

```
plotproxy.error(x,y,gout, type=1, xlim=NULL, ylim=NULL, ylab="", xlab="", main="" )
```

```
plotproxy.all(gout, ylab1="", ylab2="",xlab="", main="",
xlim=NULL, ylim1=NULL, ylim2=NULL, legposition="topleft",
YAXstyle=0, pbox=TRUE,
legnames = c('Phs', 'Pos', 'Amp', 'Prd') )
```

```
plotproxy1(x, y, gout, xlim = NULL, ylim = NULL, ylab = "", xlab = "",
main = "")
```

```
plotproxy.error11(x, y, gout, type = 1, xlim = NULL,
ylim = NULL, ylab = "", xlab = "", main = "")
```

```
plotproxy.all2(gout, ylab1 = "", ylab2 = "", xlab = "", main = "",
xlim = NULL, ylim1 = NULL, ylim2 = NULL,
legposition = "topleft", YAXstyle = 0,
pbox = TRUE, legnames = c("Phs", "Pos", "Amp",
"Prd"))
```

**Arguments**

x	original x values from file
y	original y values from file
gout	output of proxyJK
type	type of error bar plotting: 1 = bars, 2=shaded
xlim	2-vector(limit on x-axis)
ylim	2-vector(limit on xy-axis)

ylim1	2-vector(limit on xy-axis)
ylim2	2-vector(limit on xy-axis)
ylab	character, y-axis label
ylab1	character, y-axis label
ylab2	character, y-axis label
xlab	character, x-axis label
main	character, title label
legposition	legend position
YAXstyle	Style for Y-axis
pbox	logical
legnames	names for legend

**Details**

Takes output directly from program

**Value**

graphical side effects

**Author(s)**

Jonathan M. Lees<jonathan.lees@unc.edu>

**References**

Wang, T., Surge, D., and Lees, J. M., (2015) ClamR: A Statistical Evaluation of Isotopic and Temperature Records in Sclerochronologic Studies. *Palaeogeography, Palaeoclimatology, Palaeoecology*, doi:10.1016/j.palaeo.2015.07.008.

**See Also**

proxyJK

**Examples**

```
## Not run:
## example to read in your data from csv file (for non ClamR data

## fn = "donna_viking_1.csv"
## C1 = scan(file=fn, what=list(mm="", o18=""), sep=",")
## x = as.numeric(C1$mm)
## y = as.numeric(C1$o18)
## x = x[!is.na(y)]
## y = y[!is.na(y)]
```

```
data(CLAM1)

x = CLAM1$x
y = CLAM1$y

dx = 3.392

gout = proxyJK(x, y, dx)

plotproxy1(x,y,gout)

## End(Not run)
```

---

proxyA

*Optimum Wilkinson Curve Fitting*

---

### **Description**

Runs one cycle of optimum Wilkinson curve fitting for a single sinusoid fitting.

### **Usage**

```
proxyA(ax, ay, xin)
```

### **Arguments**

ax	x-axis values
ay	y-axis values
xin	starting model: c( Phs,Pos,Amp,Prd)

### **Details**

This program implements the Wilkinson and ivany approach to climate time series modeling. This is used in the more sophisticated proxyJK code.

### **Value**

Optimum model, vector of 4 values

### **Note**

Uses stats package routine optim for optimization

### **Author(s)**

Jonathan M. Lees<jonathan.lees@unc.edu>

## References

Wang, T., Surge, D., and Lees, J. M., (2015) ClamR: A Statistical Evaluation of Isotopic and Temperature Records in Sclerochronologic Studies. *Palaeogeography, Palaeoclimatology, Palaeoecology*, doi:10.1016/j.palaeo.2015.07.008.

Wilkinson, B. H. and Ivany, L. C., Paleoclimatic inference from stable isotope profiles of accretionary biogenic hardparts; a quantitative approach to the evaluation of incomplete data, *Palaeogeography, Palaeoclimatology, Palaeoecology*, vol. 185, no. 1-2, pp.95-114, 01 Sep 2002

## See Also

proxyJK

## Examples

```
data(CLAM1)
x = CLAM1$x
y = CLAM1$y
```

```
dx = 3.392
```

```
Aout = proxyA(x, y, dx)
```

---

proxyJK

*Jackknife Wilkinson Curve Fitting*

---

## Description

Perform a jackknife estimate of proxy curve fitting for time series analysis.

## Usage

```
proxyJK(x, y, dx)
```

## Arguments

x	x-axis values
y	y-axis values
dx	width of window to span in time

## Details

Routine that improves on the Wilkinson and Ivany(2002) approach to climate time series modeling. The jackknife is used to estimate the 95 percent confidence bounds for the modeled estimates. dx should be chosen to be approximately half a cycle or more.

**Value**

List:

OUT	list( par, mid, ax, predmid,JKest, JKvar, PSTILDE )
omids	output midpoints
pmids	values at output midpoints
x	input x
y	input y
predy	predicted y from spline fit

**Note**

See proxyA for a duplication of the Wilkinson codes.

**Author(s)**

Jonathan M. Lees<jonathan.lees@unc.edu>

**References**

Wang, T., Surge, D., and Lees, J. M., (2015) ClamR: A Statistical Evaluation of Isotopic and Temperature Records in Sclerochronologic Studies. *Palaeogeography, Palaeoclimatology, Palaeoecology*, doi:10.1016/j.palaeo.2015.07.008.

