Application of ClamR

Jonathan M. Lees and Ting Wang
University of North Carolina, Chapel Hill
Department of Geological Sciences
CB #3315, Mitchell Hall
Chapel Hill, NC 27599-3315
email: jonathan.lees@unc.edu
ph: (919) 962-0695

July 30, 2015

```
library(RPMG)
library(ClamR)
nomoredev<-function()
    {
        dev.off()
    }
```

1 Load CLAM1

Load the CLAM1 data at the attachment.

CLAM1.RData: one-year data interval (4.68-9.31 mm) of δ18O record of an archaeological limpet Patella vulgata (specimen QG2-1064-1) from Orkney, Scotland (Surge and Barrett, 2012).

```
###  load("./DATA/CLAM1.RData", verbose=TRUE)
data(CLAM1)
```

Plot the data

```
shell_x=CLAM1$x[38:70]
shell_y=CLAM1$y[38:70]
o181 = jpng(file='./FIGS/o181.png', width=12, height=12, bg='white')
plot(shell_x,shell_y,xlab="Distance from Margin (mm)",
     ylab=expression(delta*"18O(% VPDB)"))
nomoredev()
```

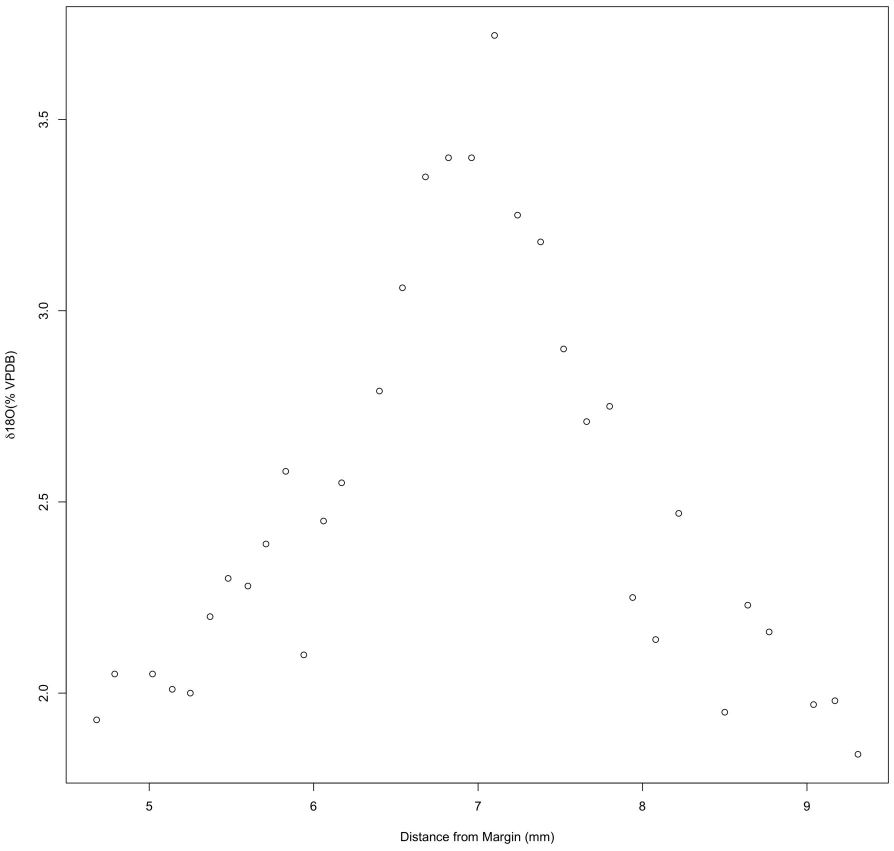

Figure 1: ClamR Example ./FIGS/o181.png

```
null device
          1
```

2 Conduct the case studies as the ordero in the manuscript.

2.1 Application of ClamR to oxygen isotope record (5.1)

The x range for the section CLAM1[38:70]is 4.68mm to 9.31mm.

```
#The x range for the section CLAM1[38:70]is 4.68mm to 9.31mm
testx1=seq(4.68, 9.31,length=33)
#add Gaussion noise
#change the parameter of simulated sine and make them closer the shell range,
###   the amplitude of sine equals half of the shell_y range, and the period
##   equals the shell_x range, and the position equals to the mean of all shell_y
###   value and the phase equals to 1/4 of the shell_x range

testy1=0.94*sin((testx1-1.16)*2*pi/4.63)+2.5+rnorm(33,mean=0,sd=0.25)
# Find the window with the minimum of least square error and plot it
#      window size should be between 1/2 and 1 wavelength, here the dx is 2*window size,
###   so dx should be (9.31-4.68)=4.63 mm to 2*4.63=9.26mm, same for the following two.
#   the average sampling density is 4.63/33 =0.140303, so we approximate it to 0.2
#   the window requires at least 4 data or samples to run, therefore,
###    the minimum dx should be 8*density approximately 1.6
window_testx1=windowsize(testx1,testy1,1.6,9.4,0.2)
#   The best window is at aroung 4.8 (close to the theoretical 4.63),
### so we selected the maximum of allowrance 4.5 mm based on the outplot plot,
##  and make all the plots for this test all together
#   Mark the best window size with a vertical line

gout_testx1 = proxyJK(testx1, testy1, 4.8)
```

Plot the panel of results:

```
FIGpanel1 = jpng(file='./FIGS/FIGpanel1.png', width=7, height=8, bg='white')
par(mfrow=c(3,2))
plot(testx1,testy1,xlim=c(4,10),ylim = c(1.5,4),
     xlab="Distance from Margin (mm)",
     ylab=expression(delta*"18O(ppm VPDB)"))
lines(testx1,0.94*sin((testx1-1.16)*2*pi/4.63)+2.5, lty="dotted")
#abline(x)
plot((window_testx1$win)/2,window_testx1$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=4.8/2, col='black')
#arrows(4.5, 0, x1 = 4.8, y1 = 0.14)
plotproxy1(testx1, testy1, gout_testx1, xlim=c(4,10),
          ylim = c(1.5,4), xlab="Distance from Margin (mm)",
          ylab=expression(delta*"18O(ppm VPDB)"), main="")
plotproxy.all(gout_testx1, YAXstyle=1, xlim=c(4,10),
             ylim1=c(0,4), ylim2=c(-15,5))
plotproxy.error(testx1, testy1, gout_testx1, type = 1,
               xlim=c(4,10),ylim = c(1.5,4),  xlab="Distance from Margin (mm)",
                ylab=expression(delta*"18O(ppm VPDB)"))
 lines(testx1,0.94*sin((testx1-1.16)*2*pi/4.63)+2.5, lty="dotted")
 plotproxy.error(testx1, testy1, gout_testx1, type = 2, xlim=c(4,10),
                ylim = c(1.5,4),  xlab="Distance from Margin (mm)",
                ylab=expression(delta*"18O(ppm VPDB)"))
 lines(testx1,0.94*sin((testx1-1.16)*2*pi/4.63)+2.5, lty="dotted")
 nomoredev()
null device
          1
```

Change the amplitude of gaussian noise into 0.1 permil VPDB noise standard deviation and repeat.

```
 #change the amplitude of gaussian noise into 0.1 permil VPDB noise standard deviation and r
 testx11=seq(4.68, 9.31,length=33)
 testy11=0.94*sin((testx11-1.16)*2*pi/4.63)+2.5+rnorm(33,mean=0,sd=0.1)
 window_testx11=windowsize(testx11,testy11,1.6,9.4,0.2)
 #The best window is at aroung 4.8 (close to the theoretical 4.63), so we selected the maxin
 #Mark the best window size with a vertical line

 gout_testx11 = proxyJK(testx11, testy11, 4.8)



 TEMP1 = jpng(file='./FIGS/TEMP1.png',width=7, height=8, bg='white')
 par(mfrow=c(3,2))
 plot(testx11,testy11,xlim=c(4,10),ylim = c(1.5,4),
      xlab="Distance from Margin (mm)",
      ylab=expression(delta*"18O(ppm VPDB)"))
 lines(testx11,0.94*sin((testx11-1.16)*2*pi/4.63)+2.5, lty="dotted")
 plot((window_testx11$win)/2,window_testx11$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=4.8/2, col='black')
 plotproxy1(testx11, testy11, gout_testx11,
           xlim=c(4,10),ylim = c(1.5,4), xlab="Distance from Margin (mm)",
           ylab=expression(delta*"18O(ppm VPDB)"), main="")
 plotproxy.all(gout_testx11, YAXstyle=1, xlim=c(4,10),
              ylim1=c(0,4), ylim2=c(-15,5))
 plotproxy.error(testx11, testy11, gout_testx11, type = 1,
                xlim=c(4,10),ylim = c(1.5,4),
                xlab="Distance from Margin (mm)",
                ylab=expression(delta*"18O(ppm VPDB)"))
 lines(testx11,0.94*sin((testx11-1.16)*2*pi/4.63)+2.5, lty="dotted")
 plotproxy.error(testx11, testy11, gout_testx11,
                type = 2, xlim=c(4,10),ylim = c(1.5,4),
                xlab="Distance from Margin (mm)",
```

