A quick tour of IMIX

Ziqiao Wang

2020/2/3

Contents

Introduction

IMIX is an R package for integration of multiple genomics data types to investigate the associations between genes and a specific outcome, including binary, continuous, survival, and categorical outcomes based on finite multivariate Gaussian mixture modelling using summary statistics. The input is summary statistics for each data type including either p-value or z-score. We support summary statistics of data outputs such as DNA methylation, copy number variation(CNV), and gene expression (RNAseq/microarray) at gene level. Nonethelss, IMIX is as flexible as it can be extended to other molecular level as long as the summary statistics of the multiple data types are coherent with each other. It provides features to select the true number of components behind the data, parameter estimation for the summary statistics via EM algorithm. The most important feature is that it evaluates the data through different covariance and mean structures of mixture modelling and selects the overall best fitting model, which in turn provides the oracle output while controlling for the global false discovery rate (FDR) at a user specified level.

This document gives a quick tour of IMIX functionalities. The tasks addressed in this package include assessment of the true number of components with respect to the data, identification of interesting genes for each data type combination, FDR control, and plotting functionality. See help(package="IMIX") for further details and references provided by citation("IMIX").

Data Preparation

Example data 1: p values for RNAseq and CNV data

Each row is a gene, each column is a data type. The dimension of the input data is 1000×2.

```
data("data_p")
dim(data_p)
```

```
## [1] 1000    2
```

```
head(data_p)
```

```
##             p.rnaseq     p.cnv
## ALOX12B 6.087266e-08 0.2283308
## DNAJC7  1.547188e-05 0.1438505
## S100A8  6.965336e-20 0.4140232
## SNORD8  9.907678e-01 0.3540862
## GSTM5   5.138138e-07 0.9842864
## VSTM2L  1.327199e-02 0.7146840
```

Example data 2: Simulate z scores for DNA methylation, RNAseq, and CNV data

The dimension is 1000×3.

```
library(MASS)
N=1000
truelabel <- sample(1:8,
                    prob = rep(0.125, 8),
                    size = N,
                    replace = TRUE)
mu1 = c(0, 5)
mu2 = c(0, 5)
mu3 = c(0, 5)
mu1_mv = c(mu1[1], mu2[1], mu3[1])
mu2_mv = c(mu1[2], mu2[1], mu3[1])
mu3_mv = c(mu1[1], mu2[2], mu3[1])
mu4_mv = c(mu1[1], mu2[1], mu3[2])
mu5_mv = c(mu1[2], mu2[2], mu3[1])
mu6_mv = c(mu1[2], mu2[1], mu3[2])
mu7_mv = c(mu1[1], mu2[2], mu3[2])
mu8_mv = c(mu1[2], mu2[2], mu3[2])

cov_sim = list()
for (i in 1:8) {
  cov_sim[[i]] = diag(3)
}
data_z = array(0, c(N, 3))
data_z[which(truelabel == 1), ] = mvrnorm(
  n = length(which(truelabel == 1)),
  mu = mu1_mv,
  Sigma = cov_sim[[1]],
  tol = 1e-6,
  empirical = FALSE
```

```
)
data_z[which(truelabel == 2), ] = mvrnorm(
  n = length(which(truelabel == 2)),
  mu = mu2_mv,
  Sigma = cov_sim[[2]],
  tol = 1e-6,
  empirical = FALSE
)
data_z[which(truelabel == 3), ] = mvrnorm(
  n = length(which(truelabel == 3)),
  mu = mu3_mv,
  Sigma = cov_sim[[3]],
  tol = 1e-6,
  empirical = FALSE
)
data_z[which(truelabel == 4), ] = mvrnorm(
  n = length(which(truelabel == 4)),
  mu = mu4_mv,
  Sigma = cov_sim[[4]],
  tol = 1e-6,
  empirical = FALSE
)
data_z[which(truelabel == 5), ] = mvrnorm(
  n = length(which(truelabel == 5)),
  mu = mu5_mv,
  Sigma = cov_sim[[5]],
  tol = 1e-6,
  empirical = FALSE
)
data_z[which(truelabel == 6), ] = mvrnorm(
  n = length(which(truelabel == 6)),
  mu = mu6_mv,
  Sigma = cov_sim[[6]],
  tol = 1e-6,
  empirical = FALSE
)
data_z[which(truelabel == 7), ] = mvrnorm(
  n = length(which(truelabel == 7)),
  mu = mu7_mv,
  Sigma = cov_sim[[7]],
  tol = 1e-6,
  empirical = FALSE
)
data_z[which(truelabel == 8), ] = mvrnorm(
  n = length(which(truelabel == 8)),
  mu = mu8_mv,
  Sigma = cov_sim[[8]],
  tol = 1e-6,
  empirical = FALSE
)

rownames(data_z)=paste0("gene",1:N)
colnames(data_z)=c("z.methy","z.ge","z.cnv")
```

```
dim(data_z)
```

```
## [1] 1000    3
```

