Correction of co-clustering practice session

Vincent Vandewalle

2024-12-02

1. Simulate data from the latent block model

Parameters of the model in the case of continuous data

```
# Parameters of the model
prop.r <- c(0.5, 0.5) # proportions for rows
prop.c <- c(0.5, 0.5) # proportions for columns
a <- vector("list", 2)
a[[1]][[1]] <- list(mu = 0, var = 1) # mean and variance for the first row and column block
a[[1]][[2]] <- list(mu = 0.5, var = 1) # mean and variance for the first row and second column block
a[[2]][[1]] <- list(mu = -0.5, var = 1) # mean and variance for the second row and first column block
a[[2]][[2]] <- list(mu = -0.25, var = 1) # mean and variance for the second row and second column block

# Specific parameter for co-clustering (mu, sigma)
param <- list(prop.r = prop.r, prop.c = prop.c, a = a)
# param
```

Function to simulated data

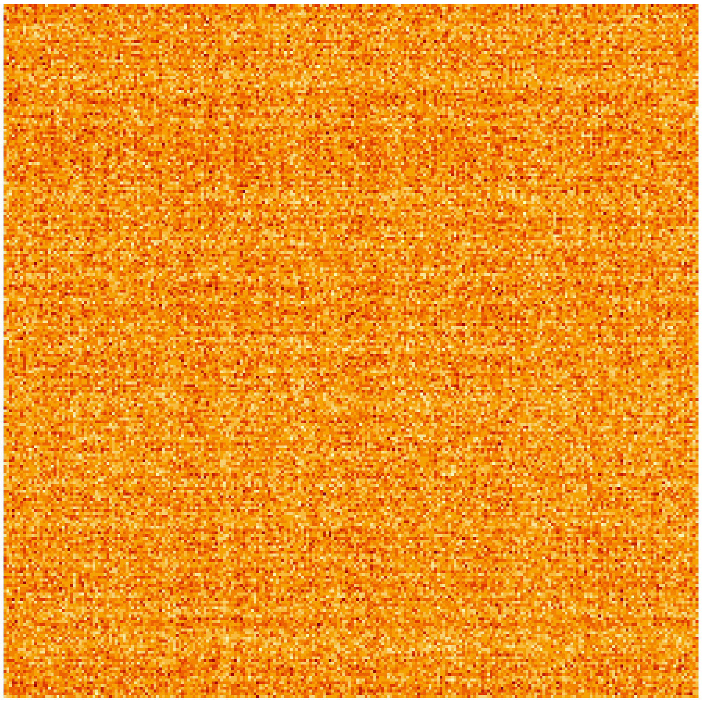
- x: data
- z: row partition
- w: column partition

The strategy is first to simulate the row and the column partition, then to simulate each entry of the data matrix given its row and its column.

```
# Fonction which simulates data from the latent block model
# param: parameters of the model
# n: number of rows
# d: number of columns
# return: list with data matrix x, row partition z and column partition w
simulate_data <- function(param, n, d) {
  K <- length(param$prop.r)
  G <- length(param$prop.c)
  z <- sample(1:K, size = n, replace = TRUE, prob = param$prop.r)
  w <- sample(1:G, size = d, replace = TRUE, prob = param$prop.c)
  x <- matrix(0, n, d)
  for (k in 1:K) {
    for (g in 1:G) {
      rows <- z == k
      cols <- w == g
      # sample data from block (k, g) with it specific mean and variance
      x[rows, cols] <- rnorm(sum(rows) * sum(cols),
                             mean = param$a[[k]][[g]]$mu,
                             sd = sqrt(param$a[[k]][[g]]$var))
    }
  }
  return(list(x = x, z = z, w = w))
}
simulated_data <- simulate_data(param, 250, 250)
```

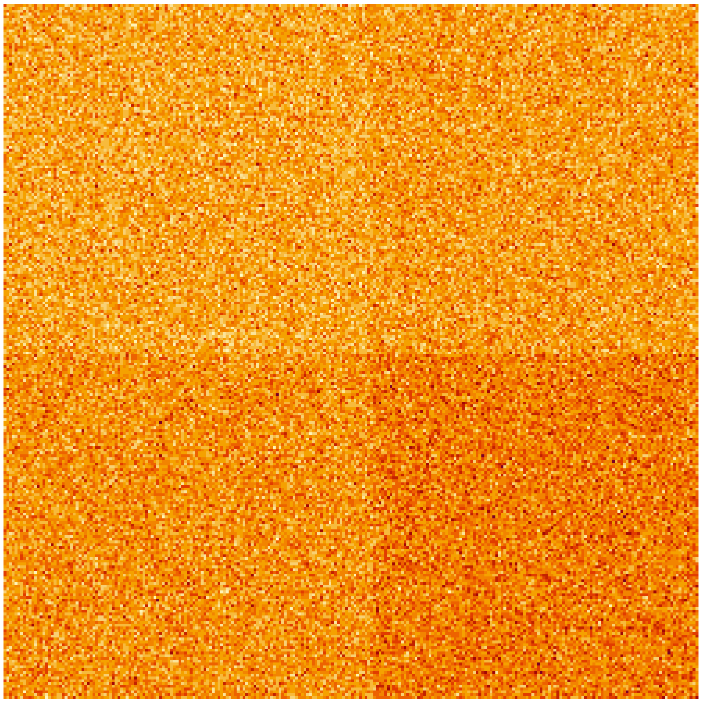
The data matrix can be plotted as follows:

```
heatmap(simulated_data$x,Colv = NA, Rowv = NA, labRow = NA , labCol = NA, scale = "none")
```



After reorderring the rows and the columns by cluster we get the following heatmap

```
heatmap(simulated_data$x[order(simulated_data$z),order(simulated_data$w)],
        Colv = NA, Rowv = NA, labRow = NA , labCol = NA, scale = "none")
```



where we see the cluster structure.