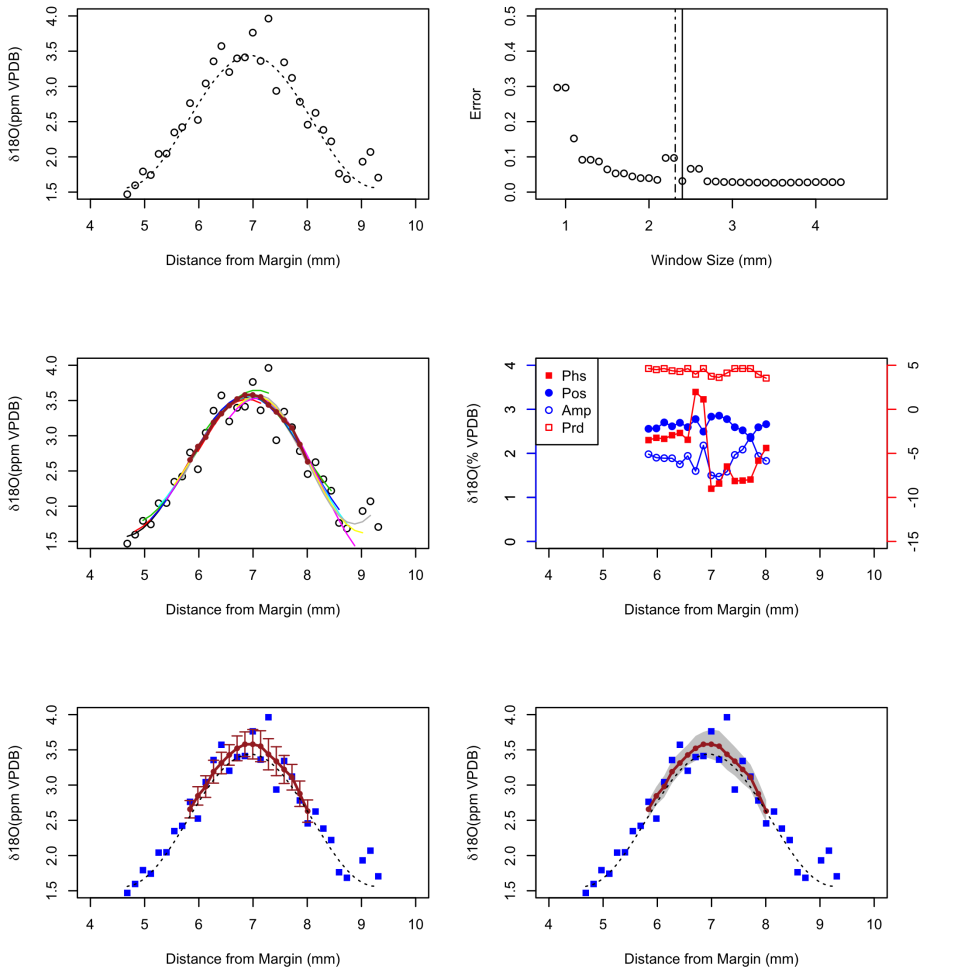

Figure 2: ClamR Example ./FIGS/FIGpanel1.png

```
             ylab=expression(delta*"18O(ppm VPDB)"))
lines(testx11,0.94*sin((testx11-1.16)*2*pi/4.63)+2.5, lty="dotted")
nomoredev()
```

```
null device
          1
```

Test the data (testx2, testy2) that has unequal distance interval.

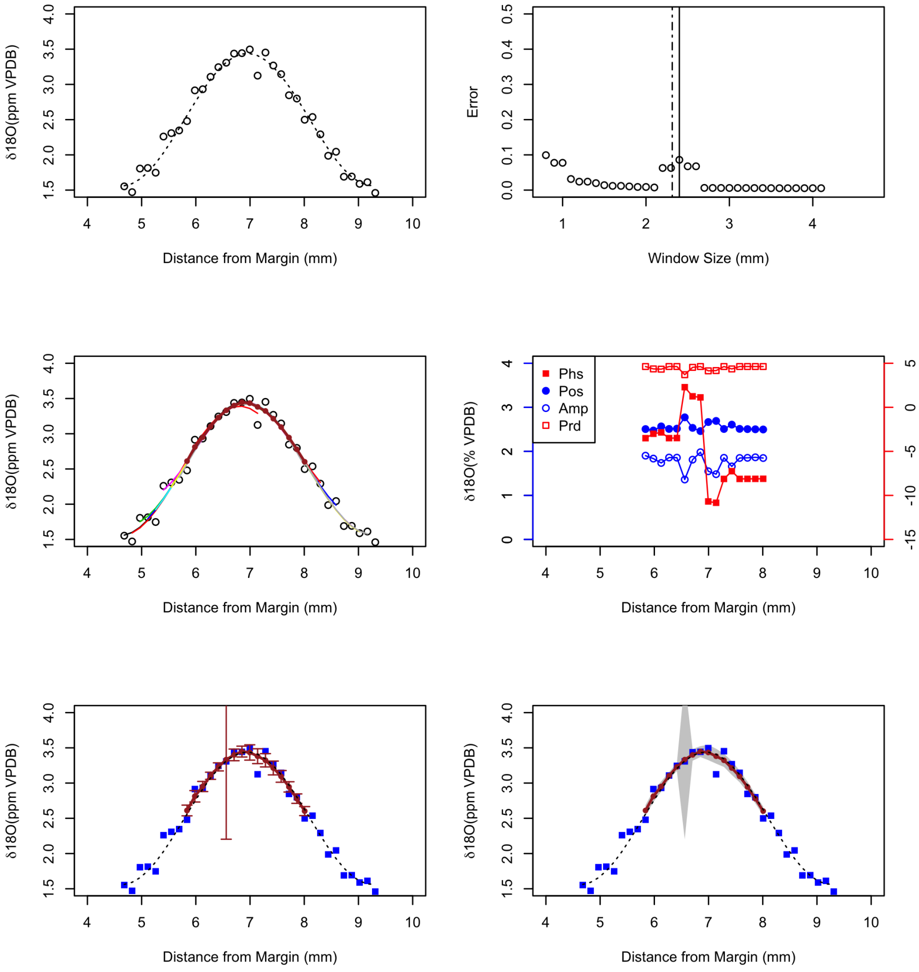

Figure 3: ClamR Example ./FIGS/TEMP1.png

```
###Test the data (testx2, testy2) that has unequal distance interval
testx2=shell_x
#testy2=sin(2*pi/3*testx2)+3+rnorm(33,mean=0,sd=0.25)
testy2=0.94*sin((testx2-1.16)*2*pi/4.63)+2.5+rnorm(33,mean=0,sd=0.25)
#Find the window with the minimum of least square error and plot it
window_testx2=windowsize(testx2,testy2,1.8,9.4,0.2)
#The best window is 4.5mm, and make all the plots for this test all together
gout_testx2 = proxyJK(testx2, testy2, 5)
```

```
TEMP2 =jpng(file='./FIGS/TEMP2.png',width=7, height=8, bg='white')
par(mfrow=c(3,2))
plot(testx2,testy2,xlab="Distance from Margin (mm)",
     ylab=expression(delta*"18O(ppm VPDB)"), xlim=c(4,10),ylim = c(1.5,4))
lines(testx2,0.94*sin((testx2-1.16)*2*pi/4.63)+2.5, lty="dotted")
plot((window_testx2$win)/2,window_testx2$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(testx2, testy2, gout_testx2,
          xlim=c(4,10),  ylim = c(1.5,4),
          xlab="Distance from Margin (mm)",
          ylab=expression(delta*"18O(ppm VPDB)"), main="")
plotproxy.all(gout_testx2, YAXstyle=1,
             xlim=c(4,10),ylim1=c(0,4), ylim2=c(-15,5))
plotproxy.error(testx2, testy2, gout_testx2, type = 1, xlim=c(4,10),
               ylim = c(1.5,4), xlab="Distance from Margin (mm)",
               ylab=expression(delta*"18O(ppm VPDB)"))
lines(testx2,0.94*sin((testx2-1.16)*2*pi/4.63)+2.5, lty="dotted")
plotproxy.error(testx2, testy2, gout_testx2, type = 2, xlim=c(4,10),
               ylim = c(1.5,4), xlab="Distance from Margin (mm)",
               ylab=expression(delta*"18O(ppm VPDB)"))
lines(testx2,0.94*sin((testx2-1.16)*2*pi/4.63)+2.5, lty="dotted")
nomoredev()
```

```
null device
          1
```

Change the amplitude of gaussian noise into 0.1 permil VPDB noise standard deviation and repeat.

```
#change the amplitude of gaussian noise into 0.1 permil VPDB noise standard deviation and r
testx22=shell_x
testy22=0.94*sin((testx22-1.16)*2*pi/4.63)+2.5+rnorm(33,mean=0,sd=0.1)
#Find the window with the minimum of least square error and plot it
window_testx22=windowsize(testx22,testy22,1.8,9.4,0.2)
#The best window is 4.5mm, and make all the plots for this test all together
gout_testx22 = proxyJK(testx22, testy22, 5)
```

```
TEMP3 = jpng(file='./FIGS/TEMP3.png',width=7, height=8, bg='white')
par(mfrow=c(3,2))
plot(testx22,testy22,xlab="Distance from Margin (mm)",
```