Model selection: select the number of components for the mixture distribution

Example 1: Two data types, p values

```
select_comp1=model_selection_component(data_p,data_type = "p",seed=20)
```

```
## Start Number of Component Selections!
## Test for 1 Component!
## Start IMIX-cor-twostep procedure!
## Successfully Done!
## Start IMIX-cor model procedure!
## Successfully Done!
## Test for 2 Components!
## Start IMIX-cor-twostep procedure!
## Successfully Done!
## Start IMIX-cor model procedure!
## Successfully Done!
## Test for 3 Components!
## Start IMIX-cor-twostep procedure!
## Successfully Done!
## Start IMIX-cor model procedure!
## Successfully Done!
## Test for 4 Components!
## Start IMIX-cor-twostep procedure!
## Successfully Done!
## Start IMIX-cor model procedure!
## Successfully Done!
## Done!
```

```
names(select_comp1)
```

```
## [1] "Component_Selected_AIC" "Component_Selected_BIC" "AIC/BIC"
## [4] "IMIX_ind_unrestrict"    "IMIX_cor_twostep"       "IMIX_cor"
```

```
select_comp1$Component_Selected_AIC
```

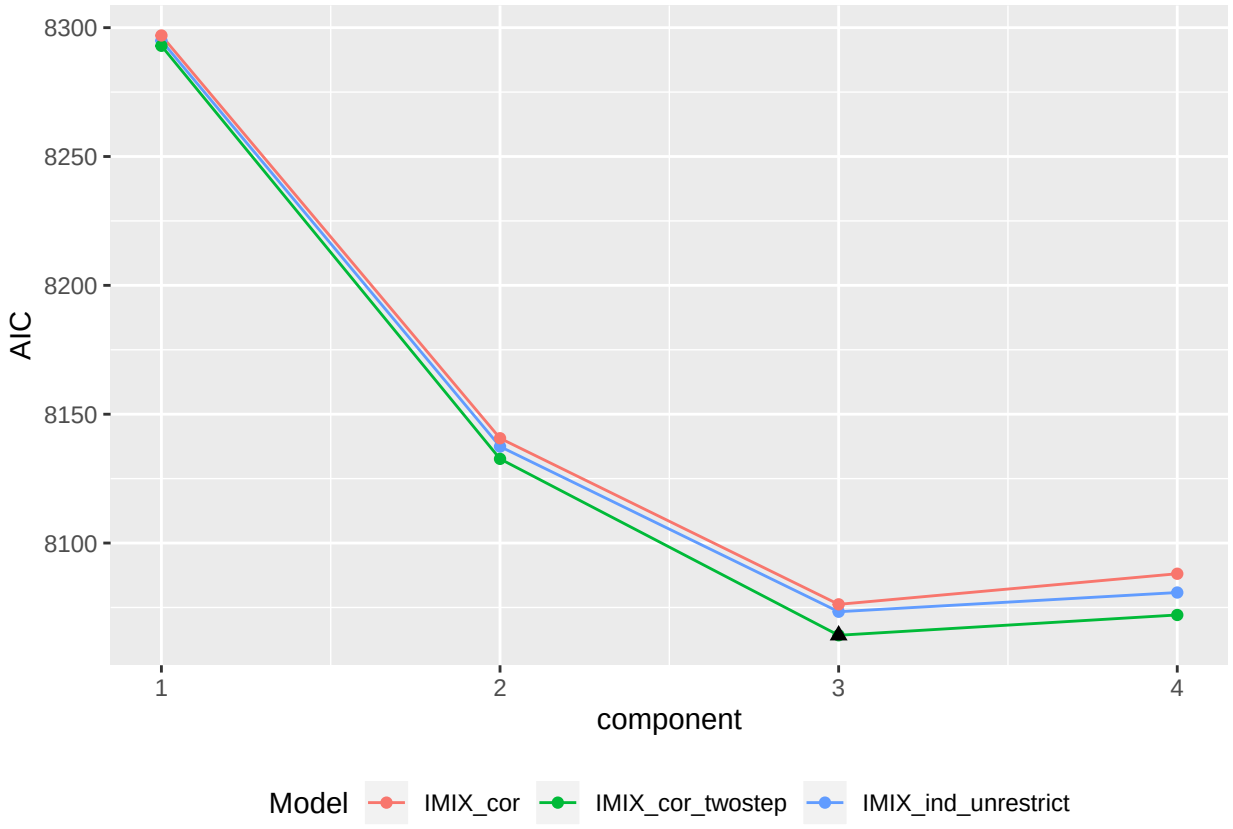
```
## [1] component3
## Levels: component1 component2 component3 component4
```

```
select_comp1$Component_Selected_BIC
```

```
## [1] component3
## Levels: component1 component2 component3 component4
```

