*

June 10, 2022

The results below are generated from an R script.

```
## Install a package manager and packages
if (!require("pacman")) {
  install.packages("pacman")
}
pacman::p_load(Rfast, foreach, doParallel, mvnfast, rstudioapi)
current_path = rstudioapi::getActiveDocumentContext()$path
setwd(dirname(current_path))

pacman::p_load_gh("pkimes/sigclust2")
shc = get("shc", env = environment(shc))

source("sequential_function.R")

# k = 3 # number of clusters (3 or 10)
# uneven = FALSE #whether or not to have uneven weights
# distribution = 't' # t distribution or normal distribution?
# iterations = 50 # number of iterations


n = 500 # total number of samples
alpha = 0.05


if (distribution=='t'){
  distribution_name = 'True distribution components: t-distrbution (df=3) mixture distribution'
  samplefunc <- function(n, mu, sigma, w){
    rmixt(n = n,mu = mu,sigma = sigma,w = w,df = 3)
  }
}else{
  distribution_name = 'True distribution: Normal mixture distribution'
  samplefunc <- function(n, mu, sigma, w){
  rmixn(n=n, mu=mu, sigma=sigma, w=w)
}
}


# formulating d, delta (dimension and distance between clusters)
if (k == 10){
  a = c(2, 20, 2, 40, 2, 60, 2, 80, 2, 100, 2, 150, 2, 200) # dim2
  b =c(8, 20, 8, 40, 8, 60, 8, 80, 8, 100, 8, 150, 8, 200) # dim8
  d_delta = matrix(c(a, b) , ncol = 2, byrow = T)
} else if (k == 3){
```

*This report is automatically generated with the R package **knitr** (version 1.37).

```
  a = c(2, 1, 2, 2, 2, 3, 2, 4, 2, 5, 2, 6, 2, 7, 2, 8, 2, 9) # dim2
  b = c(8, 1, 8, 2, 8, 3, 8, 4, 8, 5, 8, 6, 8, 7, 8, 8, 8, 9) # dim8
  d_delta = matrix(c(a, b) , ncol = 2, byrow = T)
} else {
  stop("k != 3 or 10")
}


#weights
w = rep.int(1, k)
if (uneven){
  w[1] = 1 / 4
  w[2] = 1 / 2
}
w = w / sum(w)


K = floor(sqrt(n / 2)) #num clusters to test
K = min(K, 14L) # to ensure not estimating too many clusters

coresToUse = floor(detectCores() / 2) # cores to use

# function which creates data and performs one iteration
simulation <- function(iteration) {

  mu = matrix(runif(k*d, min = 0, max = delta), nrow = k)

  # simulate data
  set.seed(18 + iteration)
  data = samplefunc(n=n, mu=mu, sigma=sigma, w=w)


  D1 = data[1:floor(n / 2), ]
  D2 = data[(floor(n / 2) + 1):n, ]

  # Estimate no.clusters
  Cluster_numbers = estimate.cluster.all(D1, D2, alpha, K)
  sigclust_splits = sum(shc(data, alpha = alpha)$nd_type == "sig")
  return(c(unlist(Cluster_numbers, use.names = F), sigclust_splits + 1L))
}


meanEstimate = matrix(nrow = nrow(d_delta), ncol = iterations)
medianEstimate = meanEstimate
meanEstimatel2 = meanEstimate
medianEstimatel2 = meanEstimate
AICEstimate = meanEstimate
BICEstimate = meanEstimate
sigclustEstimate = meanEstimate
RIFThierEstimate = meanEstimate

# For parallel computing
cl <- makeCluster(coresToUse) #not to overload computer
```

```
registerDoParallel(cl)


for (j in 1:nrow(d_delta)) {
  d = d_delta[j, 1]
  delta = d_delta[j, 2]



