R code accompanying the paper “Projection pursuit based on Gaussian mixtures and evolutionary algorithms”, JCGS, 2019

Luca Scrucca and Alessio Serafini

15 Oct 2018

```
# If needed please install the following packages from CRAN:
#
# install.packages(c("ppgmmga", "gridExtra", "rmarkdown", "mlbench", "gtable",
#                    "pgmm", "dr", "fastICA"), dependencies = TRUE)
#
# and the following packages from Bioconductor:
#
# if(!requireNamespace("BiocManager", quietly = TRUE))
#    install.packages("BiocManager")
# BiocManager::install("Biobase")
# BiocManager::install("multtest")

# To reproduce the results of the paper please use:
# - R ver. 3.5.3 (2019-03-11)
# - ppgmmga ver. 1.2 (2019-07-08)
# In R ver. >= 3.6 the following command is needed for backward compatibility:
RNGkind(sample.kind = "Rounding")
```

```
## Warning in RNGkind(sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

```
# To compile the full report with results use:
# rmarkdown::render("code.R")

library(ppgmmga)
```

```
## Package 'ppgmmga' version 1.2
```

```
library(mclust)
```

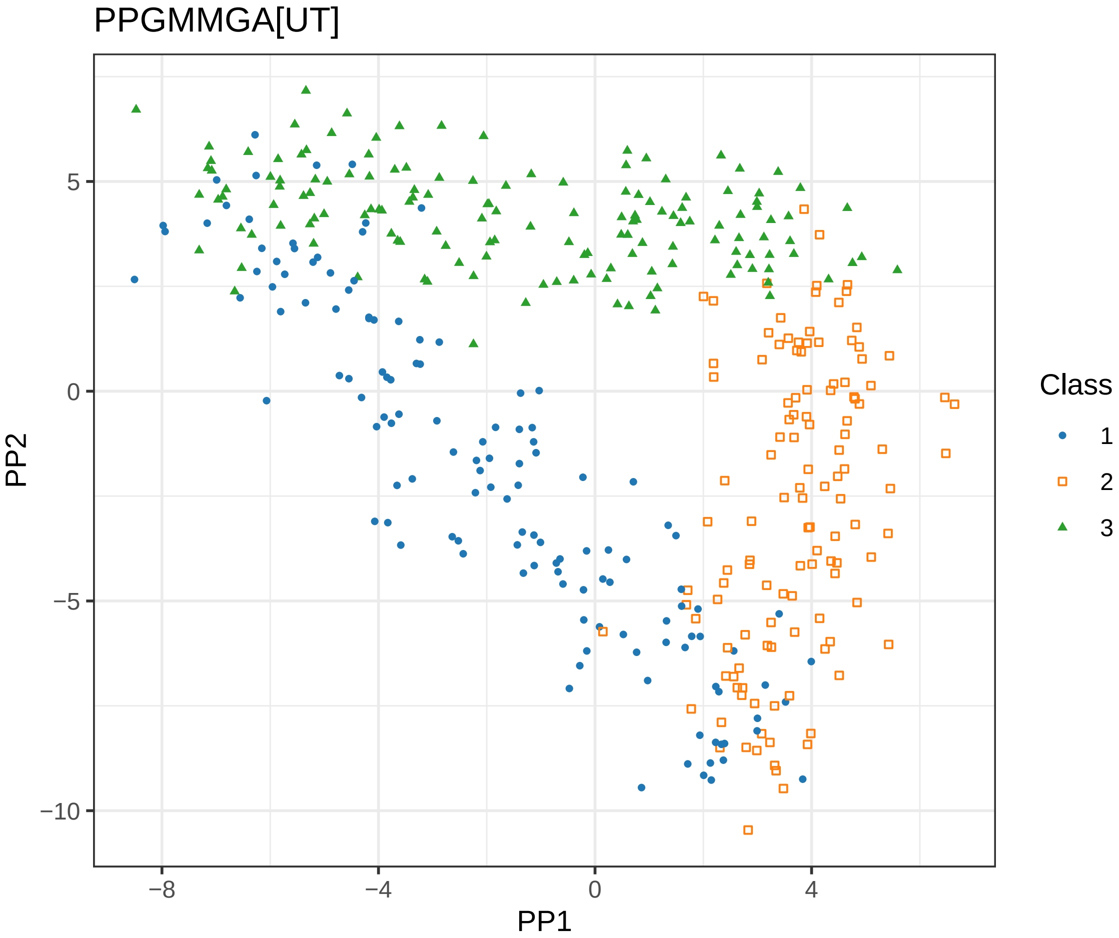
```
## Package 'mclust' version 5.4.6
## Type 'citation("mclust")' for citing this R package in publications.
```

```
library(ggplot2)
library(ggthemes)
library(gridExtra)
library(grid)
theme_set(theme_bw())
theme_update(plot.title = element_text(hjust = 0.5))
source(system.file("scripts/jcgs", "mc_negent_other_methods.R", package = "ppgmmga"))
source(system.file("scripts/jcgs", "misc.R", package = "ppgmmga"))


# Waveform data ---------------------------------------------------------

library(mlbench)
```

```
set.seed(20180124)
x <- mlbench.waveform(400)
X <- x$x
Class <- factor(x$classes)

X <- scale(X, center = TRUE, scale = FALSE)
GMM <- densityMclust(X)

PPGMMGA1 <- ppgmmga(data = X, d = 2, GMM = GMM, scale = FALSE, approx = "UT", seed = 1)
summary(PPGMMGA1, check = TRUE)
```

```
## -- ppgmmga ----------------------------
##
## Data dimensions               = 400 x 21
## Data transformation           = center
## Projection subspace dimension = 2
## GMM density estimate          = (EII,5)
## Negentropy approximation      = UT
## GA optimal negentropy         = 1.002469
## GA encoded basis solution:
##            x1       x2      x3       x4       x5       x6       x7
## [1,] 4.947688 1.564279 1.65297 1.772846 1.860352 1.978782 1.925953
##            x8       x9      x10  ...       x39      x40
## [1,] 1.962031 1.772991 1.479144       2.475268 2.603082
##
## Estimated projection basis:
##              PP1        PP2
##  [1,]  0.0000000  0.0026661
##  [2,]  0.0000000 -0.1032041
##  [3,]  0.0000000 -0.1728110
##  [4,]  0.0000000 -0.2560134
##  [5,]  0.0000000 -0.3494888
##  [6,]  0.0000000 -0.3582193
##  [7,]  0.0000000 -0.3691507
##  [8,]  0.2261044 -0.2153611
##  [9,]  0.3692394 -0.0320902
## [10,]  0.3662940  0.1139038
## [11,]  0.3958025  0.2805435
## [12,]  0.2215663  0.3183615
## [13,]  0.0667459  0.3401403
## [14,] -0.1494979  0.2741385
## [15,] -0.3070799  0.2243247
## [16,] -0.2986673  0.1412286
## [17,] -0.3712442  0.0713744
## [18,] -0.2787678  0.0180108
## [19,] -0.1999961  0.0103275
## [20,] -0.0820795 -0.0029283
## [21,]  0.0065177 -0.0104015
##
## Monte Carlo Negentropy approximation check:
##                            UT
## Approx Negentropy 1.002469110
## MC Negentropy     1.021002839
## MC se             0.003108763
## Relative accuracy 0.981847524
```

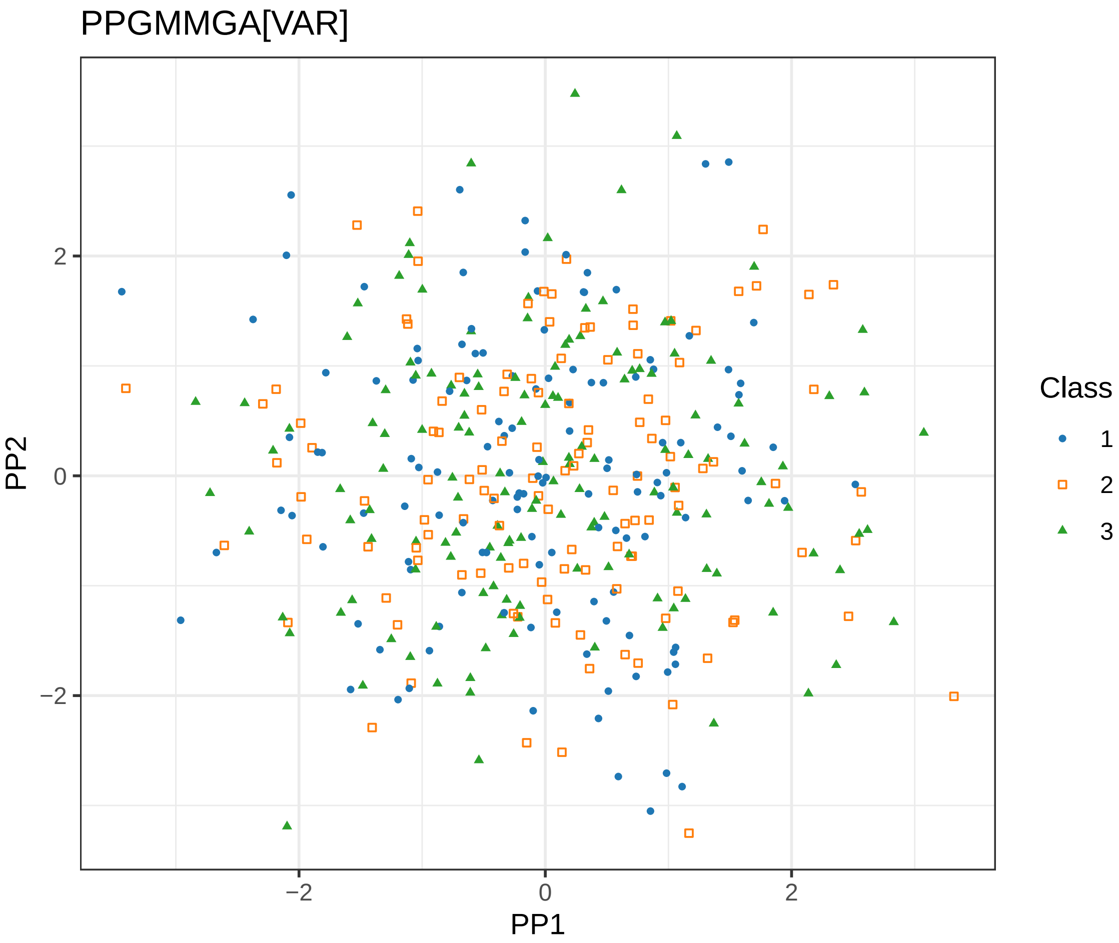
```
plot(PPGMMGA1, Class, drawAxis = FALSE) + ggtitle("PPGMMGA[UT]")
```



```
PPGMMGA2 <- ppgmmga(data = X, d = 2, GMM = GMM, scale = FALSE, approx = "VAR", seed = 2)
summary(PPGMMGA2, check = TRUE)
```

