Supplementary Material C – Data and Software

Data Files

We considered 10 years of hourly data (January 2003 to December 2012) collected from 16 tide gauge stations around the coast of Australia. The hourly data have been obtained from the research quality database of *The University of Hawaii Sea Level Centre (UHSLC)* (Caldwell. 2015). The measurements of sea levels are with reference to the zero point of the respective tide benchmarks. Hourly missing values were replaced using growth rates of the series after which the series were de-seasonalised and then daily maxima and minima series (interval time series (ITS)) were constructed.

- The hourly data with estimated missing values are in the file **1sea_level_16.xlsx**.
- The daily interval time series constructed from **1sea_level_16.xlsx** are in the file, **2daily_its16.xlsx**.
- We also obtained the daily mean series for each station from **1sea_level_16.xlsx** and this data are in the file, **3daily_means16.xlsx**.

Caldwell, P. C., Merrifield, M. A., Thompson, P. R. (2015), Sea level measured by tide gauges from global oceans -- the Joint Archive for Sea Level holdings (NCEI Accession 0019568), Version 5.5, NOAA National Centers for Environmental Information, Dataset, doi:10.7289/V5V40S7W.

R and Matlab Scripts

1) R Script for Point–to-Point Distance Method

```
#The data is stored in the matrix MARES which come from the data file: 2daily_its16.xlsx
DISTH <- matrix(0,nrow=16,ncol=16)
DISTE <- matrix(0,nrow=16,ncol=16)
DISTM <- matrix(0,nrow=16,ncol=16)
DISTMT <- matrix(0,nrow=16,ncol=16)

# calculating Point-to-point distances
haus<-function(max1,min1,max2,min2)
{
 max(abs(max1-max2),abs(min1-min2))
}
eucli<-function(max1,min1,max2,min2)
{
 sqrt((max1-max2)^2+(min1-min2)^2)
}

mallows<-function(max1,min1,max2,min2)
{
sqrt(((min1+max1)/2-(min2+max2)/2)^2+1/3*((max1-min1)/2-(max2-min2)/2)^2)
}

mallowsT<-function(max1,min1,max2,min2)
{
 sqrt(((min1+max1)/2-(min2+max2)/2)^2+1/6*((max1-min1)/2-(max2-min2)/2)^2)
}
```

```
for(i1 in 1:15)
  for(i2 in (i1+1):16)
  {
    dh <- de <- dm <- dmT <- 0
    for(j in 1:nrow(MARES))
    {
      dh <- dh+haus(MARES[j,2*(i1-1)+1],MARES[j,2*i1],MARES[j,2*(i2-
1)+1],MARES[j,2*i2])
      de <- de+eucli(MARES[j,2*(i1-1)+1],MARES[j,2*i1],MARES[j,2*(i2-
1)+1],MARES[j,2*i2])
      dm <- dm+mallows(MARES[j,2*(i1-1)+1],MARES[j,2*i1],MARES[j,2*(i2-
1)+1],MARES[j,2*i2])
      dmT <- dmT+mallowsT(MARES[j,2*(i1-1)+1],MARES[j,2*i1],MARES[j,2*(i2-
1)+1],MARES[j,2*i2])
    }
    DISTH[i1,i2]<- DISTH[i2,i1]<-dh/nrow(MARES)
    DISTE[i1,i2]<- DISTE[i2,i1]<-de/nrow(MARES)
    DISTM[i1,i2]<- DISTM[i2,i1]<-dm/nrow(MARES)
    DISTMT[i1,i2]<- DISTMT[i2,i1]<-dmT/nrow(MARES)
  }

#hierarchical clustering
library(stats)
HIER_E <- hclust(as.dist(DISTE), method = "ward.D2", members = NULL)
plot(HIER_E)
HIER_H <- hclust(as.dist(DISTH), method = "ward.D2", members = NULL)
plot(HIER_H)
HIER_M <- hclust(as.dist(DISTM), method = "ward.D2", members = NULL)
plot(HIER_M)
HIER_MT <- hclust(as.dist(DISTMT), method = "ward.D2", members = NULL)
plot(HIER_MT)

#dynamical clustering
library(symbolicDA)
D=DClust(as.dist(DISTE), 3, iter=100)
D=DClust(as.dist(DISTH), 3, iter=100)
D=DClust(as.dist(DISTM), 3, iter=100)
D=DClust(as.dist(DISTMT), 3, iter=100)
```

