Large PSM clustering

Paul DW Kirk

02/05/2021

Large PSM clustering

In order to get this Rmd file to run, I first changed the maximum memory available to R by following the top answer given at: https://stackoverflow.com/questions/51295402/r-on-macos-error-vector-memory-exhausted-limit-reached

It would be worth checking to see if this (i.e. editing .Renviron) would fix the issue being encountered when using the maxpear function.

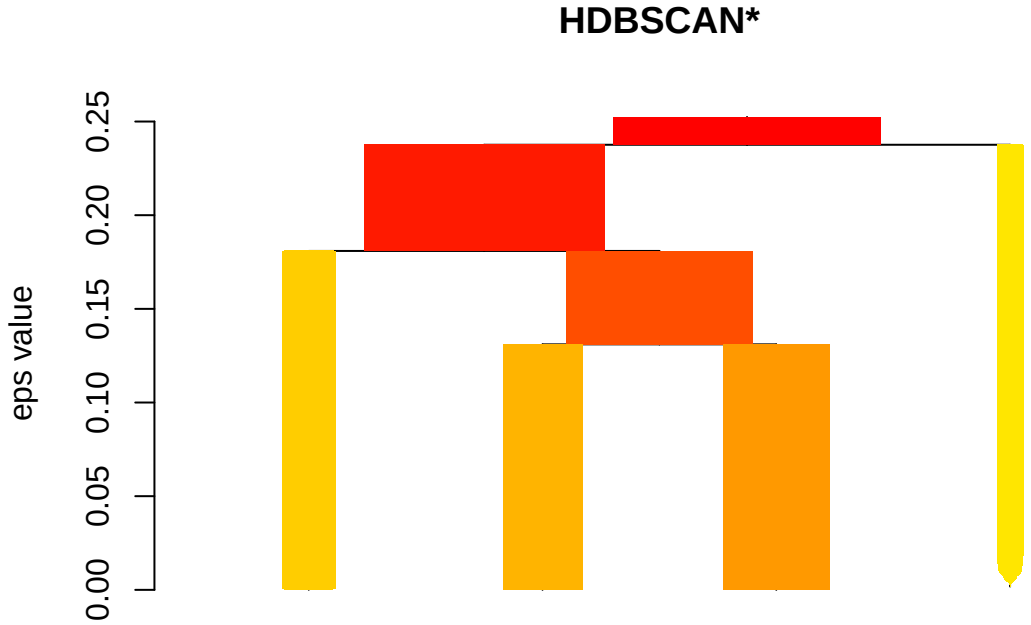
```
rm(list = ls())

generatePSM <- function(n, seed = 1)
{
  nSamplesCluster1 <- round(0.1*n)
  nSamplesCluster2 <- round(0.2*n)
  nSamplesCluster3 <- round(0.3*n)
  nSamplesCluster4 <- n - (nSamplesCluster1 + nSamplesCluster2 + nSamplesCluster3)

  clusters <- c(rep(1,nSamplesCluster1), rep(2,nSamplesCluster2),
                rep(3,nSamplesCluster3), rep(4,nSamplesCluster4))

  pilotPSM <- matrix(0, nrow = n, ncol = n)
  pilotPSM[1:nSamplesCluster1, 1:nSamplesCluster1]                   <- 1
  pilotPSM[(nSamplesCluster1+1):(nSamplesCluster1+nSamplesCluster2),
           (nSamplesCluster1+1):(nSamplesCluster1+nSamplesCluster2)] <- 1
  pilotPSM[(nSamplesCluster1+nSamplesCluster2+1):
             (nSamplesCluster1+nSamplesCluster2+nSamplesCluster3),
           (nSamplesCluster1+nSamplesCluster2+1):
             (nSamplesCluster1+nSamplesCluster2+nSamplesCluster3)]   <- 1
  pilotPSM[(nSamplesCluster1+nSamplesCluster2+nSamplesCluster3+1):n,
           (nSamplesCluster1+nSamplesCluster2+nSamplesCluster3+1):n] <- 1

  set.seed(seed)
  noiseMatrix <- matrix(rbeta(n*n,0.1,2), nrow = n, ncol = n)

  pilotPSM    <- abs(pilotPSM - noiseMatrix)
  newPSM      <- (pilotPSM + t(pilotPSM))/2
  return(list(psm = newPSM, clusters = clusters))
}

n <- 1000
psmResults   <- generatePSM(n)
trueClusters <- psmResults$clusters
```

```
psm1000      <- psmResults$psm


#Make symmetric
graphics.off()
pheatmap::pheatmap(psm1000, cluster_rows = F, cluster_cols = F)
hist(psm1000)
```

