Reproducibility: Directional Spectra-based Clustering for Visualizing Patterns of Ocean Waves and Winds

Carolina Euan

5/2/2018

In this document we reproduce the simulation results. First we load the source functions and libraries. (Non meaningful code is omited on pdf)

```
library(HMClust)
here()
```

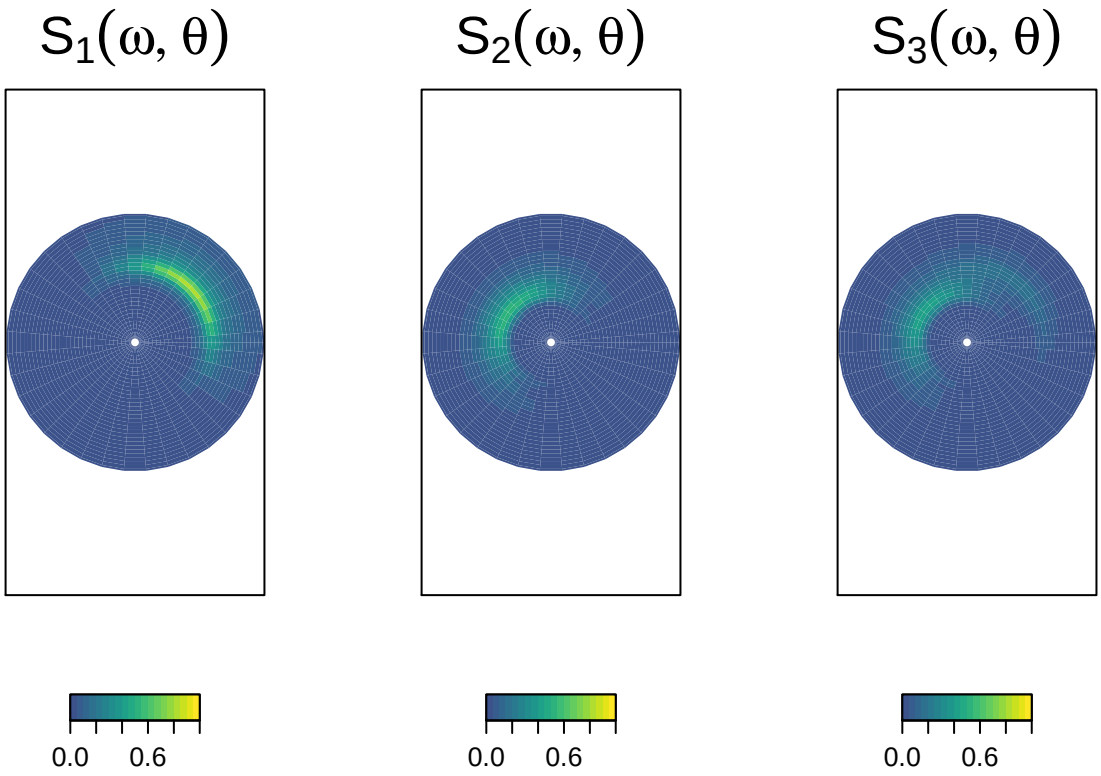
```
## [1] "/Users/euancacd/Dropbox/Postdoc/Research/Waves/PaPer/JCGS/CodeSubmit 2"
```

```
source("Source.R")
load("MCStudy.RData")
```