2. Estimation based on hand crafted SEM algorithm

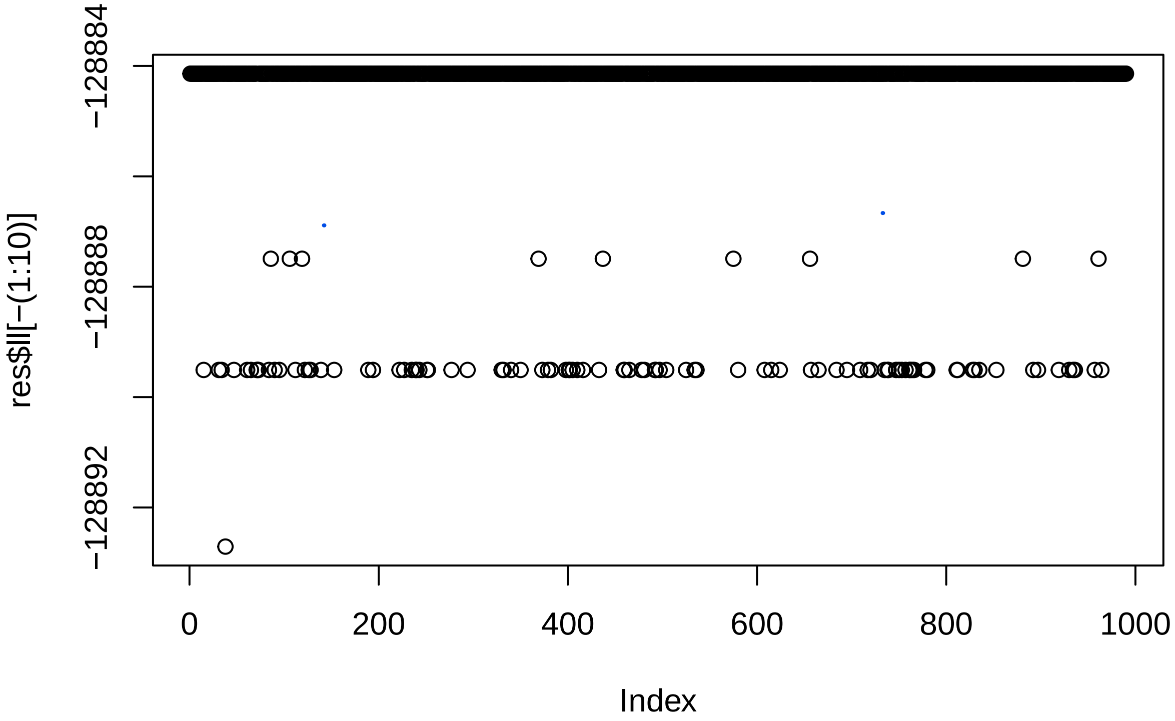
Definition of functions for parameters estimation

```
# Calculation of row group probabilities given column groups (default),
# or vice versa (used for the stochastic sampling step)
# x: data matrix
# w: column partition
# param: parameters of the model
# row: if TRUE, the function calculates the row group probabilities given column groups, otherwise the
# return: matrix of posterior probabilities
posterior <- function(x, w, param, row = TRUE) {
  n <- nrow(x)
  d <- ncol(x)
  K <- length(param$prop.r)
  G <- length(param$prop.c)
  # Rows' groups with fixed column groups
  if (row) {
    # lnf: joint log-likelihood of each row with fixed value of the column cluster and considering diff
    lnf <- matrix(0, n, K)
    for (k in 1:K) {
      for (g in 1:G) {
        data <- x[, w == g, drop = FALSE]
        lnf[, k] <- lnf[, k] + rowSums(dnorm(data, param$a[[k]][[g]]$mu,
                                              sqrt(param$a[[k]][[g]]$var), log = TRUE))
      }
      lnf[, k] <- lnf[, k] + log(param$prop.r[k])
    }
  }
  # Columns' groups with fixed row groups
  else {
    z <- w # for more coherence
    lnf <- matrix(0, d, G)
    for (g in 1:G) {
      for (k in 1:K) {
        data <- x[z == k, , drop = FALSE]
        lnf[, g] <- lnf[, g] + colSums(dnorm(data, param$a[[k]][[g]]$mu,
                                             sqrt(param$a[[k]][[g]]$var), log = TRUE))
      }
      lnf[, g] <- lnf[, g] + log(param$prop.c[g])
    }
  }
  # Calculation of posterior probabilities
  t <- prop.table(exp(sweep(lnf, 1, apply(lnf, 1, max), "-")), 1)
  return(t)
}

# Mstep
mstep <- function(x, z, w) {
  K <- max(z)
  G <- max(w)
  param <- NULL
  param$prop.r <- prop.table(table(z))
  param$prop.c <- prop.table(table(w))
  param$a <- vector("list", K)
  for (k in 1:K) {
    param$a[[k]] <- vector("list", G)
    for (g in 1:G) {
      data <- as.vector(x[z == k, w == g])
      param$a[[k]][[g]]$mu <- mean(data)
      param$a[[k]][[g]]$var <- var(data) * (length(data) - 1) / length(data)
    }
  }
  return(param)
}


# Computation of the completed log-likelihood
loglikelihood <- function(x, z, w, param) {
  K <- length(param$prop.r)
  G <- length(param$prop.c)
  ll <- 0
  for (k in 1:K) {
    for (g in 1:G) {
      data <- as.vector(x[z == k, w == g])
      ll <- ll + sum(dnorm(data, param$a[[k]][[g]]$var,
                           sqrt(param$a[[k]][[g]]$var), log = TRUE))
    }
  }
  return(ll)
}
```

```r
# Beware of the disappearance of modality
# Version S1 - M1 - S2 - M2 (another possible version S1 - S2 - M)
# Possibility of semi-supervised learning by fixing the row or column partition
sem <- function(x, z, w, param, niter = 100, missz = TRUE, missw = TRUE) {
  ll <- NULL
  for (i in 1:niter) {
    K <- length(param$prop.r)
    G <- length(param$prop.c)
    if (missz) {
      tik <- posterior(x, w, param)
      if (any(is.na(tik))) {
        break
      }
      # Sampling of the row partition
      z <- apply(tik, 1, function(y) sample(K, 1, prob = y))
      z <- as.numeric(as.factor(z)) # Handling modalities disappearance
    }
    # Update of the parameters given the row partition and the column partition
    param <- mstep(x, z, w)
    if (missw) {
      sjg <- posterior(x, z, param, row = FALSE)
      if (any(is.na(sjg))) {
        break
      }
      w <- apply(sjg, 1, function(y) sample(G, 1, prob = y))
      w <- as.numeric(as.factor(w)) # Handling modalities disappearance
    }
    param <- mstep(x, z, w)
    # Completion of log-likelihood tracking: useful for monitoring the convergence of the algorithm
    ll <- c(ll, loglikelihood(x, z, w, param))
  }
  return(list(z = z, w = w, param = param, ll = ll))
}

# Initialisation by random partition
init <- function(x, K, G) {
  n <- nrow(x)
  d <- ncol(x)
  z <- sample(K, n, replace = TRUE)
  w <- sample(G, d, replace = TRUE)
  param <- mstep(x, z, w)
  return(list(z = z, w = w, param = param))
}
```

Parameters estimation by SEM

```r
init.val <- init(simulated_data$x, 2, 2)
res <- sem(simulated_data$x, init.val$z, init.val$w, init.val$param, 1000)
plot(res$ll[-(1:10)]) # Minitoring of the completed log-likelihood
```



```r
res$param # Estimated parameters
```

```
## $prop.r
## z
##     1     2
## 0.504 0.496
##
## $prop.c
## w
##     1     2
## 0.532 0.468
##
## $a
## $a[[1]]
## $a[[1]][[1]]
## $a[[1]][[1]]$mu
## [1] -0.4938501
##
## $a[[1]][[1]]$var
## [1] 1.011374
##
##
## $a[[1]][[2]]
## $a[[1]][[2]]$mu
## [1] -0.255893
##
## $a[[1]][[2]]$var
## [1] 0.9861257
##
##
##
```

```
## $a[[2]]
## $a[[2]][[1]]
## $a[[2]][[1]]$mu
## [1] -0.0004345198
##
## $a[[2]][[1]]$var
## [1] 0.9838473
##
##
## $a[[2]][[2]]
## $a[[2]][[2]]$mu
## [1] 0.5017202
##
## $a[[2]][[2]]$var
## [1] 1.009523
```

Comparison of the partitions

```
table(simulated_data$z , res$z)
```

```
##
##       1   2
##   1   0 124
##   2 126   0
```

```
table(simulated_data$w , res$w)
```

```
##
##       1   2
##   1 133   0
##   2   0 117
```

Thus the true rows and columns partitions are exactly recovered!