```
## -- ppgmmga -----------------------------
##
## Data dimensions                 = 400 x 21
## Data transformation             = center
## Projection subspace dimension = 2
## GMM density estimate            = (EII,5)
## Negentropy approximation        = VAR
## GA optimal negentropy           = 0.2332543
## GA encoded basis solution:
##            x1      x2       x3       x4       x5       x6      x7       x8
## [1,] 5.047285 2.09815 1.700378 1.409182 1.581833 1.611557 1.44716 1.748681
##            x9      x10  ...       x39     x40
## [1,] 1.313699 1.772397       1.676822 2.242114
##
## Estimated projection basis:
##              PP1        PP2
##  [1,] -0.2708339 -0.4251899
##  [2,]  0.0942513 -0.0326857
##  [3,] -0.2293498  0.2724225
##  [4,] -0.1983561  0.0224020
##  [5,]  0.0617098 -0.1252308
##  [6,] -0.3685554  0.1457683
##  [7,]  0.1851251  0.2644924
##  [8,] -0.0996772  0.0836680
##  [9,]  0.3930107  0.1438695
## [10,] -0.1397710 -0.1253368
## [11,] -0.0048142 -0.1104520
## [12,] -0.2819305  0.1365490
## [13,] -0.1598240  0.2142221
## [14,]  0.2098509  0.1348208
## [15,] -0.1483759  0.1491140
## [16,]  0.1042038  0.2623540
## [17,] -0.0344610 -0.0977059
## [18,] -0.0093336 -0.3094121
## [19,]  0.1378850  0.3627416
## [20,] -0.1116634 -0.2890206
## [21,] -0.5032487  0.2945498
##
## Monte Carlo Negentropy approximation check:
##                           VAR
## Approx Negentropy 0.233254344
## MC Negentropy     0.238876730
## MC se             0.003149867
## Relative accuracy 0.976463235
```

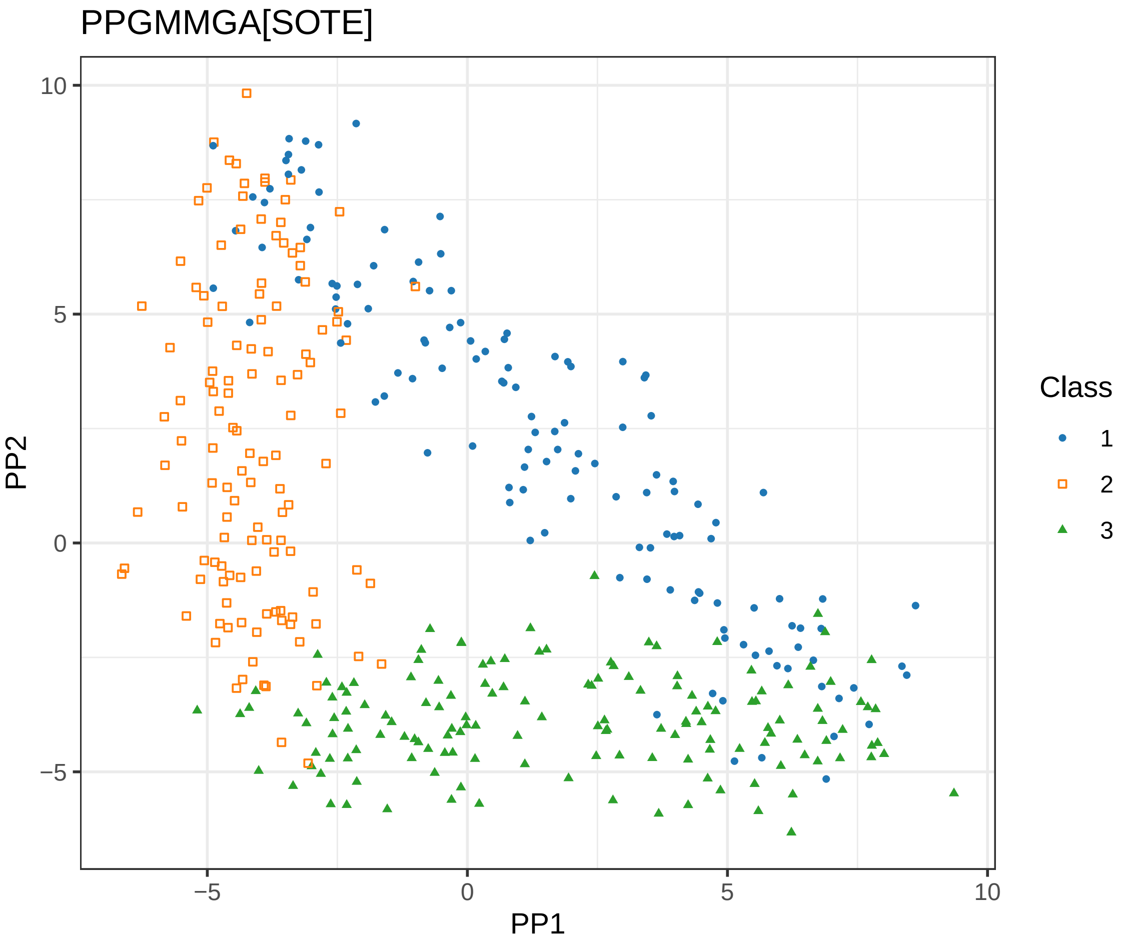
```
plot(PPGMMGA2, Class, drawAxis = FALSE) + ggtitle("PPGMMGA[VAR]")
```



```
PPGMMGA3 <- ppgmmga(data = X, d = 2, GMM = GMM, scale = FALSE, approx = "SOTE", seed = 3)
summary(PPGMMGA3, check = TRUE)
```

```
## -- ppgmmga -----------------------------
##
## Data dimensions                    = 400 x 21
## Data transformation                = center
## Projection subspace dimension = 2
## GMM density estimate               = (EII,5)
## Negentropy approximation           = SOTE
## GA optimal negentropy              = 1.005812
## GA encoded basis solution:
##              x1        x2        x3        x4        x5        x6        x7
## [1,] 2.308028 1.563483 1.652205 1.769854 1.856781 1.983334 1.943112
##              x8        x9       x10  ...        x39       x40
## [1,] 1.998647 1.827142 1.544723       2.553373 3.141593
##
## Estimated projection basis:
##                PP1        PP2
##  [1,]  0.0000000  0.0000000
##  [2,]  0.0000000  0.0000000
##  [3,]  0.0000000  0.1760548
##  [4,]  0.0000000  0.2639578
##  [5,]  0.0000000  0.3601405
##  [6,]  0.0000000  0.3588297
##  [7,] -0.1644906  0.3478786
##  [8,] -0.2506852  0.1855117
##  [9,] -0.3673343 -0.0156234
## [10,] -0.3434626 -0.1605137
## [11,] -0.3497617 -0.3311279
## [12,] -0.1734570 -0.3452100
## [13,] -0.0183523 -0.3468925
## [14,]  0.1845164 -0.2544383
## [15,]  0.3318653 -0.1835839
## [16,]  0.3123605 -0.1025645
## [17,]  0.3757966 -0.0220160
## [18,]  0.2756083  0.0183152
## [19,]  0.1970853  0.0149899
## [20,]  0.0813167  0.0129555
## [21,] -0.0073129  0.0102109
##
## Monte Carlo Negentropy approximation check:
##                               SOTE
## Approx Negentropy 1.00581240
## MC Negentropy     1.02226057
## MC se             0.00312136
## Relative accuracy 0.98391000
```

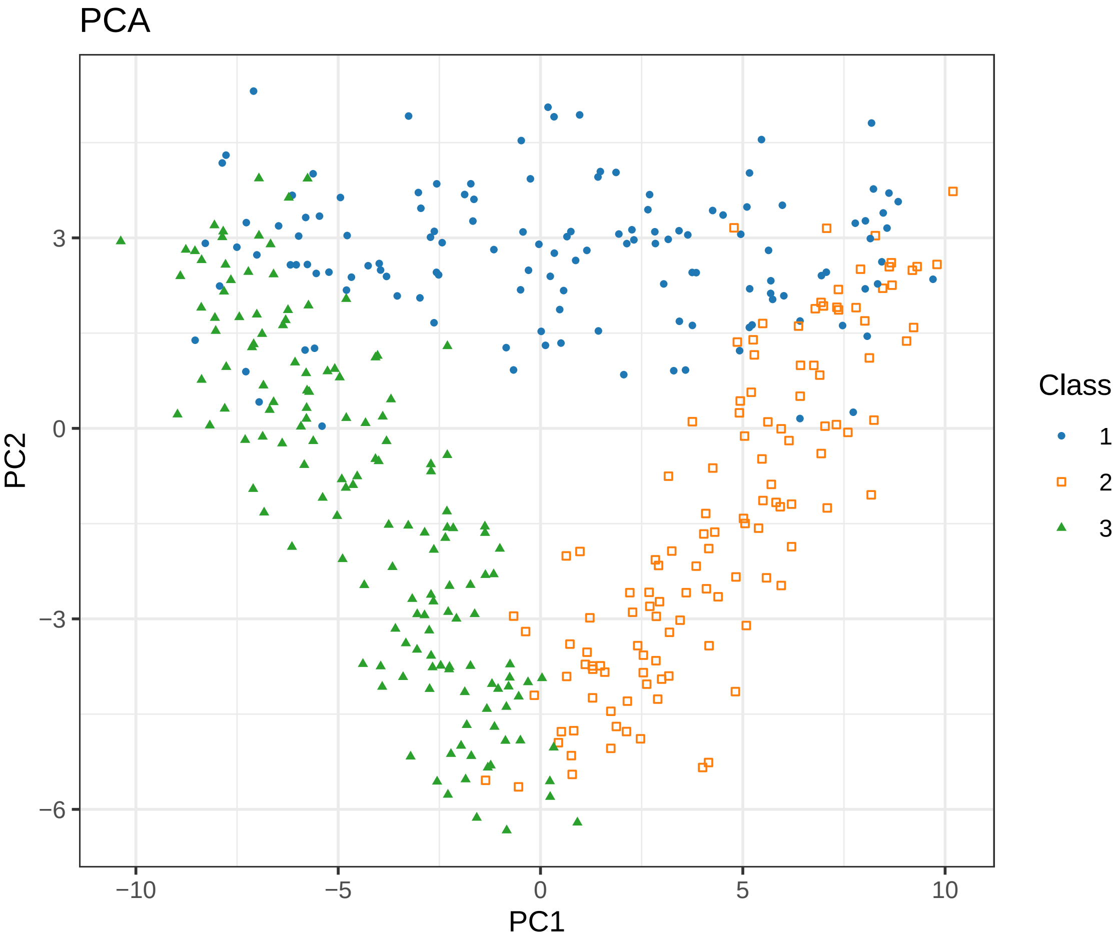
```
plot(PPGMMGA3, Class, drawAxis = FALSE) + ggtitle("PPGMMGA[SOTE]")
```



```
PCA <- NegentropyPCA(PPGMMGA1)
PCA[c("Negentropy", "se")]
```

```
## $Negentropy
## [1] 1.026846
##
## $se
## [1] 0.003112106
```

```
PPGMMPCA <- PPGMMGA1; PPGMMPCA$approx <- "PCA"
PPGMMPCA$basis <- PCA$basis
PPGMMPCA$Z <- PCA$Z
plot(PPGMMPCA, Class, drawAxis = FALSE) + ggtitle("PCA")
```



```
ICA <- NegentropyFASTICA(PPGMMGA1)
```

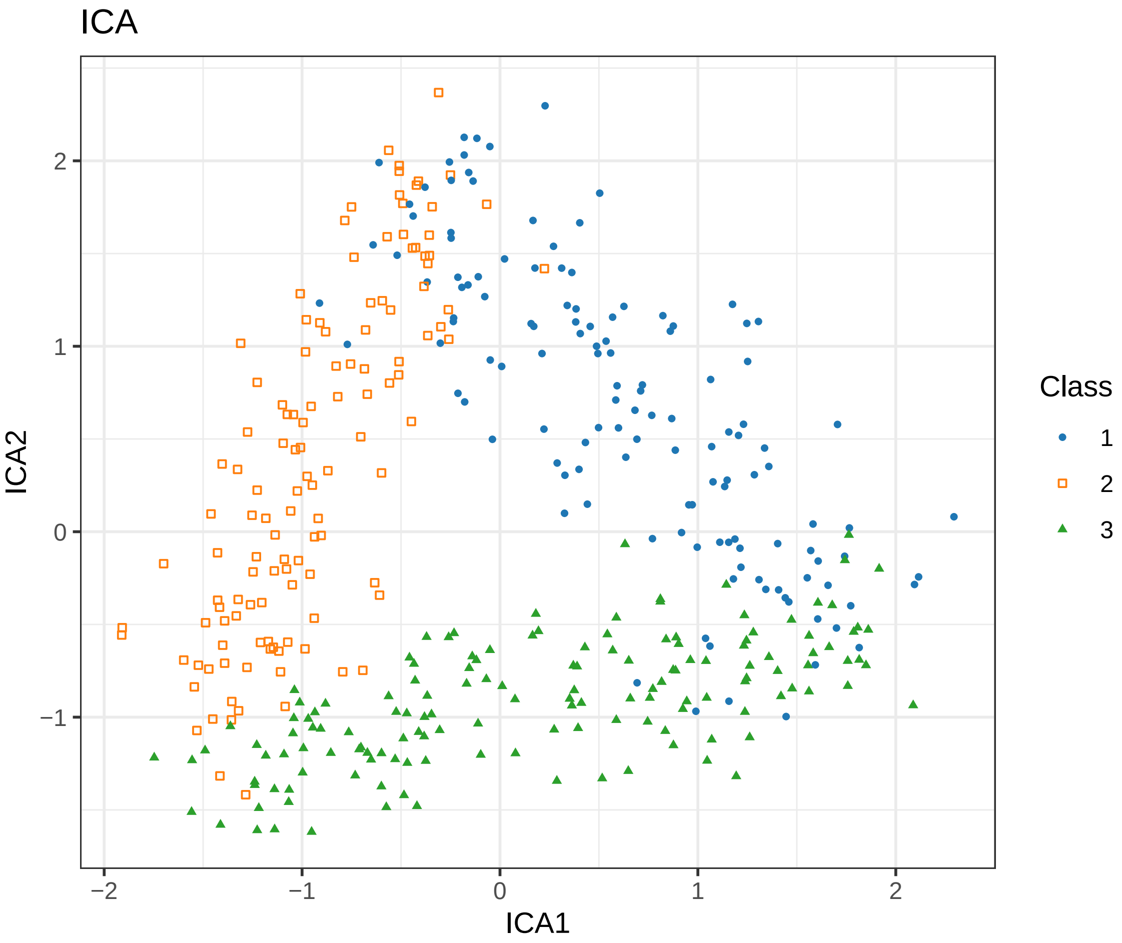
```
## Loading required package: fastICA
```