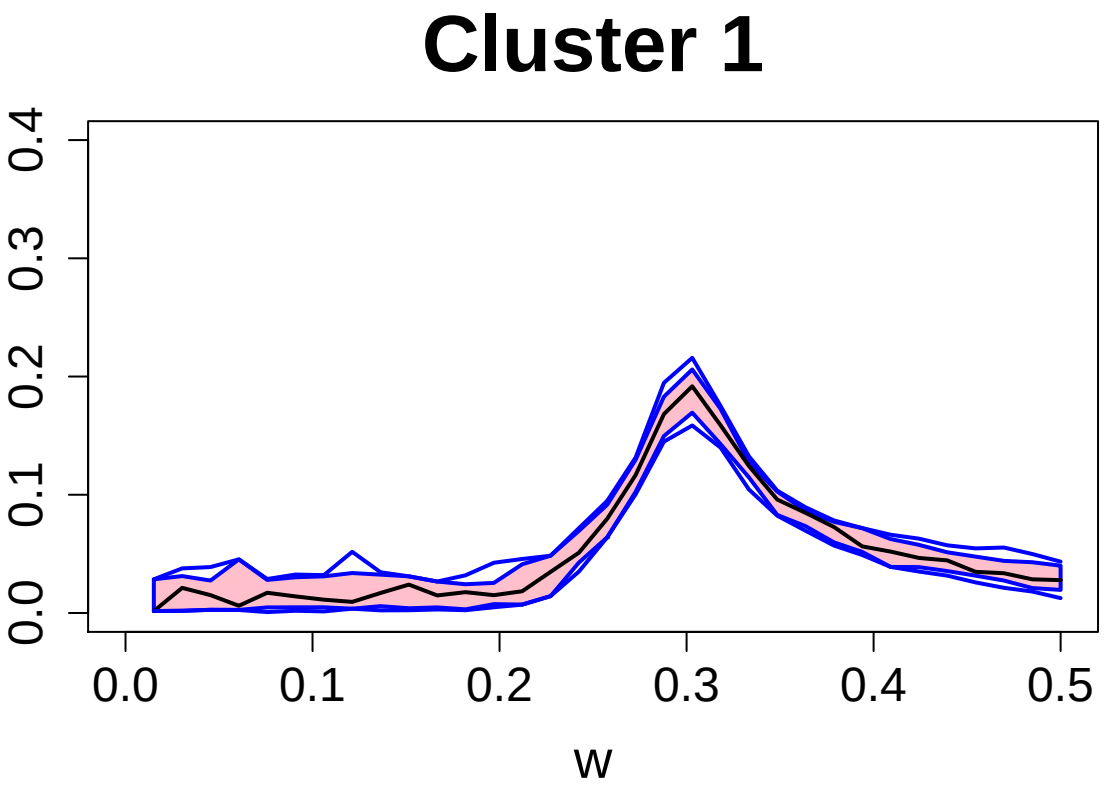
Simulated Example

Clustering visualization based on Example 2.

```
#Example 2
Sd.Mat<-Aux.Cov.Mat(w,theta,lambda = c(3,10))
Sd.Mat<-sqrt(.05)*Sd.Mat
set.seed(4485)
m<-50
Sim.S1<-Sim.DirSpec(w,theta,hs=1,tp=1/(.3*2*pi),theta0=pi/4,smax=5,n=m,gam = 2,sd.mat = Sd.Mat)
Sim.S2<-Sim.DirSpec(w,theta,hs=1,tp=1/(.2*2*pi),theta0=4*pi/5,smax=3,n=m,gam = 1,sd.mat = )
Sim.S3<-Sim.DirSpeck(w,theta,list(Sim.S1$S.w.theta,Sim.S2$S.w.theta),n=m,sd.mat=Sd.Mat)
norms<-c(normalize(x=w,y=theta,normfx = TRUE,D=2,fx =Sim.S1$S.w.theta),
         normalize(x=w,y=theta,normfx = TRUE,D=2,fx =Sim.S2$S.w.theta),
         normalize(x=w,y=theta,normfx = TRUE,D=2,fx =Sim.S3$S.w.theta))
par(mfrow=c(1,3))
image2D(x=Xmat,y=Ymat,z= Sim.S1$S.w.theta/norms[1],col=viridis_pal(begin = .25)(20),asp=1,
        colkey = list(side = 1, length = 0.5,cex.axis=1.3) ,theta = 0, cex.main=2.5,font.main=2,
        axes=FALSE,main=expression(S[1](omega,theta)),xlab=" ", ylab=" ",zlim=c(0,1))
image2D(x=Xmat,y=Ymat,z= Sim.S2$S.w.theta/norms[2],cex.main=2.5,font.main=2,
        col=viridis_pal(begin =.25)(20),asp=1,
        colkey = list(side = 1, length = 0.5,cex.axis=1.3) ,theta = 0,
        axes=FALSE,main=expression(S[2](omega,theta)),xlab=" ", ylab=" ",zlim=c(0,1))
image2D(x=Xmat,y=Ymat,z= Sim.S3$S.w.theta/norms[3],cex.main=2.5,font.main=2,
        colkey = list(side = 1, length = 0.5,cex.axis=1.3) ,theta = 0,
        col=viridis_pal(begin = .25)(20),asp=1,
        axes=FALSE,main=expression(S[3](omega,theta)),xlab=" ", ylab=" ",zlim=c(0,1))
```

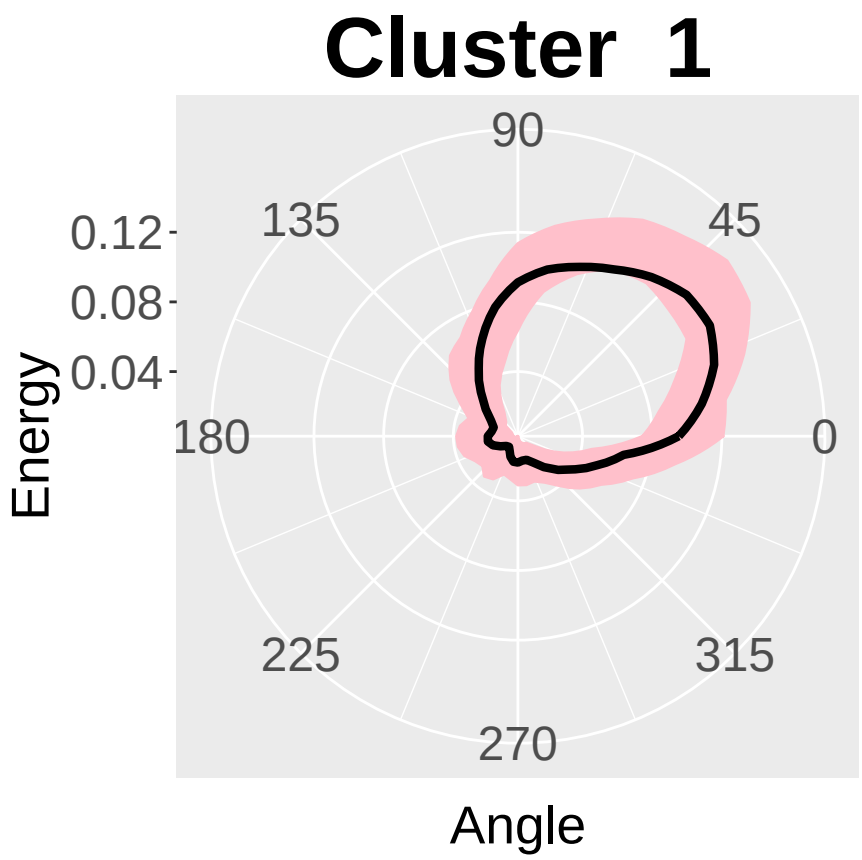



```
#Clustering visualization
fbplot(w/(2*pi),fit=f1,col="pink",xlim = c(0,.5),
       ylim=c(0,0.4),xlab="w",ylab="",cex.axis=1.5,cex.lab=1.7,
       main="Cluster 1",cex.main=2.5)
```

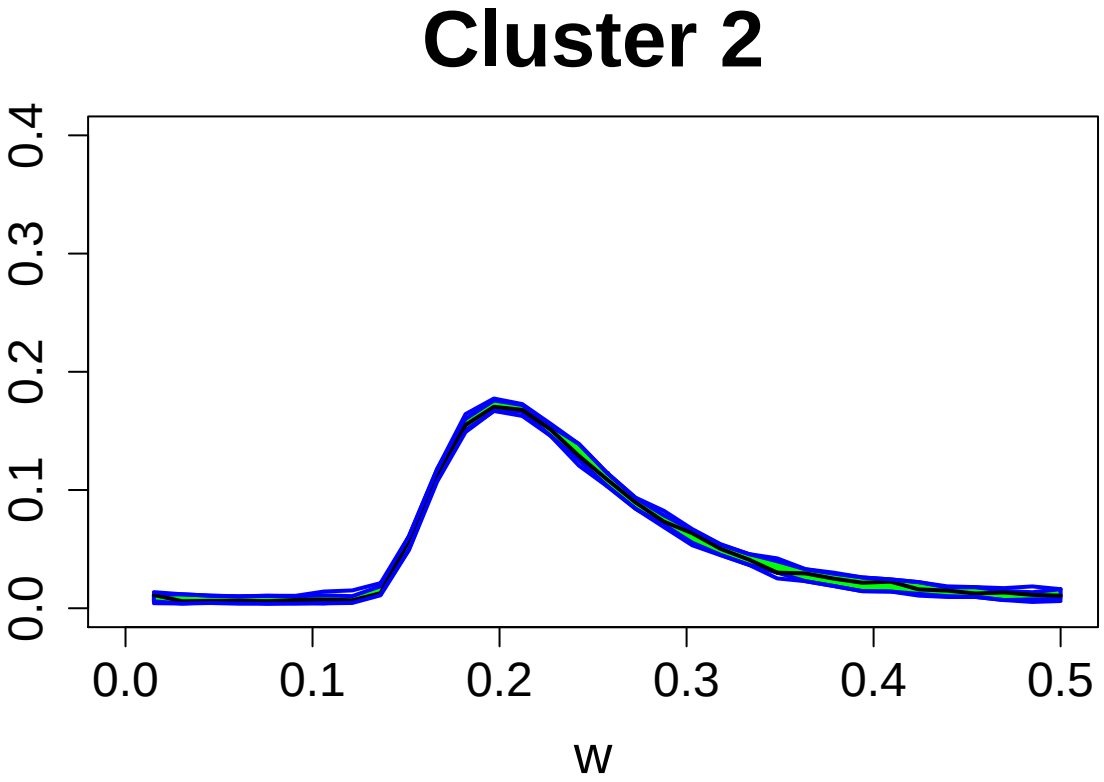


```
## $depth
##  [1] 0.4341620 0.4093754 0.3186395 0.4038837 0.4274335 0.3181447 0.3769697
##  [8] 0.2945455 0.4194682 0.3592084 0.2874706 0.4191218 0.2874706 0.2615955
## [15] 0.2494743 0.3411503 0.4024985 0.4246135 0.3591095 0.3237848 0.2678293
## [22] 0.4020037 0.3470377 0.3922078 0.3617316 0.4234261 0.3931973 0.2707483
## [29] 0.3421892 0.4158071 0.3641558 0.2981571 0.4155597 0.3685591 0.3805318
## [36] 0.4103649 0.2986024 0.3494125 0.3745455 0.4007174 0.3401608 0.4224861
## [43] 0.2966234 0.3403587 0.3932468 0.3389734 0.3937910 0.3150773 0.4156092
## [50] 0.3487693
##
## $outpoint
## integer(0)
##
## $medcurve
## [1] 1
```

```
WrapBP(theta,g1,colorP = "pink",index = 1)
```