2) **Matlab and R Scripts for Time Domain Features**

Matlab Script

```
%Determines the mean, variance, skewness, kurtosis and trend of each of the radius and
%centre series (ITS)
%The radius and centre series are determined for each ITS from the data file,
%2 daily_its16.xlsx and stored in the arrays rad16 and cen16
len=length(rad16);
rc=[rad16 cen16];
% Determination of trend
x1=1:len; x2=zeros(len,1);
x2=1+x2; x=[x1' x2];
reg=zeros(32,2);
resrc=zeros(len,32);
bse=zeros(17,1);
t=len;
for k=1:32
   y=rc(:,k);
   [bb,bbint,rr,rrint,sstats]=regress(y,x);
   reg(k,1)=bb(1);
   reg(k,2)=bb(2);
   bse(k)=(bbint(2,2)-bb(2))/tinv(0.975, t-2);
   resrc(:,k)=rr;
end
% Determination of moments
sk=skewness(rc)'; kur=kurtosis(rc)'; mres=mean(rc)'; vres=var(rc)';

rc_clus=[bse(1:16)  mres(1:16) vres(1:16) sk(1:16,:) kur(1:16,:) bse(17:32)  mres(17:32)
vres(17:32) sk(17:32,:) kur(17:32,:)];
```

R Script

```
% These features,  rc_clus are stored in the file, ts16_feat.xlsx and read into the following R-
Script
library(openxlsx)
rc <-read.xlsx("ts16_feat.xlsx")
src=scale(rc[,2:11]) #radius and centre
W=src
#W=src[,1:5] #radius
#W=src[,6:10] #centre

#clustering
library(stats)
d=dist(W, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)
x= hclust(d, method = "ward.D2")
plot(x)
x = hclust(d, method = "average")
plot(x)
x = hclust(d, method = "complete")
plot(x)
kmeans(W,2)
```

Matlab Script

```
%Determines the mean, variance, skewness, kurtosis and trend of each the daily mean series.
%These series are in the file 3daily_mean16.xlsx and are stored in seadm16.xlsx after the
%text had been removed.

W<-read.xlsx("seadm16.xlsx")
len=length(W);
x=1:len;
xx=[ones(len,1) x'];
reg=zeros(16,2);
resm=zeros(len,16);
bse=zeros(17,1);
t=len;

%%  Determination of trend
for k=1:16
   y=W(:,k);
   [bb,bbint,rr,rrint,sstats]=regress(y,xx);
   reg(k,1)=bb(1);
   reg(k,2)=bb(2);
   bse(k)=(bbint(2,2)-bb(2))/tinv(0.975, t-2);
   resm(:,k)=rr;
end
sk=skewness(W)';
kur=kurtosis(W)';
mres=mean(W)';
vres=var(W)';
m_clus=[bse(1:16)  mres(1:16) vres(1:16) sk(1:16,:) kur(1:16,:)];
```

R Script

```
#These features, m_clus are stored in the file tsmf16.xlsx and read into the following R-
Script:
library(openxlsx)
fmn <-read.xlsx("tsmf16.xlsx")
sfmn=scale(fmn[,2:6]) #daily means
W=sfmn

#clustering
library(stats)

d=dist(W, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)
x= hclust(d, method = "ward.D2")
plot(x)
x = hclust(d, method = "average")
plot(x)
x = hclust(d, method = "complete")
plot(x)
kmeans(W,2)
```