The model selected 3 components out of 4. Then we visualize it.

```
plot_component(select_comp1,type="AIC")
```



Example 2: Three data types, z scores

```
select_comp2=model_selection_component(data_z,data_type = "z")
```

```
## Start Number of Component Selections!
## Test for 1 Component!
## Start IMIX-cor-twostep procedure!
## Successfully Done!
## Start IMIX-cor model procedure!
## Successfully Done!
## Test for 2 Components!
## Start IMIX-cor-twostep procedure!
## Successfully Done!
## Start IMIX-cor model procedure!
## Successfully Done!
## Test for 3 Components!
## Start IMIX-cor-twostep procedure!
## Successfully Done!
## Start IMIX-cor model procedure!
```

```
## Successfully Done!
## Test for 4 Components!
## Start IMIX-cor-twostep procedure!
## Successfully Done!
## Start IMIX-cor model procedure!
## Successfully Done!
## Test for 5 Components!
## Start IMIX-cor-twostep procedure!
## Successfully Done!
## Start IMIX-cor model procedure!
## Successfully Done!
## Test for 6 Components!
## Start IMIX-cor-twostep procedure!
## Successfully Done!
## Start IMIX-cor model procedure!
## Successfully Done!
## Test for 7 Components!
## Start IMIX-cor-twostep procedure!
## Successfully Done!
## Start IMIX-cor model procedure!
## Successfully Done!
## Test for 8 Components!
## Start IMIX-cor-twostep procedure!
## Successfully Done!
## Start IMIX-cor model procedure!
## Successfully Done!
## Done!
```

```
names(select_comp2)
```

```
## [1] "Component_Selected_AIC" "Component_Selected_BIC" "AIC/BIC"
## [4] "IMIX_ind_unrestrict"    "IMIX_cor_twostep"       "IMIX_cor"
```

```
select_comp2$Component_Selected_AIC
```