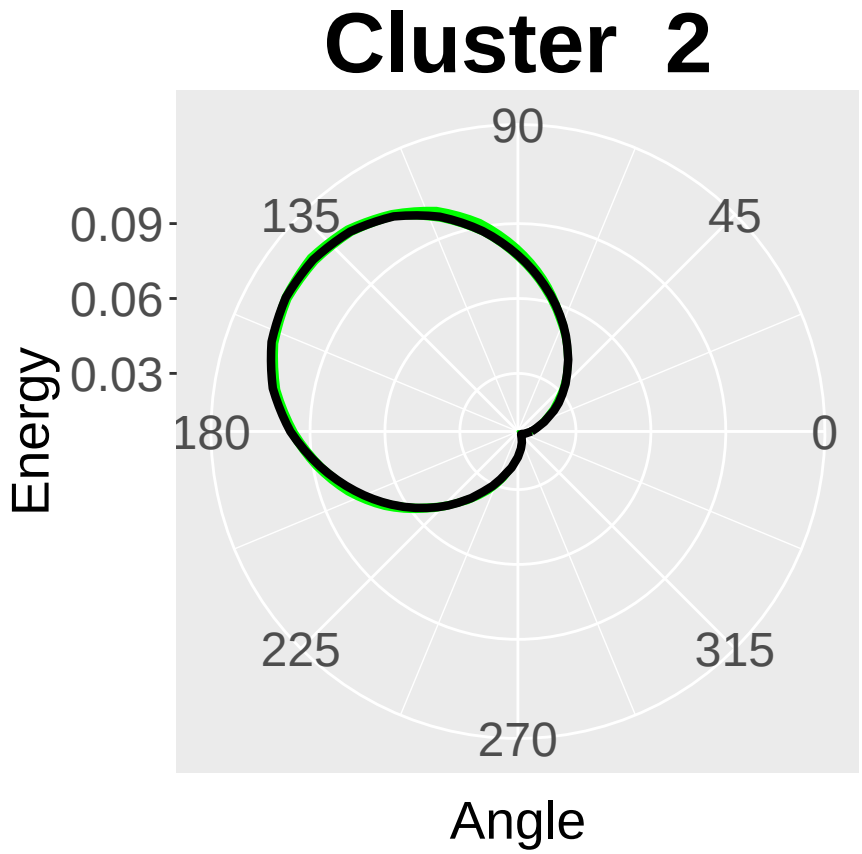



```
fbplot(w/(2*pi),fit=f2,col="green",xlim = c(0,.5),
       ylim=c(0,0.4),xlab="w",ylab="",cex.axis=1.5,cex.lab=1.7,
       main="Cluster 2",cex.main=2.5)
```

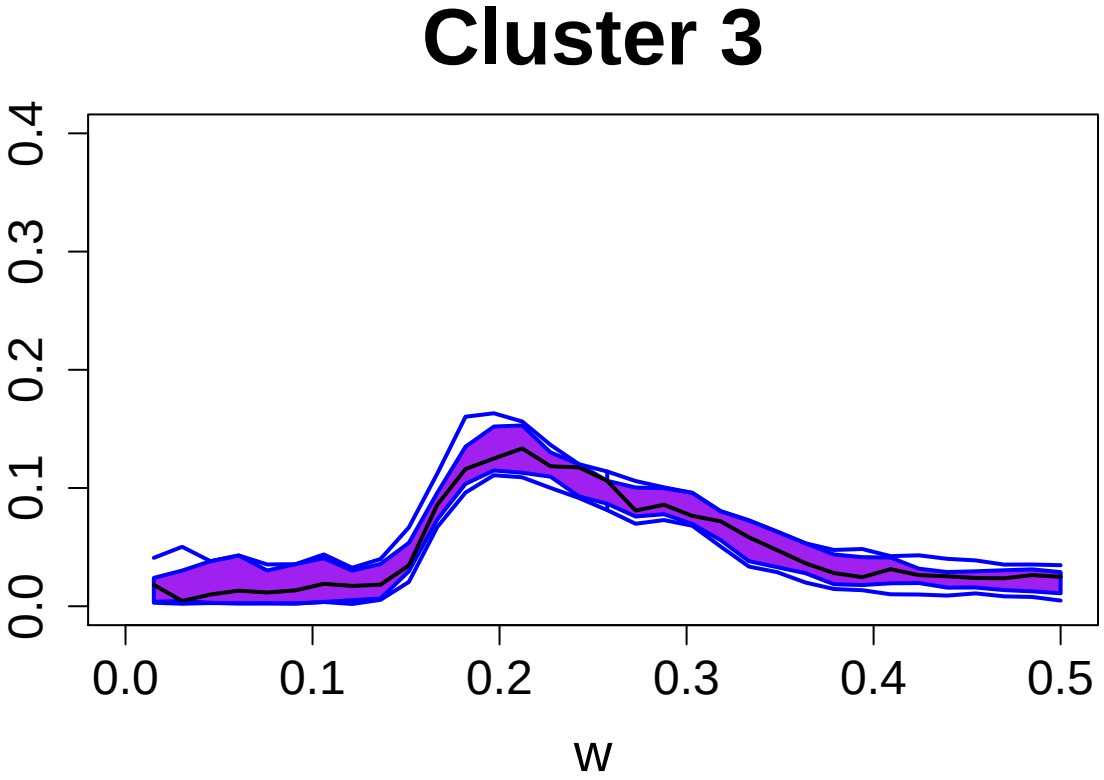



```
## $depth
##  [1] 0.3567842 0.3776623 0.3772171 0.3252195 0.3366976 0.3421892 0.3601484
##  [8] 0.3530736 0.3417440 0.3691033 0.3879530 0.3819171 0.3741991 0.3480767
## [15] 0.3511936 0.3420903 0.3856772 0.3824613 0.3385776 0.3640569 0.3768707
## [22] 0.3442177 0.3295733 0.4156092 0.3892888 0.3752381 0.3504020 0.3289301
## [29] 0.3980952 0.3692517 0.3494620 0.3603463 0.3797897 0.3617316 0.3548547
## [36] 0.3474830 0.3669759 0.3284848 0.3427335 0.3981942 0.3439703 0.3572789
## [43] 0.3414966 0.3397155 0.3875572 0.3236858 0.4367347 0.3186889 0.3330365
## [50] 0.3542610
##
## $outpoint
## integer(0)
##
## $medcurve
## [1] 47
```

```
WrapBP(theta,g2,colorP = "green",index = 2)
```