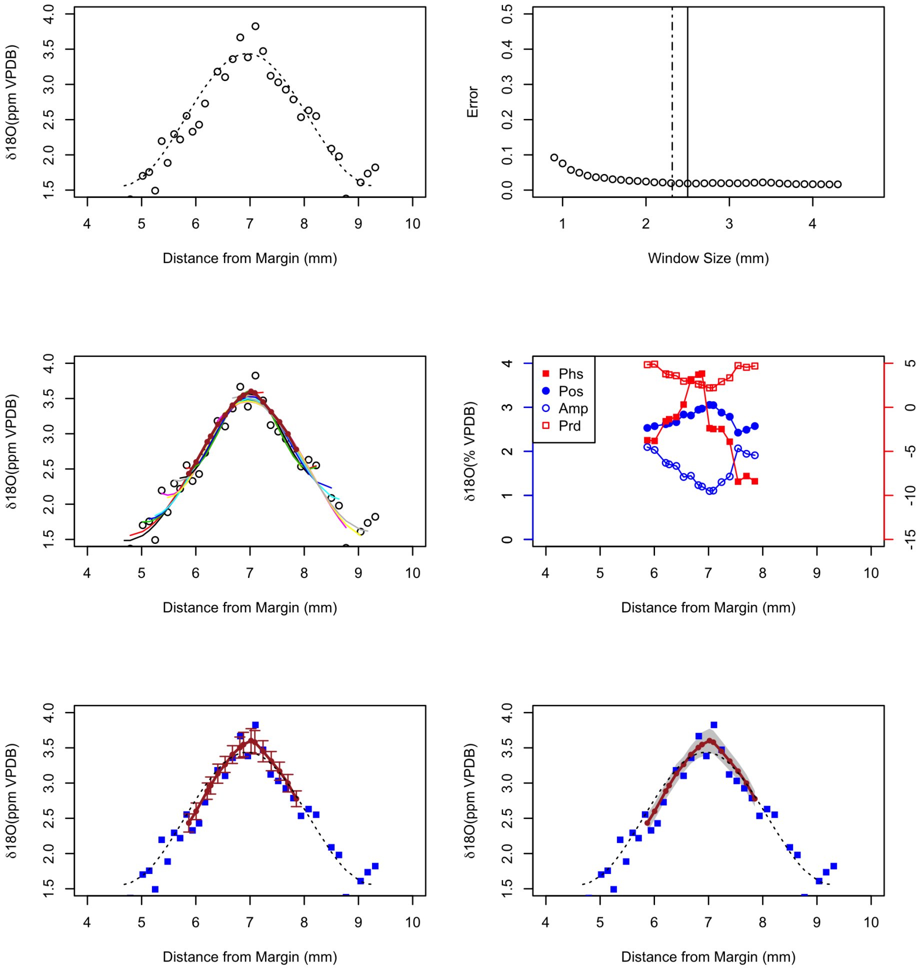

Figure 4: ClamR Example ./FIGS/TEMP2.png

```
     ylab=expression(delta*"18O(ppm VPDB)"), xlim=c(4,10),
     ylim = c(1.5,4))
lines(testx22,0.94*sin((testx22-1.16)*2*pi/4.63)+2.5, lty="dotted")
plot((window_testx22$win)/2,window_testx22$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(testx22, testy22, gout_testx22, xlim=c(4,10),  ylim = c(1.5,4),
          xlab="Distance from Margin (mm)",
          ylab=expression(delta*"18O(ppm VPDB)"), main="")
plotproxy.all(gout_testx22, YAXstyle=1,
             xlim=c(4,10),ylim1=c(0,4), ylim2=c(-15,5))
plotproxy.error(testx22, testy22, gout_testx22, type = 1, xlim=c(4,10),
               ylim = c(1.5,4), xlab="Distance from Margin (mm)",
               ylab=expression(delta*"18O(ppm VPDB)"))
lines(testx22,0.94*sin((testx22-1.16)*2*pi/4.63)+2.5, lty="dotted")
plotproxy.error(testx22, testy22, gout_testx22, type = 2,
               xlim=c(4,10),  ylim = c(1.5,4), xlab="Distance from Margin (mm)",
               ylab=expression(delta*"18O(ppm VPDB)"))
lines(testx22,0.94*sin((testx22-1.16)*2*pi/4.63)+2.5, lty="dotted")
nomoredev()
```

```
null device
          1
```

ClamR analysis of limpet oxygen isotope and test the shell data CLAM1[38:70].

```
####5.1.3 ClamR analysis of limpet oxygen isotope and test the shell data CLAM1[38:70]
window_shell=windowsize(shell_x,shell_y,1.8,9.4,0.2)
#the window size is 5mm, and make all the plots together
gout_shell = proxyJK(shell_x, shell_y, 5)



TEMP4 = jpng(file='./FIGS/TEMP4.png',width=7, height=8, bg='white')
par(mfrow=c(3,2))
plot(shell_x,shell_y,xlab="Distance from Margin (mm)",
     ylab=expression(delta*"18O(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(shell_x, shell_y, gout_shell, xlim=c(4,10), ylim = c(1.5,4),
          xlab="Distance from Margin (mm)",
          ylab=expression(delta*"18O(ppm VPDB)"), main="")
plotproxy.all(gout_shell,YAXstyle=1, xlim=c(4,10), ylim1=c(0,4),
             ylim2=c(-15,5))
plotproxy.error(shell_x, shell_y, gout_shell, type = 1, xlim=c(4,10),
               ylim = c(1.5,4), xlab="Distance from Margin (mm)",
               ylab=expression(delta*"18O(ppm VPDB)"))
plotproxy.error(shell_x, shell_y, gout_shell, type = 2, xlim=c(4,10),
               ylim = c(1.5,4), xlab="Distance from Margin (mm)",
```

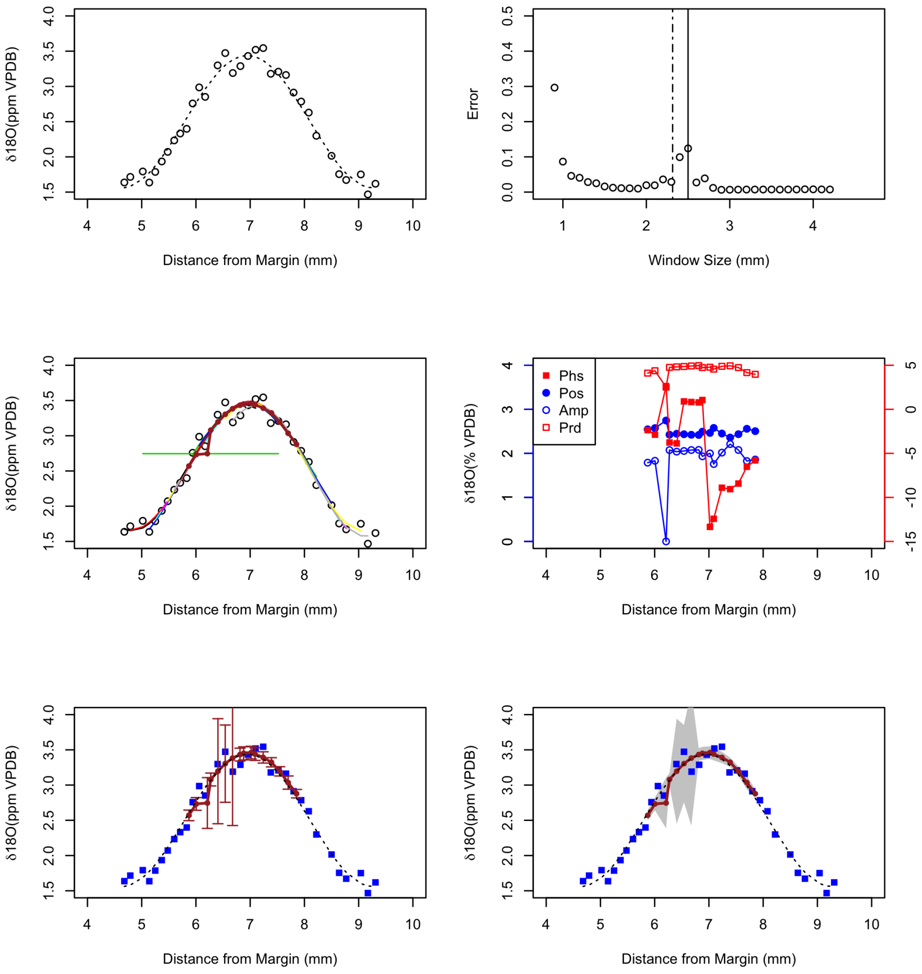

Figure 5: ClamR Example ./FIGS/TEMP3.png

```
                 ylab=expression(delta*"18O(ppm VPDB)"))
 nomoredev()

null device
          1
```