```
ICA[c("Negentropy", "se")]
```

```
## $Negentropy
## [1] 1.027086
##
## $se
## [1] 0.003138164
```

```
# trick for plotting
```

```
PPGMMICA <- PPGMMGA1; PPGMMICA$approx <- "ICA"
PPGMMICA$basis <- ICA$basis
PPGMMICA$Z <- ICA$Z
plot(PPGMMICA, Class, drawAxis = FALSE) + ggtitle("ICA")
```



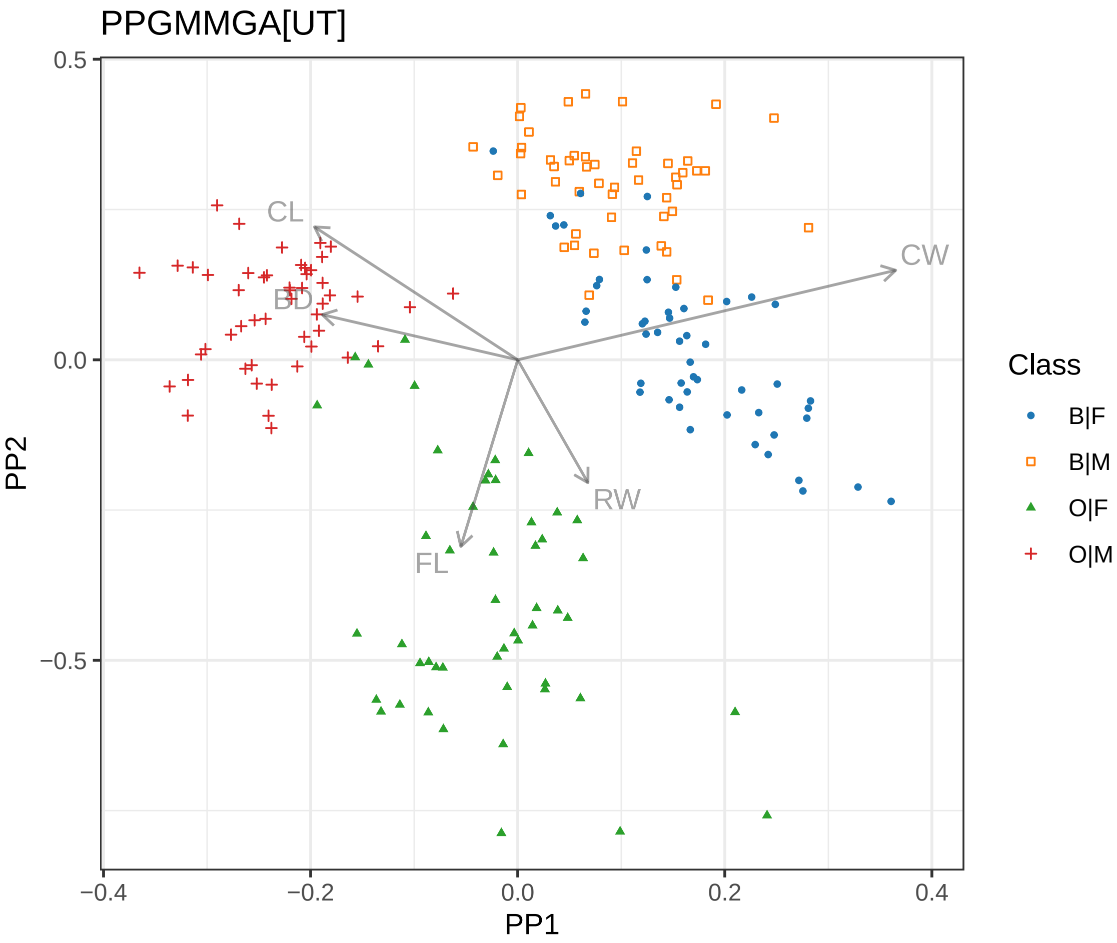
```
ppgmmga.table.angle(PPGMMGA1, PPGMMGA2, PPGMMGA3, PPGMMPCA, PPGMMICA)
```

```
##        UT      VAR      SOTE       PCA                    ICA
## UT      0 89.01658  8.142409  9.034730  9.0347304672581270069
## VAR    NA  0.00000 89.754845 89.490559 89.4905586581620298148
## SOTE   NA       NA  0.000000  6.032357  6.0323569653623128062
## PCA    NA       NA        NA  0.000000  0.0000000000003146623
## ICA    NA       NA        NA        NA  0.0000000000000000000
```

```
# Crabs data -----------------------------------------------------------

data(crabs, package = "MASS")
X <- crabs[, 4:8]
Class <- as.factor(with(crabs, paste(sp, sex, sep = "|")))

X <- scale(X, center = TRUE, scale = TRUE)
GMM <- densityMclust(X)

PPGMMGA1 <- ppgmmga(data = X, d = 2, GMM = GMM, approx = "UT", seed = 1)
summary(PPGMMGA1, check = TRUE)
```

```
## -- ppgmmga -----------------------------
##
## Data dimensions               = 200 x 5
## Data transformation           = center & scale
## Projection subspace dimension = 2
## GMM density estimate          = (VEE,6)
## Negentropy approximation      = UT
## GA optimal negentropy         = 0.6001497
## GA encoded basis solution:
##            x1       x2       x3       x4       x5       x6       x7
## [1,] 5.602055 1.990215 0.531916 2.723246 1.010578 1.694094 1.978759
##           x8
## [1,] 2.06845
##
## Estimated projection basis:
##           PP1        PP2
## FL -0.1184951 -0.6699390
## RW  0.1461940 -0.4410283
## CL -0.4232782  0.4767309
## CW  0.7871378  0.3212069
## BD -0.4072296  0.1619555
##
## Monte Carlo Negentropy approximation check:
##                            UT
## Approx Negentropy 0.600149700
## MC Negentropy     0.607751667
## MC se             0.002972412
## Relative accuracy 0.987491656
```

```
plot(PPGMMGA1, Class) + ggtitle("PPGMMGA[UT]")
```

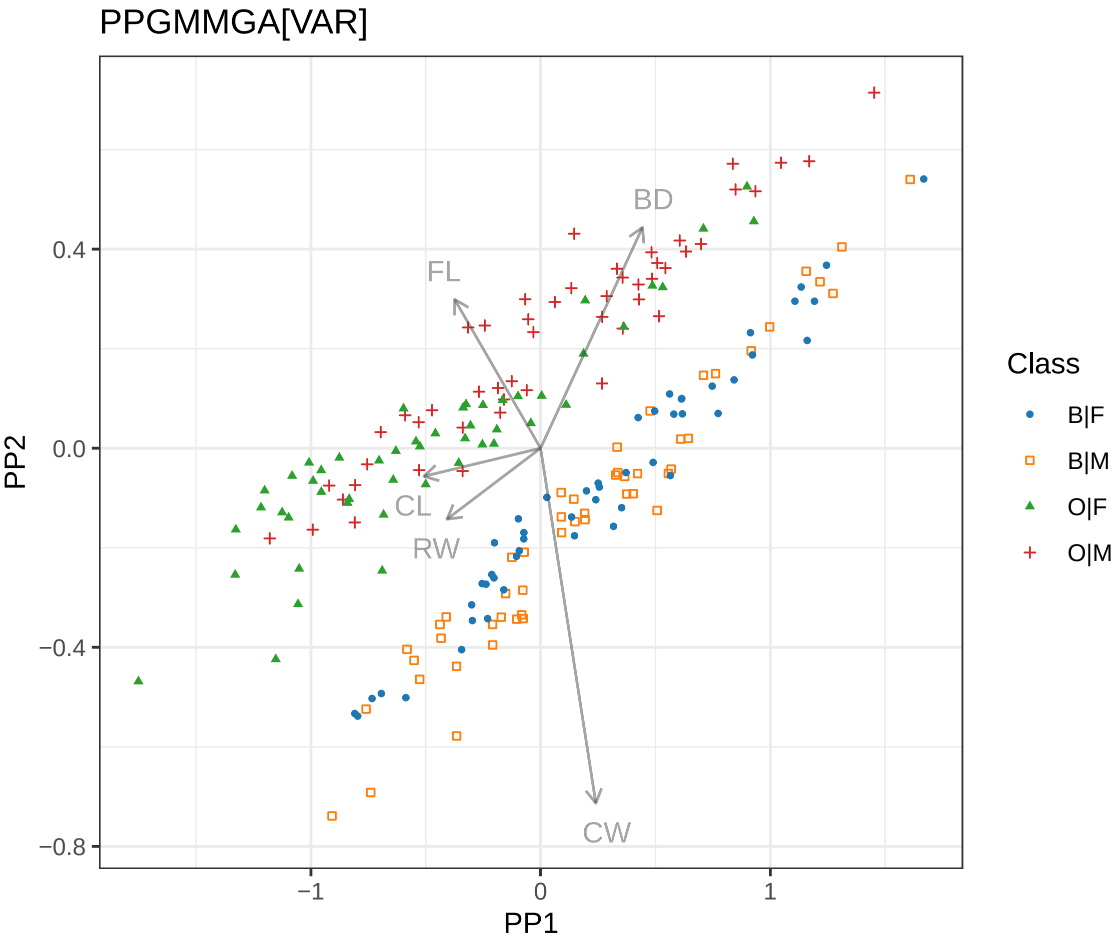


```
PPGMMGA2 <- ppgmmga(data = X, d = 2, GMM = GMM, approx = "VAR", seed = 2)
summary(PPGMMGA2, check = TRUE)
```

```
## -- ppgmmga ----------------------------
##
## Data dimensions                  = 200 x 5
## Data transformation              = center & scale
## Projection subspace dimension = 2
## GMM density estimate             = (VEE,6)
## Negentropy approximation         = VAR
## GA optimal negentropy            = 0.2716329
## GA encoded basis solution:
##            x1       x2       x3       x4       x5       x6       x7
## [1,] 3.884605 1.058383 1.260722 2.313448 3.332061 0.805733 2.049418
##            x8
## [1,] 2.367513
##
## Estimated projection basis:
##           PP1        PP2
## FL -0.4136463  0.3304305
## RW -0.4502869 -0.1575530
## CL -0.5612821 -0.0623415
## CW  0.2659398 -0.7884391
## BD  0.4902824  0.4903774
##
```

```
## Monte Carlo Negentropy approximation check:
##                           VAR
## Approx Negentropy 0.271632884
## MC Negentropy     0.462154683
## MC se             0.002837045
## Relative accuracy 0.587753179
```

```
plot(PPGMMGA2, Class) + ggtitle("PPGMMGA[VAR]")
```

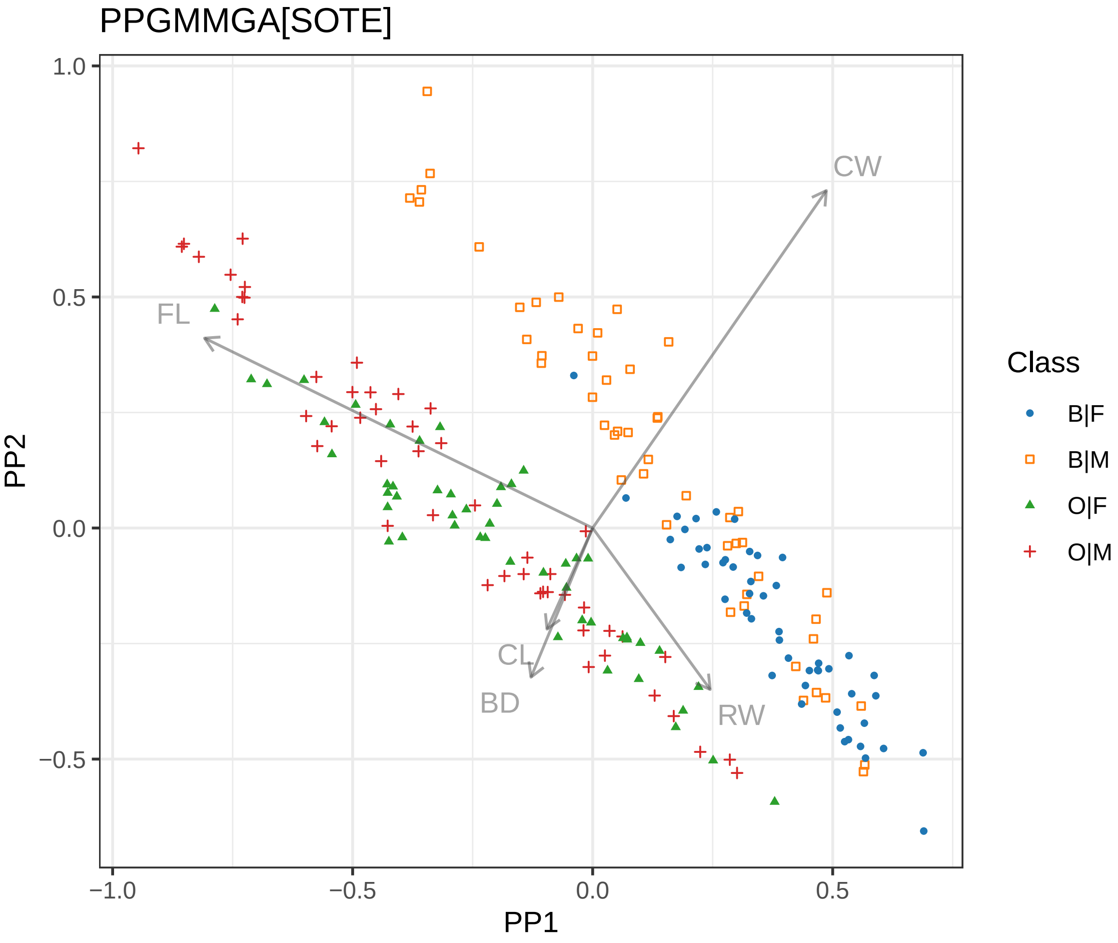