3) R Script for Wavelet Variance Method

```
# Determines the wavelet variances of the radius and centre series of each ITS.
#The radius and centre series are determined for each ITS from the file
#2daily_its16.xlsx and stored in the files rad16.xlsx and cen16.xlsx

library(openxlsx)
library(waveslim)
library(stats)

#read in radius and centre series
rad <-read.xlsx("rad16.xlsx")
cen <-read.xlsx("cen16.xlsx")
rv=matrix(nrow=16,ncol=8)
cv=matrix(nrow=16,ncol=8)

#calculate wavelet variances of radius and centre series and put together
for(k in 1:16)
{x=rad[,k]
wx=modwt(x, wf = "la8", n.levels = 8, boundary = "periodic")
wv=wave.variance(wx, type="gaussian", p=0.025)
wv=as.matrix(wv)
wvr=t(wv[1:8,1])
rv[k,]=wvr}

for(k in 1:16)
{x=cen[,k]
wx=modwt(x, wf = "la8", n.levels = 8, boundary = "periodic")
wv=wave.variance(wx, type="gaussian", p=0.025)
wv=as.matrix(wv)
wvr=t(wv[1:8,1])
cv[k,]=wvr}

rcv=cbind(rv,cv)
W=rcv
#clustering
d=dist(W, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)
x = hclust(d, method = "ward.D2")
plot(x)
x = hclust(d, method = "average")
plot(x)
x = hclust(d, method = "complete")
plot(x)
kmeans(W,2)
```

Determines the wavelet variances of each of the daily mean series
#read in mean daily series
These series are in the file **3daily_mean16.xlsx** which are stored in **seadm16.xlsx** after text had been removed.

```
library(openxlsx)
library(waveslim)
library(stats)

dailym <-read.xlsx("seadm16.xlsx")

#calculate wavelet variances of mean daily series
dmv=matrix(nrow=16,ncol=8)
for(k in 1:16)
{x=dailym[,k]
wx=modwt(x, wf = "la8", n.levels = 8, boundary = "periodic")
wv=wave.variance(wx, type="gaussian", p=0.025)
wv=as.matrix(wv)
wvr=t(wv[1:8,1])
dmv[k,]=wvr}
W=dmv


#clustering
d=dist(W, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)
x = hclust(d, method = "ward.D2")
plot(x)
x = hclust(d, method = "average")
plot(x)
x = hclust(d, method = "complete")
plot(x)
kmeans(W,2)
```

4) R Script for Space-Time Coefficients Method

```
#Estimation of STAR model parameters
#Example: p = 1
#The lower and upper interval bounds from 2daily_its16.xlsx were stored in files xl.txt
and xu.txt respectively

library(systemfit)
library(mAr)
N <- length(xu)
d1xu1 <- xu[-N]
xu1   <- xu[-1]
d1xl1 <- xl[-N]
xl1   <- xl[-1]
eq1 <- xu1 ~ d1xu1 + xl1 + d1xl1
eq2 <- xl1 ~ d1xl1 + xu1 + d1xu1
inst <- ~ d1xu1 + d1xl1
Mrest <- matrix(0,3,8)
Mrest[1,2] <- 1
Mrest[1,6] <- -1
Mrest[2,3] <- 1
Mrest[2,7] <- -1
Mrest[3,4] <- 1
Mrest[3,8] <- -1
vrest <- c(0,0,0)
model1 <- systemfit(list(eq1, eq2), "3SLS", inst=inst, restrict.matrix=Mrest,
            restrict.rhs=vrest)
summary(model1)
rcov1 <- model1$residCov
sc1 <- log(det(rcov1))+log(N)/N*5  #Schwarz's criterion
aic1 <- log(det(rcov1))+2/N*5  #AIC
coef1 <- coef(model1) #Parameter estimates

#The estimated coefficients (coef1) are standardised and stored in the file: stdst16.xlsx
library(openxlsx)
library(stats)

st <-read.xlsx("stdst16.xlsx")
#clustering
W=as.matrix(st)
d=dist(W, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)
x= hclust(d, method = "ward.D2")
plot(x)
x = hclust(d, method = "average")
plot(x)
x = hclust(d, method = "complete")
plot(x)
kmeans(W,2)
```