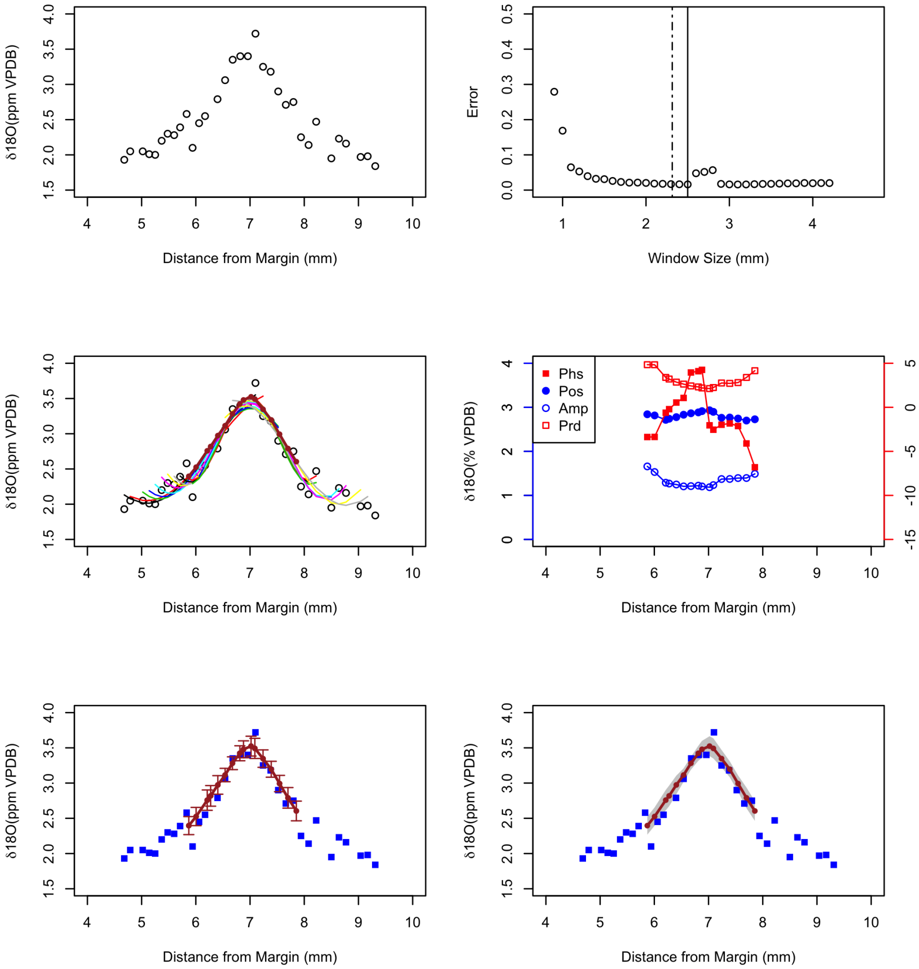

Figure 6: ClamR Example ./FIGS/TEMP4.png

3 Otolith

ClamR analysis of otolith oxygen isotope and application to Bruce and Linda Ivany otolith δ18O data.

otolith.csv: δ18O record of an aragonite otolith from the early Oligocene Rosefield Clay in the US Gulf Coast (Ivany, 2000).

```
####5.1.4 ClamR analysis of otolith oxygen isotope and application to
###   Bruce and Linda Ivany otolith $\delta 18\text{O}$ data.
```

```
 ##   otolith=read.csv(file="./DATA/otolith.csv",header=TRUE)

 data(otolith)
 otolith_d<-otolith$distance
 otolith_o<-otolith$d18O
 #half of the wavelength (minimum to minimum distance) is 0.69, the whole wavelength is 1.38
 #sampling density is 0.03920, round it to 0.1, so eight points is 0.8mm
 window_otolith=windowsize(otolith_d,otolith_o,0.8,2.8,0.1)
 #set window size to 1.3mm
 gout_otolith = proxyJK(otolith_d, otolith_o, 1.3)



 TEMP5=jpng(file='./FIGS/TEMP5.png',width=7, height=8, bg='white')
 par(mfrow=c(3,2))
 plot(otolith_d,otolith_o,xlab="Distance from Margin (mm)",
      ylab=expression(delta*"18O(ppm VPDB)"), xlim=c(0.3,2.9), ylim=c(-1.5,1))
 plot((window_otolith$win)/2,window_otolith$error,xlab="Window Size (mm)",
      ylab="Error", xlim=c(0.8/2,2.8/2), ylim=c(0,0.5))
 abline(v=1.38/2, lty="dotdash", col='black')
 abline(v=1.3/2, col='black')
 plotproxy1(otolith_d, otolith_o, gout_otolith, xlab="Distance from Margin (mm)",
           ylab=expression(delta*"18O(ppm VPDB)"),main="",
           xlim=c(0.3,2.9), ylim=c(-1.5,1))
 plotproxy.all(gout_otolith, YAXstyle=1, xlim=c(0.3,2.9),
              ylim1=c(-1,2), ylim2=c(-4,4))
 plotproxy.error(otolith_d, otolith_o, gout_otolith, type = 1,
                xlab="Distance from Margin (mm)",
                ylab=expression(delta*"18O(ppm VPDB)"),
                xlim=c(0.3,2.9), ylim=c(-1.5,1))
 plotproxy.error(otolith_d, otolith_o, gout_otolith, type = 2,
                xlab="Distance from Margin (mm)",
                ylab=expression(delta*"18O(ppm VPDB)"),
                xlim=c(0.3,2.9), ylim=c(-1.5,1))
 nomoredev()
null device
          1
```

4 Application of ClamR to reconstructed temperature record (5.2)

The data 'rwplimpet' are estimated temperatures based on the oxygen isotope time series from an archaeological limpet Patella vulgata (specimen103a-39-1)