The comparison can also be automatically performed by using the Adjusted Rand Index which is invariant up to label permutation.

```
library("mclust")
```

```
## Package 'mclust' version 6.1
## Type 'citation("mclust")' for citing this R package in publications.
```

```
##
## Attaching package: 'mclust'
```

```
## The following object is masked _by_ '.GlobalEnv':
##
##     mstep
```

```
adjustedRandIndex(simulated_data$z , res$z)
```

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

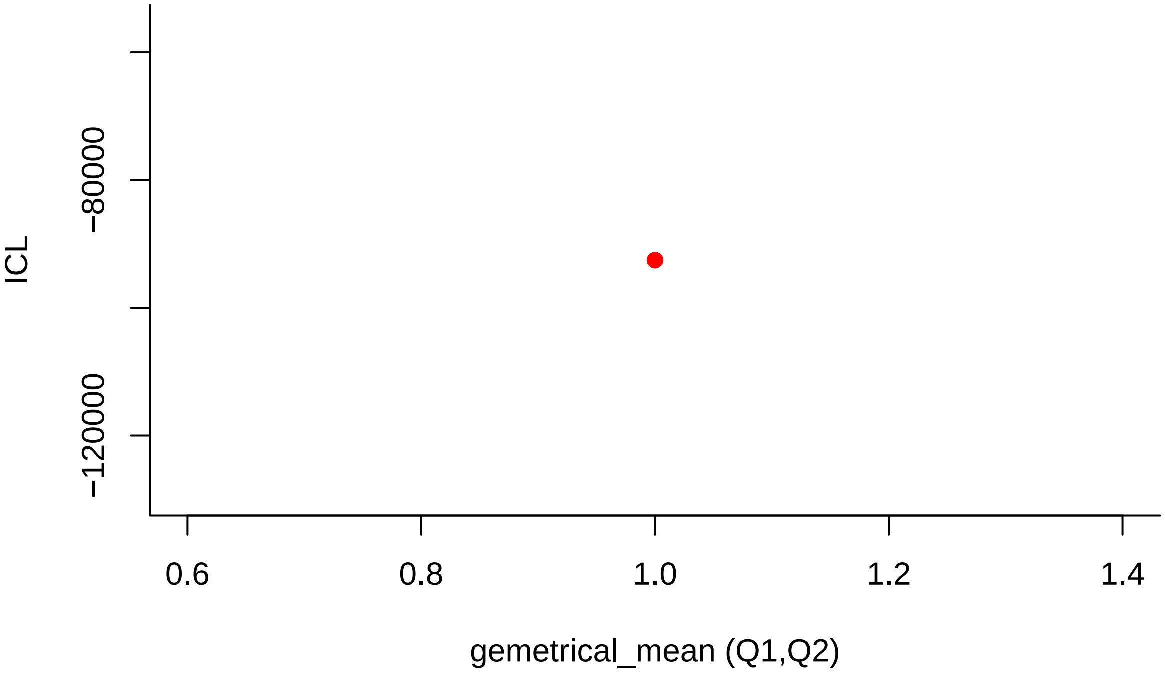
```
adjustedRandIndex(simulated_data$w , res$w)
```

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

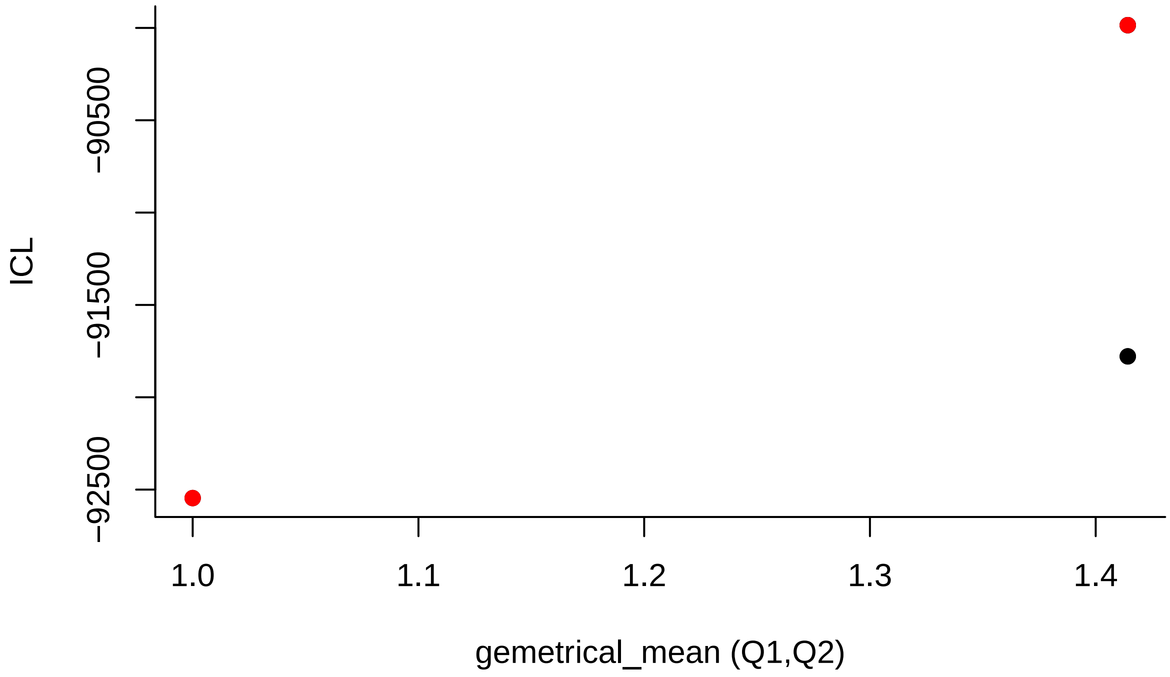
3. Test on the package `blockmodels`

```
# install.packages("blockmodels")
library(blockmodels)
bm_model = BM_gaussian(membership_type = "LBM", adj = simulated_data$x)
bm_model$estimate()
```

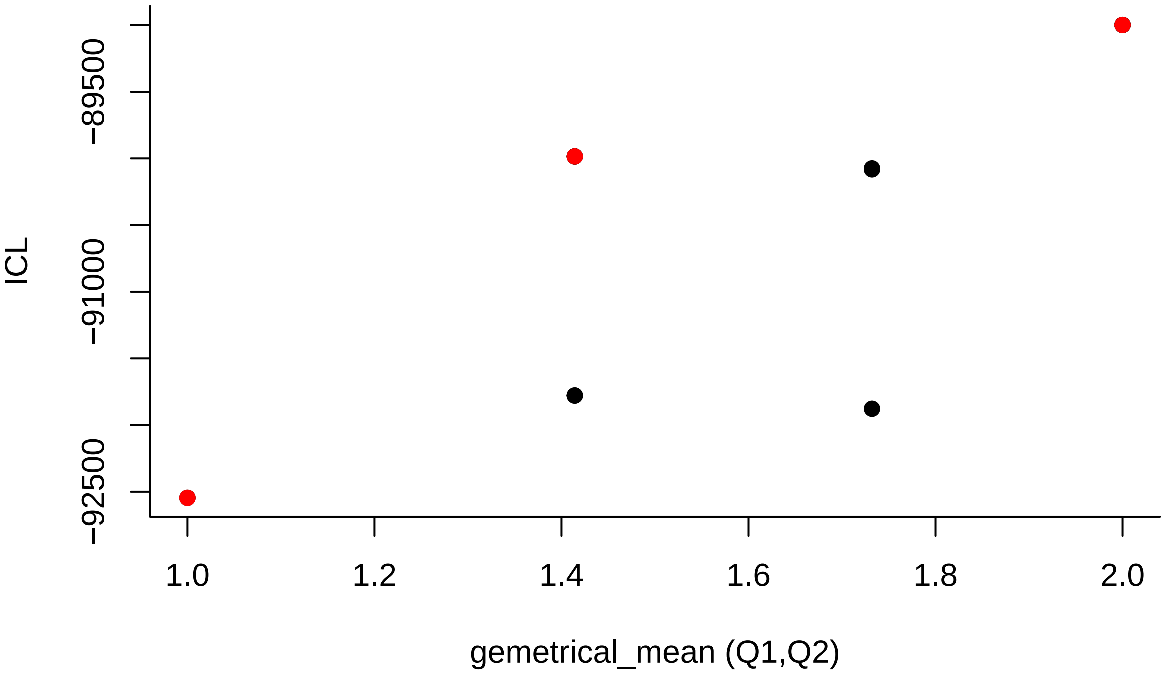
```
## -> Estimation for 2 groups (1+1)
##                 -> 1 initializations provided
##                 -> 0 initializations already used
##             -> Estimation with 1 initializations
## Executing 1 jobs in parallel                                          -> Better ICL criterion found
##                     -> new ICL: -92545.9328147575
##                     -> old ICL: NA
##                 -> 1 row groups, 1 col groups
```