```
PPGMMGA3 <- ppgmmga(data = X, d = 2, GMM = GMM, approx = "SOTE", seed = 3)
summary(PPGMMGA3, check = TRUE)
```

```
## -- ppgmmga -----------------------------
##
## Data dimensions                  = 200 x 5
## Data transformation              = center & scale
## Projection subspace dimension = 2
## GMM density estimate             = (VEE,6)
## Negentropy approximation         = SOTE
## GA optimal negentropy            = 0.5684237
## GA encoded basis solution:
##            x1       x2       x3      x4 x5       x6       x7       x8
## [1,] 5.006415 1.701216 1.050783 1.68292  0 1.213038 2.798521 0.700012
##
## Estimated projection basis:
```

```
##            PP1        PP2
## FL -0.8183454  0.4158548
## RW  0.2477974 -0.3536123
## CL -0.0962744 -0.2213935
## CW  0.4926716  0.7391007
## BD -0.1300507 -0.3267081
##
## Monte Carlo Negentropy approximation check:
##                             SOTE
## Approx Negentropy 0.568423669
## MC Negentropy     0.490400886
## MC se             0.002830511
## Relative accuracy 1.159100004
```

```
plot(PPGMMGA3, Class) + ggtitle("PPGMMGA[SOTE]")
```

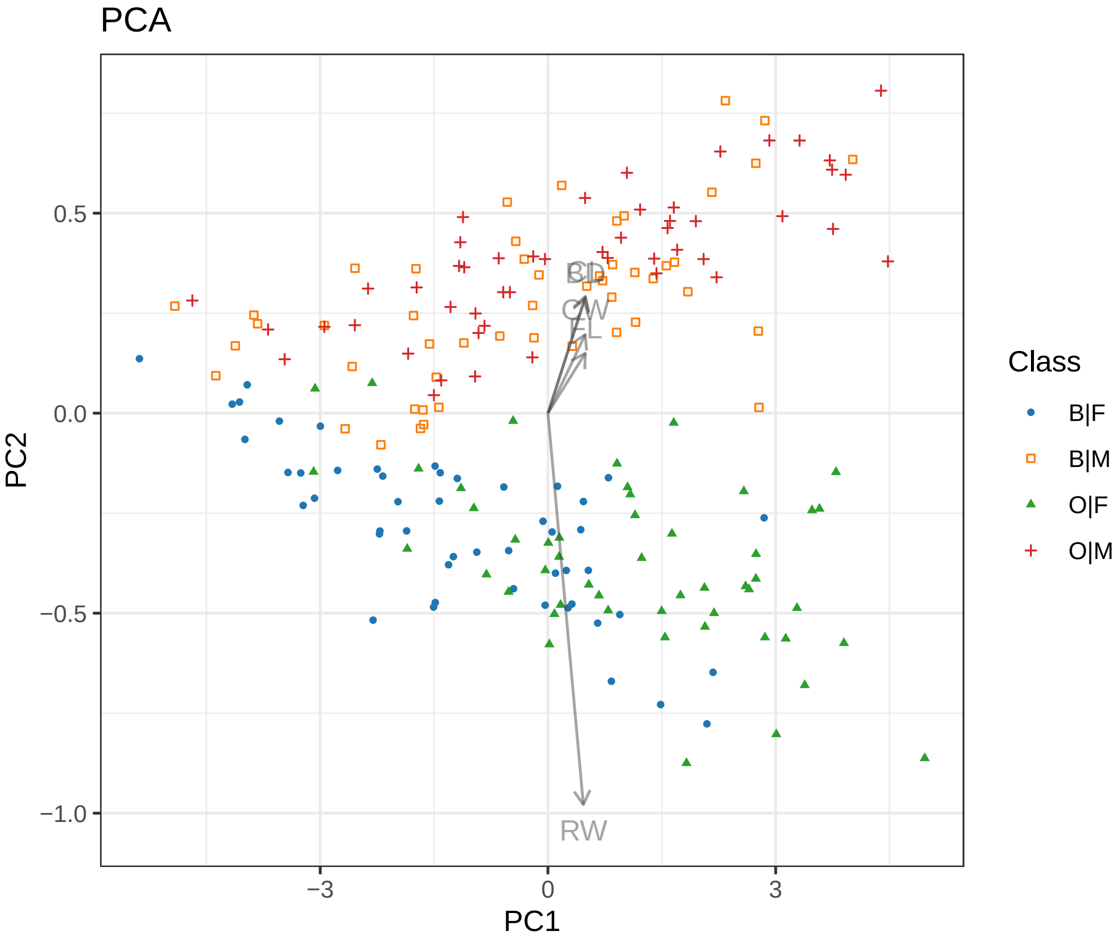


```
PCA <- NegentropyPCA(PPGMMGA1)
PCA[c("Negentropy", "se")]
```

```
## $Negentropy
## [1] 0.1933726
##
## $se
## [1] 0.002615653
```

```
PPGMMPCA <- PPGMMGA1; PPGMMPCA$approx <- "PCA"
```

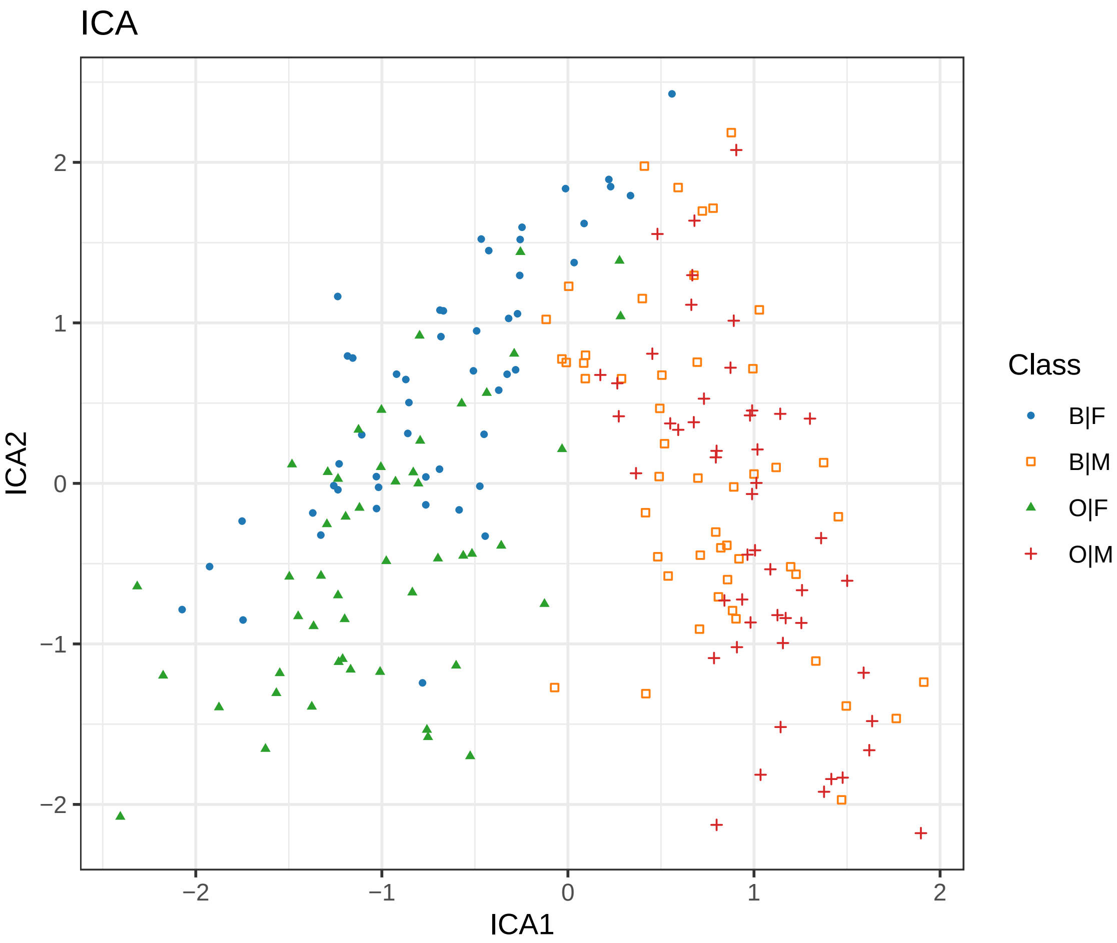
```
PPGMMPCA$basis <- PCA$basis
PPGMMPCA$Z <- PCA$Z
plot(PPGMMPCA, Class) + ggtitle("PCA")
```



```
ICA <- NegentropyFASTICA(PPGMMGA1)
ICA[c("Negentropy", "se")]
```

```
## $Negentropy
## [1] 0.1898331
##
## $se
## [1] 0.00266059
```

```
# trick for plotting
PPGMMICA <- PPGMMGA1; PPGMMICA$approx <- "ICA"
PPGMMICA$basis <- ICA$basis
PPGMMICA$Z <- ICA$Z
plot(PPGMMICA, Class, drawAxis = FALSE) + ggtitle("ICA")
```



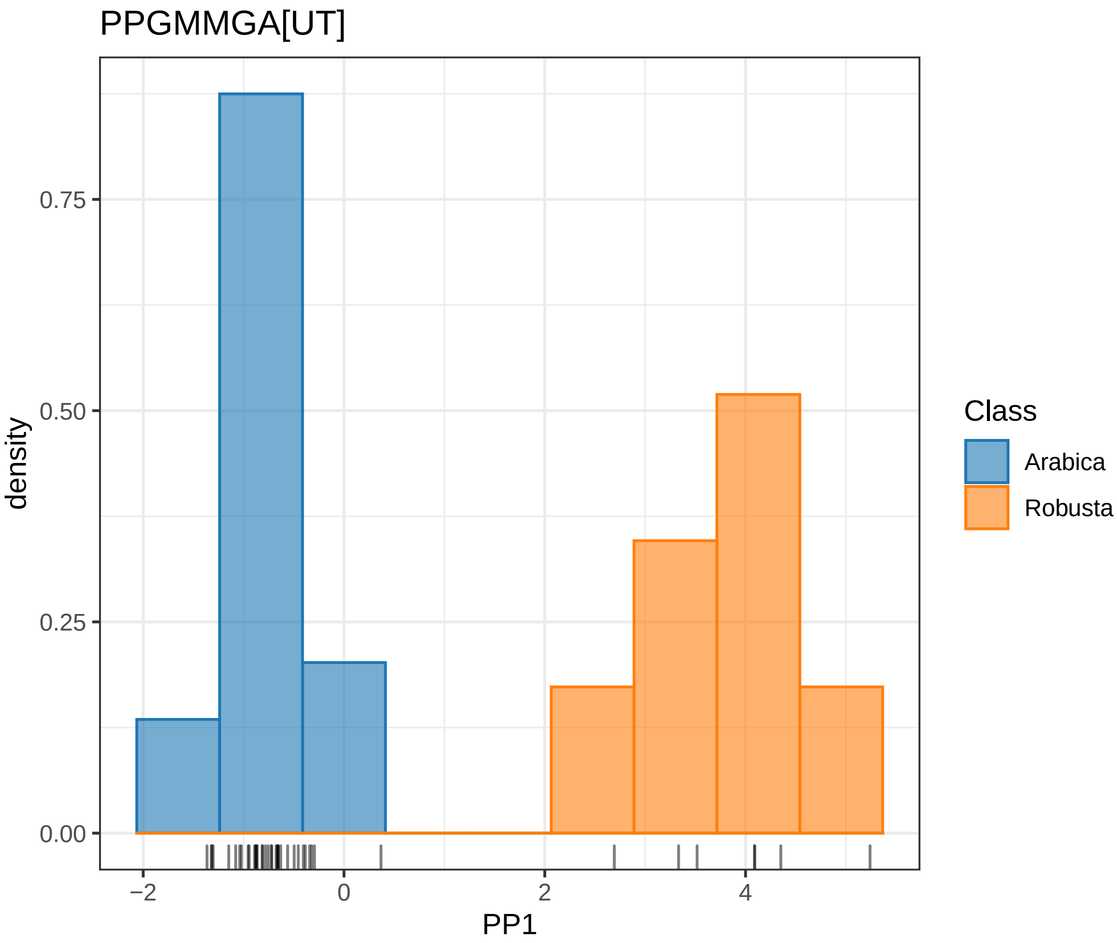
```
ppgmmga.table.angle(PPGMMGA1, PPGMMGA2, PPGMMGA3, PPGMMPCA, PPGMMICA)
```