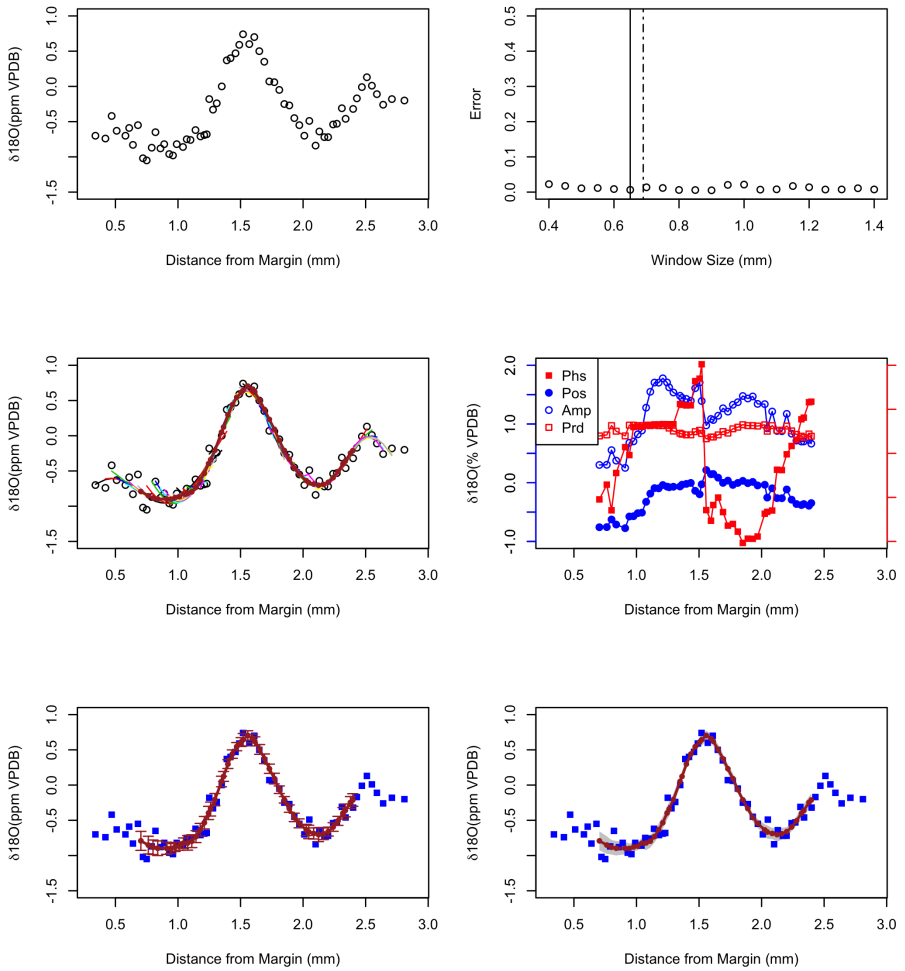

Figure 7: ClamR Example ./FIGS/TEMP5.png

from Croig Cave (Wang et al., 2012)

```
####5.2: Application of ClamR to reconstructed temperature record
##5.2.1. ClamR analysis of temperatures reconstructed from shell oxygen isotope
####Application to estimated temperatures from archaeological
###    RWP shell 103a-39-1 with 3 years of temperature data

##   rwp_limpet=read.csv(file="./DATA/rwp_limpet.csv",header=TRUE)
```

```
data(rwp_limpet)
#the distance of the first year from the left is named as distance1, so is the temperature
distance1 <- rwp_limpet$distance1[1:29]
temp1 <- rwp_limpet$temp1[1:29]
#0.24 to 2.08
#analysis of distance1 and temp1, 1.84 (half of the wavelength) to 3.68 (the wavelenth from
#density 1.84/(29-4-3-1)=0.0876, approximate to 0.1; initial dx is 8*0.1=0.8
window_temp1=windowsize(distance1,temp1,0.8,3.7,0.1)
gout_rwp1 = proxyJK(distance1, temp1, 1.7)
```

```
TEMP6= jpng(file='./FIGS/TEMP6.png',width=7, height=8, bg='white')
par(mfrow=c(3,2))
plot(distance1,temp1,xlab="Distance from Margin (mm)",
     ylab=expression(paste("Temperature ("^"o","C)")),
     xlim=c(0,3), ylim=c(3,18))
plot(window_temp1$win,window_temp1$error,xlab="Window Size (mm)",
     ylab="Error", xlim=c(0.8,3.7),ylim=c(0,10))
abline(v=1.84, lty="dotdash", col='black')
abline(v=1.7, col='black')
plotproxy1(distance1, temp1, gout_rwp1, xlab="Distance from Margin (mm)",
          ylab=expression(paste("Temperature ("^"o","C)")),xlim=c(0,3),
          ylim=c(3,18), main="")
plotproxy.all(gout_rwp1, YAXstyle=1,xlim=c(0,3),
             ylab1=expression(paste("Temperature ("^"o","C)")),ylim1=c(0,15),
             ylim2=c(-10,10),legposition="bottomright")
plotproxy.error(distance1, temp1, gout_rwp1, type = 1, xlab="Distance from Margin (mm)",
               ylab=expression(paste("Temperature ("^"o","C)")),xlim=c(0,3), ylim=c(3,18))
plotproxy.error(distance1, temp1, gout_rwp1, type = 2, xlab="Distance from Margin (mm)",
               ylab=expression(paste("Temperature ("^"o","C)")),xlim=c(0,3), ylim=c(3,18))
nomoredev()
```

```
null device
          1
```

the distance of the second year from the left is named as distance2, so is the temperature:

```
#the distance of the second year from the left is named as distance2, so is the temperature
distance2<-rwp_limpet$distance2[1:30]
temp2<-rwp_limpet$temp2[1:30]
#2.08 to 3.85
#analysis of distance1 and temp1, 1.77 (half of the wavelength) to 3.54 (the wavelenth from
#density 1.77/(29-3-2-1)=0.077, approximate to 0.1; initial dx is 8*0.1=0.8
```

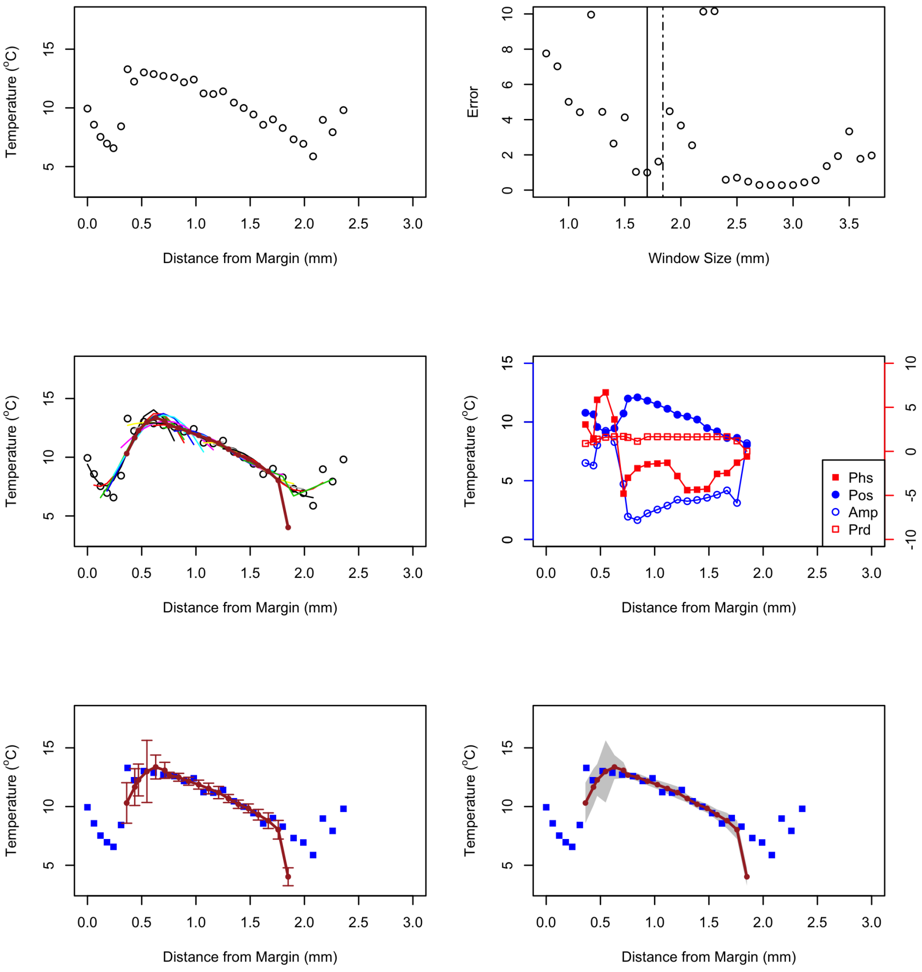

Figure 8: ClamR Example ./FIGS/TEMP6.png

```
window_temp2=windowsize(distance2,temp2,0.8,3.6,0.1)
gout_rwp2 = proxyJK(distance2, temp2, 1.9)
```

```
TEMP7=jpng(file='./FIGS/TEMP7.png',width=7, height=8, bg='white')
par(mfrow=c(3,2))
```

```
plot(distance2,temp2,xlab="Distance from Margin (mm)",
     ylab=expression(paste("Temperature ("^"o","C)")),xlim=c(1,4),
     ylim=c(3,18))
plot(window_temp2$win,window_temp2$error,xlab="Window Size (mm)",
     ylab="Error",xlim=c(0.8,3.7),ylim=c(0,10))
abline(v=1.77, lty="dotdash", col='black')
abline(v=1.9, col='black')
plotproxy1(distance2, temp2, gout_rwp2, xlab="Distance from Margin (mm)",
          ylab=expression(paste("Temperature ("^"o","C)")),
          xlim=c(1,4), ylim=c(3,18), main="")
plotproxy.all(gout_rwp2, YAXstyle=1,xlim=c(1,4),
             ylab1=expression(paste("Temperature ("^"o","C)")),
             ylim1=c(0,15), ylim2=c(-10,10),legposition="bottomright")
plotproxy.error(distance2, temp2, gout_rwp2, type = 1, xlab="Distance from Margin (mm)",
               ylab=expression(paste("Temperature ("^"o","C)")),xlim=c(1,4), ylim=c(3,18))
plotproxy.error(distance2, temp2, gout_rwp2, type = 2, xlab="Distance from Margin (mm)",
               ylab=expression(paste("Temperature ("^"o","C)")),xlim=c(1,4), ylim=c(3,18))
nomoredev()
```

```
null device
          1
```

the distance of the third year from the left is named as distance3, so is the temperature:

```
#the distance of the third year from the left is named as distance3, so is the temperature
distance3<-rwp_limpet$distance3[1:28]
temp3<-rwp_limpet$temp3[1:28]
#3.85 to 6.11
#analysis of distance1 and temp1, 2.26 (half of the wavelength) to 4.52 (the wavelenth from
#density 2.26/(28-3-2-1)=0.102, approximate to 0.1; initial dx is 8*0.1=0.8, but we are not
#analysis of distance3 and temp3, 1.13 to 2.26 for the wavelenth range
window_temp3=windowsize(distance3,temp3,1.4,4.5,0.1)
gout_rwp3 = proxyJK(distance3, temp3, 2.8)
```

```
TEMP8 = jpng(file='./FIGS/TEMP8.png',width=7, height=8, bg='white')
par(mfrow=c(3,2))
plot(distance3,temp3,xlab="Distance from Margin (mm)",
     ylab=expression(paste("Temperature ("^"o","C)")),
     xlim=c(3.6,6.6), ylim=c(3,18))
plot(window_temp3$win,window_temp3$error,xlab="Window Size (mm)",
     ylab="Error",,xlim=c(1.4,4.5),ylim=c(0,10))
abline(v=2.26, lty="dotdash", col='black')
abline(v=2.8, col='black')
```