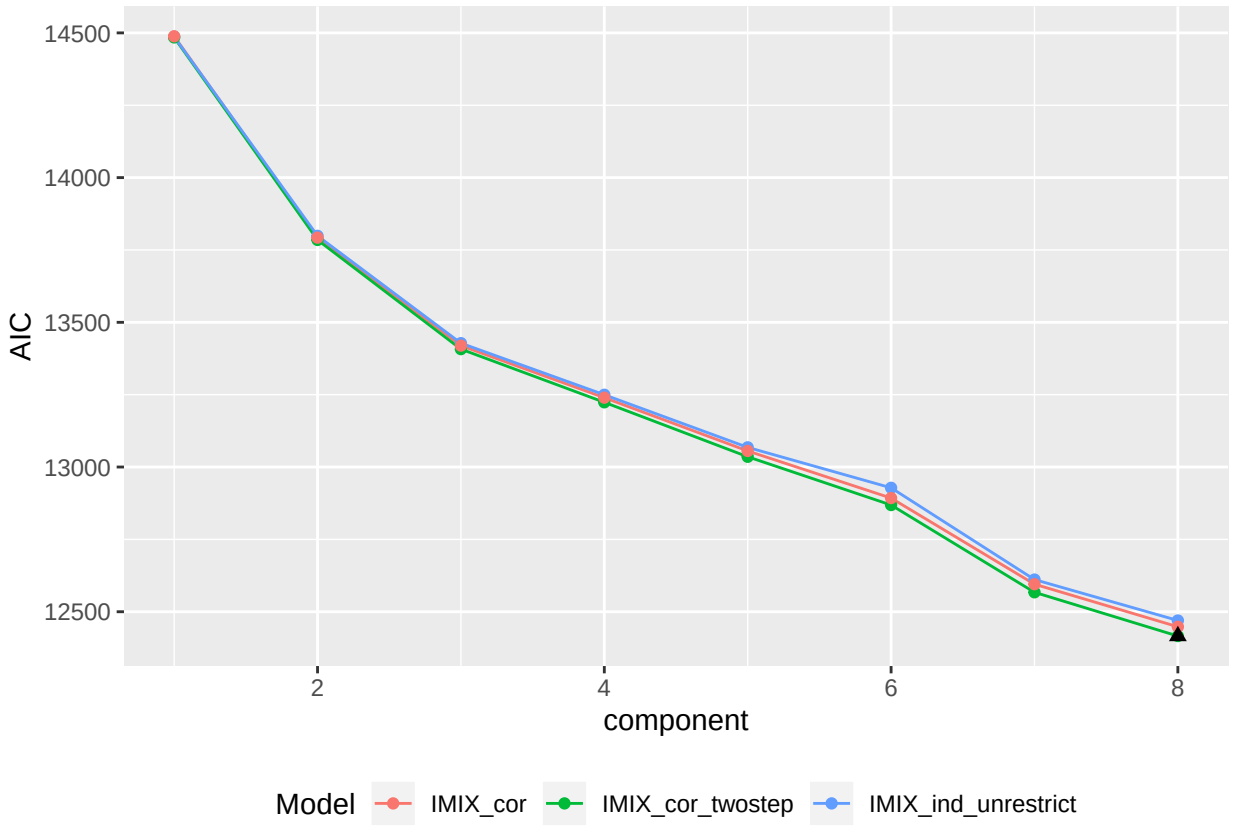
```
## [1] component8
## 8 Levels: component1 component2 component3 component4 ... component8
```

```
select_comp2$Component_Selected_BIC
```

```
## [1] component8
## 8 Levels: component1 component2 component3 component4 ... component8
```

The model selected all 8 components. Then we visualize it.

```
plot_component(select_comp2,type="AIC")
```



Integrative genomics test for two and three omics data types

Example 1: Two data types, p values

Initial values for the p transformed z score mixture model, this step can be skipped

```
mu_input=list() # generate an initial list for mean
mu_input[[1]]=c(0,0) # mean vector for component 1, data 1 null and data 2 null
mu_input[[2]]=c(3,0) # mean vector for component 2, data 1 nonnull and data 2 null
mu_input[[3]]=c(0,3) # mean vector for component 3, data 1 null and data 2 nonnull
mu_input[[4]]=c(3,3) # mean vector for component 4, data 1 nonnull and data 2 nonnull

cov_input=list() # generate an initial list for the covariance matrices
for(i in 1:4){
  cov_input[[i]]=diag(3)
}

p_input=rep(0.25,4) # initial value for the proportion of components
```

