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")
```

```
## mixtools package, version 1.2.0, Released 2020-02-05
## This package is based upon work supported by the National Science Foundation under Grant
No.  SES-0518772.
##
## Attaching package:  'mixtools'
## The following object is masked from 'package:grid':
##
##     depth
## The following object is masked from 'package:mclust':
##
##     dmvnorm
## The following objects are masked from 'package:Rfast':
##
##     dmvnorm, rmvnorm
##
## Attaching package:  'sigclust'
## The following object is masked from 'package:sigclust2':
##
##     sigclust
```

```
# 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 = mus,sigma = sigmas,w = w,df = 3)
  }
}else{
  distribution_name = 'True distribution: Normal mixture distribution'
  samplefunc <- function(n, mu, sigma, w){
  rmixn(n=n, mu=mus, sigma=sigmas, w=w)
}
```

```r
}


# formulating d, delta (dimension and distance between clusters)
if (k == 10){
  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 if (k == 3){
  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) {
  # simulate data
  set.seed(18 + iteration)
  data = samplefunc(n=n, mu=mus, sigma=sigmas, 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
```

```r
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]



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


  #mus = zeros(k, d)
  #mus[1,1] = delta
  #mus[2,2] = -delta
  #mus[3,2] = delta
  mus = outer(rep.int(1L, k), seq.int(d)) + delta * seq.int(0, k - 1L)