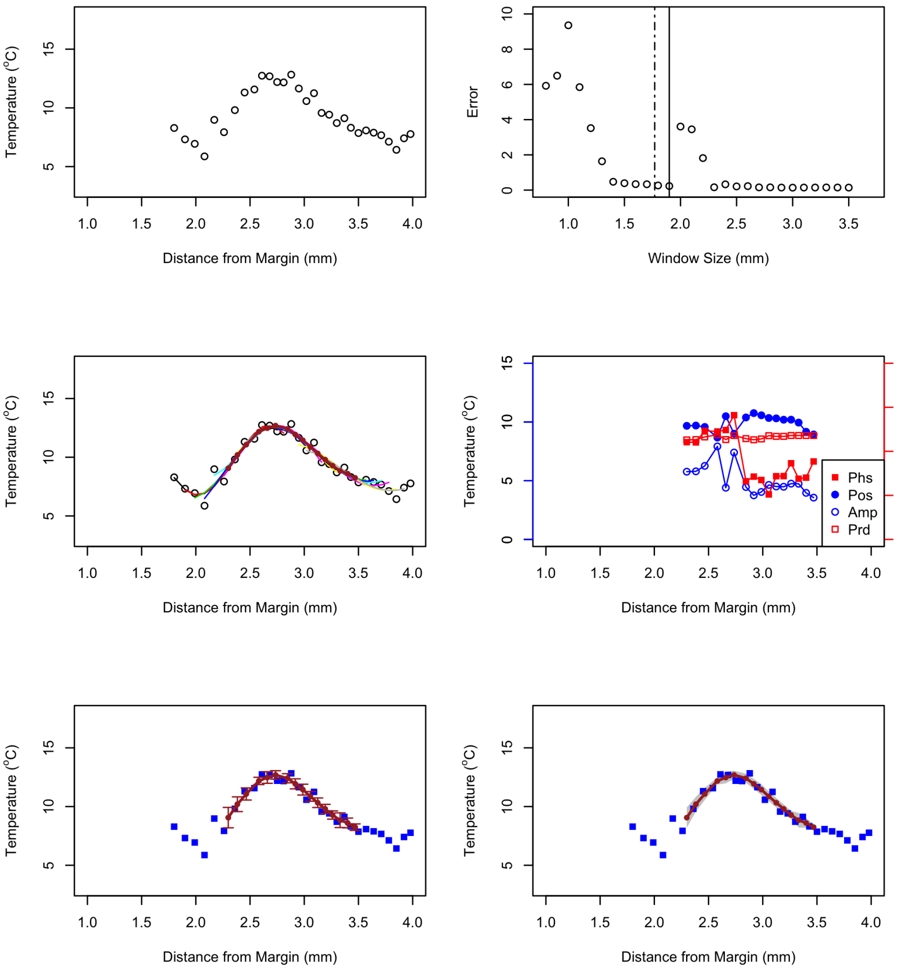

Figure 9: ClamR Example ./FIGS/TEMP7.png

```
plotproxy1(distance3, temp3, gout_rwp3, xlab="Distance from Margin (mm)",
          ylab=expression(paste("Temperature ("^"o","C)")),
          xlim=c(3.6,6.6), ylim=c(3,18), main="")
plotproxy.all(gout_rwp3, YAXstyle=1,xlim=c(3.6,6.6),
             ylab1=expression(paste("Temperature ("^"o","C)")),
             ylim1=c(0,15), ylim2=c(-10,10),legposition="bottomright")
plotproxy.error(distance3, temp3, gout_rwp3, type = 1,
               xlab="Distance from Margin (mm)",
               ylab=expression(paste("Temperature ("^"o","C)")),
               xlim=c(3.6,6.6), ylim=c(3,18))
plotproxy.error(distance3, temp3, gout_rwp3, type = 2,
               xlab="Distance from Margin (mm)",
               ylab=expression(paste("Temperature ("^"o","C)")),
               xlim=c(3.6,6.6), ylim=c(3,18))
nomoredev()
```

```
null device
          1
```

5 Climate record at Croig Cave

climatecc.csv: 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, http://www.cdd.noaa.gov, Smith and Reynolds, 2004)

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

## climate=read.csv(file="./DATA/climate_cc.csv",header=TRUE)

data(climate)
climate_month <- climate$overall
climate_temp <- climate$Temperature
#window size should be 12
gout_climate = proxyJK(climate_month, climate_temp, 12)



TEMP9 = jpng(file='./FIGS/TEMP9.png',width=7, height=8, bg='white')
par(mfrow=c(3,2))
plot(climate_month,climate_temp,xlab="month",
     ylab=expression(paste("Temperature ("^"o","C)")))
#
plotproxy.all(gout_climate, YAXstyle=1,
              ylab1=expression(paste("Temperature ("^"o","C)")),
              ylim1=c(0,15), legposition="bottomright")
plotproxy.error(climate_month, climate_temp, gout_climate, type = 1,
                xlab="month",ylab=expression(paste("Temperature ("^"o","C)")))
plotproxy.error(climate_month, climate_temp, gout_climate, type = 2,
                xlab="month",ylab=expression(paste("Temperature ("^"o","C)")))
nomoredev()
```

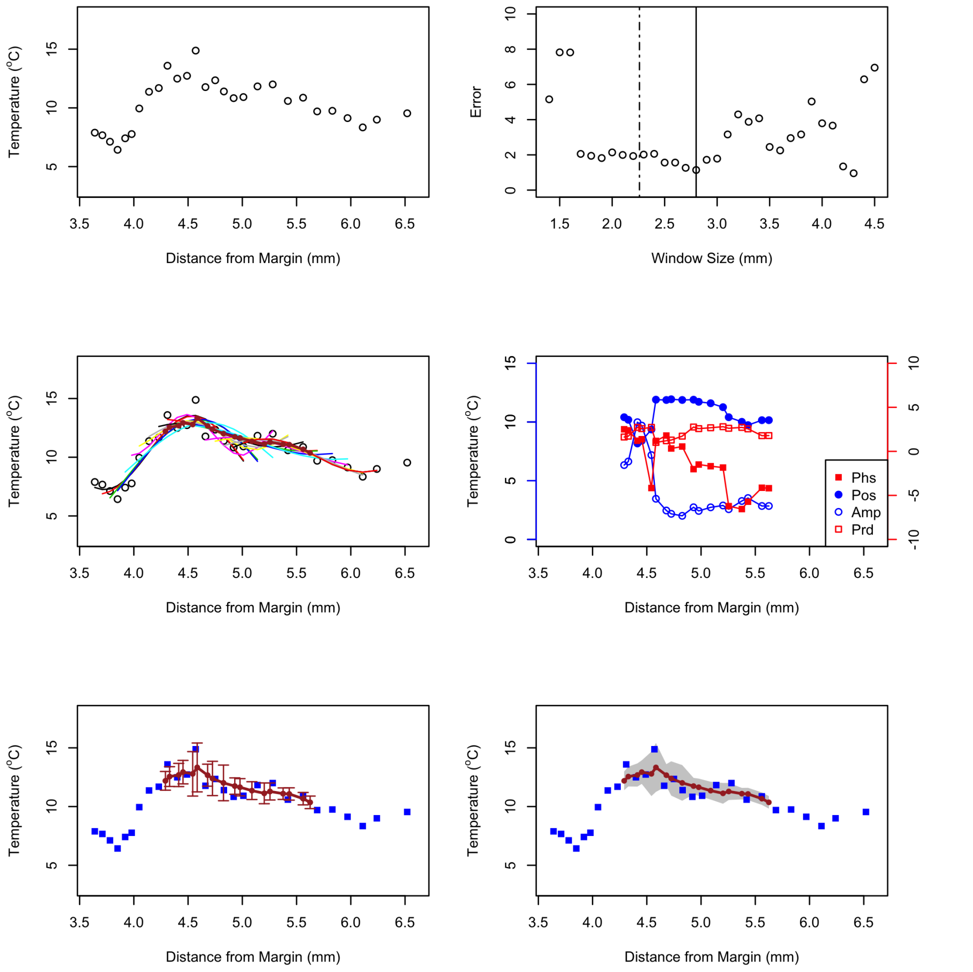

Figure 10: ClamR Example ./FIGS/TEMP8.png

```
null device
          1
```

```
TEMP10 = jpng(file='./FIGS/TEMP10.png',width=8, height=7, bg='white')
par(mfrow=c(2,1))
plotproxy.error11(climate_month, climate_temp, gout_climate, type=2,
```

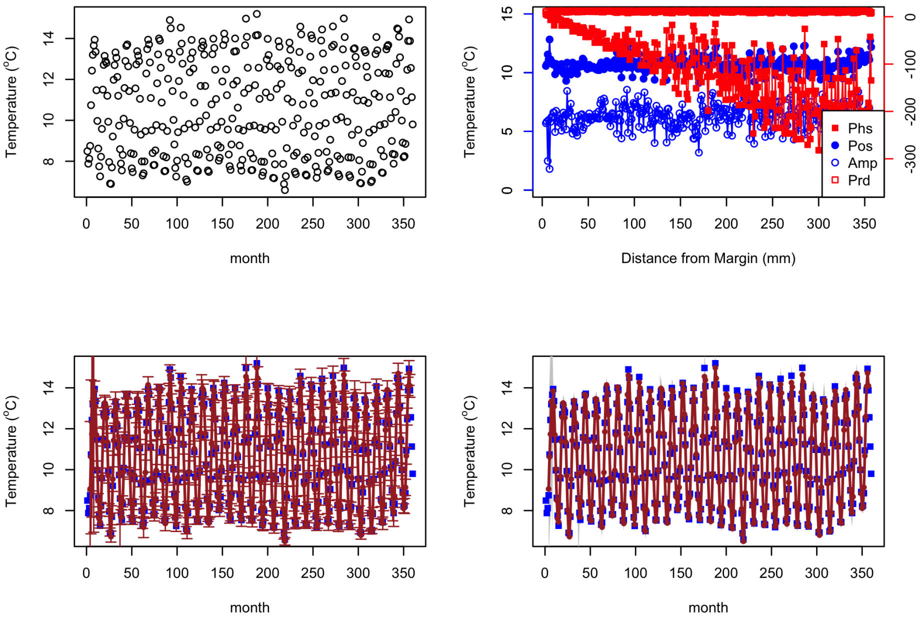

Figure 11: ClamR Example ./FIGS/TEMP9.png