```
##      UT      VAR     SOTE      PCA                    ICA
## UT    0 76.23147 66.29325 88.11079 88.1107920736300229692
## VAR  NA  0.00000 85.69386 88.20356 88.2035595841385884341
## SOTE NA       NA  0.00000 88.34848 88.3484806999108798209
## PCA  NA       NA       NA  0.00000  0.0000000000000706712
## ICA  NA       NA       NA       NA  0.0000000000000000000
```

```
# Coffee data ----------------------------------------------------------

data("coffee", package = "pgmm")
X <- coffee[,-(1:2)]
names(X)[8] <- c("Caffeine")
Class <- factor(coffee$Variety, levels = 1:2, labels = c("Arabica", "Robusta"))

X <- scale(X, center = TRUE, scale = TRUE)
GMM <- densityMclust(X)

PPGMMGA1 <- ppgmmga(data = X, d = 1, GMM = GMM, approx = "UT", seed = 1)
summary(PPGMMGA1, check = TRUE)
```

```
## -- ppgmmga -----------------------------
##
## Data dimensions                 = 43 x 12
## Data transformation             = center & scale
## Projection subspace dimension = 1
## GMM density estimate          = (VEI,3)
## Negentropy approximation      = UT
## GA optimal negentropy         = 1.073236
## GA encoded basis solution:
##            x1       x2       x3       x4       x5       x6       x7
## [1,] 3.974492 1.336151 1.470531 1.514307 1.715505 0.984944 2.792978
##            x8       x9      x10      x11
## [1,] 1.478995 0.252026 0.997868 1.750353
##
## Estimated projection basis:
##                            PP1
## Water               -0.0413378
## Bean Weight         -0.0375860
## Extract Yield       -0.0101412
## ph Value             0.0366314
## Free Acid            0.2624216
## Mineral Content      0.0249465
## Fat                 -0.7487171
## Caffeine             0.5286202
## Trigonelline        -0.1393259
## Chlorogenic Acid     0.0546360
## Neochlorogenic Acid  0.0973544
## Isochlorogenic Acid  0.2324976
##
## Monte Carlo Negentropy approximation check:
##                            UT
## Approx Negentropy 1.073235748
## MC Negentropy     1.072869540
## MC se             0.003234728
## Relative accuracy 1.000341335
```

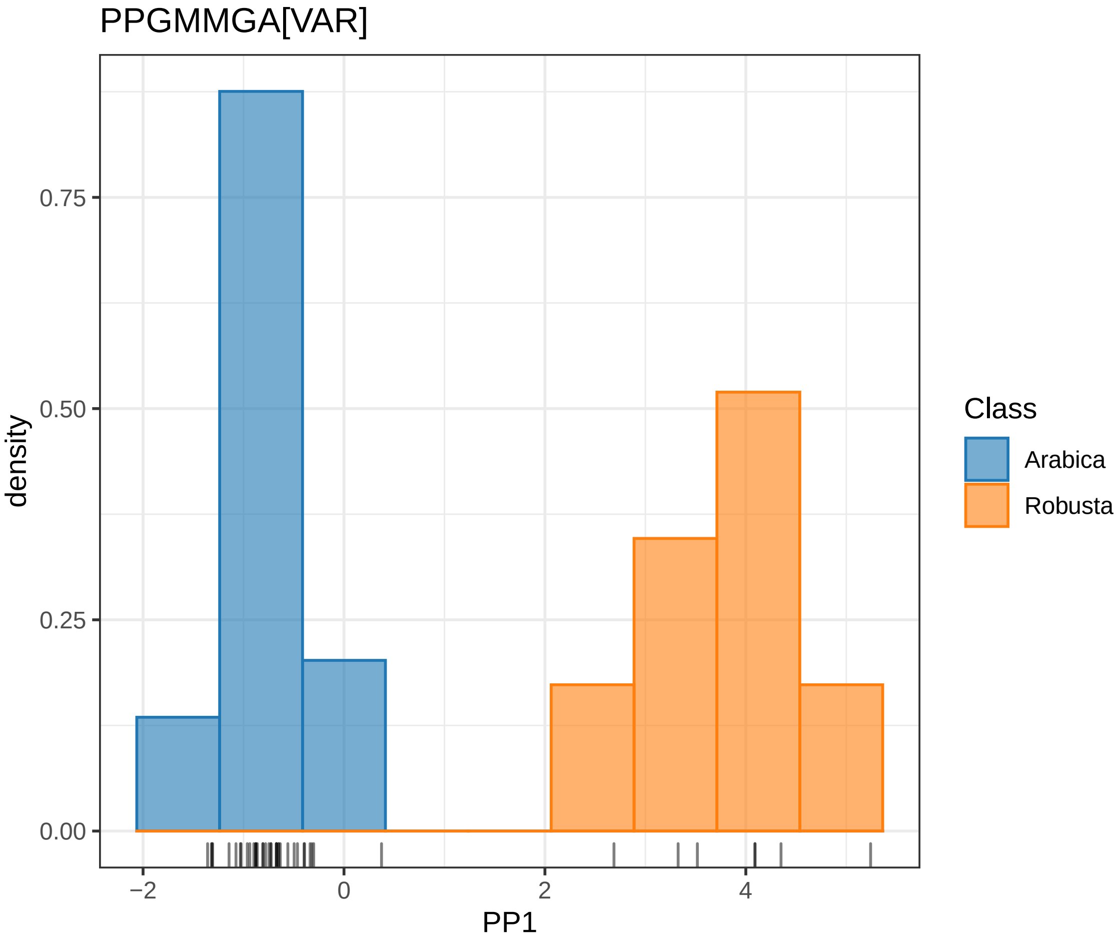
```
plot(PPGMMGA1, Class, bins = 9) + ggtitle("PPGMMGA[UT]")
```



```
PPGMMGA2 <- ppgmmga(data = X, d = 1, GMM = GMM, approx = "VAR", seed = 2)
summary(PPGMMGA2, check = TRUE)
```

```
## -- ppgmmga ----------------------------
##
## Data dimensions                = 43 x 12
## Data transformation            = center & scale
## Projection subspace dimension = 1
## GMM density estimate           = (VEI,3)
## Negentropy approximation       = VAR
## GA optimal negentropy          = 1.073043
## GA encoded basis solution:
##            x1       x2       x3       x4       x5       x6       x7
## [1,] 3.985399 1.335038 1.470806 1.514711 1.715593 0.982995 2.795832
##            x8       x9      x10      x11
## [1,] 1.479609 0.260608 0.975056 1.747732
##
## Estimated projection basis:
##                           PP1
## Water                -0.0421133
## Bean Weight          -0.0374603
## Extract Yield        -0.0100781
## ph Value              0.0388146
## Free Acid             0.2593937
## Mineral Content       0.0245482
## Fat                  -0.7483482
## Caffeine              0.5300507
## Trigonelline         -0.1393792
## Chlorogenic Acid      0.0542327
## Neochlorogenic Acid  0.0970621
## Isochlorogenic Acid  0.2335802
##
## Monte Carlo Negentropy approximation check:
##                              VAR
## Approx Negentropy 1.073043207
## MC Negentropy     1.074049801
## MC se             0.003226833
## Relative accuracy 0.999062805
```

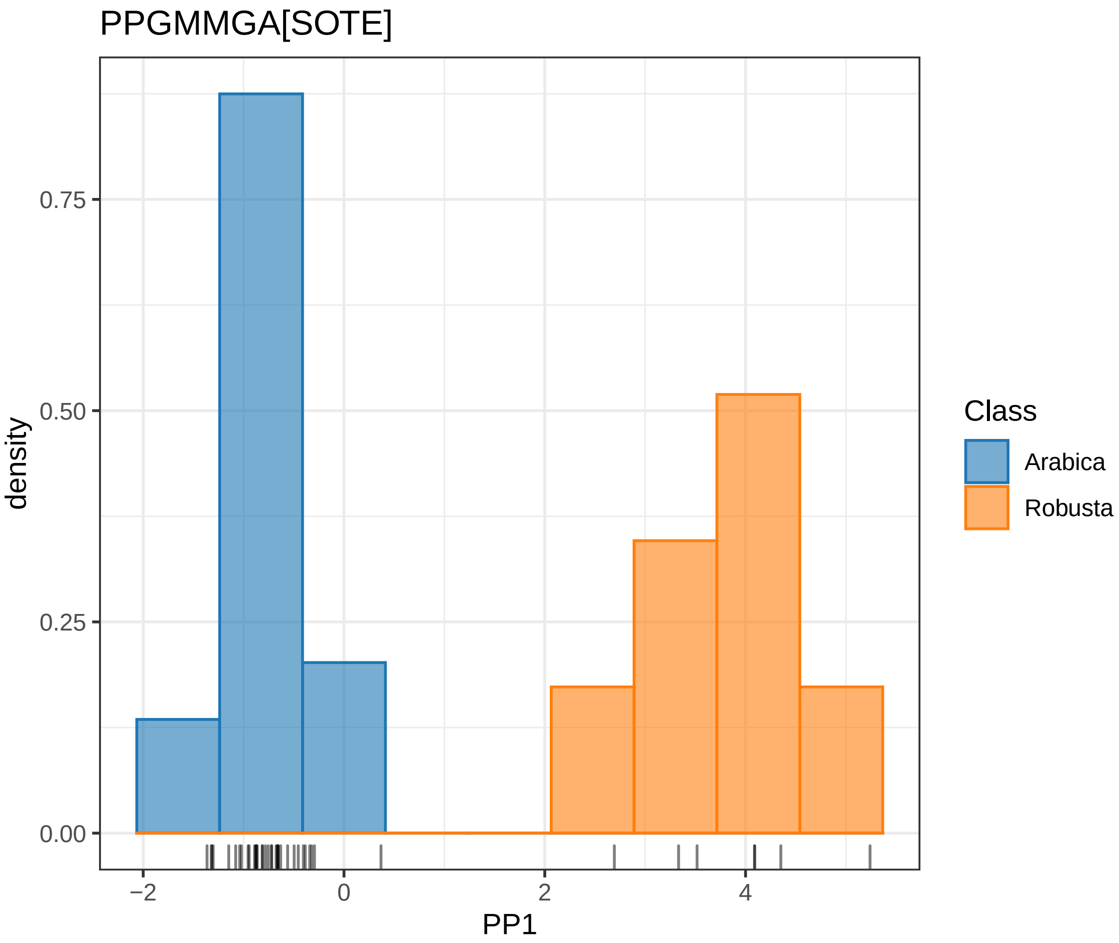
```
plot(PPGMMGA2, Class, bins = 9) + ggtitle("PPGMMGA[VAR]")
```



```
PPGMMGA3 <- ppgmmga(data = X, d = 1, GMM = GMM, approx = "SOTE", seed = 3)
summary(PPGMMGA3, check = TRUE)
```