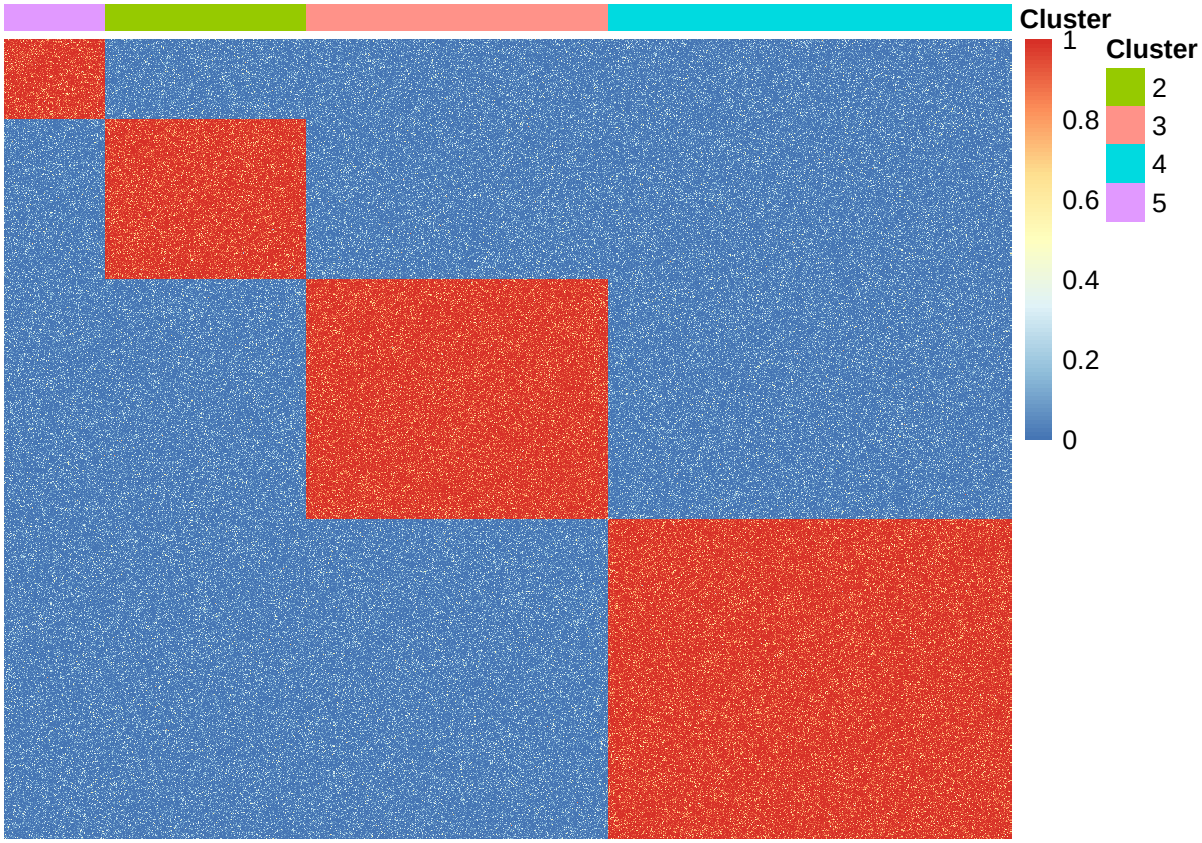
Illustrate the use of HDBSCAN

We check to see how we can use HDBSCAN to perform clustering on the basis of the PSM.

```
library("ggplot2")
library("dbscan")

myDist         <- as.dist(1 - psm1000)

hdbscanResults <- hdbscan(myDist, minPts = round(0.05*n))

plot(hdbscanResults) # A summary dendrogram, showing the relationships between the clusters
```




```
# Generate annotations for rows and columns
annotation_col = data.frame(
  Cluster = factor(hdbscanResults$cluster+1L)
)

rownames(annotation_col) <- rownames(psm1000) <-
  colnames(psm1000) <- paste0("V", seq(1,n))

pheatmap::pheatmap(psm1000, cluster_rows = F, cluster_cols = F,
                   annotation_col = annotation_col,
                   show_rownames = F,
                   show_colnames = F)
```




```
print(mcclust::arandi(trueClusters, hdbscanResults$cluster))
```

```
## [1] 1
```