```
                 xlab="month",ylab=expression(paste("Temperature ("^"o","C)")))
 plotproxy.all2(gout_climate, YAXstyle=1,
              ylab1=expression(paste("Temperature ("^"o","C)")),
              ylim1=c(0,15), legposition="bottomright")
 nomoredev()
```

```
null device
          1
```

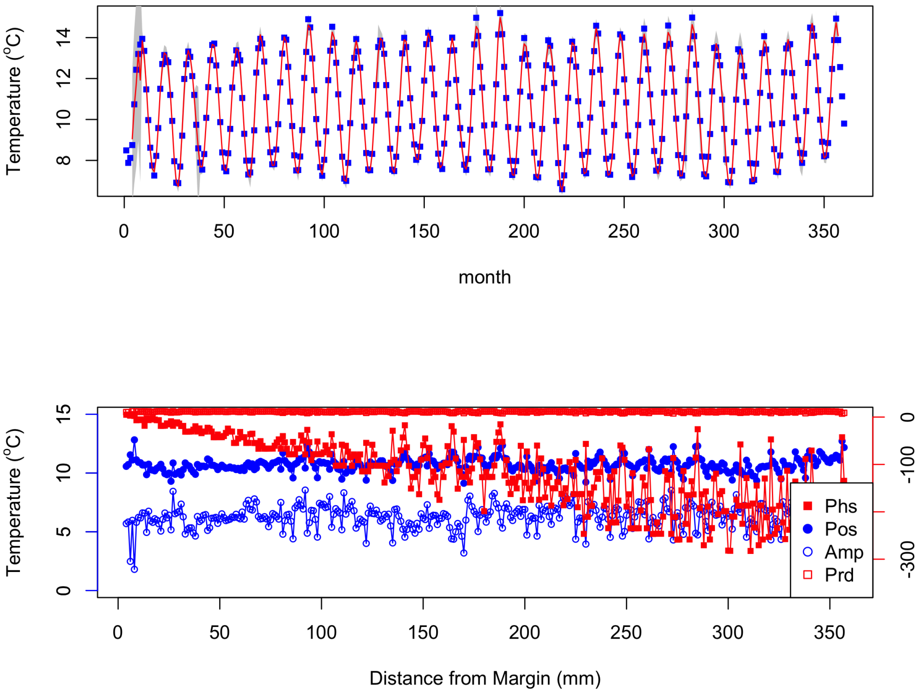

Figure 12: ClamR Example ./FIGS/TEMP10.png

6 Elliot Data Example

elliot.yr1.shell.csv and elliot.yr2.shell.csv: both data sets are selected from the δ18O record of a modern Mercenaria mercenaria shell collected live from Cedar Key in northern Florida and measured and analyzed by Elliot et al. (2003). elliot1.csv records one summer (including the most negative δ_{18}O). elliot2.csv records one winter (including the most positive δ_{18}O).

elliotyr1temp.csv and elliotyr2temp.csv: 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 δ18O are also derived from the local instrumental data. elliotyr1temp.csv is the predicted δ18O for the summer interval of elliot1.csv. elliotyr2temp.csv is the predicted δ18O for the winter interval of elliot2.csv.

```
## elliot_yr1_shell=read.csv(file="./DATA/elliot_yr1_shell.csv",header=TRUE)
##  elliot_yr1_temp=read.csv(file="./DATA/elliot_yr1_temp.csv",header=TRUE)
```

```
 data(elliot_yr1)
 date1<-elliot_yr1$date1
 d18o1<-elliot_yr1$d18o1
 date_temp1<-elliot_yr1$date_temp1
 d18o_pred1<-elliot_yr1$d18o_pred1
 elliot_yr1 = list(date1=date1, d18o1=d18o1, date_temp1=date_temp1, d18o_pred1=d18o_pred1)
```

```
 window_elliot1=windowsize(date1,d18o1,0.8,2.16,0.1)
 #set window size to 1.2mm
 gout_elliot1 = proxyJK(date1, d18o1, 1.2)
 TEMP11 =  jpng(file='TEMP11.png', width=7, height=8, bg='white' )
 par(mfrow=c(3,2))
 plot(date1,d18o1,xlab="Age(years)",
      ylab=expression(delta*"18O(ppm VPDB)"), xlim=c(1994.9,1996.2), ylim=c(-2.5,2))
 plot((window_elliot1$win)/2,window_elliot1$error,
      xlab="Window Size (years)", ylab="Error", xlim=c(0.8/2,2.16/2), ylim=c(0,0.5))
 abline(v=1.08/2, lty="dotdash", col='black')
 abline(v=1.2/2, col='black')
 plotproxy1(date1, d18o1, gout_elliot1, xlab="Age(years)",
            ylab=expression(delta*"18O(ppm VPDB)"),main="",
            xlim=c(1994.9,1996.2), ylim=c(-2.5,2))
 plotproxy.all(gout_elliot1, YAXstyle=1, xlim=c(1994.9,1996.2),
               ylim1=c(-2,4), ylim2=c(-5,5),xlab="Age(years)")
 plotproxy.error(date1, d18o1, gout_elliot1, type = 1,
                 xlab="Age(years)",ylab=expression(delta*"18O(ppm VPDB)"),
                 xlim=c(1994.9,1996.2), ylim=c(-2.5,2))
 lines(date_temp1,d18o_pred1, lty="dotted")
 plotproxy.error(date1, d18o1, gout_elliot1, type = 2, xlab="Age(years)",
                 ylab=expression(delta*"18O(ppm VPDB)"),xlim=c(1994.9,1996.2), ylim=c(-2.5,2
 lines(date_temp1,d18o_pred1, lty="dotted")
 nomoredev()
null device
          1
```

```
 ## elliot_yr2_shell=read.csv(file="./DATA/elliot_yr2_shell.csv",header=TRUE)
 ##  elliot_yr2_temp=read.csv(file="./DATA/elliot_yr2_temp.csv",header=TRUE)

 data(elliot_yr2)
 date2<-elliot_yr2$date2
 d18o2<-elliot_yr2$d18o2
```

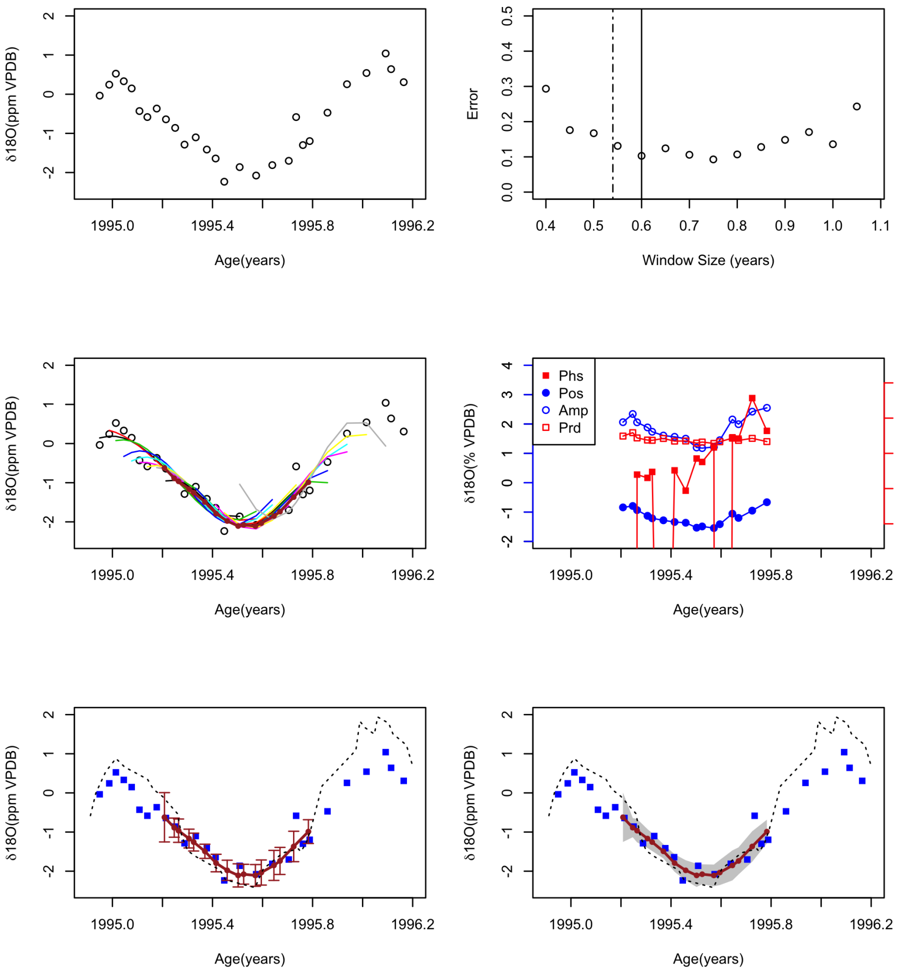

Figure 13: ClamR Example TEMP11.png