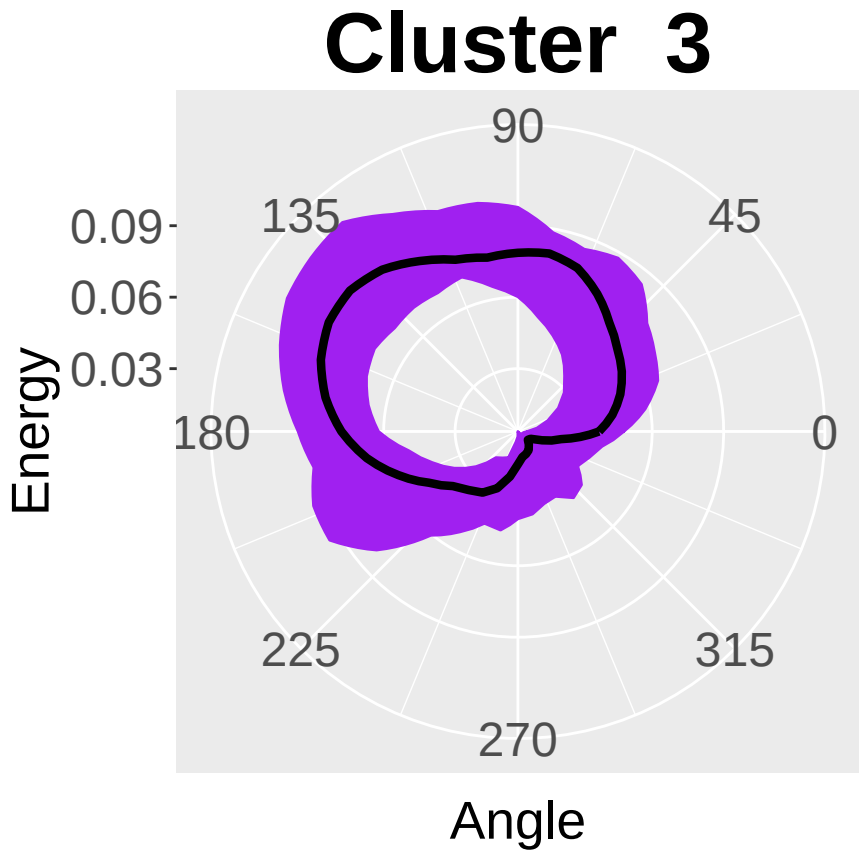



```
fbplot(w/(2*pi),fit=f3,col="purple",xlim = c(0,.5),
       ylim=c(0,0.4),xlab="w",ylab="",cex.axis=1.5,cex.lab=1.7,
       main="Cluster 3",cex.main=2.5)
```




```
## $depth
##  [1] 0.4334199 0.3035498 0.3511936 0.3384787 0.3388745 0.2695609 0.3203711
##  [8] 0.2806926 0.2943476 0.3076067 0.4303525 0.4134323 0.3979468 0.3341249
## [15] 0.3890414 0.3676685 0.2961781 0.4393074 0.2249351 0.3262090 0.3802845
## [22] 0.4260977 0.4150649 0.3901793 0.3083983 0.3966605 0.3585158 0.4216450
## [29] 0.3806803 0.3272975 0.4375263 0.3723191 0.4044774 0.2564997 0.3496599
## [36] 0.3032529 0.2507607 0.4360421 0.3830056 0.3934941 0.4042795 0.4267409
## [43] 0.3963142 0.3204205 0.3554484 0.3648485 0.3791960 0.3521831 0.3281385
## [50] 0.4232777
##
## $outpoint
## integer(0)
##
## $medcurve
## [1] 18
```

```
WrapBP(theta,g3,colorP = "purple",index = 3)
```