5) R Script for Correlation Matrix and Interval ACF methods

```
#The data with the lower and upper interval bounds from 2daily_its16.xlsx for each
station are stored in file minmax.txt
nstat <- 16 #number of stations
x <- ts(matrix(scan("minmax.txt"), ncol=2*nstat, byrow=T), deltat=1/365,
start=c(2003,1))
k <- 20 #maximum lag for ACF computation. It is set as 20 as an example

for (i in 1:nstat) {
   for (k1 in 1:(k+1)){
tmp <- paste("y",i,k1," <- acf(cbind(x[,(2*",i,"-1)],x[,(2*",i,")]), lag.max=",k,",
plot=F)$acf[",k1,",,]", sep="")
eval(parse(text=tmp))}
     }

#1.Squared differences between the corresponding elements in the ACF matrices for each
#k
for (i in 1:(nstat-1)) {
 for (j in (i+1):nstat) {
   for(k1 in 1:(k+1)){
 tmp <- paste("z",i,j,k1," <- (y",i,k1,"-y",j,k1,")^2", sep="")
 eval(parse(text=tmp))
   }}}

# (Square root of the) sum of the squared differences for all lags
sumsq1 <- matrix(numeric(nstat^2), ncol=nstat)
for (i in 1:(nstat-1)) {
 for (j in (i+1):nstat) {
    for(k1 in 1:(k+1)){
    tmp <- paste("sumsq1[",i,",",j,"] <- sumsq1[",i,",",j,"] + sum(z",i,j,k1,")",sep="")
    eval(parse(text=tmp))
   }}}
eucdist1 <- sumsq1^0.5

#2.Absolute differences between the corresponding elements in the ACF matrices for
each k
for (i in 1:(nstat-1)) {
 for (j in (i+1):nstat) {
   for(k1 in 1:(k+1)){
    tmp <- paste("z",i,j,k1," <- abs(y",i,k1,"-y",j,k1,")", sep="")
    eval(parse(text=tmp))
   }}}

#Sum of the absolute differences for all lags
sumsq2 <- matrix(numeric(nstat^2), ncol=nstat)
for (i in 1:(nstat-1)) {
 for (j in (i+1):nstat) {
   for(k1 in 1:(k+1)){
    tmp <- paste("sumsq2[",i,",",j,"] <- sumsq2[",i,",",j,"] + sum(z",i,j,k1,")",sep="")
```

```
    eval(parse(text=tmp))
  }}}
eucdist2 <- sumsq2

#3.Absolute differences between the corresponding elements in the ACF matrices for
#each k
#and max absolute row sum and max absolute column sum of the difference matrices
for (i in 1:(nstat-1)) {
 for (j in (i+1):nstat) {
  for(k1 in 1:(k+1)){
   tmp <- paste("z",i,j,k1," <- (abs(y",i,k1,"-y",j,k1,"))", sep="")
   eval(parse(text=tmp))
   tmp <- paste("maxrz",i,j,k1," <- max(sum(z",i,j,k1,"[1,]),sum(z",i,j,k1,"[2,]))",
sep="") #max row
   eval(parse(text=tmp))
   tmp <- paste("maxcz",i,j,k1," <- max(sum(z",i,j,k1,"[,1]),sum(z",i,j,k1,"[,2]))",
sep="") #max column
   eval(parse(text=tmp))
     }}}

#Matrix norm: max absolute row sum norm of the difference of the ACF matrices
arsumnorm <- matrix(numeric(nstat^2), ncol=nstat)
for (i in 1:(nstat-1)) {
 for (j in (i+1):nstat) {
  for(k1 in 1:(k+1)){
   tmp <- paste("arsumnorm[",i,",",j,"] <- arsumnorm[",i,",",j,"] + maxrz",i,j,k1,sep="")
   eval(parse(text=tmp))
  }}}

#Matrix norm: max absolute column sum norm of the difference of the ACF matrices
acsumnorm <- matrix(numeric(nstat^2), ncol=nstat)
for (i in 1:(nstat-1)) {
 for (j in (i+1):nstat) {
  for(k1 in 1:(k+1)){
   tmp <- paste("acsumnorm[",i,",",j,"] <- acsumnorm[",i,",",j,"] + maxcz",i,j,k1,sep="")
   eval(parse(text=tmp))
  }}}

#4.Distance measure based on interval ACF
intmeans <- numeric(nstat) #Vector with the station means
for (j in 1:nstat)  intmeans[j] <- mean((x[,(2*j-1)]+x[,(2*j)])/2)
intvars <- numeric(nstat) #Vector with the station variances
for (j in 1:nstat)  intvars[j] <- mean(x[,(2*j-1)]^2+x[,(2*j-1)]*x[,(2*j)]+x[,(2*j)]^2)/3-
intmeans[j]^2
acvm <- matrix(numeric(nstat*k), ncol=k) #Autocovariance matrix (one row for each
#station)
acfm <- matrix(numeric(nstat*k), ncol=k) #Autocorrelation matrix (one row for each
#station)
for (k1 in 1:k) {
w <- matrix(numeric(4*nstat*(nrow(x)-k1)), ncol=4*nstat)
```

```
for (j in 1:nstat) w[,(4*(j-1)+1):(4*j)] <- ts.intersect(x[,(2*j-1)],x[,(2*j)],lag(x[,(2*j-1)],
k1),lag(x[,(2*j)],k1))
intcov <- numeric(nstat)
intcor <- numeric(nstat)
for (j in 1:nstat) {intcov[j] <- sum(2*(w[,(4*(j-1)+1)]-intmeans[j])*(w[,(4*(j-1)+3)]-
intmeans[j])+
(w[,(4*(j-1)+1)]-intmeans[j])*(w[,(4*(j-1)+4)]-intmeans[j])+
(w[,(4*(j-1)+2)]-intmeans[j])*(w[,(4*(j-1)+3)]-intmeans[j])+
2*(w[,(4*(j-1)+2)]-intmeans[j])*(w[,(4*(j-1)+4)]-intmeans[j]))/(6*nrow(x))
#autocovariance
}
intcor <- intcov/intvars #autocorrelation
acvm[,k1] <- intcov
acfm[,k1] <- intcor
}
# (Square root of the) squared acf differences
distacf <- matrix(numeric(nstat^2), ncol=nstat)
#Distance matrix
for (i in 1:nstat) {
  for (j in 1:nstat) distacf[i,j] <- sqrt((acfm[i,]-acfm[j,])%*%(acfm[i,]-acfm[j,]))
  }
```