**See Also**

proxyA

**Examples**

```
## Not run:

##### this is for reading in data
##### fn = "/home/lees/DONNA/donna_viking_1.csv"

## fn = "donna_viking_1.csv"
##### C1 = scan(file=fn, what=list(mm="", o18=""), sep=",")
##### x = as.numeric(C1$mm)
##### y = as.numeric(C1$o18)
##### x = x[!is.na(y)]
##### y = y[!is.na(y)]

data(CLAM1)
x = CLAM1$x
y = CLAM1$y
```

```
dx = 3.392

gout = proxyJK(x, y, dx)

plotproxy1(x, y, gout)

par(mfrow=c(2,1))

plotproxy.error(x, y, gout, type = 1)
plotproxy.error(x, y, gout, type = 2)

par(mfrow=c(2,1))
plotproxy.error(x, y, gout, type = 2)

plotproxy.all2(gout, YAXstyle=1 )

## End(Not run)
```

---

RESCALE

*Rescale a vector to fit in a certain range*

---

### Description

Rescale a vector to fit in a certain range

### Usage

```
RESCALE(x, nx1, nx2, minx, maxx)
```

### Arguments

x	vector
nx1	new minimum
nx2	new maximum
minx	old min
maxx	old max

### Details

Used for graphics.

### Value

scale vector is returned

**Author(s)**

Jonathan M. Lees<jonathan.lees.edu>

**Examples**

```
x = rnorm(10)
RESCALE(x, 3, 9, min(x), max(x) )
```

---

rwp\_limpet

*Reconstructed Temperature Record*

---

**Description**

Application to estimated temperatures from archaeological RWP shell 103a-39-1 with 3 years of temperature data.

**Usage**

```
data("rwp_limpet")
```

**Format**

A data frame with 74 observations on the following 8 variables.

distance\_all a numeric vector  
temp\_all a numeric vector  
distance1 a numeric vector  
temp1 a numeric vector  
distance2 a numeric vector  
temp2 a numeric vector  
distance3 a numeric vector  
temp3 a numeric vector

**Source**

RWP shell 103a-39-1

**References**

Wang, T., Surge, D., and Lees, J. M., (2015) ClamR: A Statistical Evaluation of Isotopic and Temperature Records in Sclerochronologic Studies. *Palaeogeography, Palaeoclimatology, Palaeoecology*, doi:10.1016/j.palaeo.2015.07.008.

**Examples**

```
data(rwp_limpet)
plot(rwp_limpet$distance_all, rwp_limpet$temp_all, type='b')
```



---

SinMod *Sine Model of climate time series*

---

**Description**

Sine Model of climate time series

**Usage**

```
SinMod(x, myEx, dC2)
```

**Arguments**

x	input model x consisting of 4 values, phase, position, amplitude, period
myEx	externally defined X-values
dC2	externally defined observations at X

**Details**

This is the function used in optimization of sinusoidal fits to climate data.

**Value**

squared sum of difference between observed and predicted

**Note**

Uses stats package for optimization

**Author(s)**

Jonathan M. Lees<jonathan.lees@unc.edu>

**References**

Wang, T., Surge, D., and Lees, J. M., (2015) ClamR: A Statistical Evaluation of Isotopic and Temperature Records in Sclerochronologic Studies. *Palaeogeography, Palaeoclimatology, Palaeoecology*, doi:10.1016/j.palaeo.2015.07.008.

**See Also**

optim, proxyA, proxyJK

**Examples**

```

data(CLAM1)
x = CLAM1$x
y = CLAM1$y
dx = 3.392

A1 = proxyA(x, y, dx)

RMSout = SinMod(A1$par, x, y)

```

---

windowsize

*Find Window Size*


---

**Description**

Estimate optimal window size for seasonal time series analysis.

**Usage**

```
windowsize(x, y, winmin, winmax, winstep)
```

**Arguments**

x	original x values from file
y	original y values from file
winmin	Minimum window size
winmax	Maximum window size
winstep	step size

**Value**

win	optimum window length
error	error for win

**Author(s)**

Jonathan M. Lees<[jonathan.lees@unc.edu](mailto:jonathan.lees@unc.edu)>

**References**

Wang, T., Surge, D., and Lees, J. M., (2015) ClamR: A Statistical Evaluation of Isotopic and Temperature Records in Sclerochronologic Studies. *Palaeogeography, Palaeoclimatology, Palaeoecology*, doi:10.1016/j.palaeo.2015.07.008.

**Examples**

```
## Not run:  
data(CLAM1)  
shellx=CLAM1$x[38:70]  
shelly=CLAM1$y[38:70]  
  
window_shell=window_size(shellx,shelly,1.8,9.4,0.2)  
  
## End(Not run)
```

# Index

## \* datasets

- CLAM1, 3
- climate, 4
- elliott\_yr1, 5
- elliott\_yr2, 6
- otolith, 9
- rwp\_limpet, 16

## \* misc

- error.bar, 7
- NextPow2, 8
- plotproxy.error, 10
- proxyA, 12
- proxyJK, 13
- RESCALE, 15
- SinMod, 17
- window\_size, 18

## \* package

- ClamR-package, 2

CLAM1, 3

ClamR (ClamR-package), 2

ClamR-package, 2

climate, 4

elliott\_yr1, 5

elliott\_yr2, 6

error.bar, 7

NextPow2, 8

otolith, 9

plotproxy.all (plotproxy.error), 10

plotproxy.all2 (plotproxy.error), 10

plotproxy.error, 10

plotproxy.error11 (plotproxy.error), 10

plotproxy1 (plotproxy.error), 10

proxyA, 12

proxyJK, 13

RESCALE, 15

rwp\_limpet, 16

SinMod, 17

window\_size, 18