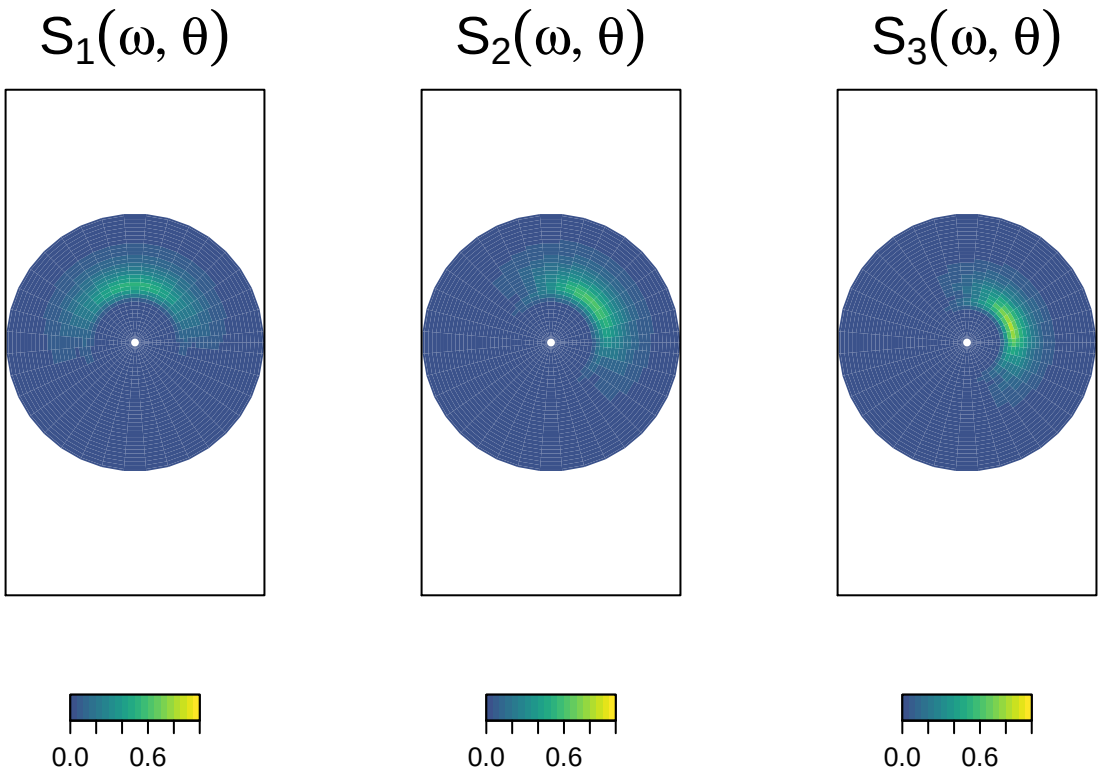
Monte Carlo Study

Based on load Monte Carlo simulations results we obtained the following plots. The code to generate the MC Simulations is included at the end but it will take around 4 hours to run.

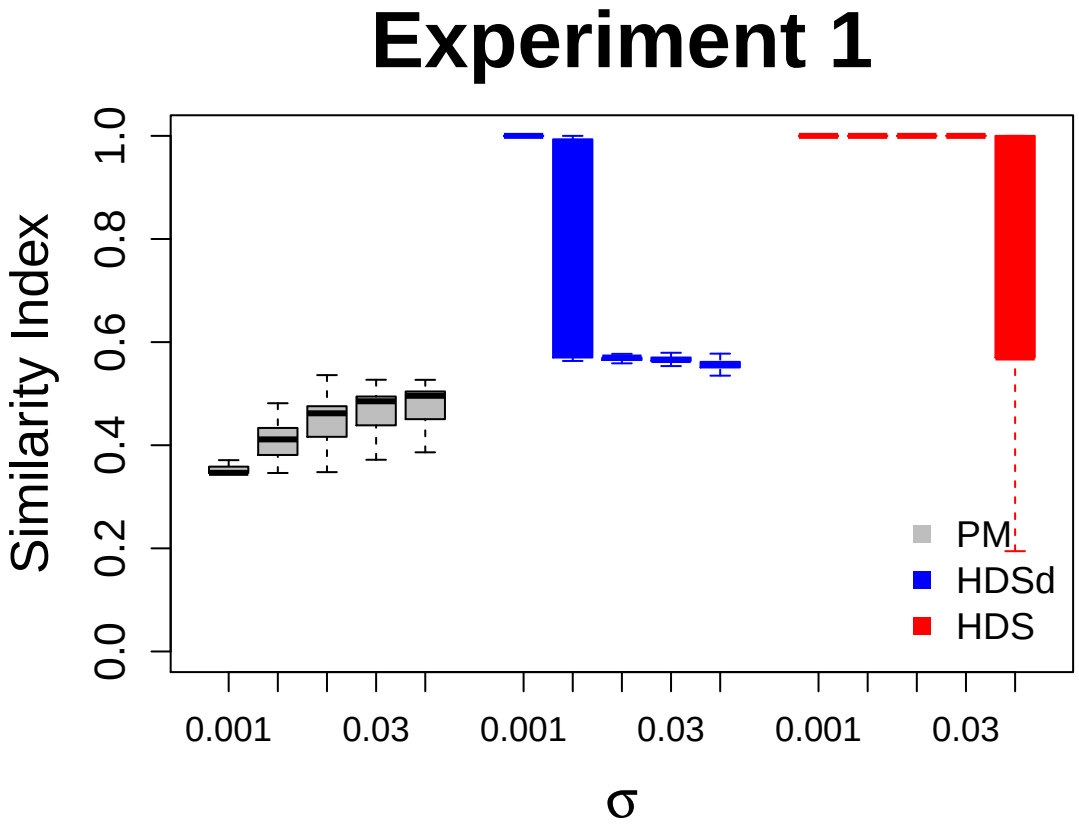
Monte Carlo Study 1

```
## DS plots
Sd.Mat<-Aux.Cov.Mat(w,theta,lambda = c(.5,10))
Sd.Mat<-sqrt(.0001)*Sd.Mat
Sim.S1<-Sim.DirSpec(w,theta,hs=1,tp=1/(.22*2*pi),theta0=pi/2,smax=3,n=1,gam = 1,sd.mat = Sd.Mat)
Sim.S2<-Sim.DirSpec(w,theta,hs=1,tp=1/(.22*2*pi),theta0=pi/4,smax=5,n=1,gam = 1,sd.mat = Sd.Mat)
Sim.S3<-Sim.DirSpec(w,theta,hs=1,tp=1/(.18*2*pi),theta0=pi/8,smax=5,n=1,gam=1,sd.mat = Sd.Mat)
norms<-c(normalize(x=w,y=theta,normfx = TRUE,D=2,fx =Sim.S1$S.w.theta),
         normalize(x=w,y=theta,normfx = TRUE,D=2,fx =Sim.S2$S.w.theta),
         normalize(x=w,y=theta,normfx = TRUE,D=2,fx =Sim.S3$S.w.theta))
par(mfrow=c(1,3))
image2D(x=Xmat,y=Ymat,z= Sim.S1$S.w.theta/norms[1],col=viridis_pal(begin = .25)(20),asp=1,
        colkey = list(side = 1, length = 0.5,cex.axis=1.3) ,theta = 0, cex.main=2.5,
        font.main=2,axes=FALSE,main=expression(S[1](omega,theta)),xlab=" ", ylab=" ",
        zlim=c(0,1))
image2D(x=Xmat,y=Ymat,z= Sim.S2$S.w.theta/norms[2],cex.main=2.5,font.main=2,col=viridis_pal(begin = .25
        colkey = list(side = 1, length = 0.5,cex.axis=1.3) ,theta = 0,
```

```
        axes=FALSE,main=expression(S[2](omega,theta)),xlab=" ", ylab=" ",zlim=c(0,1))
image2D(x=Xmat,y=Ymat,z= Sim.S3$S.w.theta/norms[3],cex.main=2.5,font.main=2,
        colkey = list(side = 1, length = 0.5,cex.axis=1.3) ,theta = 0, col=viridis_pal(begin = .25)(20)
        axes=FALSE,main=expression(S[3](omega,theta)),xlab=" ", ylab=" ",zlim=c(0,1))
```




```
##
tau<-sqrt(c(.000001,.0001,.0004,.0009,.0025))
par(mfrow=c(1,1),mar=c(4,5,4,4))
boxplot(at=c(1:5,7:11,13:17),
        do.call("cbind", COMPLETE.RES.CASE1)[,c(1,4,7,10,13,3,6,9,12,15,2,5,8,11,14)],
        col=c(rep("gray",5),rep("blue",5),rep("red",5)),ylim=c(0,1),border=c(rep("black",5),
        rep("blue",5),rep("red",5)), axes=FALSE,xlab=expression(sigma),cex.lab=1.7,
        main="Experiment 1",cex.main=2.5,ylab="Similarity Index",outline=FALSE)
axis(2,cex.axis=1.3)
box()
axis(1, at=c(1:5,7:11,13:17),labels = rep(tau,3),cex.axis=1.1)
legend("bottomright",col=c("gray","blue","red"),pch=15,c("PM","HDSd","HDS"),bty="n",
       cex=1.2,
       xjust=1)
```