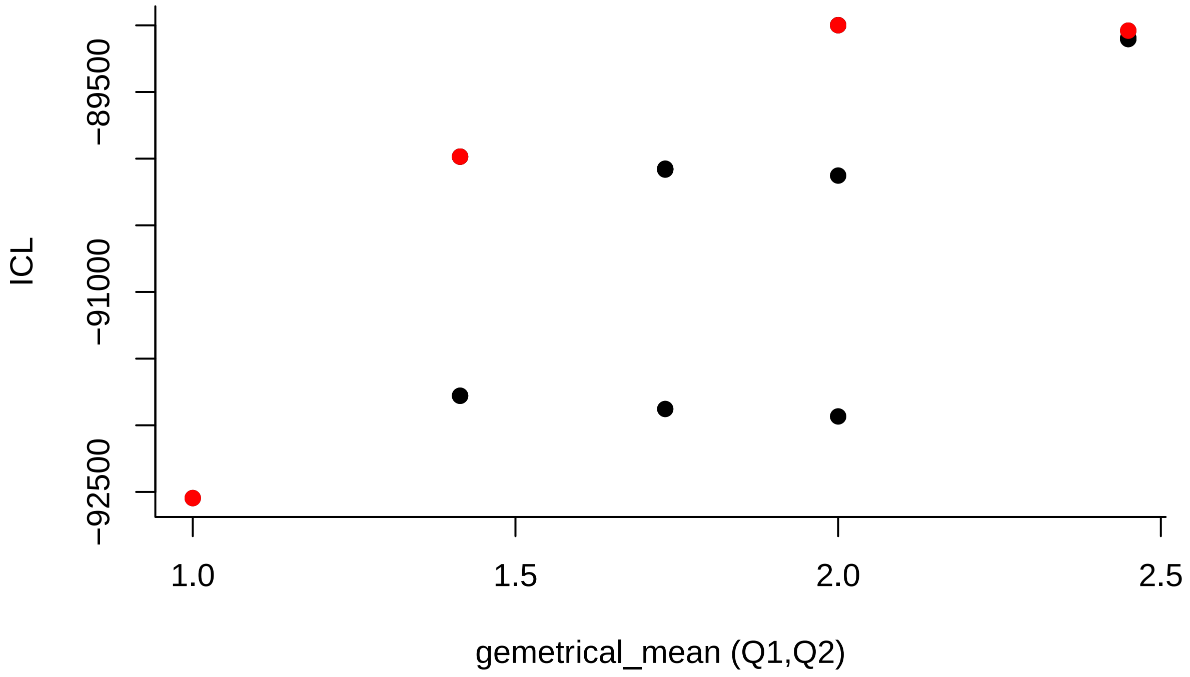
```
## -> Computation of eigen decomposition used for initalizations
##     -> for rows
##     -> for cols
##
## -> Pass 1
##     -> With ascending number of groups
##         -> For 3 groups
##             -> Selecting initialization
##                 -> Init from spectral clustering
##                 -> Init from splitting groups from 2 groups
##                 -> 4 initializations provided
##                 -> 0 initializations already used
##             -> Estimation with 4 initializations
## Executing 4 jobs in parallel                                          -> Better ICL criterion found
##                     -> new ICL: -89985.1876467787
##                     -> old ICL: NA
##                 -> 2 row groups, 1 col groups
```



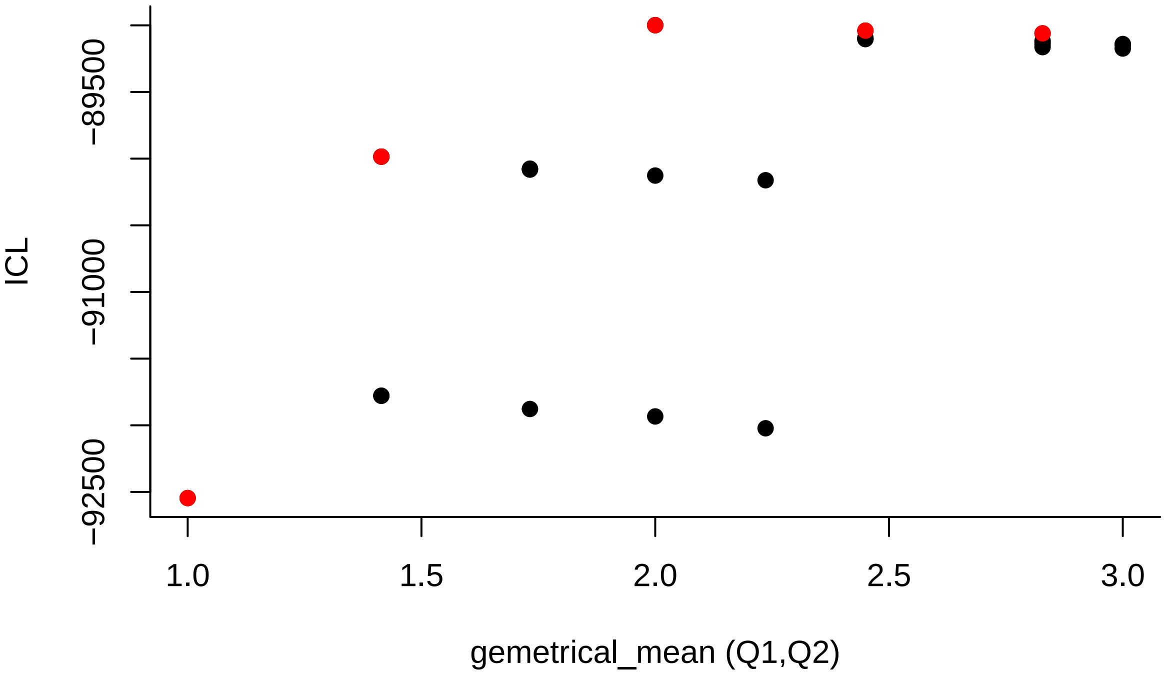
```
##         -> For 4 groups
##             -> Selecting initialization
##                 -> Init from spectral clustering
##                 -> Init from splitting groups from 3 groups
##                 -> 6 initializations provided
##                 -> 0 initializations already used
##             -> Estimation with 6 initializations
## Executing 6 jobs in parallel                                    -> Better ICL criterion foun
##                     -> new ICL: -88998.8949265216
##                     -> old ICL: NA
##                 -> 2 row groups, 2 col groups
```