Start the test

```
test1=IMIX(data_input=data_p,data_type="p",mu_ini=mu_input,cov_ini=cov_input,p_ini=p_input,alpha=0.1)
```

```
## Start IMIX-ind procedure!
## Successfully Done!
## Start IMIX-cor-twostep procedure!
## Successfully Done!
## Start IMIX-cor model procedure!
## Successfully Done!
## Start IMIX-cor-restrict procedure!
## Successfully Done!
## Start Model Selection!
## Start Adaptive FDR Control!
## All Done!
```

Result outputs of example 1

```
test1$estimatedFDR # Print the estimated global FDR for each component
```

```
## $estimated_mFDR_comp2
## [1] 0.09948009
##
## $estimated_mFDR_comp3
## [1] 0
##
## $estimated_mFDR_comp4
## [1] 0.09991635
```

```
test1$`AIC/BIC` # The AIC and BIC values for each model
```

```
##                        AIC      BIC
## IMIX_ind          8068.002 8121.987
## IMIX_cor_twostep  8067.853 8141.469
## IMIX_cor          8083.875 8196.753
## IMIX_cor_restrict 8075.222 8168.470
```

```
test1$`Selected Model` # The best fitted model selected by AIC
```

```
## [1] "IMIX_cor_twostep"
```

```
str(test1$IMIX_cor_twostep)
```

```
## List of 8
##  $ posterior prob      : num [1:1000, 1:4] 1.03e-03 1.93e-02 6.20e-10 7.28e-01 1.15e-03 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:1000] "ALOX12B" "DNAJC7" "S100A8" "SNORD8" ...
##   .. ..$ : chr [1:4] "component1" "component2" "component3" "component4"
##  $ Full LogLik all     : num [1:29] -52129 -4116 -4028 -4022 -4021 ...
##  $ Full MaxLogLik final: num -4019
##  $ iterations          : num 29
##  $ pi                  : num [1:4] 0.3679 0.58 0.0111 0.041
##  $ mu                  :List of 4
##   ..$ : num [1:2] 0.617 0.475
##   ..$ : num [1:2] 3.758 0.475
##   ..$ : num [1:2] 0.617 4.802
##   ..$ : num [1:2] 3.76 4.8
```

```
##  $ cov                 :List of 4
##   ..$ : num [1:2, 1:2] 1.495 0.133 0.133 1.33
##   ..$ : num [1:2, 1:2] 4.975 -0.152 -0.152 1.367
##   ..$ : num [1:2, 1:2] 3.545 -0.25 -0.25 0.168
##   ..$ : num [1:2, 1:2] 5.967 -0.476 -0.476 1.049
##  $ g                   : num 4
```

```
dim(test1$significant_genes_with_FDRcontrol)
```

```
## [1] 1000    3
```

```
head(test1$significant_genes_with_FDRcontrol)
```

```
##              localFDR class_withoutFDRcontrol class_FDRcontrol
## GLB1     4.181068e-05                       4                4
## SLC22A13 5.310557e-05                       4                4
## XIRP1    6.938574e-05                       4                4
## SCN10A   8.972468e-05                       4                4
## ZNF620   1.707402e-04                       4                4
## CHCHD4   9.825373e-04                       4                4
```

The results for each gene, this includes localFDR, classes with global FDR control at $\alpha = 0.1$ and classes without global FDR control. Here the class labels corresponds to 1=(ge-,cnv-),2=(ge+,cnv-),3=(ge-,cnv+),4=(ge+,cnv+). We could see that component 3 is missing here after controlling for FDR, and there are only 9 genes in component 3 before we control for FDR. This result is coherent with the model selection result.