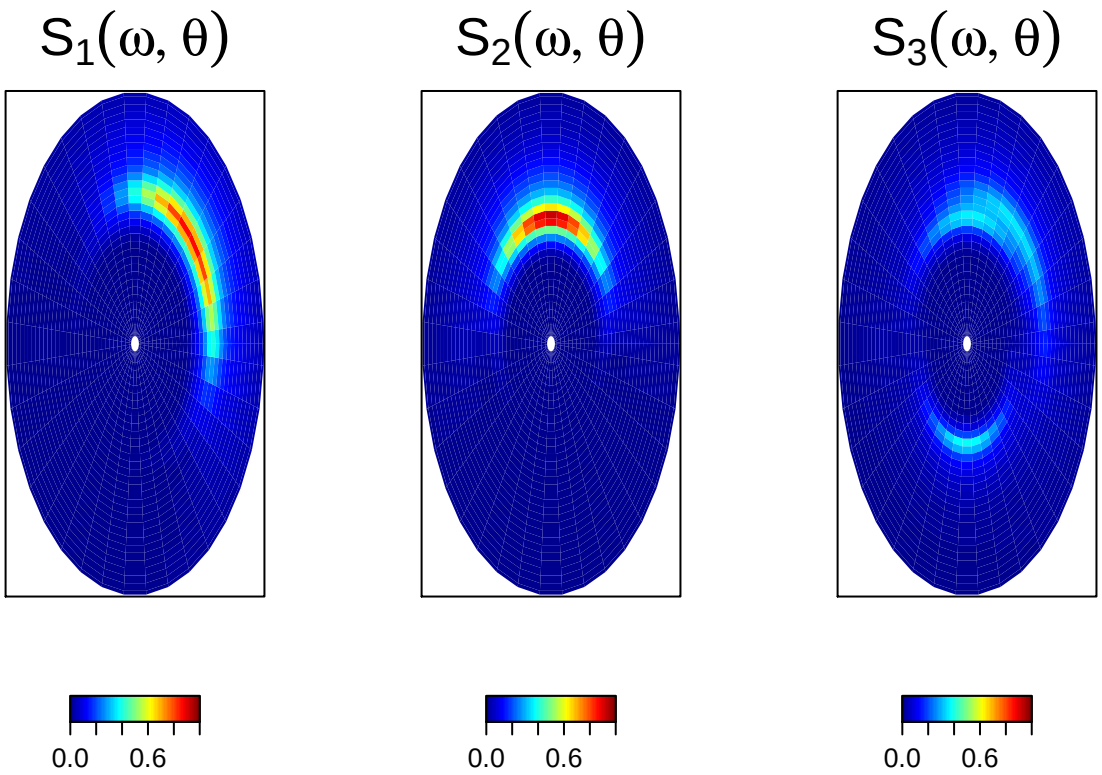


Monte Carlo Study 2

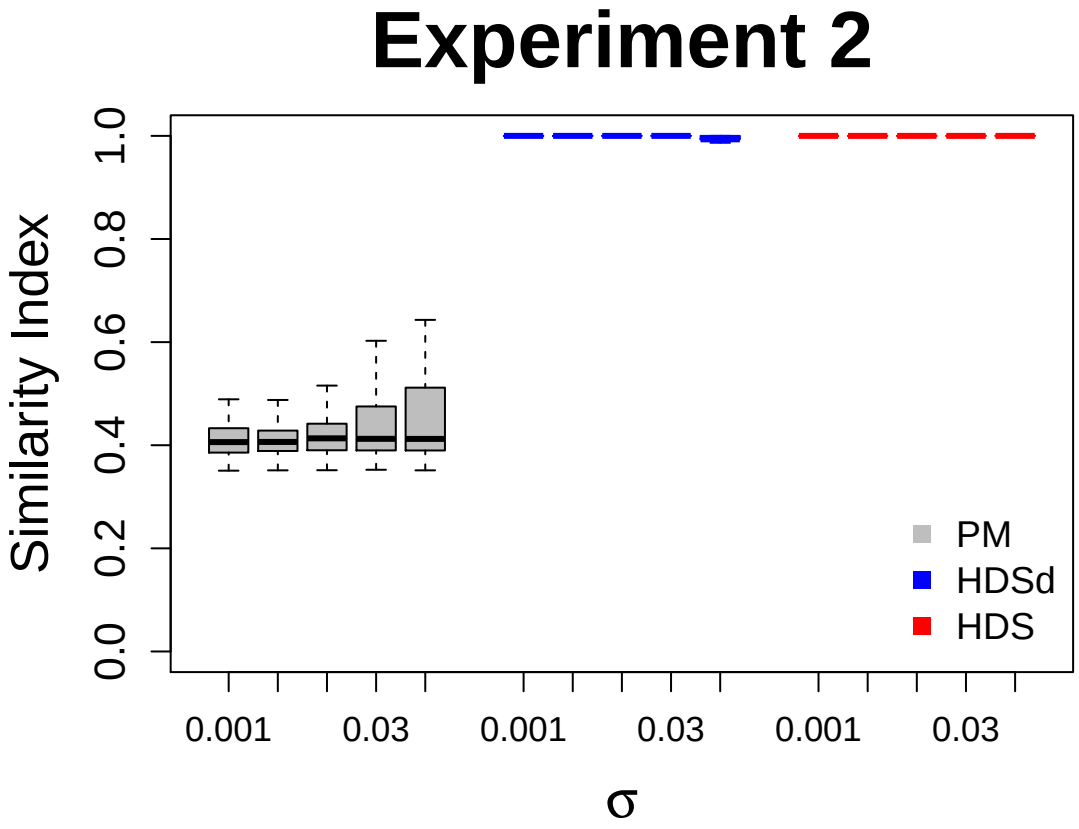
```
#DS plots
Sd.Mat<-Aux.Cov.Mat(w,theta,lambda = c(.5,10))
Sd.Mat<-sqrt(.0001)*Sd.Mat
Sim.S1<-Sim.DirSpec(w,theta,hs=1,tp=1/(.3*2*pi),theta0=pi/4,smax=5,n=2,gam = 2)
Sim.S2<-Sim.DirSpec(w,theta,hs=2,tp=1/(.25*2*pi),theta0=pi/2,smax=5,n=2,gam = 2)
Sim.SS<-Sim.DirSpec(w,theta,hs=1,tp=1/(.2*2*pi),theta0=3*pi/2,smax=5,n=2,gam = 2)

SS<-list(normalize(Sim.S1$S.w.theta,x = w,y = theta,D = 2),
         normalize(Sim.S2$S.w.theta,x = w,y = theta,D = 2),
         normalize(Sim.SS$S.w.theta,x = w,y = theta,D = 2))
Sim.S3<-Sim.DirSpeck(w = w,theta = theta,SpecList = SS,n = 2,plot = FALSE,sd.mat = Sd.Mat )
norms<-c(normalize(x=w,y=theta,normfx = TRUE,D=2,fx =Sim.S1$S.w.theta),
         normalize(x=w,y=theta,normfx = TRUE,D=2,fx =Sim.S2$S.w.theta),
         normalize(x=w,y=theta,normfx = TRUE,D=2,fx =Sim.S3$S.w.theta))
par(mfrow=c(1,3))
image2D(x=Xmat,y=Ymat,z= Sim.S1$S.w.theta/norms[1]
        ,colkey = list(side = 1, length = 0.5,cex.axis=1.3) ,theta = 0, cex.main=2.5,
        font.main=2,axes=FALSE,main=expression(S[1](omega,theta)),xlab=" ", ylab=" ",
        zlim=c(0,1))
image2D(x=Xmat,y=Ymat,z= Sim.S2$S.w.theta/norms[2],cex.main=2.5,font.main=2,
        colkey = list(side = 1, length = 0.5,cex.axis=1.3) ,theta = 0,
        axes=FALSE,main=expression(S[2](omega,theta)),xlab=" ", ylab=" ",zlim=c(0,1))
image2D(x=Xmat,y=Ymat,z= Sim.S3$S.w.theta/norms[3],cex.main=2.5,font.main=2,
```

```
        colkey = list(side = 1, length = 0.5,cex.axis=1.3) ,theta = 0,
        axes=FALSE,main=expression(S[3](omega,theta)),xlab=" ", ylab=" ",zlim=c(0,1))
```




```
par(mfrow=c(1,1),mar=c(4,5,4,4))
boxplot(at=c(1:5,7:11,13:17),
        do.call("cbind", COMPLETE.RES.CASE2)[,c(1,4,7,10,13,3,6,9,12,15,2,5,8,11,14)],
        col=c(rep("gray",5),rep("blue",5),rep("red",5)),ylim=c(0,1),border=c(rep("black",5),
        rep("blue",5),rep("red",5)), axes=FALSE,xlab=expression(sigma),cex.lab=1.7,
        main="Experiment 2",cex.main=2.5,ylab="Similarity Index",outline=FALSE)
axis(2,cex.axis=1.3)
box()
axis(1, at=c(1:5,7:11,13:17),labels = rep(tau,3),cex.axis=1.1)
legend("bottomright",col=c("gray","blue","red"),pch=15,c("PM","HDSd","HDS"),bty="n",
       cex=1.2,
       xjust=1)
```




```
#Code to replicate the MC simulations
##############################
# Monte carlo study 1 #######
##############################
tau<-sqrt(c(.000001,.0001,.0004,.0009,.0025))
n1<-50
n2<-50
n3<-50
M<-1000
library("doParallel")
cl <- min(k, detectCores() - 1)
registerDoParallel(cl)

TG<-matrix(1:150,ncol=3,byrow = FALSE)
COMPLETE.RES.CASE1<-vector("list",5)
PS.MEAN.CASE1<-vector("list",5)