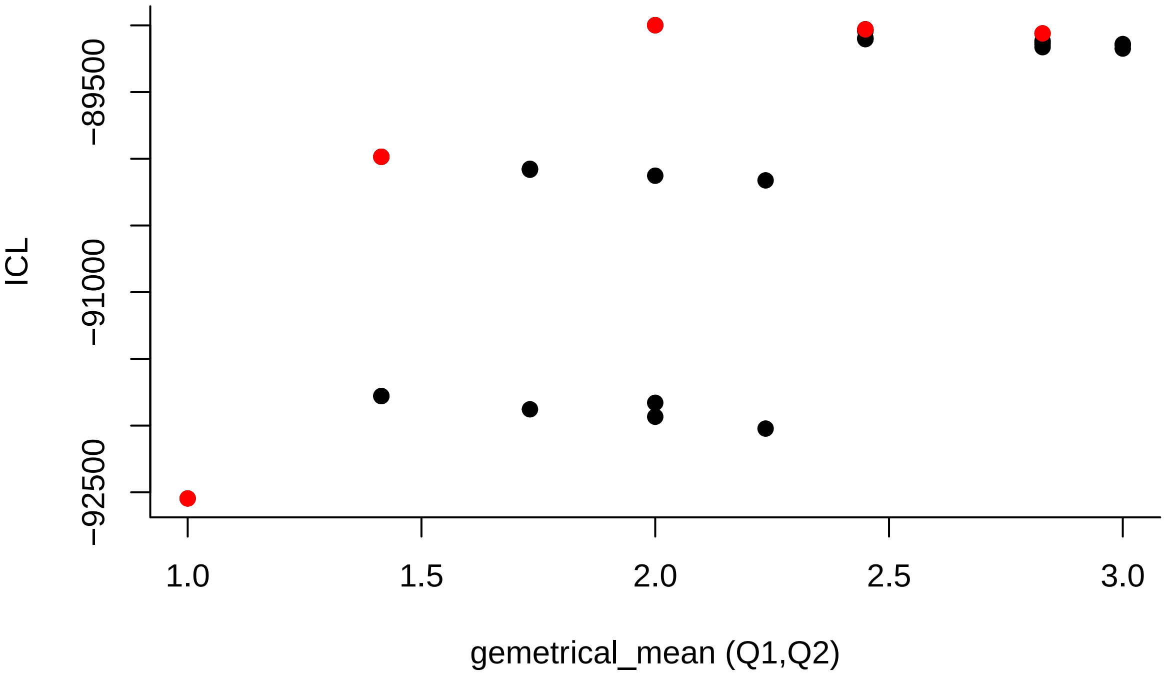
```
##          -> For 5 groups
##              -> Selecting initialization
##                  -> Init from spectral clustering
##                  -> Init from splitting groups from 4 groups
##                  -> 8 initializations provided
##                  -> 0 initializations already used
##              -> Estimation with 8 initializations
## Executing 8 jobs in parallel                                  -> Better ICL criterion foun
##                      -> new ICL: -89039.9681614792
##                      -> old ICL: NA
##                  -> 2 row groups, 3 col groups
```



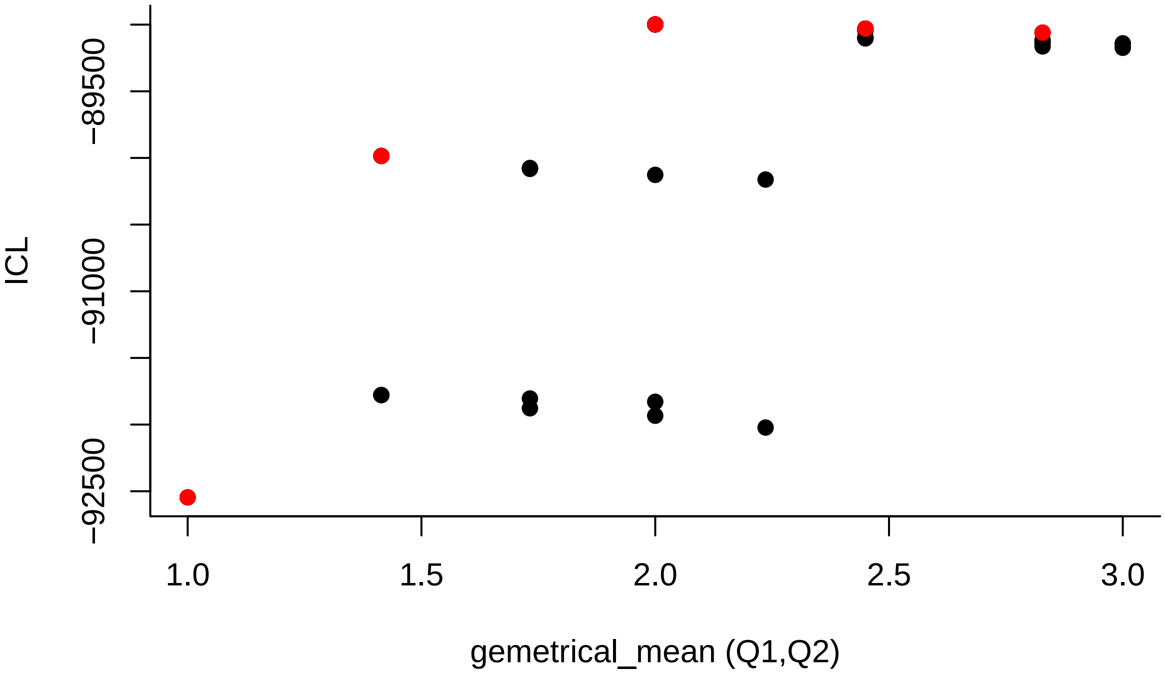
```
##         -> For 6 groups
##             -> Selecting initialization
##                 -> Init from spectral clustering
##                 -> Init from splitting groups from 5 groups
##                 -> 10 initializations provided
##                 -> 0 initializations already used
##             -> Estimation with 10 initializations
## Executing 10 jobs in parallel                                              -> Better ICL criterion fou
##                     -> new ICL: -89060.248320396
##                     -> old ICL: NA
##                 -> 2 row groups, 4 col groups
```



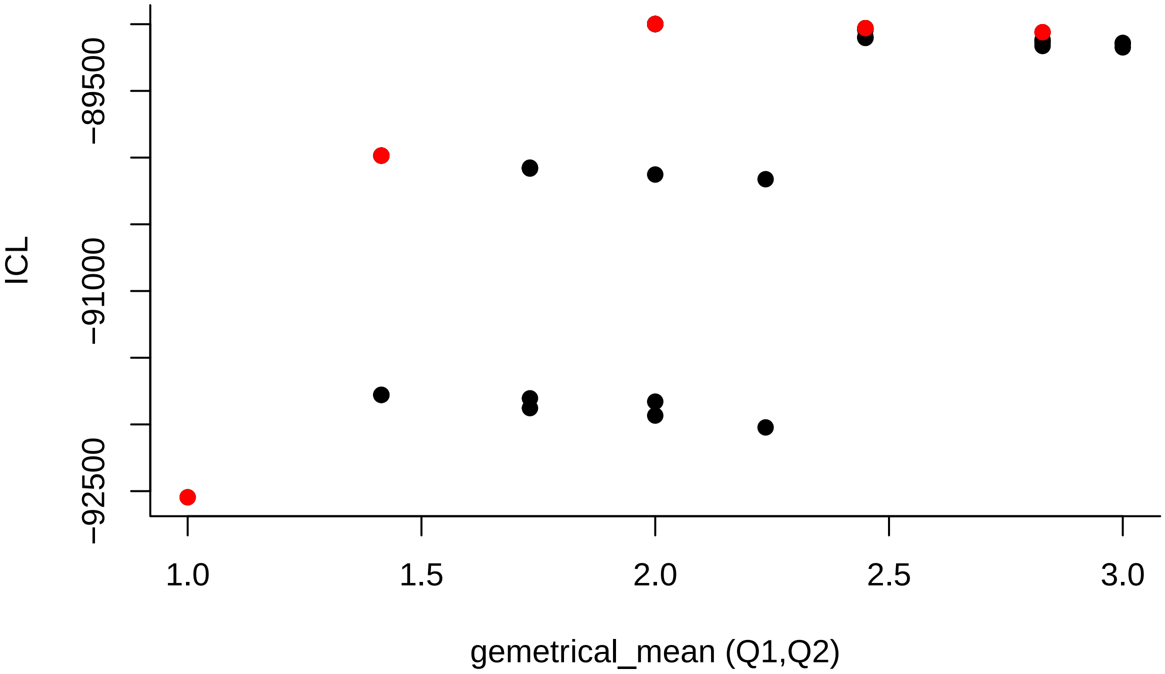
```
##     -> With descending number of groups
##         -> For 5 groups
##             -> Selecting intializations
##                 -> Init from merging groups from 6 groups
##                 -> 7 initializations provided
##                 -> 0 initializations already used
##             -> Estimation with 7 initializations
## Executing 7 jobs in parallel                                            -> Better ICL criterion foun
##                     -> new ICL: -89029.9610351145
##                     -> old ICL: -89039.9681614792
##                 -> 2 row groups, 3 col groups
```