Assess how HDBSCAN scales with increasing n

```
library(tictoc)

#rangeOfSampleSizes <- c(100, 500, 1000, 1500, 2500, 5000, 10000, 20000,
#                        50000, 1e5, 2e5, 5e5, 1e6, 5e6, 1e7, 1e8)

rangeOfSampleSizes <- c(100, 500, 1000, 1500, 2500, 5000, 10000, 27000)

nRepetitions  <- 5

timingsMatrix <- ariMatrix <- matrix(nrow = length(rangeOfSampleSizes),
                                     ncol = nRepetitions)

for(i in 1:length(rangeOfSampleSizes))
{
  print(paste("i = ",  i))
  for(j in 1:nRepetitions)
  {
    print(paste("j = ",  j))

    n                  <- rangeOfSampleSizes[i]
    print(n)
```

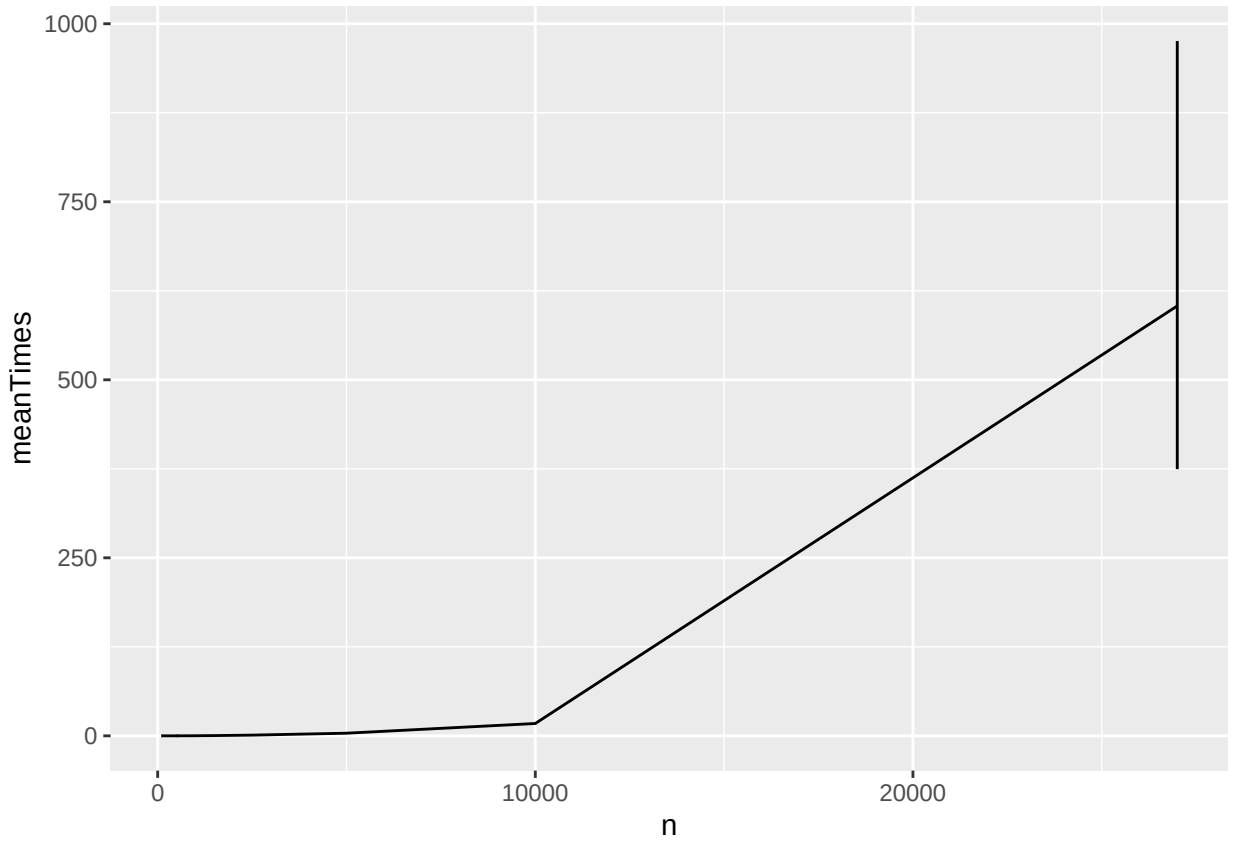
```
    psmResults          <- generatePSM(n)
    trueClusters        <- psmResults$clusters
    psm                 <- psmResults$psm

    myDist              <- as.dist(1 - psm)

    tic()
    hdbscanResults      <- hdbscan(myDist, minPts = round(0.05*n))
    timed               <- toc(quiet = T)
    timingsMatrix[i,j] <- timed$toc - timed$tic

    ariMatrix[i,j]      <- mcclust::arandi(trueClusters, hdbscanResults$cluster)

    print(timingsMatrix[i,j])


  }


}
```

```
## [1] "i =  1"
## [1] "j =  1"
## [1] 100
## [1] 0.004
## [1] "j =  2"
## [1] 100
## [1] 0.004
## [1] "j =  3"
## [1] 100
## [1] 0.005
## [1] "j =  4"
## [1] 100
## [1] 0.006
## [1] "j =  5"
## [1] 100
## [1] 0.005
## [1] "i =  2"
## [1] "j =  1"
## [1] 500
## [1] 0.049
## [1] "j =  2"
## [1] 500
## [1] 0.037
## [1] "j =  3"
## [1] 500
## [1] 0.042
## [1] "j =  4"
## [1] 500
## [1] 0.048
## [1] "j =  5"
## [1] 500
## [1] 0.048
## [1] "i =  3"
```