#Distance matrices for correlation and Interval ACF methods, **eucd1, eucd2, #arsumnorm , acsumnorm, distacf** are stored in the files, **ssq365.xlsx, ab365.xlsx, #maxrow365.xlsx, maxcol365.xlsx, acf365.xlsx,** respectively.

```
library(openxlsx)
#read in distances
distssq <- read.xlsx("ssq365.xlsx") #Frobenius
distab <- read.xlsx("ab365.xlsx")  #Absolute value sum
distmrow <- read.xlsx("maxrow365.xlsx") # Maximum absolute row sum
distmcol <- read.xlsx("maxcol365.xlsx") #maximu absolute column sum
distacf <- read.xlsx("acf365.xlsx")  #interval ACF


#clustering
HIER_acf <- hclust(as.dist(distacf), method = "ward.D2", members = NULL)
plot(HIER_acf)
HIER_ssq <- hclust(as.dist(distssq), method = "ward.D2", members = NULL)
plot(HIER_ssq)
HIER_ab <- hclust(as.dist(distab), method = "ward.D2", members = NULL)
plot(HIER_ab)
HIER_mr <- hclust(as.dist(distmrow), method = "ward.D2", members = NULL)
plot(HIER_mr)
HIER_mc <- hclust(as.dist(distmcol), method = "ward.D2", members = NULL)
plot(HIER_mc)
```

6) R Script for Observation-based and ACF methods for Daily Mean Series

```
#These series are in the file 3daily_mean16.xlsx and are stored in the file
#seadm16.xlsx after the text had been removed.
library(openxlsx)
dailym <-read.xlsx("seadm16.xlsx")

#clustering using observations of daily means series
W=dailym
#W=as.matrix(dailym)
WW=scale(t(W))
d=dist(WW, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)
x = hclust(d, method = "ward.D2")
plot(x)
x = hclust(d, method = "average")
plot(x)
x = hclust(d, method = "complete")
plot(x)
kmeans(W,2)

#clustering using ACF of daily mean series
nlag=365;
dmacf=matrix(nrow = nlag, ncol=16)
for (i in 1:16)
  {da = acf(W[,i],nlag, plot=FALSE)
  dmacf[,i]=as.matrix(da$acf[2:(nlag+1)])}

WW=scale(t(dmacf))
d=dist(WW, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)
x = hclust(d, method = "ward.D2")
plot(x)
x = hclust(d, method = "average")
plot(x)
x = hclust(d, method = "complete")
 plot(x)
kmeans(W,2)
```