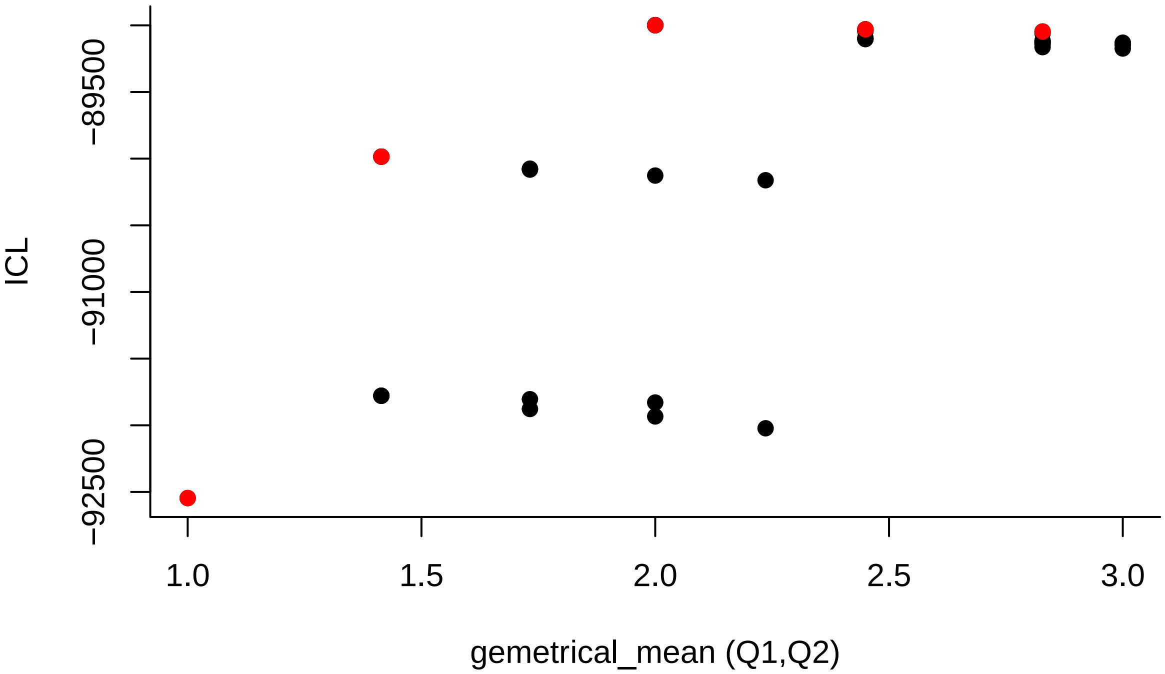
```
##         -> For 4 groups
##             -> Selecting intializations
##                 -> Init from merging groups from 5 groups
##                 -> 4 initializations provided
##                 -> 0 initializations already used
##             -> Estimation with 4 initializations
## Executing 4 jobs in parallel                                            -> Useless, no better ICL cri
##                     -> better ICL found: -88998.9013931815
##                     -> old ICL: -88998.8949265216
```



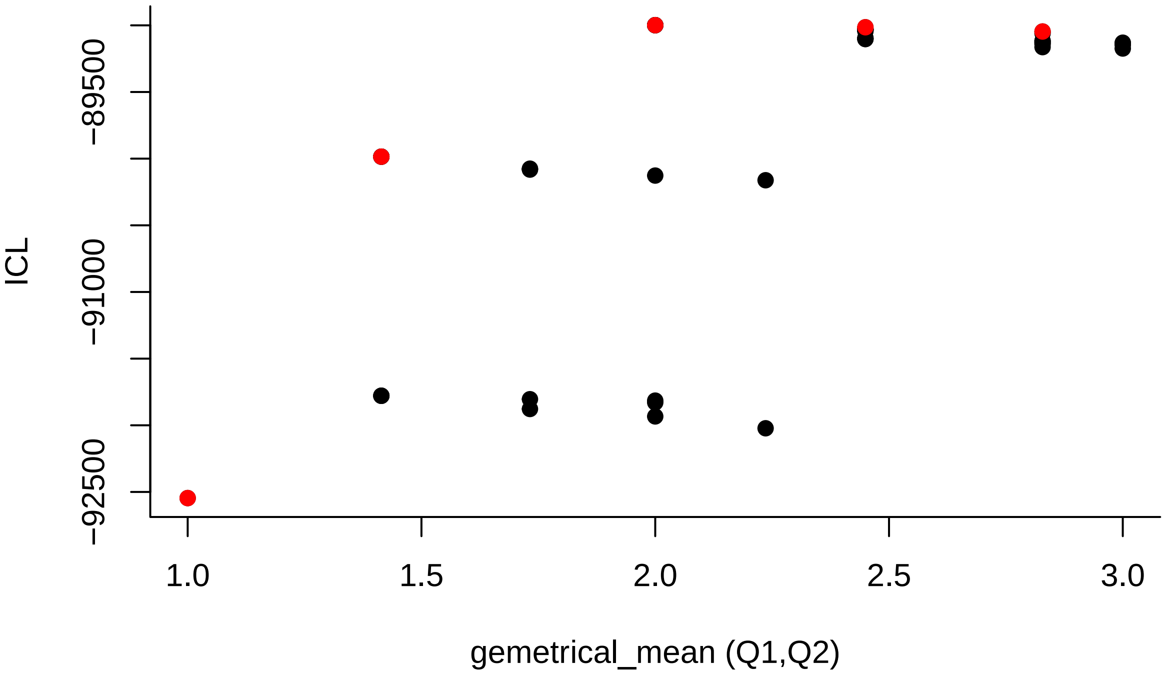
```
##         -> For 3 groups
##             -> Selecting intializations
##                 -> Init from merging groups from 4 groups
##                 -> 2 initializations provided
##                 -> 0 initializations already used
##             -> Estimation with 2 initializations
## Executing 2 jobs in parallel                                            -> Useless, no better ICL cri
##                     -> better ICL found: -89985.1876467787
##                     -> old ICL: -89985.1876467787
```