Example 2: Three data types, tranformed z values

IMIX test without specifying the initial values of the parameters

```
test2=IMIX(data_input=data_z,data_type="z",p_ini=rep(0.125,8),alpha=0.05,verbose = TRUE)
```

```
## Assign initial values!
## number of iterations= 14
## number of iterations= 18
## number of iterations= 10
## Start IMIX-ind procedure!
## iter=1: loglik=-211319.340967286
## iter=2: loglik=-6219.91821213171
## iter=3: loglik=-6215.44586785553
## iter=4: loglik=-6215.42741217281
## iter=5: loglik=-6215.42396915171
## iter=6: loglik=-6215.42320911141
## iter=7: loglik=-6215.42303794989
## iter=8: loglik=-6215.42299899458
## iter=9: loglik=-6215.42299007738
## iter=10: loglik=-6215.42298802982
## iter=11: loglik=-6215.42298755888
## Successfully Done!
```

```
## Start IMIX-cor-twostep procedure!
## iter=1: loglik=-206958.792194342
## iter=2: loglik=-6216.39282934126
## iter=3: loglik=-6186.76208046601
## iter=4: loglik=-6185.38763403367
## iter=5: loglik=-6185.15454824807
## iter=6: loglik=-6185.1111235976
## iter=7: loglik=-6185.10323560445
## iter=8: loglik=-6185.10181927698
## iter=9: loglik=-6185.10156485357
## iter=10: loglik=-6185.1015189111
## iter=11: loglik=-6185.10151056272
## iter=12: loglik=-6185.10150903662
## iter=13: loglik=-6185.10150875617
## Successfully Done!
## Start IMIX-cor model procedure!
## iter=1: loglik=-206958.792194342
## iter=2: loglik=-6216.39282934126
## iter=3: loglik=-6186.85221175454
## iter=4: loglik=-6185.52731019493
## iter=5: loglik=-6185.29839161
## iter=6: loglik=-6185.25323167881
## iter=7: loglik=-6185.24415403539
## iter=8: loglik=-6185.24226976979
## iter=9: loglik=-6185.24187425847
## iter=10: loglik=-6185.24179973062
## iter=11: loglik=-6185.24179379169
## iter=12: loglik=-6185.24179981246
## iter=13: loglik=-6185.24180565216
## iter=14: loglik=-6185.2418095062
## iter=15: loglik=-6185.24181174259
## iter=16: loglik=-6185.24181295885
## iter=17: loglik=-6185.24181359466
## Successfully Done!
## Start IMIX-cor-restrict procedure!
## iter=1: loglik=-206958.792194342
## iter=2: loglik=-6216.39282934126
## iter=3: loglik=-6186.76106409091
## iter=4: loglik=-6185.33637067415
## iter=5: loglik=-6185.06265026996
## iter=6: loglik=-6185.004357692
## iter=7: loglik=-6184.9924301221
## iter=8: loglik=-6184.99002486867
## iter=9: loglik=-6184.98953718544
## iter=10: loglik=-6184.98943701856
## iter=11: loglik=-6184.98941615792
## iter=12: loglik=-6184.98941175934
## iter=13: loglik=-6184.98941082216
## Successfully Done!
## Start Model Selection!
## Start Adaptive FDR Control!
## All Done!
```