```
## -- ppgmmga -----------------------------
##
## Data dimensions                = 43 x 12
## Data transformation            = center & scale
## Projection subspace dimension = 1
## GMM density estimate           = (VEI,3)
## Negentropy approximation      = SOTE
## GA optimal negentropy         = 1.07323
## GA encoded basis solution:
##             x1       x2       x3       x4       x5       x6        x7
## [1,] 0.8329046 1.805445 1.671062 1.627284 1.426087 2.156657 0.3485631
##            x8       x9      x10      x11
## [1,] 1.662615 2.889511 2.143055 1.390861
##
## Estimated projection basis:
##                            PP1
## Water               -0.0413558
## Bean Weight         -0.0376020
## Extract Yield       -0.0101674
## ph Value             0.0365959
## Free Acid            0.2623787
## Mineral Content      0.0249475
## Fat                 -0.7487262
## Caffeine             0.5286264
## Trigonelline        -0.1393256
## Chlorogenic Acid     0.0546344
## Neochlorogenic Acid  0.0973543
## Isochlorogenic Acid  0.2325015
##
## Monte Carlo Negentropy approximation check:
##                          SOTE
## Approx Negentropy 1.073229545
## MC Negentropy     1.075293394
## MC se             0.003211312
## Relative accuracy 0.998080664
```

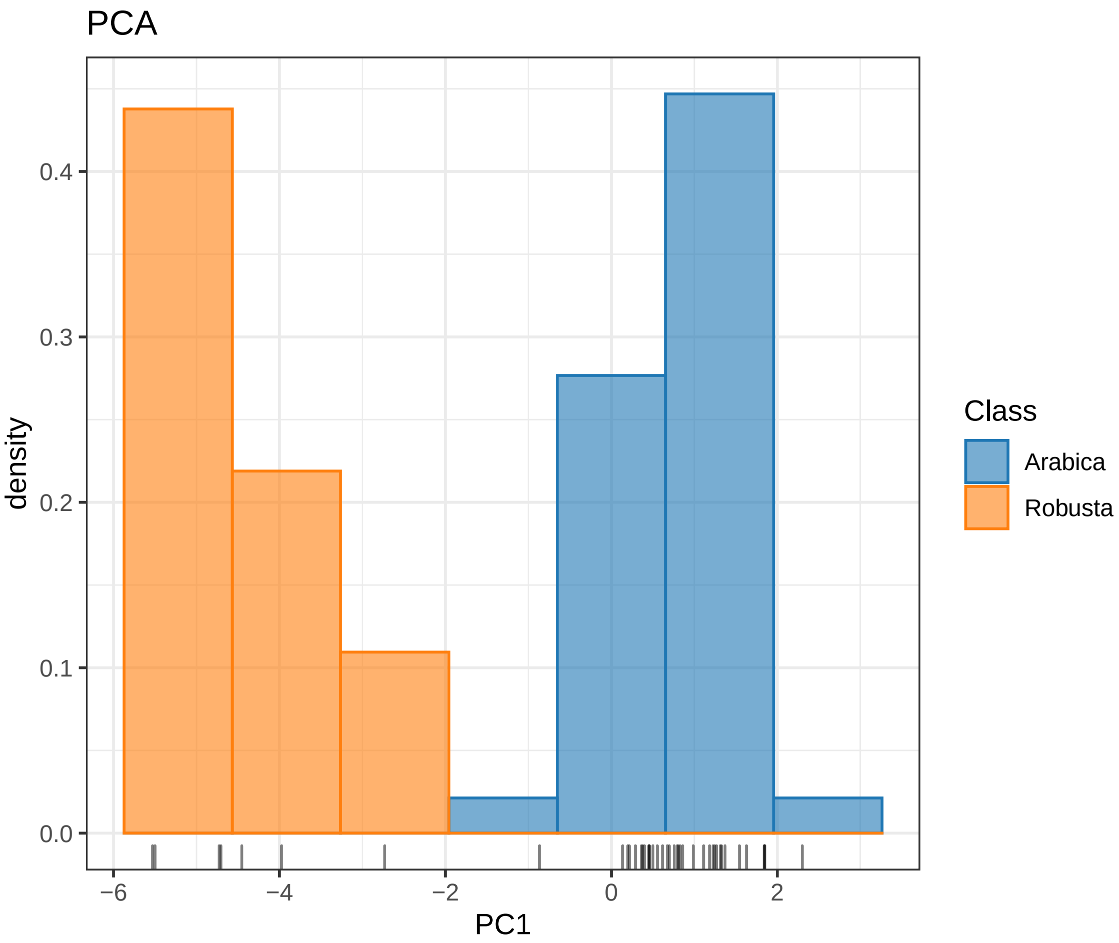
```
plot(PPGMMGA3, Class, bins = 9) + ggtitle("PPGMMGA[SOTE]")
```



```
PCA <- NegentropyPCA(PPGMMGA1)
PCA[c("Negentropy", "se")]
```

```
## $Negentropy
## [1] 0.7730807
##
## $se
## [1] 0.003177824
```

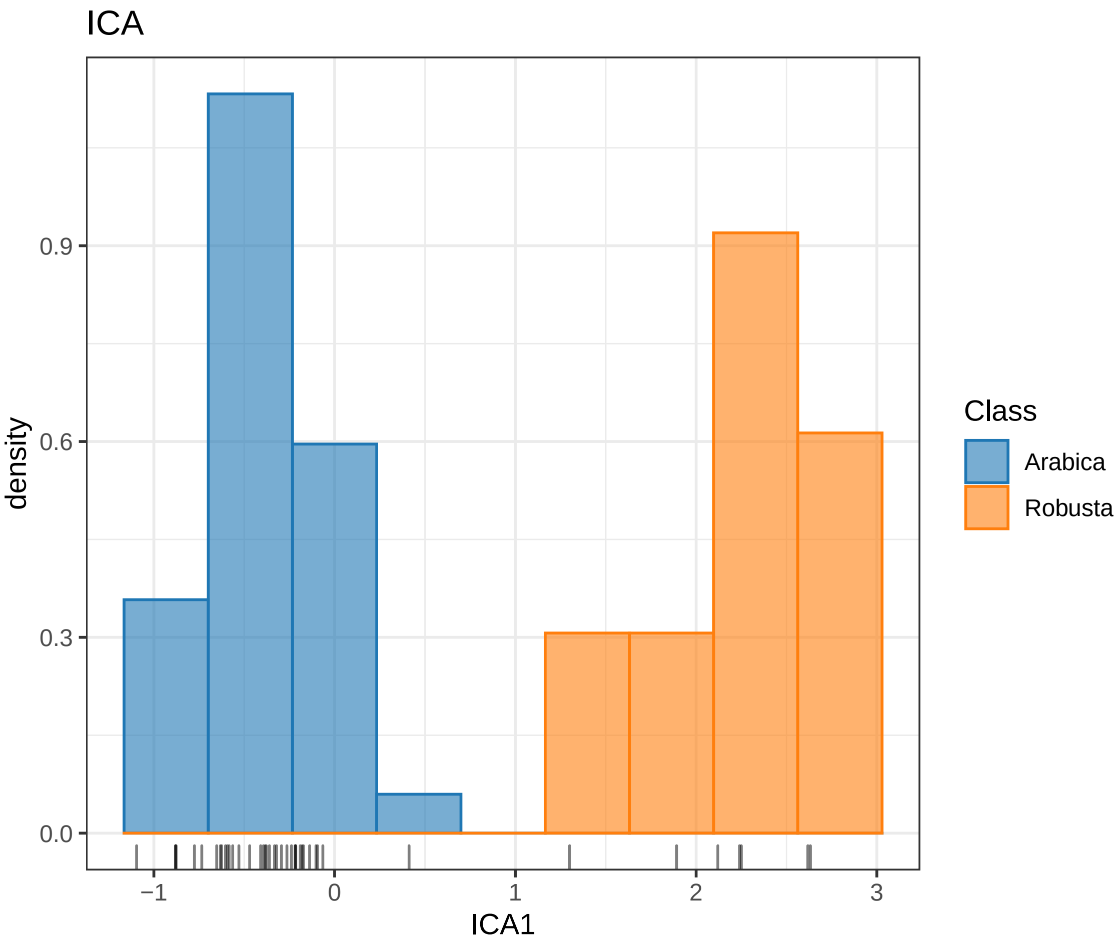
```
PPGMMPCA <- PPGMMGA1; PPGMMPCA$approx <- "PCA"
PPGMMPCA$basis <- PCA$basis
PPGMMPCA$Z <- PCA$Z
plot(PPGMMPCA, Class, nbins = 9) + ggtitle("PCA")
```



```
ICA <- NegentropyFASTICA(PPGMMGA1)
ICA[c("Negentropy", "se")]
```

```
## $Negentropy
## [1] 0.7681797
##
## $se
## [1] 0.003178032
```

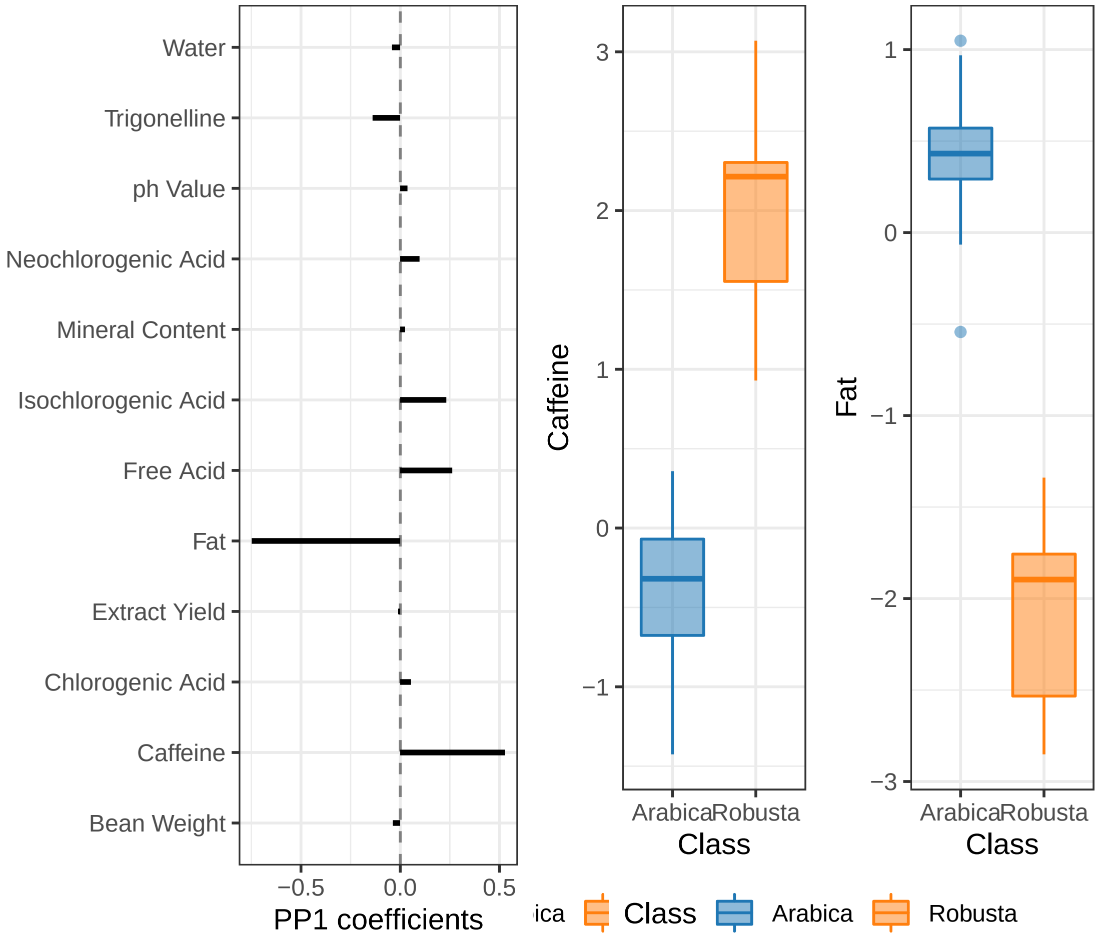
```
# trick for plotting
PPGMMICA <- PPGMMGA1; PPGMMICA$approx <- "ICA"
PPGMMICA$basis <- ICA$basis
PPGMMICA$Z <- ICA$Z
plot(PPGMMICA, Class, bins = 9) + ggtitle("ICA")
```



```
ppgmmga.table.angle(PPGMMGA1, PPGMMGA2, PPGMMGA3, PPGMMPCA, PPGMMICA)
```

```
##      UT       VAR        SOTE      PCA          ICA
## UT    0 0.2452342 0.003848958 34.12101 3.412101e+01
## VAR  NA 0.0000000 0.244290645 34.16538 3.416538e+01
## SOTE NA        NA 0.000000000 34.12119 3.412119e+01
## PCA  NA        NA          NA  0.00000 5.565848e-14
## ICA  NA        NA          NA       NA 0.000000e+00
```