  #sigma = lapply(c(3,1,1), function(x) diag(x, nrow=d))
  sigma = lapply(rep.int(1, k), function(x)
    diag(x, nrow = d))




  estimates <-
    foreach(
      i = 1:iterations,
      .combine = cbind,
      .inorder = F,
      .packages = c("mclust", "Rfast", "mvnfast", "MASS"),
      .verbose = F
    ) %dopar% {
      simulation(i)
    }

  # format data into table
  meanEstimate[j, ] = estimates[1, ]
  medianEstimate[j, ] = estimates[2, ]
  meanEstimatel2[j, ] = estimates[3, ]
  medianEstimatel2[j, ] = estimates[4, ]
  BICEstimate[j, ] = estimates[5, ]
  AICEstimate[j, ] = estimates[6, ]
  RIFThierEstimate[j, ] = estimates[7, ]
  sigclustEstimate[j, ] = estimates[8, ]
  df = stack(data.frame(
    cbind(
      "Mean" = meanEstimate[j, ],
      "Meanl2" = meanEstimatel2[j, ],
      "Median" = medianEstimate[j, ],
      "Medianl2" = medianEstimatel2[j, ],
      "AIC" = AICEstimate[j, ],
      "BIC" = BICEstimate[j, ],
      "RIFT.hc" = RIFThierEstimate[j, ],
      "shc" = sigclustEstimate[j, ]
    )
  ))
  print(paste0("(dimension, delta) = (", d, ",", delta, ")"))
  colnames(df) = c("ESTIMATE" , "METHOD")
  tableEstimates = with(df, table(METHOD, ESTIMATE))
  print(tableEstimates)
}
```

```
## [1] "(dimension, delta) = (2,20)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
##   Mean      1 10 13 15 20 19 12  5  1  2  1  1  0  0  0  0
##   Meanl2    1 11 12 15 20 19 12  5  2  2  0  1  0  0  0  0
##   Median    0  0  0  3 12 10 22 17  5  9  6  3  5  8  0  0
##   Medianl2  0  0  0  3 13 10 22 17  6  8  6  3  4  8  0  0
##   AIC       0  0  3  1  5 11 24 21 12 12  6  5  0  0  0  0
##   BIC       0  1  7  7 18 17 21 15  8  4  1  1  0  0  0  0
##   RIFT.hc   8  9 20 30 20 10  3  0  0  0  0  0  0  0  0  0
##   shc       0  1  0  0  6 11 14 19 20 17  6  3  1  0  1  1
## [1] "(dimension, delta) = (2,40)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
##   Mean      0  2  1  2 10  4 16 23 26 12  3  1  0  0  0
##   Meanl2    0  2  1  2 10  4 16 23 26 12  3  1  0  0  0
##   Median    0  0  0  0  0  5  9 23 25 14  7  2  0 15  0
##   Medianl2  0  0  0  0  0  5  9 23 26 14  6  2  0 15  0
##   AIC       0  0  0  0  0  0  4 13 23 30 13  8  4  5  0
##   BIC       0  0  0  2  2  3  6 17 25 31  6  4  1  3  0
##   RIFT.hc   6  3  6 10 20 18 18 14  4  1  0  0  0  0  0
##   shc       1  0  1  0  0  1  3  9 22 36 15  8  1  2  1
## [1] "(dimension, delta) = (2,60)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  8  9 10 11 12 13 14
##   Mean      0  0  1  1  5  5 13 27 16 19  8  3  2  0
##   Meanl2    0  0  1  1  5  5 13 27 16 19  8  3  2  0
##   Median    0  0  0  0  0  1  7 20 15 15  3  3  3 33
##   Medianl2  0  0  0  0  0  1  8 22 13 14  3  2  3 34
##   AIC       0  0  0  0  0  0  2  6 21 23 23 16  7  2
##   BIC       0  0  0  0  0  0  2 11 21 26 20 13  6  1
##   RIFT.hc  11  1  7  6  7  6 16 20 18  8  0  0  0  0
##   shc       1  0  0  0  0  0  0  1 19 43 29  7  0  0