for(indtau in 1:5){
  Sd.Mat<-Aux.Cov.Mat(w,theta,lambda = c(.5,10))
  Sd.Mat<-sqrt(tau[indtau])*Sd.Mat

  ClustRes <- foreach(icount(M), .combine = rbind) %dopar%
  {

    Sim.S1<-Sim.DirSpec(w,theta,hs=1,tp=1/(.22*2*pi),theta0=pi/2,smax=3,n=n1,
                        gam = 1,sd.mat = Sd.Mat)
    Sim.S2<-Sim.DirSpec(w,theta,hs=1,tp=1/(.22*2*pi),theta0=pi/4,smax=5,n=n2,
                        gam = 1,sd.mat = Sd.Mat)
    Sim.S3<-Sim.DirSpec(w,theta,hs=1,tp=1/(.18*2*pi),theta0=pi/8,smax=5,n=n3,
```

```
                        gam=1,sd.mat = Sd.Mat)

    DS1<-Sim.S1$Sim.S.w.theta
    DS2<-Sim.S2$Sim.S.w.theta
    DS3<-Sim.S3$Sim.S.w.theta

    jDS1<-lapply(DS1,FUN="normalize",x=w,D=2,y=theta)
    jDS2<-lapply(DS2,FUN="normalize",x=w,D=2,y=theta)
    jDS3<-lapply(DS3,FUN="normalize",x=w,D=2,y=theta)

    PS1<-lapply(jDS1, "Partitioning",w=w,theta=theta,tol=.05,min.area=.2)
    PS2<-lapply(jDS2, "Partitioning",w=w,theta=theta,tol=.05,min.area=.2)
    PS3<-lapply(jDS3, "Partitioning",w=w,theta=theta,tol=.05,min.area=.2)

    MeanP<-c(mean(unlist(lapply(PS1, nrow))),mean(unlist(lapply(PS2, nrow))),
             mean(unlist(lapply(PS2, nrow))))

    SID1<-rep(1:n1,times=unlist(lapply(PS1,nrow)))
    SID2<-rep(1:n2,times=unlist(lapply(PS2,nrow)))+n1
    SID3<-rep(1:n3,times=unlist(lapply(PS3,nrow)))+(n1+n2)

    PS<-cbind(rep(1,length(SID1)),SID1,do.call("rbind", PS1))
    PS<-rbind(PS,cbind(rep(2,length(SID2)),SID2,do.call("rbind", PS2)))
    PS<-rbind(PS,cbind(rep(3,length(SID3)),SID3,do.call("rbind", PS3)))

    HPS<-agnes(PS[,c("w.p","theta.p")])
    GPS<-cutree(HPS,k=3)

    jDS<-c(jDS1,jDS2,jDS3)
    gs<-matrix(unlist(lapply(jDS,colMeans)),nrow = length(theta),byrow = FALSE)

    HDSClustDO<-HSM(S=gs,Merger = 2,w = theta,parallel = FALSE)
    HDSClust<-HM2D(x=w,y=theta,fxy = jDS, parallel = TRUE, normalize = FALSE)

    G_PS<-matrix(NA,ncol = 3,nrow=150)
    G_HDSDO<-matrix(NA,ncol = 3,nrow=150)
    G_HDS<-matrix(NA,ncol = 3,nrow=150)
    GDSDO<-cutk(HDSClustDO,k=3)
    GDS<-cutk(HDSClust,kg=3)

    for(indk in 1:3){
      G_PS[unique(PS[GPS==indk,2]),indk]<-unique(PS[GPS==indk,2])
      G_HDSDO[GDSDO[[indk]],indk]<-GDSDO[[indk]]
      G_HDS[GDS[[indk]],indk]<-GDS[[indk]]
    }

    RESULTS<-c(Sim(G_PS,TG),Sim(G_HDS,TG),Sim(G_HDSDO,TG),MeanP)
    RESULTS
  }
  COMPLETE.RES.CASE1[[indtau]]<-ClustRes[,1:3]
  PS.MEAN.CASE1[[indtau]]<-ClustRes[,4:6]
}
```

```
# Monte carlo study 2 #######
#two one mode one bimodal  ##
##############################
tau<-sqrt(c(.000001,.0001,.0004,.0009,.0025))
n1<-50
n2<-50
n3<-50
M<-1000
k<-n1+n2+n3
library("doParallel")
cl <- min(k, detectCores() - 1)
registerDoParallel(cl)

TG<-matrix(1:150,ncol=3,byrow = FALSE)