```
df <- data.frame(variable = rownames(PPGMMGA1$basis), coefs = PPGMMGA1$basis[,1])
plot1 <- ggplot(df, aes(x = variable)) +
  geom_hline(yintercept = 0, colour = gray(1/2), lty = 2) +
  geom_linerange(aes(ymin = ifelse(coefs < 0, coefs, 0),
                     ymax = ifelse(coefs > 0, coefs, 0)),
                 lwd = 1, position = position_dodge(width = 1/2)) +
  xlab("") + ylab("PP1 coefficients") +
  coord_flip() +
  theme_bw()
df <- cbind(data.frame(X, check.names = FALSE)[c("Caffeine", "Fat")], Class)
plot2 <- ggplot(df, aes(Class, Caffeine, fill = Class, color = Class)) +
  geom_boxplot(outlier.shape = 19, alpha = 1/2) +
  scale_fill_tableau("Classic 10") +
  scale_colour_tableau("Classic 10")
plot3 <- ggplot(df, aes(Class, Fat, fill = Class, color = Class)) +
  geom_boxplot(outlier.shape = 19, alpha = 1/2) +
  scale_fill_tableau("Classic 10") +
  scale_colour_tableau("Classic 10")
plots <- grid.arrange(plot1,
                      multiple_ggplot_sharedLegend(plot2, plot3, nrow = 1,
                                                   position = "bottom"),
                      nrow = 1, widths = c(1,1))
```



```
# AIS data --------------------------------------------------------------

data(ais, package = "dr")
X <- ais[,2:12]
Class <- factor(ifelse(ais$Sex == 0, "M", "F"))

X <- scale(X, center = TRUE, scale = TRUE)
GMM <- densityMclust(X)

# d = 1
PPGMMGA1 <- ppgmmga(data = X, d = 1, GMM = GMM, approx = "UT", seed = 1)
summary(PPGMMGA1, check = TRUE)
```

```
## -- ppgmmga ----------------------------
##
## Data dimensions                 = 202 x 11
## Data transformation             = center & scale
## Projection subspace dimension = 1
## GMM density estimate          = (EVE,5)
## Negentropy approximation      = UT
## GA optimal negentropy         = 0.2716425
## GA encoded basis solution:
##            x1      x2       x3       x4       x5       x6       x7
## [1,] 2.578943 1.71223 1.570521 1.084439 1.532505 1.574244 1.506406
##            x8       x9      x10
## [1,] 1.552572 1.512312 1.401586
##
## Estimated projection basis:
##             PP1
## Ht   -0.4580431
## Wt    0.7263071
## LBM  -0.1467000
## RCC  -0.0510051
## WCC  -0.0159048
## Hc   -0.0562744
## Hg    0.0030150
## Ferr -0.0335047
## BMI  -0.4627420
## SSF  -0.0002721
## Bfat  0.1409627
##
## Monte Carlo Negentropy approximation check:
##                              UT
## Approx Negentropy 0.271642502
## MC Negentropy     0.223630430
## MC se             0.001839646
## Relative accuracy 1.214693824
```

```
PPGMMGA2 <- ppgmmga(data = X, d = 1, GMM = GMM, approx = "VAR", seed = 2)
summary(PPGMMGA2, check = TRUE)
```

```
## -- ppgmmga ----------------------------
##
## Data dimensions               = 202 x 11
## Data transformation           = center & scale
## Projection subspace dimension = 1
## GMM density estimate          = (EVE,5)
## Negentropy approximation      = VAR
## GA optimal negentropy         = 0.2283456
## GA encoded basis solution:
##           x1       x2       x3        x4       x5       x6       x7
## [1,] 2.50211 1.248956 1.744734 0.9432044 1.575921 1.631473 1.549986
##            x8       x9      x10
## [1,] 1.597335 1.616311 1.758922
##
## Estimated projection basis:
##             PP1
## Ht   -0.4418482
## Wt    0.5940894
## LBM   0.1409526
## RCC   0.0343272
## WCC   0.0200273
## Hc   -0.0157082
## Hg    0.0458607
## Ferr  0.0038760
## BMI  -0.5486422
## SSF   0.1641757
## Bfat -0.3163130
##
## Monte Carlo Negentropy approximation check:
##                           VAR
## Approx Negentropy 0.228345588
## MC Negentropy     0.307133717
## MC se             0.002420424
## Relative accuracy 0.743472879
```

```
PPGMMGA3 <- ppgmmga(data = X, d = 1, GMM = GMM, approx = "SOTE", seed = 3)
summary(PPGMMGA3, check = TRUE)
```

```
## -- ppgmmga ------------------------------
##
## Data dimensions               = 202 x 11
## Data transformation           = center & scale
## Projection subspace dimension = 1
## GMM density estimate          = (EVE,5)
## Negentropy approximation      = SOTE
## GA optimal negentropy         = 0.5528521
## GA encoded basis solution:
##            x1       x2       x3       x4       x5       x6       x7
## [1,] 2.829624 1.458553 1.559532 1.273174 1.552948 1.577181 1.534831
##            x8       x9      x10
## [1,] 1.570474 1.553346 1.080926
##
## Estimated projection basis:
##             PP1
## Ht   -0.2570312
## Wt    0.7969976
## LBM  -0.4465301
## RCC  -0.0165622
## WCC  -0.0003059
## Hc   -0.0341522
## Hg    0.0060642
## Ferr -0.0169542
## BMI  -0.2913842
## SSF  -0.0111927
## Bfat -0.1120075
##
## Monte Carlo Negentropy approximation check:
##                          SOTE
## Approx Negentropy 0.552852125
## MC Negentropy     0.099712482
## MC se             0.002023242
## Relative accuracy 5.544462602
```

```
PCA <- NegentropyPCA(PPGMMGA1)
PCA[c("Negentropy", "se")]
```

```
## $Negentropy
## [1] 0.1255747
```

```
##
## $se
## [1] 0.001943279
```

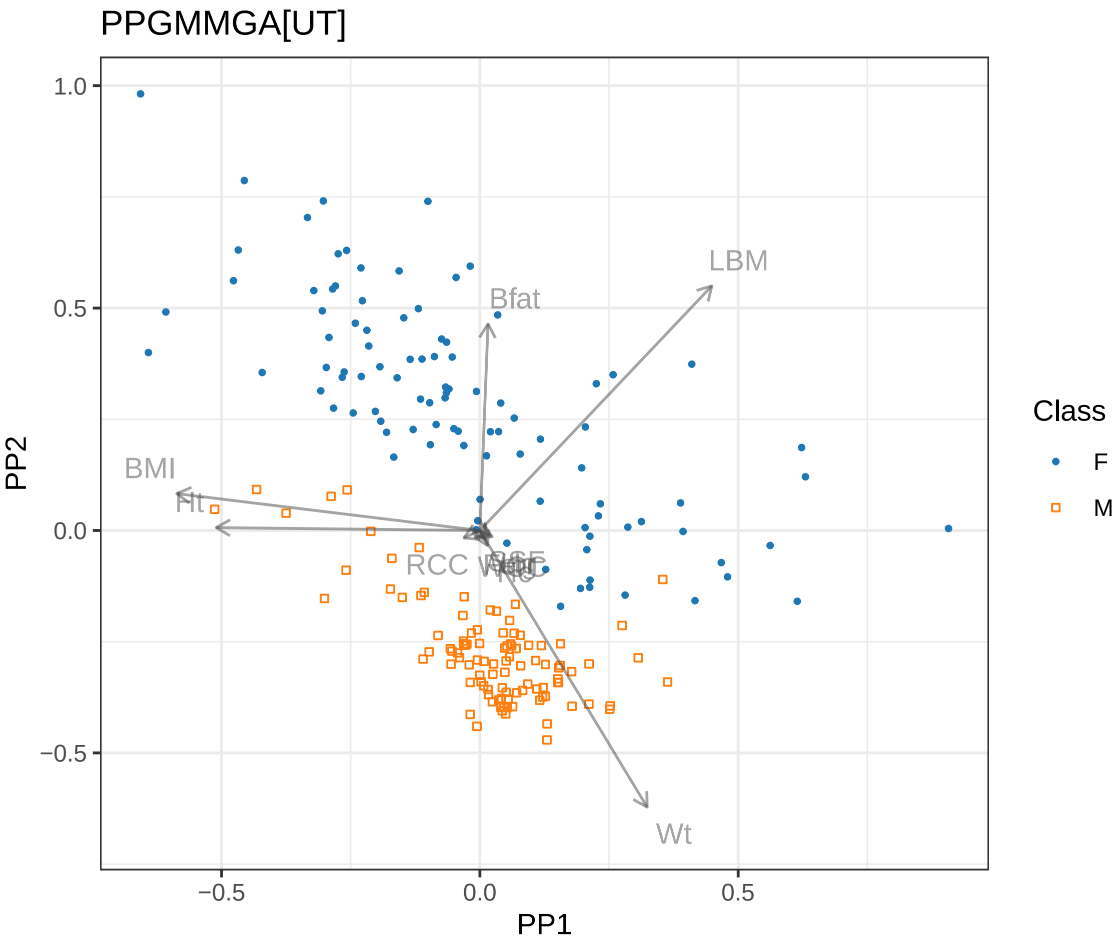
```
PPGMMPCA <- PPGMMGA1; PPGMMPCA$approx <- "PCA"
PPGMMPCA$basis <- PCA$basis
PPGMMPCA$Z <- PCA$Z

ppgmmga.table.angle(PPGMMGA1, PPGMMGA2, PPGMMGA3, PPGMMPCA)
```

```
##      UT      VAR     SOTE      PCA
## UT    0 34.78798 27.80871 81.52592
## VAR  NA  0.00000 44.13108 86.59210
## SOTE NA       NA  0.00000 87.06107
## PCA  NA       NA       NA  0.00000
```

```
# d = 2
PPGMMGA1 <- ppgmmga(data = X, d = 2, GMM = GMM, approx = "UT", seed = 1)
summary(PPGMMGA1, check = TRUE)
```

```
## -- ppgmmga -----------------------------
##
## Data dimensions               = 202 x 11
## Data transformation           = center & scale
## Projection subspace dimension = 2
## GMM density estimate          = (EVE,5)
## Negentropy approximation      = UT
## GA optimal negentropy         = 0.5225648
## GA encoded basis solution:
##            x1       x2       x3        x4       x5       x6       x7
## [1,] 2.136126 1.587227 1.591097 0.9096454 1.577396 1.602194 1.590791
##            x8       x9      x10  ...       x19      x20
## [1,] 1.586196 1.529604 2.209708       1.561785 1.395024
##
## Estimated projection basis:
##             PP1        PP2
## Ht   -0.5339346  0.0068092
## Wt    0.3387279 -0.6504408
## LBM   0.4697174  0.5751101
## RCC  -0.0324651 -0.0175501
## WCC   0.0121411 -0.0227540
## Hc    0.0157670 -0.0354448
## Hg    0.0247683 -0.0148731
## Ferr  0.0052070 -0.0175084
## BMI  -0.6138163  0.0867238
## SSF   0.0202965 -0.0100373
## Bfat  0.0164301  0.4856925
##
## Monte Carlo Negentropy approximation check:
##                            UT
## Approx Negentropy 0.522564776
## MC Negentropy     0.543037957
## MC se             0.003740006
## Relative accuracy 0.962298804
```

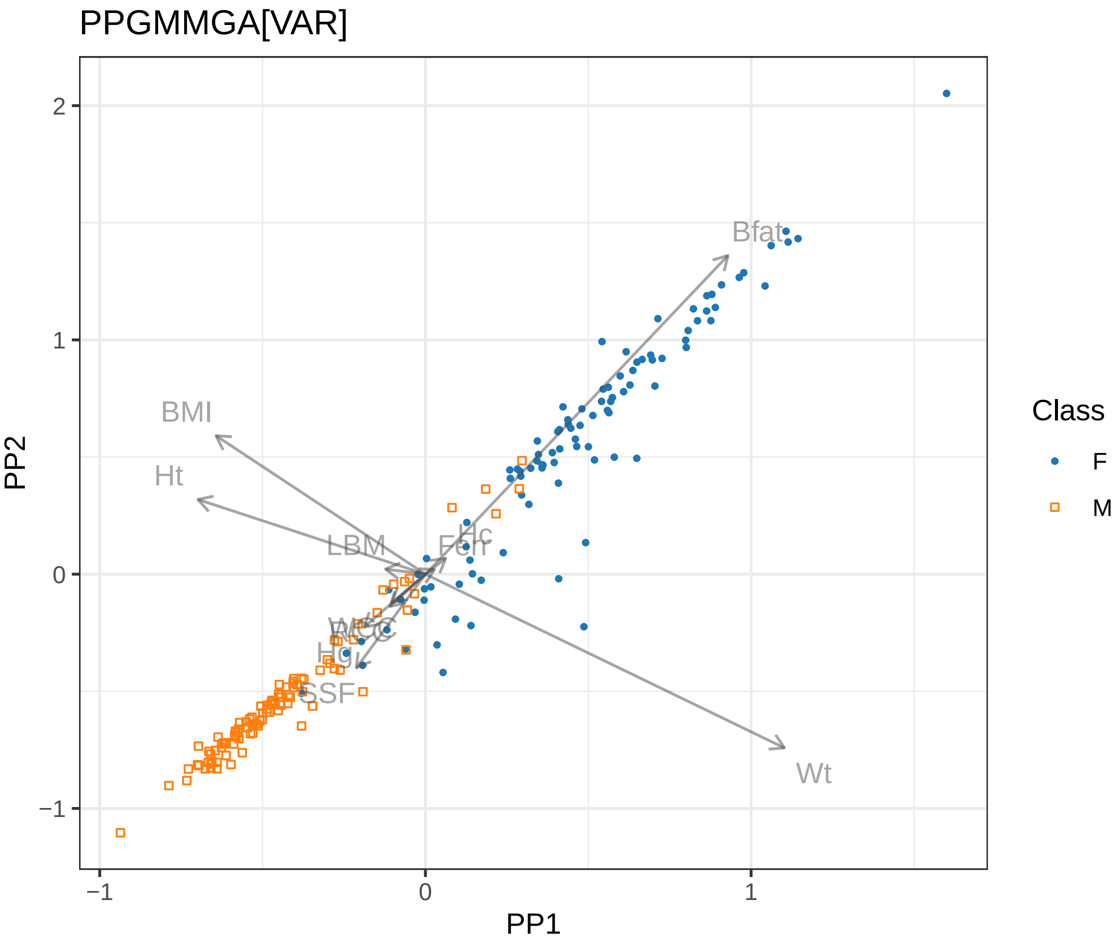
```
plot(PPGMMGA1, Class) + ggtitle("PPGMMGA[UT]")
```