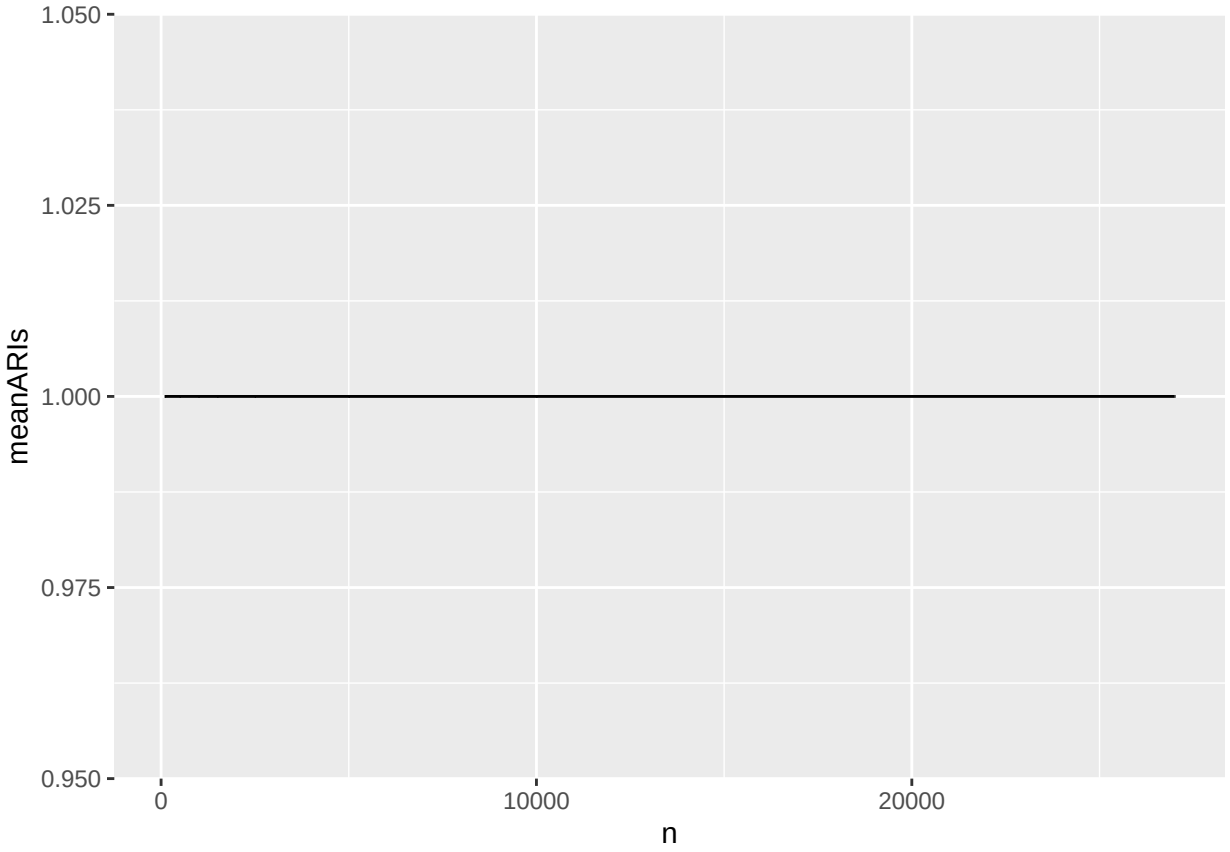
```
## [1] "j =  1"
## [1] 1000
## [1] 0.177
## [1] "j =  2"
## [1] 1000
## [1] 0.129
## [1] "j =  3"
## [1] 1000
## [1] 0.194
## [1] "j =  4"
## [1] 1000
## [1] 0.187
## [1] "j =  5"
## [1] 1000
## [1] 0.171
## [1] "i =  4"
## [1] "j =  1"
## [1] 1500
## [1] 0.351
## [1] "j =  2"
## [1] 1500
## [1] 0.349
## [1] "j =  3"
## [1] 1500
## [1] 0.414
## [1] "j =  4"
## [1] 1500
## [1] 0.416
## [1] "j =  5"
## [1] 1500
## [1] 0.427
## [1] "i =  5"
## [1] "j =  1"
## [1] 2500
## [1] 1.096
## [1] "j =  2"
## [1] 2500
## [1] 0.889
## [1] "j =  3"
## [1] 2500
## [1] 1.063
## [1] "j =  4"
## [1] 2500
## [1] 0.984
## [1] "j =  5"
## [1] 2500
## [1] 1.001
## [1] "i =  6"
## [1] "j =  1"
## [1] 5000
## [1] 4.198
## [1] "j =  2"
## [1] 5000
## [1] 3.605
```

```
## [1] "j =  3"
## [1] 5000
## [1] 3.635
## [1] "j =  4"
## [1] 5000
## [1] 3.511
## [1] "j =  5"
## [1] 5000
## [1] 3.689
## [1] "i =  7"
## [1] "j =  1"
## [1] 