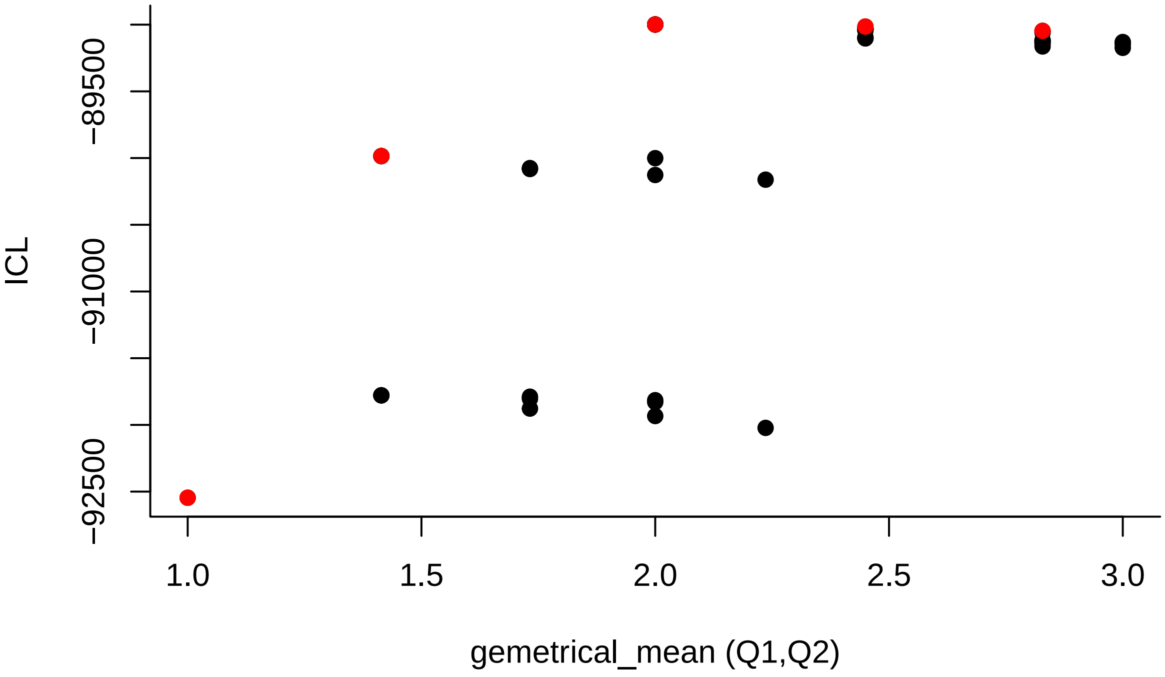
```
## -> Pass 2
##     -> With ascending number of groups
##         -> For 3 groups
##             -> Selecting initialization
##                 -> Init from splitting groups from 2 groups
##             -> already done
##         -> For 4 groups
##             -> Selecting initialization
##                 -> Init from splitting groups from 3 groups
##             -> already done
##         -> For 5 groups
##             -> Selecting initialization
##                 -> Init from splitting groups from 4 groups
##             -> already done
##         -> For 6 groups
##             -> Selecting initialization
##                 -> Init from splitting groups from 5 groups
##                 -> 5 initializations provided
##                 -> 0 initializations already used
##             -> Estimation with 5 initializations
## Executing 5 jobs in parallel                                            -> Better ICL criterion foun
##                     -> new ICL: -89046.4636557518
##                     -> old ICL: -89060.248320396
##                 -> 2 row groups, 4 col groups
```



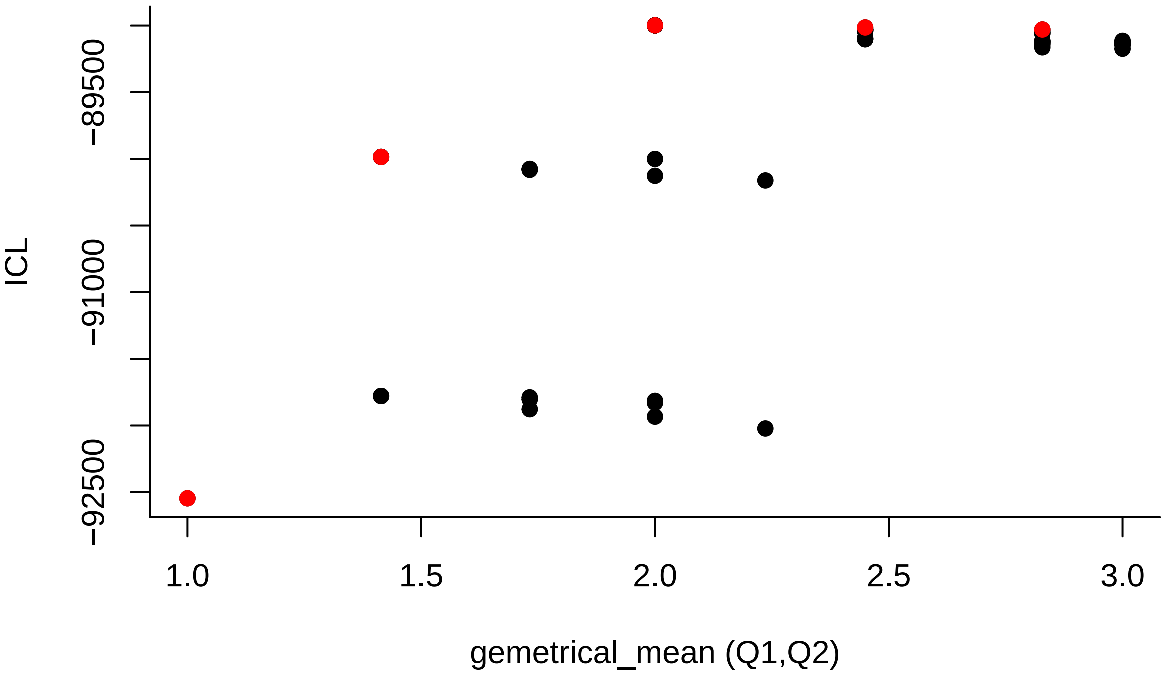
```
##     -> With descending number of groups
##         -> For 5 groups
##             -> Selecting intializations
##                 -> Init from merging groups from 6 groups
##                 -> 7 initializations provided
##                 -> 1 initializations already used
##             -> Estimation with 6 initializations
## Executing 6 jobs in parallel                                            -> Better ICL criterion foun
##                     -> new ICL: -89014.5741334993
##                     -> old ICL: -89029.9610351145
##                 -> 2 row groups, 3 col groups
```