## [1] "(dimension, delta) = (2,80)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  8  9 10 11 12 13 14
##   Mean      0  1  1  1  2  5 22 19 18 19  5  6  1  0
##   Meanl2    0  1  1  1  2  5 22 19 18 19  5  6  1  0
##   Median    0  0  0  0  3  8 11 13 19 13  4  0  4 25
##   Medianl2  0  0  0  0  3  9 12 11 20 13  3  1  4 24
##   AIC       0  0  0  0  0  0 10 10 15 24 20 16  4  1
##   BIC       0  0  1  0  0  0  5 12 17 23 24 14  3  1
##   RIFT.hc  16  1  3  3 10  6 12 13 24  9  3  0  0  0
##   shc       1  0  0  0  0  0  0  0 11 55 21 11  1  0
## [1] "(dimension, delta) = (2,100)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
##   Mean      0  0  0  1  3  9 19 22 23 13  6  2  1  1  0
##   Meanl2    0  0  0  1  3  9 19 22 23 13  6  1  2  1  0
##   Median    0  0  0  1  2  8 21 18 14  6  6  3  7 14  0
##   Medianl2  0  0  0  1  2 12 19 17 14  4  7  2  7 15  0
##   AIC       0  0  0  0  0  0  8 15 25 22 13 11  4  2  0
##   BIC       0  0  0  0  0  1  6 14 26 26 13 10  2  2  0
```

```
##   RIFT.hc  15  3  6  8  5  8 11 13 17 12  2  0  0  0  0
##   shc       0  0  0  0  0  0  1  1  7 46 30 13  1  0  1
## [1] "(dimension, delta) = (2,150)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  8  9 10 11 12 13 14
##   Mean      0  1  0  3  3  3 15 29 25 13  7  1  0  0
##   Meanl2    0  1  0  3  3  3 15 29 25 12  7  1  0  1
##   Median    0  0  0  0  3 13 23 21 21  2  3  1  4  9
##   Medianl2  0  0  0  0  4 14 24 21 17  4  3  1  6  6
##   AIC       0  0  0  0  0  2  9 19 26 20 15  4  3  2
##   BIC       0  0  0  0  0  3  8 19 27 21 16  3  2  1
##   RIFT.hc  16  3 10  9  6  2  2 18 16 16  2  0  0  0
##   shc       0  0  0  0  0  0  0  0  7 43 28 18  3  1
## [1] "(dimension, delta) = (2,200)"
##           ESTIMATE
## METHOD      1  3  4  5  6  7  8  9 10 11 12 13 14
##   Mean      0  0  4  2  7 15 22 27 16  5  2  0  0
##   Meanl2    0  0  4  2  7 15 21 28 16  4  2  1  0
##   Median    0  0  0  6 10 17 27 20  6  2  5  3  4
##   Medianl2  0  0  0  6 12 20 22 20  5  2  5  5  3
##   AIC       0  0  0  0  1  9 15 32 20 15  5  2  1
##   BIC       0  0  0  0  2  8 16 31 21 15  4  2  1
##   RIFT.hc  16  5  6  6  7  6  5 15 32  2  0  0  0
##   shc       0  0  0  0  0  0  0  3 35 41 11  8  2
## [1] "(dimension, delta) = (8,20)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 27
##   Mean      2  2  1  1  0  1  0  0  3 34 42  6  4  4  0  0  0  0  0  0  0  0  0  0
##   Meanl2    2  2  1  1  0  1  0  0  3 34 42  6  4  4  0  0  0  0  0  0  0  0  0  0
##   Median    0  0  0  0  0  0  3 19 40 30  5  3  0  0  0  0  0  0  0  0  0  0  0  0
##   Medianl2  0  0  0  0  0  0  3 19 40 30  5  3  0  0  0  0  0  0  0  0  0  0  0  0
##   AIC       0  0  0  0  0  0  0  0  1  6 61 12 14  6  0  0  0  0  0  0  0  0  0  0
##   BIC       0  0  0  0  0  0  0  0  1 25 68  5  1  0  0  0  0  0  0  0  0  0  0  0
##   RIFT.hc   0  0  0  0  5 11 29 24 15 14  2  0  0  0  0  0  0  0  0  0  0  0  0  0
##   shc       0  0  0  