COMPLETE.RES.CASE2<-vector("list",5)
PS.MEAN.CASE2<-vector("list",5)
for(indtau in 1:5){
  Sd.Mat<-Aux.Cov.Mat(w,theta,lambda = c(.5,10))
  Sd.Mat<-sqrt(tau[indtau])*Sd.Mat
  ClustRes <- foreach(icount(M), .combine = rbind) %dopar%
  {
    Sim.S1<-Sim.DirSpec(w,theta,hs=1,tp=1/(.3*2*pi),theta0=pi/4,smax=5,n=n1,
                        gam = 2)
    Sim.S2<-Sim.DirSpec(w,theta,hs=2,tp=1/(.25*2*pi),theta0=pi/2,smax=5,n=n2,
                        gam = 2)
    Sim.SS<-Sim.DirSpec(w,theta,hs=1,tp=1/(.2*2*pi),theta0=3*pi/2,smax=5,n=1,
                        gam = 2)

    SS<-list(normalize(Sim.S1$S.w.theta,x = w,y = theta,D = 2),
             normalize(Sim.S2$S.w.theta,x = w,y = theta,D = 2),
             normalize(Sim.SS$S.w.theta,x = w,y = theta,D = 2))
    Sim.S3<-Sim.DirSpeck(w = w,theta = theta,SpecList = SS,n = n3,sd.mat = Sd.Mat )


    DS1<-Sim.S1$Sim.S.w.theta
    DS2<-Sim.S2$Sim.S.w.theta
    DS3<-Sim.S3$Sim.S.w.theta

    jDS1<-lapply(DS1,FUN="normalize",x=w,D=2,y=theta)
    jDS2<-lapply(DS2,FUN="normalize",x=w,D=2,y=theta)
    jDS3<-lapply(DS3,FUN="normalize",x=w,D=2,y=theta)

    PS1<-lapply(jDS1, "Partitioning",w=w,theta=theta,tol=.05,min.area=.2)
    PS2<-lapply(jDS2, "Partitioning",w=w,theta=theta,tol=.05,min.area=.2)
    PS3<-lapply(jDS3, "Partitioning",w=w,theta=theta,tol=.05,min.area=.2)

    MeanP<-c(mean(unlist(lapply(PS1, nrow))),mean(unlist(lapply(PS2, nrow))),
             mean(unlist(lapply(PS2, nrow))))

    SID1<-rep(1:n1,times=unlist(lapply(PS1,nrow)))
    SID2<-rep(1:n2,times=unlist(lapply(PS2,nrow)))+n1
    SID3<-rep(1:n3,times=unlist(lapply(PS3,nrow)))+(n1+n2)
```

```
    PS<-cbind(rep(1,length(SID1)),SID1,do.call("rbind", PS1))
    PS<-rbind(PS,cbind(rep(2,length(SID2)),SID2,do.call("rbind", PS2)))
    PS<-rbind(PS,cbind(rep(3,length(SID3)),SID3,do.call("rbind", PS3)))

    HPS<-agnes(PS[,c("w.p","theta.p")])
    GPS<-cutree(HPS,k=3)

    jDS<-c(jDS1,jDS2,jDS3)
    gs<-matrix(unlist(lapply(jDS,colMeans)),nrow = length(theta),byrow = FALSE)

    HDSClustDO<-HSM(S=gs,Merger = 2,w = theta,parallel = FALSE)
    HDSClust<-HM2D(x=w,y=theta,fxy = jDS, parallel = TRUE, normalize = FALSE)

    G_PS<-matrix(NA,ncol = 3,nrow=150)
    G_HDSDO<-matrix(NA,ncol = 3,nrow=150)
    G_HDS<-matrix(NA,ncol = 3,nrow=150)
    GDSDO<-cutk(HDSClustDO,k=3)
    GDS<-cutk(HDSClust,kg=3)

    for(indk in 1:3){
      G_PS[unique(PS[GPS==indk,2]),indk]<-unique(PS[GPS==indk,2])
      G_HDSDO[GDSDO[[indk]],indk]<-GDSDO[[indk]]
      G_HDS[GDS[[indk]],indk]<-GDS[[indk]]
    }

    RESULTS<-c(Sim(G_PS,TG),Sim(G_HDS,TG),Sim(G_HDSDO,TG),MeanP)
    RESULTS
  }
  COMPLETE.RES.CASE2[[indtau]]<-ClustRes[,1:3]
  PS.MEAN.CASE2[[indtau]]<-ClustRes[,4:6]
}
##
```