10000
## [1] 17.36
## [1] "j =  2"
## [1] 10000
## [1] 17.143
## [1] "j =  3"
## [1] 10000
## [1] 16.61
## [1] "j =  4"
## [1] 10000
## [1] 17.419
## [1] "j =  5"
## [1] 10000
## [1] 18.283
## [1] "i =  8"
## [1] "j =  1"
## [1] 27000
## [1] 374.604
## [1] "j =  2"
## [1] 27000
## [1] 498.47
## [1] "j =  3"
## [1] 27000
## [1] 687.672
## [1] "j =  4"
## [1] 27000
## [1] 482.008
## [1] "j =  5"
## [1] 27000
## [1] 975.741
```

```
timingsToPlot <- data.frame(n = rangeOfSampleSizes, meanTimes = rowMeans(timingsMatrix), lower = apply(t

arisToPlot <- data.frame(n = rangeOfSampleSizes, meanARIs = rowMeans(ariMatrix), lower = apply(ariMatri


pTimings <- ggplot(timingsToPlot, aes(n, meanTimes)) + geom_line() + geom_errorbar(aes(ymin = lower, yma

pARIs <- ggplot(arisToPlot, aes(n, meanARIs)) + geom_line() + geom_errorbar(aes(ymin = lower, ymax = upp


plot(pTimings)
```




```
plot(pARIs)
```


1.050
1.025
1.000
0.975
0.950
meanARIs
0
10000
20000
n