0  0  0  0  0  0  0  0  0  0  4  4 14 11 12 11 21 11  3  8  1
## [1] "(dimension, delta) = (8,40)"
##           ESTIMATE
## METHOD     -1  1  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 27
##   Mean      0  1  1  1  1  2  2  5  4 21 36  8  6 12  0  0  0  0  0  0  0  0  0  0  0
##   Meanl2    0  1  1  1  1  2  2  5  4 21 36  8  6 12  0  0  0  0  0  0  0  0  0  0  0
##   Median    0  0  0  0  0  0  0  0 30 43 22  4  1  0  0  0  0  0  0  0  0  0  0  0  0
##   Medianl2  0  0  0  0  0  0  0  0 34 45 18  3  0  0  0  0  0  0  0  0  0  0  0  0  0
##   AIC       0  0  0  0  0  0  0  0  0  5 26 23 14 32  0  0  0  0  0  0  0  0  0  0  0
##   BIC       0  0  0  0  0  0  0  0  0 12 67 17  2  2  0  0  0  0  0  0  0  0  0  0  0
##   RIFT.hc   1  0  0  0  0  1  6 25 32 31  4  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   shc       0  0  0  0  0  0  0  0  0  0  0  0  0  3  3  8 10 15 12 20 10  7  5  6  1
## [1] "(dimension, delta) = (8,60)"
##           ESTIMATE
## METHOD      1  4  5  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 27
##   Mean      1  1  2  3  4  4 27 37 10  3  8  0  0  0  0  0  0  0  0  0  0  0
##   Meanl2    1  1  2  3  4  4 27 37 10  3  8  0  0  0  0  0  0  0  0  0  0  0
##   Median    0  0  0  0  0 13 51 31  3  1  1  0  0  0  0  0  0  0  0  0  0  0
##   Medianl2  0  0  0  0  0 14 56 29  0  0  1  0  0  0  0  0  0  0  0  0  0  0
##   AIC       0  0  0  0  0  0  8 35 13 22 22  0  0  0  0  0  0  0  0  0  0  0
```

```
##   BIC       0  0  0  0  0  0 14 61 13  8  4  0  0  0  0  0  0  0  0  0  0  0
##   RIFT.hc   0  0  1  2 19 39 33  6  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   shc       0  0  0  0  0  0  0  0  0  0  2  3  7 12 11 15 17 12 12  5  3  1
## [1] "(dimension, delta) = (8,80)"
##           ESTIMATE
## METHOD     -1  2  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 27
##   Mean      0  1  3  1  5  9 27 35  7  6  6  0  0  0  0  0  0  0  0  0  0  0
##   Meanl2    0  1  3  1  5  9 27 35  7  6  6  0  0  0  0  0  0  0  0  0  0  0
##   Median    0  0  0  0  3  4 39 36 12  2  4  0  0  0  0  0  0  0  0  0  0  0
##   Medianl2  0  0  0  0  3  6 47 34  4  2  4  0  0  0  0  0  0  0  0  0  0  0
##   AIC       0  0  0  0  0  0  4 39 16 19 22  0  0  0  0  0  0  0  0  0  0  0
##   BIC       0  0  0  0  0  0 14 58 14  7  7  0  0  0  0  0  0  0  0  0  0  0
##   RIFT.hc   3  0  0  1 10 40 41  5  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   shc       0  0  0  0  0  0  0  0  0  0  2  2  7 15 12 12 17 12 10  5  5  1
## [1] "(dimension, delta) = (8,100)"
##           ESTIMATE
## METHOD     -1  2  4  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 26
##   Mean      0  1  1  2  4  5  8 32 29  6  6  6  0  0  0  0  0  0  0  0  0  0  0
##   Meanl2    0  1  1  2  4  5  8 32 29  6  6  6  0  0  0  0  0  0  0  0  0  0  0
##   Median    0  0  0  0  1  0  6 35 35 10  3 10  0  0  0  0  0  0  0  0  0  0  0
##   Medianl2  0  0  0  0  1  0  