  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) = (2,1)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  8 10
##   Mean     52 47  1  0  0  0  0  0
##   Meanl2   58 42  0  0  0  0  0  0
##   Median    0 80 13  3  0  2  1  1
##   Medianl2  1 79 13  3  0  2  1  1
##   AIC       2 57 36  4  1  0  0  0
##   BIC       6 81 13  0  0  0  0  0
##   RIFT.hc   5 74 19  1  1  0  0  0
##   shc      97  2  1  0  0  0  0  0
## [1] "(dimension, delta) = (2,2)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  8  9 10 11
##   Mean     60 26  9  5  0  0  0  0  0  0  0
##   Meanl2   61 24 10  5  0  0  0  0  0  0  0
##   Median    3 29 32 23  6  3  2  1  1  0  0
##   Medianl2  4 30 33 21  6  2  2  1  1  0  0
##   AIC       5 11 25 50  7  1  1  0  0  0  0
##   BIC      26 30 27 17  0  0  0  0  0  0  0
##   RIFT.hc  12 42 38  7  1  0  0  0  0  0  0
##   shc       9 53 20  4  1  3  3  0  3  3  1
## [1] "(dimension, delta) = (2,3)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  9 10
##   Mean     21 14 18 47  0  0  0  0  0
##   Meanl2   23 14 16 47  0  0  0  0  0
##   Median    1  3 21 62  9  0  3  0  1
##   Medianl2  1  3 22 61  9  0  3  0  1
##   AIC       0  0  1 91  8  0  0  0  0
##   BIC       3  1  2 93  1  0  0  0  0
##   RIFT.hc  38 28 16 15  3  0  0  0  0
##   shc       0  4 83 10  0  2  0  1  0
## [1] "(dimension, delta) = (2,4)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  8  9 14
##   Mean      5  7 37 51  0  0  0  0  0
##   Meanl2    6  8 36 50  0  0  0  0  0
##   Median    0  0 17 66 14  1  0  1  1
##   Medianl2  0  0 18 66 13  1  0  1  1
##   AIC       0  0  2 78 16  2  1  1  0
##   BIC       0  0  8 88  4  0  0  0  0
##   RIFT.hc  33 31 16 18  2  0  0  0  0
##   shc       0  0 94  6  0  0  0  0  0
## [1] "(dimension, delta) = (2,5)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  9 10 11 12
##   Mean      4  1 35 59  1  0  0  0  0  0  0
##   Meanl2    4  1 35 59  1  0  0  0  0  0  0
##   Median    0  0 12 70 11  0  2  1  1  1  2
##   Medianl2  0  0 13 71  9  0  3  1  1  1  1
##   AIC       0  0  3 62 29  5  1  0  0  0  0
##   BIC       0  0  8 83  8  1  0  0  0  0  0
##   RIFT.hc  20 38 15 20  6  1  0  0  0  0  0
##   shc       0  0 95  5  0  0  0  0  0  0  0
## [1] "(dimension, delta) = (2,6)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  8 11 12
##   Mean      3  0 41 50  5  1  0  0  0  0
##   Meanl2    3  0 42 51  4  0  0  0  0  0
##   Median    0  0  5 70 17  4  2  1  1  0
##   Medianl2  0  0  6 71 16  4  1  0  1  1
##   AIC       0  0  2 56 34  7  1  0  0  0
##   BIC       0  0  9 78 13  0  0  0  0  0
##   RIFT.hc   8 42  5 32 12  1  0  0  0  0
##   shc       0  0 96  4  0  0  0  0  0  0
## [1] "(dimension, delta) = (2,7)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  8  9 10 11
##   Mean      2  1 41 50  5  1  0  0  0  0  0
##   Meanl2    2  1 42 49  5  1  0  0  0  0  0
##   Median    0  0  4 65 19  8  1  0  1  1  1
##   Medianl2  0  0  4 69 17  6  1  0  1  1  1
##   AIC       0  0  1 58 28  9  2  1  0  0  1
##   BIC       0  0  9 75 15  1  0  0  0  0  0
##   RIFT.hc   5 28 11 43 10  3  0  0  0  0  0
##   shc       0  0 96  4  0  0  0  0  0  0  0
## [1] "(dimension, delta) = (2,8)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  8 10 11 12
##   Mean      2  1 45 44  6  2  0  0  0  0  0
##   Meanl2    2  1 47 43  5  2  0  0  0  0  0
##   Median    0  0  1 65 25  4  1  1  1  1  1
##   Medianl2  0  0  1 68 23  3  1  1  1  1  1
##   AIC       0  0  4 60 21 11  3  0  0  1  0
##   BIC       0  0  8 74 16  2  0  0  0  0  0
##   RIFT.hc   2 21  9 47 17  4  0  0  0  0  0
##   shc       0  0 96  4  0  0  0  0  0  0  0
## [1] "(dimension, delta) = (2,9)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  9 11
##   Mean      2  0 43 48  2  5  0  0  0
##   Meanl2    2  0 45 47  3  3  0  0  0
##   Median    0  0  0 64 24  9  1  1  1
##   Medianl2  0  0  1 64 25  7  1  1  1
##   AIC       0  0  2 53 25 18  2  0  0
##   BIC       0  0 11 72 12  5  0  0  0
##   RIFT.hc    1 21  6 52 15  5  0  0  0
##   shc        0  0 96  4  0  0  0  0  0
## [1] "(dimension, delta) = (8,1)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 29
##   Mean     21 28 12  9 23  7  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   Meanl2   21 28 12  9 23  7  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   Median    1 31 48 16  3  