```
date_temp2<-elliot_yr2$date_temp2
d18o_pred2<-elliot_yr2$d18o_pred2
elliot_yr2 = list(date2=date2, d18o2=d18o2, date_temp2=date_temp2, d18o_pred2=d18o_pred2)
#half of the wavelength (minimum to minimum distance) is 1.19, the whole wavelength is 2.38
#sampling density is 1.19/(35-5-4-1)=0.0476, round it to 0.1, so eight points is 0.8mm
window_elliot2=windowsize(date2,d18o2,0.8,2.38,0.1)
#set window size to 1.2mm
gout_elliot2 = proxyJK(date2, d18o2, 1.2)
```

```
 TEMP12 =  jpng(file='TEMP12.png', width=7, height=8, bg='white' )
 par(mfrow=c(3,2))
 plot(date2,d18o2,xlab="date(year)",ylab=expression(delta*"18O(ppm VPDB)"), xlim=c(1995.2,19
 plot((window_elliot2$win)/2,window_elliot2$error,xlab="Window Size (mm)", ylab="Error", xli
 abline(v=1.19/2, lty="dotdash", col='black')
 abline(v=1.2/2, col='black')
 plotproxy1(date2, d18o2, gout_elliot2, xlab="date(year)",ylab=expression(delta*"18O(ppm VPD
 plotproxy.all(gout_elliot2, YAXstyle=1, xlim=c(1995.2,1996.85), ylim1=c(-2,4), ylim2=c(-5,5
 plotproxy.error(date2, d18o2, gout_elliot2, type = 1, xlab="date(year)",ylab=expression(del
 lines(date_temp2,d18o_pred2, lty="dotted")
 plotproxy.error(date2, d18o2, gout_elliot2, type = 2, xlab="date(year)",ylab=expression(del
 lines(date_temp2,d18o_pred2, lty="dotted")
 nomoredev()
```

```
null device
          1
```

```
 ##plot them together

 TEMP13 =  jpng(file='TEMP13.png', width=7, height=8, bg='white' )
 par(mfrow=c(2,2))
 plot(date1,d18o1,xlab="Age(years)",ylab=expression(delta*"18O(ppm VPDB)"), xlim=c(1994.9,19
 lines(date_temp1,d18o_pred1, lty="dotted")
 plotproxy.error(date1, d18o1, gout_elliot1, type = 2, xlab="Age(years)",ylab=expression(del
 lines(date_temp1,d18o_pred1, lty="dotted")
 plot(date2,d18o2,xlab="Age(years)",ylab=expression(delta*"18O(ppm VPDB)"), xlim=c(1995.2,19
 lines(date_temp2,d18o_pred2, lty="dotted")
 plotproxy.error(date2, d18o2, gout_elliot2, type = 2, xlab="Age(years)",ylab=expression(del
 lines(date_temp2,d18o_pred2, lty="dotted")
 nomoredev()
```

```
null device
          1
```

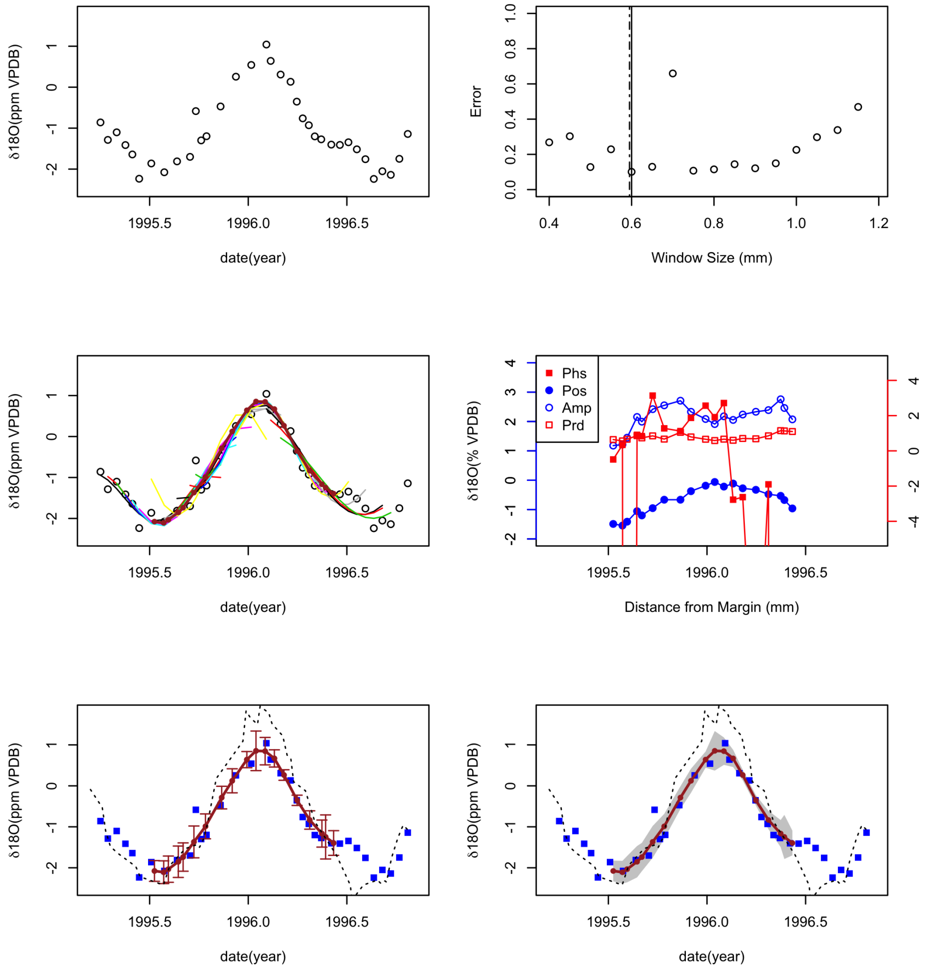

Figure 14: ClamR Example TEMP12.png

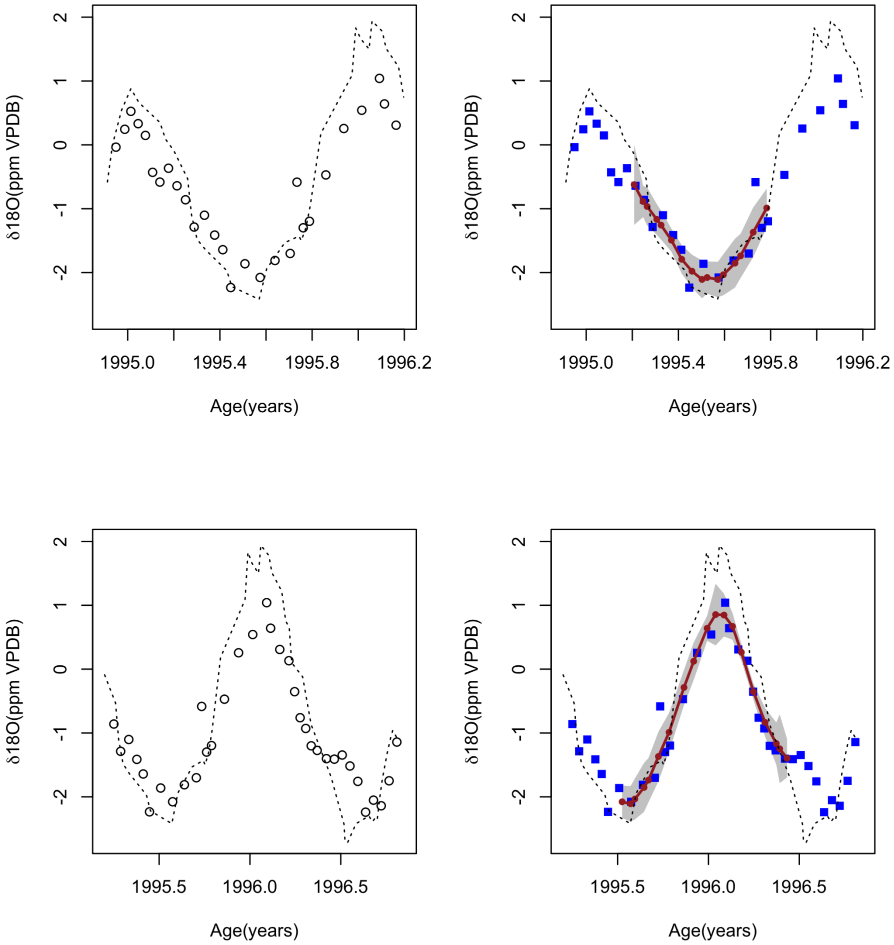

Figure 15: ClamR Example TEMP13.png