6 43 31  8  1 10  0  0  0  0  0  0  0  0  0  0  0
##   AIC       0  0  0  0  0  0  0  8 35 26 20 11  0  0  0  0  0  0  0  0  0  0  0
##   BIC       0  0  0  0  0  0  0 15 55 18  9  3  0  0  0  0  0  0  0  0  0  0  0
##   RIFT.hc   3  1  0  0  1 16 36 34  9  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   shc       0  0  0  0  0  0  0  0  0  0  0  2  4  8 13 11 12 17 13  9  5  5  1
## [1] "(dimension, delta) = (8,150)"
##           ESTIMATE
## METHOD      1  2  4  5  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 26
##   Mean      1  1  3  2  0  7  7 42 29  5  3  0  0  0  0  0  0  0  0  0  0  0  0
##   Meanl2    1  1  3  2  0  7  7 42 29  5  3  0  0  0  0  0  0  0  0  0  0  0  0
##   Median    0  0  0  0  0  0  7 27 28  9  1 28  0  0  0  0  0  0  0  0  0  0  0
##   Medianl2  0  0  0  0  0  0  7 36 24  4  1 28  0  0  0  0  0  0  0  0  0  0  0
##   AIC       0  0  0  0  0  0  0 10 55 25  8  2  0  0  0  0  0  0  0  0  0  0  0
##   BIC       0  0  0  0  0  0  0 14 62 18  6  0  0  0  0  0  0  0  0  0  0  0  0
##   RIFT.hc   0  0  0  0  1 12 33 42 12  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   shc       0  0  0  0  0  0  0  0  0  0  0  2  3  7 10 15 12 21 13  8  7  1  1
## [1] "(dimension, delta) = (8,200)"
##           ESTIMATE
## METHOD     -1  1  3  4  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 27
##   Mean      0  2  1  1  3  0  2  5 56 25  5  0  0  0  0  0  0  0  0  0  0  0  0  0
##   Meanl2    0  2  1  1  3  0  2  5 56 25  5  0  0  0  0  0  0  0  0  0  0  0  0  0
##   Median    0  0  0  0  0  0  0  5 26 17  2  3 47  0  0  0  0  0  0  0  0  0  0  0
##   Medianl2  0  0  0  0  0  0  0  5 27 17  1  3 47  0  0  0  0  0  0  0  0  0  0  0
##   AIC       0  0  0  0  0  0  0  0 14 53 26  4  3  0  0  0  0  0  0  0  0  0  0  0
##   BIC       0  0  0  0  0  0  0  0 14 60 22  4  0  0  0  0  0  0  0  0  0  0  0  0
##   RIFT.hc   1  0  0  0  0  1  6 30 49 13  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   shc       0  0  0  0  0  0  0  0  0  0  0  0  3  1 10 12 12 12 19 11  9  7  3  1

#stop cluster (parallel computing)
stopCluster(cl)

print(distribution_name )

## [1] "True distribution components: t-distrbution (df=3) mixture distribution"
```

```
print(paste(k, 'true clusters:'))

## [1] "10 true clusters:"

print('Cluster weights:')

## [1] "Cluster weights:"

print(w)

##  [1] 0.02857143 0.05714286 0.11428571 0.11428571 0.11428571 0.11428571 0.11428571
##  [8] 0.11428571 0.11428571 0.11428571
```

The R session information (including the OS info, R version and all packages used):

```
sessionInfo()

## R version 4.1.2 (2021-11-01)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Monterey 12.0.1
##
## Matrix products: default
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_GB.UTF-8/en_GB.UTF-8/en_GB.UTF-8/C/en_GB.UTF-8/en_GB.UTF-8