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   Medianl2  1 31 48 16  3  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   AIC       0  0  9 16 43 25  5  0  1  0  1  0  0  0  0  0  0  0  0  0  0
##   BIC       0  0 58 24 12  4  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0
##   RIFT.hc   1  1 78 20  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   shc       0  0  0  3  8  9 13  8  7  6  9 12  9  5  2  3  1  2  1  1  1
## [1] "(dimension, delta) = (8,2)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  8  9 10 11 12 15
##   Mean     15  4  2 32 25 16  5  1  0  0  0  0  0
##   Meanl2   15  4  2 32 25 16  5  1  0  0  0  0  0
##   Median    0  0  1 81 13  5  0  0  0  0  0  0  0
##   Medianl2  0  0  1 81 13  5  0  0  0  0  0  0  0
##   AIC       0  0  0 16 16 44 20  3  1  0  0  0  0
##   BIC       0  0  0 62 18 18  1  0  1  0  0  0  0
##   RIFT.hc   1 59 16 14  8  2  0  0  0  0  0  0  0
##   shc       0  0  2  6 11 20 20 21  9  3  4  3  1
## [1] "(dimension, delta) = (8,3)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  8  9 10 12
##   Mean     12  0  5 13 31 26 12  1  0  0  0
##   Meanl2   12  0  5 13 31 26 12  1  0  0  0
##   Median    0  0  0 74 18  8  0  0  0  0  0
##   Medianl2  0  0  0 74 18  8  0  0  0  0  0
##   AIC       0  0  0  4 22 26 37 11  0  0  0
##   BIC       0  0  0 43 30 17  7  3  0  0  0
##   RIFT.hc   2 33 22 30 11  2  0  0  0  0  0
##   shc       0  0  1 10 17 23 18 14 10  6  1
## [1] "(dimension, delta) = (8,4)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  8  9 10 11 12
##   Mean     12  0  5  8 24 31 16  4  0  0  0  0
##   Meanl2   12  0  5  8 24 31 16  4  0  0  0  0
##   Median    0  0  0 72 17  9  2  0  0  0  0  0
##   Medianl2  0  0  0 72 17  9  2  0  0  0  0  0
##   AIC       0  0  0  4 12 22 45 16  0  1  0  0
##   BIC       0  0  0 25 29 29 14  3  0  0  0  0
##   RIFT.hc   6 17 12 30 26  9  0  0  0  0  0  0
##   shc       0  0  3  8 22 25 22 13  5  0  1  1
## [1] "(dimension, delta) = (8,5)"
##           ESTIMATE
## METHOD      1  2  3  4  5  6  7  8  9 10 11
##   Mean     11  0  4  8 20 26 29  2  0  0  0
##   Meanl2   11  0  4  8 20 26 29  2  0  0  0
##   Median    0  0  0 53 32 12  3  0  0  0  0
##   Medianl2  0  0  0 53 32 12  3  0  0  0  0
##   AIC       0  0  0  2  9 18 60  9  2  0  0
##   BIC        0  0  0 18 30 25 27  0  0  0  0
##   RIFT.hc    7 24  4 37 14 13  1  0  0  0  0
##   shc        0  0  4 11 21 22 19 17  2  3  1
## [1] "(dimension, delta) = (8,6)"
##           ESTIMATE
## METHOD       1  2  3  4  5  6  7  8  9 10
##   Mean      10  0  7  5 17 37 21  2  1  0
##   Meanl2    10  0  7  5 17 37 21  2  1  0
##   Median     0  0  0 38 44 16  2  0  0  0
##   Medianl2   0  0  0 38 44 16  2  0  0  0
##   AIC        0  0  0  5  5 19 58 11  1  1
##   BIC        0  0  0  9 24 37 28  2  0  0
##   RIFT.hc    6 25  1 47 10 11  0  0  0  0
##   shc        0  0  2 11 24 24 21 12  4  2
## [1] "(dimension, delta) = (8,7)"
##           ESTIMATE
## METHOD       1  2  3  4  5  6  7  8  9 10 11
##   Mean      10  0  7  4 17 28 31  2  1  0  0
##   Meanl2    10  0  7  4 17 28 31  2  1  0  0
##   Median     0  0  1 29 39 27  4  0  0  0  0
##   Medianl2   0  0  1 30 38 27  4  0  0  0  0
##   AIC        0  0  0  3  3 18 59 14  3  0  0
##   BIC        0  0  0  8 16 39 35  2  0  0  0
##   RIFT.hc   11 20  0 58  7  4  0  0  0  0  0
##   shc        0  0  4 10 24 18 29  8  4  2  1
## [1] "(dimension, delta) = (8,8)"
##           ESTIMATE
## METHOD       1  2  3  4  5  6  7  8  9 10 11
##   Mean      10  0  8  5  9 35 29  3  1  0  0
##   Meanl2    10  0  8  5  9 35 29  3  1  0  0
##   Median     0  0  1 27 33 34  5  0  0  0  0
##   Medianl2   0  0  1 27 35 32  5  0  0  0  0
##   AIC        0  0  0  2  2 19 52 18  5  2  0
##   BIC        0  0  0  6 11 39 42  2  0  0  0
##   RIFT.hc   14 17  0 58  8  3  0  0  0  0  0
##   shc        0  0  3 12 21 20 28  8  4  3  1
## [1] "(dimension, delta) = (8,9)"
##           ESTIMATE
## METHOD       1  2  3  4  5  6  7  8  9 10 11
##   Mean      10  0  9  3  7 37 30  3  1  0  0
##   Meanl2    10  0  9  3  7 37 30  3  1  0  0
##   Median     0  0  0 26 29 39  6  0  0  0  0
##   Medianl2   0  0  0 26 31 39  4  0  0  0  0
##   AIC        0  0  0  1  3 21 55 17  3  0  0
##   BIC        0  0  0  1  9 36 50  4  0  0  0
##   RIFT.hc   12 15  0 65  6  2  0  0  0  0  0
##   shc        0  0  3 14 20 25 20 10  3  4  1
```