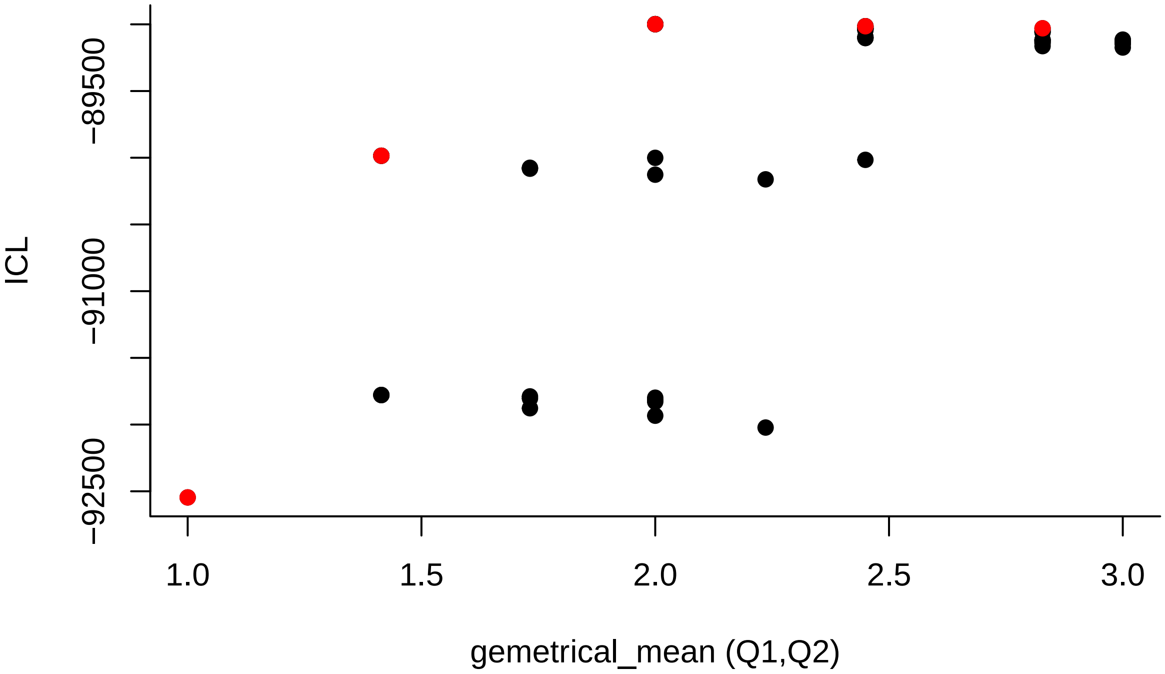
```
##         -> For 4 groups
##             -> Selecting intializations
##                 -> Init from merging groups from 5 groups
##                 -> 4 initializations provided
##                 -> 1 initializations already used
##             -> Estimation with 3 initializations
## Executing 3 jobs in parallel                                        -> Useless, no better ICL cri
##                     -> better ICL found: -88998.9014457485
##                     -> old ICL: -88998.8949265216
```



```
##         -> For 3 groups
##              -> Selecting intializations
##                  -> Init from merging groups from 4 groups
##              -> Already done
## -> Pass 3
##      -> With ascending number of groups
##          -> For 3 groups
##              -> Selecting initialization
##                  -> Init from splitting groups from 2 groups
##              -> already done
##          -> For 4 groups
##              -> Selecting initialization
##                  -> Init from splitting groups from 3 groups
##              -> already done
##          -> For 5 groups
##              -> Selecting initialization
##                  -> Init from splitting groups from 4 groups
##              -> already done
##          -> For 6 groups
##              -> Selecting initialization
##                  -> Init from splitting groups from 5 groups
##                  -> 5 initializations provided
##                  -> 0 initializations already used
##              -> Estimation with 5 initializations
## Executing 5 jobs in parallel                                  -> Better ICL criterion found
##                      -> new ICL: -89029.8642616346
##                      -> old ICL: -89046.4636557518
##                  -> 2 row groups, 4 col groups
```



```
##      -> With descending number of groups
##          -> For 5 groups
##              -> Selecting intializations
##                  -> Init from merging groups from 6 groups
##                  -> 7 initializations provided
##                  -> 2 initializations already used
##              -> Estimation with 5 initializations
## Executing 5 jobs in parallel                                     -> Better ICL criterion foun
##                      -> new ICL: -89014.4811877352
##                      -> old ICL: -89014.5741334993
##                  -> 2 row groups, 3 col groups
```

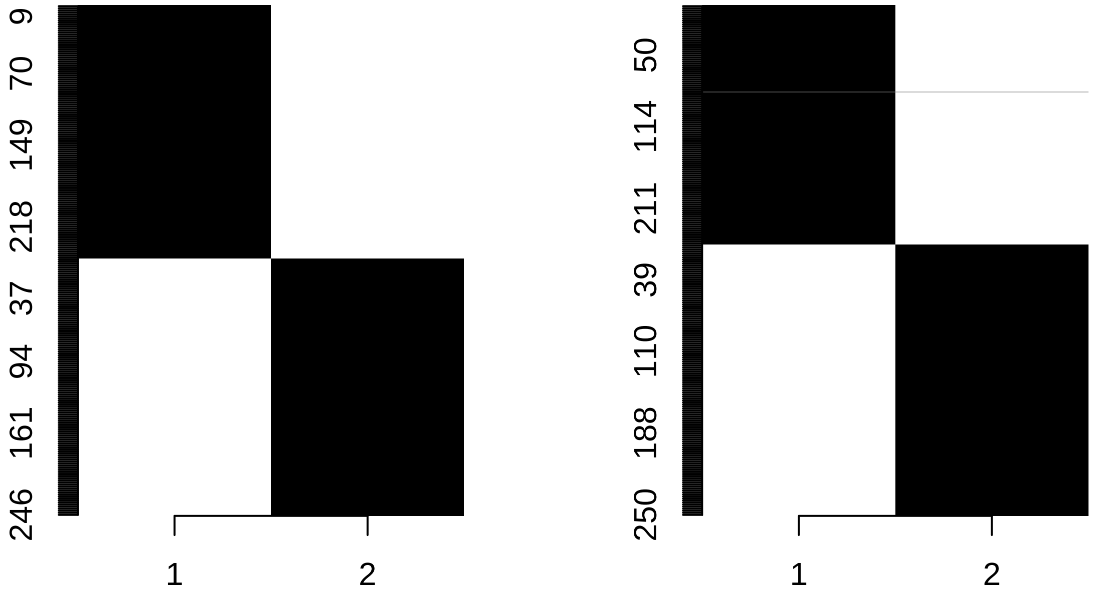


```
##          -> For 4 groups
##              -> Selecting intializations
##                  -> Init from merging groups from 5 groups
##              -> Already done
##          -> For 3 groups
##              -> Selecting intializations
##                  -> Init from merging groups from 4 groups
##              -> Already done
## -> Pass 4
##      -> With ascending number of groups
##          -> For 3 groups
##              -> Selecting initialization
##                  -> Init from splitting groups from 2 groups
##              -> already done
##          -> For 4 groups
##              -> Selecting initialization
##                  -> Init from splitting groups from 3 groups
##              -> already done
##          -> For 5 groups
##              -> Selecting initialization
##                  -> Init from splitting groups from 4 groups
##              -> already done
##          -> For 6 groups
##              -> Selecting initialization
##                  -> Init from splitting groups from 5 groups
##              -> already done
##     -> With descending number of groups
##         -> For 5 groups
##             -> Selecting intializations
##                 -> Init from merging groups from 6 groups
##             -> Already done
##         -> For 4 groups
##             -> Selecting intializations
##                 -> Init from merging groups from 5 groups
##             -> Already done
##         -> For 3 groups
##             -> Selecting intializations
##                 -> Init from merging groups from 4 groups
##             -> Already done
```

```
which.max(bm_model$ICL)
```

```
## [1] 4
```

```
bm_model$memberships[[4]]$plot()
```



```
bm_model$model_parameters
```

```
## [[1]]
## NULL
##
## [[2]]
## [[2]]$n_parameters
## [1] 2
##
## [[2]]$mu
##              [,1]
## [1,] -0.07642439
##
## [[2]]$sigma2
## [1] 1.131155
##
##
```

```
## [[3]]
## [[3]]$n_parameters
## [1] 3
##
## [[3]]$mu
##            [,1]
## [1,]  0.2343231
## [2,] -0.3822433
##
## [[3]]$sigma2
## [1] 1.036123
##
##
## [[4]]
## [[4]]$n_parameters
## [1] 5
##
## [[4]]$mu
##            [,1]          [,2]
## [1,]  0.5014776 -0.0001582536
## [2,] -0.2555951 -0.4934023466
##
## [[4]]$sigma2
## [1] 0.9979564
##
##
## [[5]]
## [[5]]$n_parameters
## [1] 7
##
## [[5]]$mu
##            [,1]          [,2]       [,3]
## [1,]  0.5016485 -0.0001957045  0.2521499
## [2,] -0.2555152 -0.4934241233 -0.3708482
##
## [[5]]$sigma2
## [1] 0.9979605
##
##
## [[6]]
## [[6]]$n_parameters
## [1] 9
##
## [[6]]$mu
##            [,1]          [,2]       [,3]       [,4]
## [1,]  0.5017410 -0.0002225528  0.2508116  0.2482205
## [2,] -0.2554717 -0.4934403908 -0.3763885 -0.3680393
##
## [[6]]$sigma2
## [1] 0.9979692
```