##
## attached base packages:
## [1] grid      parallel  stats     graphics  grDevices utils     datasets  methods
## [9] base
##
## other attached packages:
##  [1] knitr_1.37         sigclust_1.1.0     mixtools_1.2.0     gridExtra_2.3
##  [5] ggplot2_3.3.5      MASS_7.3-54        pracma_2.3.6       mclust_5.4.9
##  [9] sigclust2_1.2.4    rstudioapi_0.13    mvnfast_0.2.7      doParallel_1.0.16
## [13] iterators_1.0.13   foreach_1.5.1      Rfast_2.0.6        RcppZiggurat_0.1.6
## [17] Rcpp_1.0.8         pacman_0.5.1
##
## loaded via a namespace (and not attached):
##   [1] colorspace_2.0-2       ellipsis_0.3.2         dynamicTreeCut_1.63-1
##   [4] rprojroot_2.0.3        htmlTable_2.4.0        XVector_0.34.0
##   [7] base64enc_0.1-3        ggdendro_0.1.23        fs_1.5.2
##  [10] remotes_2.4.2          bit64_4.0.5            AnnotationDbi_1.56.2
##  [13] fansi_0.5.0            codetools_0.2-18       splines_4.1.2
##  [16] cachem_1.0.6           impute_1.68.0          pkgload_1.2.4
##  [19] Formula_1.2-4          WGCNA_1.70-3           cluster_2.1.2
##  [22] kernlab_0.9-29         GO.db_3.14.0           png_0.1-7
##  [25] compiler_4.1.2         httr_1.4.2             backports_1.4.1
##  [28] Matrix_1.3-4           fastmap_1.1.0          cli_3.3.0
##  [31] htmltools_0.5.2        prettyunits_1.1.1      tools_4.1.2
##  [34] gtable_0.3.0           glue_1.6.1             GenomeInfoDbData_1.2.7
##  [37] dplyr_1.0.7            ggthemes_4.2.4         Biobase_2.54.0
##  [40] vctrs_0.4.1            Biostrings_2.62.0      preprocessCore_1.56.0
##  [43] xfun_0.30              fastcluster_1.2.3      stringr_1.4.0
##  [46] ps_1.7.0               brio_1.1.3             testthat_3.1.4
```

```
##  [49] lifecycle_1.0.1        devtools_2.4.3         zlibbioc_1.40.0
##  [52] scales_1.1.1           RColorBrewer_1.1-2     memoise_2.0.1
##  [55] rpart_4.1-15           segmented_1.3-4        latticeExtra_0.6-29
##  [58] stringi_1.7.6          RSQLite_2.2.10         highr_0.9
##  [61] S4Vectors_0.32.3       desc_1.4.1             checkmate_2.0.0
##  [64] BiocGenerics_0.40.0    pkgbuild_1.3.1         GenomeInfoDb_1.30.1
##  [67] rlang_1.0.2            pkgconfig_2.0.3        matrixStats_0.61.0
##  [70] bitops_1.0-7           evaluate_0.15          lattice_0.20-45
##  [73] purrr_0.3.4            htmlwidgets_1.5.4      bit_4.0.4
##  [76] tidyselect_1.1.1       processx_3.5.3         magrittr_2.0.2
##  [79] R6_2.5.1               IRanges_2.28.0         generics_0.1.1
##  [82] Hmisc_4.6-0            DBI_1.1.2              pillar_1.6.4
##  [85] foreign_0.8-81         withr_2.4.3            survival_3.2-13
##  [88] KEGGREST_1.34.0        RCurl_1.98-1.6         nnet_7.3-16
##  [91] tibble_3.1.6           crayon_1.4.2           utf8_1.2.2
##  [94] jpeg_0.1-9             usethis_2.1.6          data.table_1.14.2
##  [97] blob_1.2.2             callr_3.7.0            digest_0.6.29
## [100] stats4_4.1.2           munsell_0.5.0          sessioninfo_1.2.2
```

```
Sys.time()
```

```
## [1] "2022-06-08 21:44:55 BST"
```