```r
#stop cluster (parallel computing)
stopCluster(cl)

print(distribution_name )
```

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

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

## [1] "3 true clusters:"

print('Cluster weights:')

## [1] "Cluster weights:"

print(w)

## [1] 0.3333333 0.3333333 0.3333333
```

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

```r
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] sigclust_1.1.0     mixtools_1.2.0     gridExtra_2.3      ggplot2_3.3.5
##  [5] MASS_7.3-54        pracma_2.3.6       mclust_5.4.9       sigclust2_1.2.4
##  [9] mvnfast_0.2.7      doParallel_1.0.16  iterators_1.0.13   foreach_1.5.1
## [13] Rfast_2.0.6        RcppZiggurat_0.1.6 Rcpp_1.0.8         rstudioapi_0.13
## [17] knitr_1.37         pacman_0.5.1
##
## loaded via a namespace (and not attached):
##  [1] segmented_1.3-4        bitops_1.0-7           matrixStats_0.61.0
##  [4] bit64_4.0.5            RColorBrewer_1.1-2     httr_1.4.2
##  [7] GenomeInfoDb_1.30.1    dynamicTreeCut_1.63-1  tools_4.1.2
## [10] backports_1.4.1        utf8_1.2.2             R6_2.5.1
## [13] rpart_4.1-15           Hmisc_4.6-0            DBI_1.1.2
## [16] BiocGenerics_0.40.0    colorspace_2.0-2       nnet_7.3-16
## [19] withr_2.4.3            tidyselect_1.1.1       bit_4.0.4
## [22] compiler_4.1.2         preprocessCore_1.56.0  WGCNA_1.70-3
## [25] cli_3.2.0              Biobase_2.54.0         htmlTable_2.4.0
## [28] ggdendro_0.1.23        scales_1.1.1           checkmate_2.0.0
## [31] stringr_1.4.0          digest_0.6.29          foreign_0.8-81
## [34] XVector_0.34.0         base64enc_0.1-3        jpeg_0.1-9
## [37] pkgconfig_2.0.3        htmltools_0.5.2        fastmap_1.1.0
## [40] highr_0.9              ggthemes_4.2.4         htmlwidgets_1.5.4
## [43] rlang_1.0.2            RSQLite_2.2.10         impute_1.68.0
## [46] generics_0.1.1         dplyr_1.0.7            RCurl_1.98-1.6
## [49] magrittr_2.0.2         GO.db_3.14.0           GenomeInfoDbData_1.2.7
## [52] Formula_1.2-4          Matrix_1.3-4           munsell_0.5.0
## [55] S4Vectors_0.32.3       fansi_0.5.0            lifecycle_1.0.1
## [58] stringi_1.7.6          zlibbioc_1.40.0        blob_1.2.2
## [61] crayon_1.4.2           lattice_0.20-45        Biostrings_2.62.0
## [64] splines_4.1.2          KEGGREST_1.34.0        pillar_1.6.4
## [67] fastcluster_1.2.3      codetools_0.2-18       stats4_4.1.2
## [70] glue_1.6.1             evaluate_0.15          latticeExtra_0.6-29
## [73] data.table_1.14.2      png_0.1-7              vctrs_0.4.1
## [76] gtable_0.3.0           purrr_0.3.4            kernlab_0.9-29
## [79] cachem_1.0.6           xfun_0.30              survival_3.2-13
## [82] tibble_3.1.6           AnnotationDbi_1.56.2   memoise_2.0.1
## [85] IRanges_2.28.0         cluster_2.1.2          ellipsis_0.3.2
```

```
Sys.time()
```

```
## [1] "2022-06-03 11:40:30 BST"
```