Results of example 2

```
test2$estimatedFDR
```

```
## $estimated_mFDR_comp2
## [1] 0.04751399
##
## $estimated_mFDR_comp3
## [1] 0.04496918
##
## $estimated_mFDR_comp4
## [1] 0.04561218
##
## $estimated_mFDR_comp5
## [1] 0.04645422
##
## $estimated_mFDR_comp6
## [1] 0.04820897
##
## $estimated_mFDR_comp7
## [1] 0.04624412
##
## $estimated_mFDR_comp8
## [1] 0.04481877
```

```
test2$`AIC/BIC`
```

```
##                         AIC      BIC
## IMIX_ind          12468.85 12562.09
## IMIX_cor_twostep  12480.20 12750.13
## IMIX_cor          12528.48 12916.20
## IMIX_cor_restrict 12491.98 12791.35
```

```
test2$`Selected Model`
```

```
## [1] "IMIX_ind"
```

```
str(test2$IMIX_ind)
```

```
## List of 7
##  $ posterior prob      :'data.frame':    1000 obs. of  8 variables:
##   ..$ component1: num [1:1000] 2.00e-09 9.99e-01 4.80e-07 7.79e-13 7.95e-04 ...
##   ..$ component2: num [1:1000] 3.98e-17 3.12e-04 3.59e-16 3.37e-08 2.75e-10 ...
##   ..$ component3: num [1:1000] 1.53e-13 5.43e-04 1.00 2.33e-05 9.99e-01 ...
##   ..$ component4: num [1:1000] 1.00 5.16e-06 8.95e-15 1.08e-16 1.95e-10 ...
##   ..$ component5: num [1:1000] 3.01e-21 1.68e-07 7.41e-10 1.00 3.42e-07 ...
##   ..$ component6: num [1:1000] 2.22e-08 1.80e-09 7.48e-24 5.22e-12 7.54e-17 ...
##   ..$ component7: num [1:1000] 8.92e-05 3.27e-09 2.17e-08 3.77e-09 2.86e-07 ...
##   ..$ component8: num [1:1000] 1.39e-12 7.99e-13 1.27e-17 1.28e-04 7.75e-14 ...
##  $ Full LogLik all     : num [1:11] -211319 -6220 -6215 -6215 -6215 ...
##  $ Full MaxLogLik final: num -6215
##  $ iterations          : num 11
##  $ pi                  : num [1:8] 0.111 0.114 0.126 0.115 0.128 ...
##  $ mu                  : num [1:6] 0.0159 5.0838 -0.0568 4.9876 -0.0206 ...
##  $ sigma               : num [1:6] 0.985 0.974 0.939 1.037 0.947 ...
```

```
dim(test2$significant_genes_with_FDRcontrol)
```

```
## [1] 1000    3
```

```
head(test2$significant_genes_with_FDRcontrol)
```

```
##             localFDR class_withoutFDRcontrol class_FDRcontrol
## gene212 7.676829e-09                       8               8
## gene599 3.789346e-08                       8               8
## gene764 4.317928e-08                       8               8
## gene140 6.726940e-08                       8               8
## gene949 6.740396e-08                       8               8
## gene785 1.251474e-07                       8               8
```

For the output includes each model parameter estimations, the model selection AIC and BIC values and the best selected model. The estimated FDR corresponding to the prespecified $\alpha = 0.05$ threshold, the local FDR and class labels for each gene, both without FDR control and based on the FDR control at $\alpha = 0.05$. Here the class labels corresponds to 1=(meth-,ge-,cnv-),2=(meth+,ge-,cnv-),3=(meth-,ge+,cnv-),4=(meth-,ge-,cnv+),5=(meth+,ge+,cnv-),6=(meth+,ge-,cnv+),7=(meth-,ge+,cnv+),8=(meth+,ge+,cnv+).

```
sessionInfo()
```

```
## R version 3.6.1 (2019-07-05)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Sierra 10.12.6
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
##
## locale:
## [1] zh_CN.UTF-8/zh_CN.UTF-8/zh_CN.UTF-8/C/zh_CN.UTF-8/zh_CN.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] MASS_7.3-51.5 mclust_5.4.5  IMIX_0.1.0
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.3        pillar_1.4.3      compiler_3.6.1    mixtools_1.1.0
##  [5] tools_3.6.1       digest_0.6.23     lattice_0.20-38   evaluate_0.14
##  [9] tibble_2.1.3      gtable_0.3.0      pkgconfig_2.0.3   rlang_0.4.4
## [13] Matrix_1.2-17     rstudioapi_0.10   yaml_2.2.0        mvtnorm_1.0-12
## [17] xfun_0.11         stringr_1.4.0     dplyr_0.8.3       knitr_1.26
## [21] segmented_1.1-0   grid_3.6.1        tidyselect_1.0.0  glue_1.3.1
## [25] R6_2.4.1          survival_2.44-1.1 rmarkdown_2.0     ggplot2_3.2.1
## [29] purrr_0.3.3       magrittr_1.5      splines_3.6.1     scales_1.0.0
## [33] htmltools_0.3.6   assertthat_0.2.1  colorspace_1.4-1  labeling_0.3
## [37] stringi_1.4.5     lazyeval_0.2.2    munsell_0.5.0     crayon_1.3.4
```