```
PPGMMGA2 <- ppgmmga(data = X, d = 2, GMM = GMM, approx = "VAR", seed = 2)
summary(PPGMMGA2, check = TRUE)
```

```
## -- ppgmmga -----------------------------
##
## Data dimensions                  = 202 x 11
## Data transformation              = center & scale
## Projection subspace dimension = 2
## GMM density estimate             = (EVE,5)
## Negentropy approximation         = VAR
## GA optimal negentropy            = 0.6644692
## GA encoded basis solution:
##            x1       x2       x3       x4       x5       x6       x7
## [1,] 2.576889 2.126279 1.428576 1.121862 1.591549 1.428802 1.617856
##            x8       x9      x10  ...        x19      x20
## [1,] 1.493145 1.487974 1.477187       0.7094031 1.214405
##
## Estimated projection basis:
##             PP1        PP2
## Ht   -0.3965755  0.1810347
## Wt    0.6259853 -0.4212152
## LBM  -0.0695712  0.0122557
## RCC  -0.0617853 -0.0783384
## WCC  -0.0581105 -0.0685847
## Hc    0.0352792  0.0384462
## Hg   -0.1072089 -0.1285989
## Ferr  0.0157236  0.0114947
## BMI  -0.3650283  0.3355181
## SSF  -0.1204297 -0.2272049
## Bfat  0.5273530  0.7723153
##
## Monte Carlo Negentropy approximation check:
##                             VAR
## Approx Negentropy 0.664469205
## MC Negentropy     0.861798543
## MC se             0.003870203
## Relative accuracy 0.771026141
```

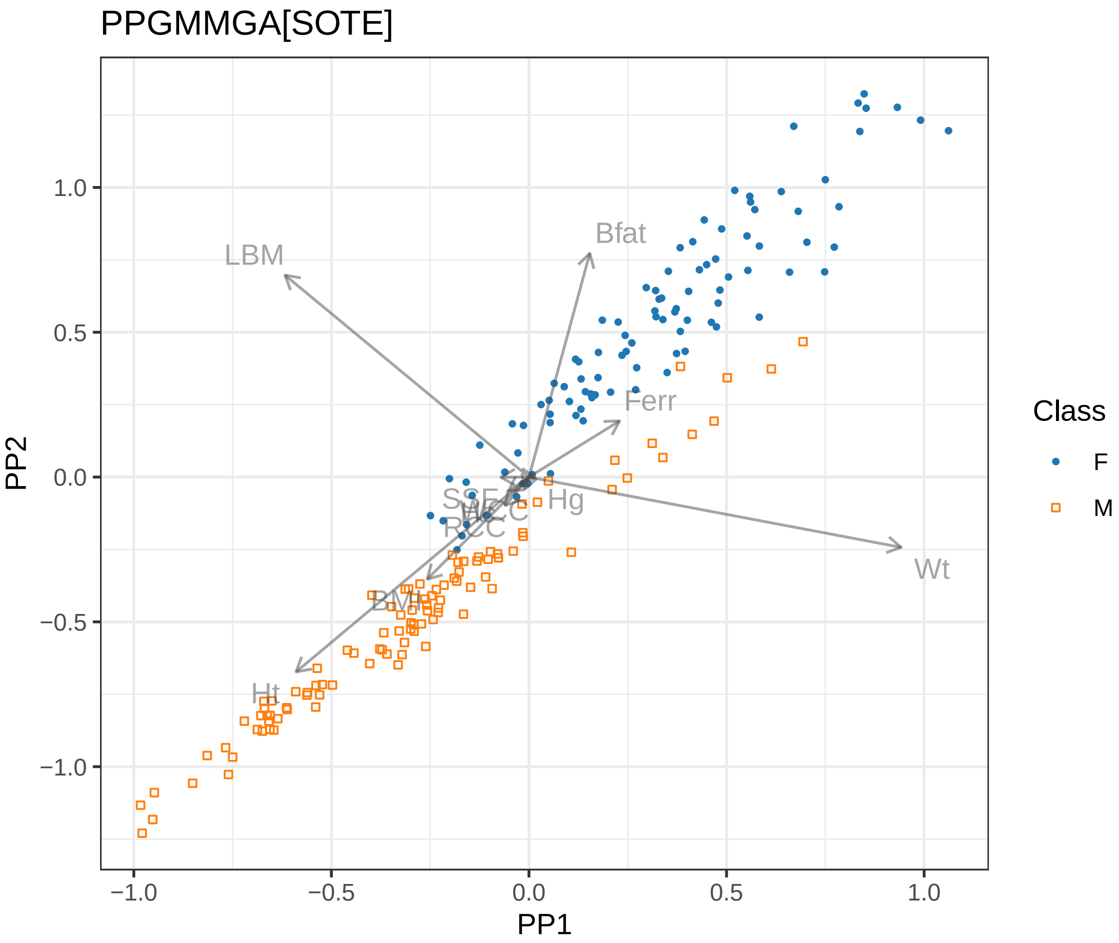
```
plot(PPGMMGA2, Class) + ggtitle("PPGMMGA[VAR]")
```



```
PPGMMGA3 <- ppgmmga(data = X, d = 2, GMM = GMM, approx = "SOTE", seed = 3)
summary(PPGMMGA3, check = TRUE)
```

```
## -- ppgmmga ----------------------------
##
## Data dimensions                  = 202 x 11
## Data transformation              = center & scale
## Projection subspace dimension = 2
## GMM density estimate             = (EVE,5)
## Negentropy approximation         = SOTE
## GA optimal negentropy         = 1.042832
## GA encoded basis solution:
##            x1       x2       x3       x4      x5       x6       x7
## [1,] 2.582654 1.686772 1.517114 1.374397 1.74885 1.582927 1.530416
##            x8       x9      x10  ...       x19       x20
## [1,] 1.562764 1.523765 1.063616       1.658494 0.7380267
##
## Estimated projection basis:
##             PP1        PP2
## Ht   -0.4428896 -0.5055384
## Wt    0.7080881 -0.1825068
## LBM  -0.4640835  0.5240683
## RCC  -0.0449702 -0.0739108
## WCC  -0.0076830 -0.0309834
## Hc   -0.0386473 -0.0359108
## Hg    0.0116134 -0.0015633
## Ferr  0.1722937  0.1456523
## BMI  -0.1935485 -0.2649471
## SSF  -0.0532966 -0.0005336
## Bfat  0.1157156  0.5808166
##
## Monte Carlo Negentropy approximation check:
##                          SOTE
## Approx Negentropy 1.042831789
## MC Negentropy     0.235172122
## MC se             0.003039366
## Relative accuracy 4.434334221
```

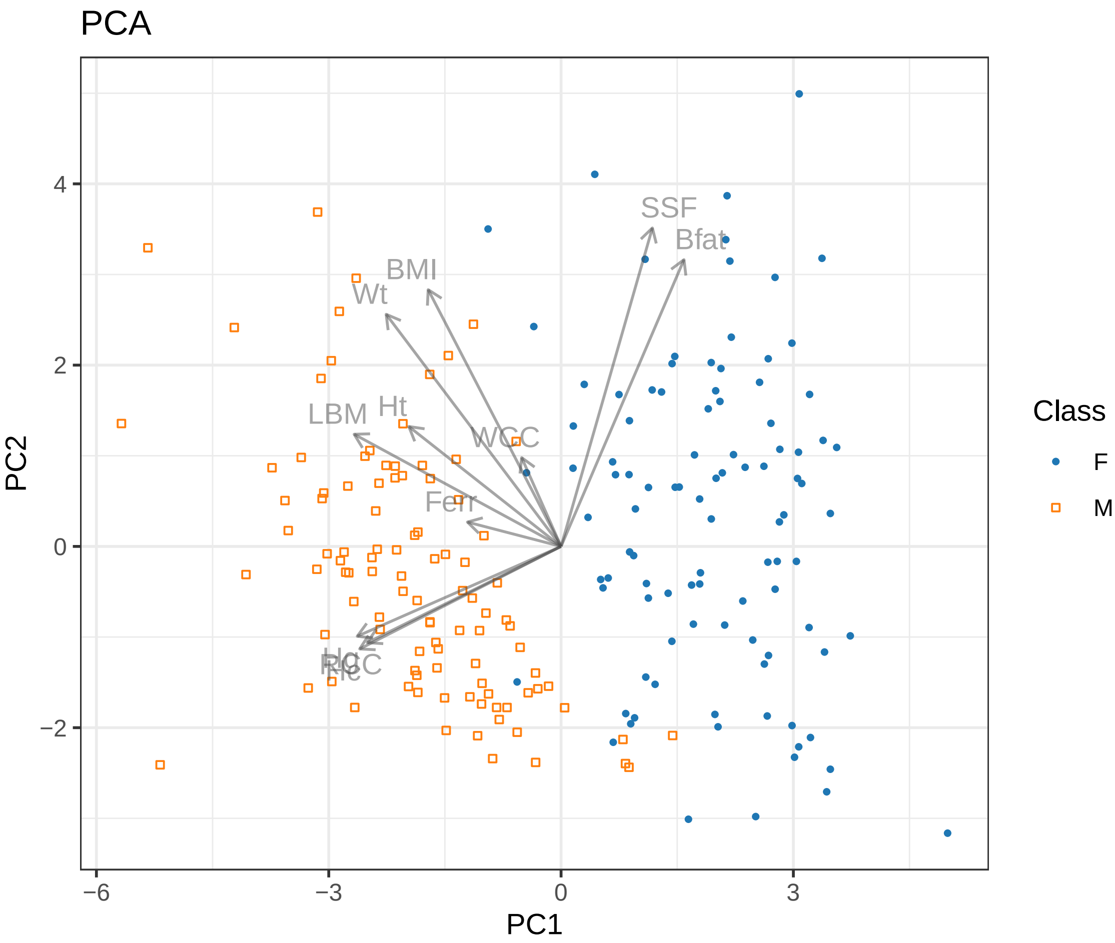
```
plot(PPGMMGA3, Class) + ggtitle("PPGMMGA[SOTE]")
```



```
PCA <- NegentropyPCA(PPGMMGA1)
PCA[c("Negentropy", "se")]
```

```
## $Negentropy
## [1] 0.2703419
##
## $se
## [1] 0.003189416
```

```
PPGMMPCA <- PPGMMGA1; PPGMMPCA$approx <- "PCA"
PPGMMPCA$basis <- PCA$basis
PPGMMPCA$Z <- PCA$Z
plot(PPGMMPCA, Class) + ggtitle("PCA")
```



```
ppgmmga.table.angle(PPGMMGA1, PPGMMGA2, PPGMMGA3, PPGMMPCA)
```

```
##      UT      VAR     SOTE      PCA
## UT    0 70.37841 46.07293 85.28613
## VAR  NA  0.00000 61.38057 88.51716
## SOTE NA       NA  0.00000 89.65081
## PCA  NA       NA       NA  0.00000
```

```
# Leukemia data ---------------------------------------------------------

data(golub, package = "multtest")
X <- t(as.matrix(golub))
X <- scale(X, center = TRUE, scale = FALSE)
dim(X)
```

```
## [1]   38 3051
```

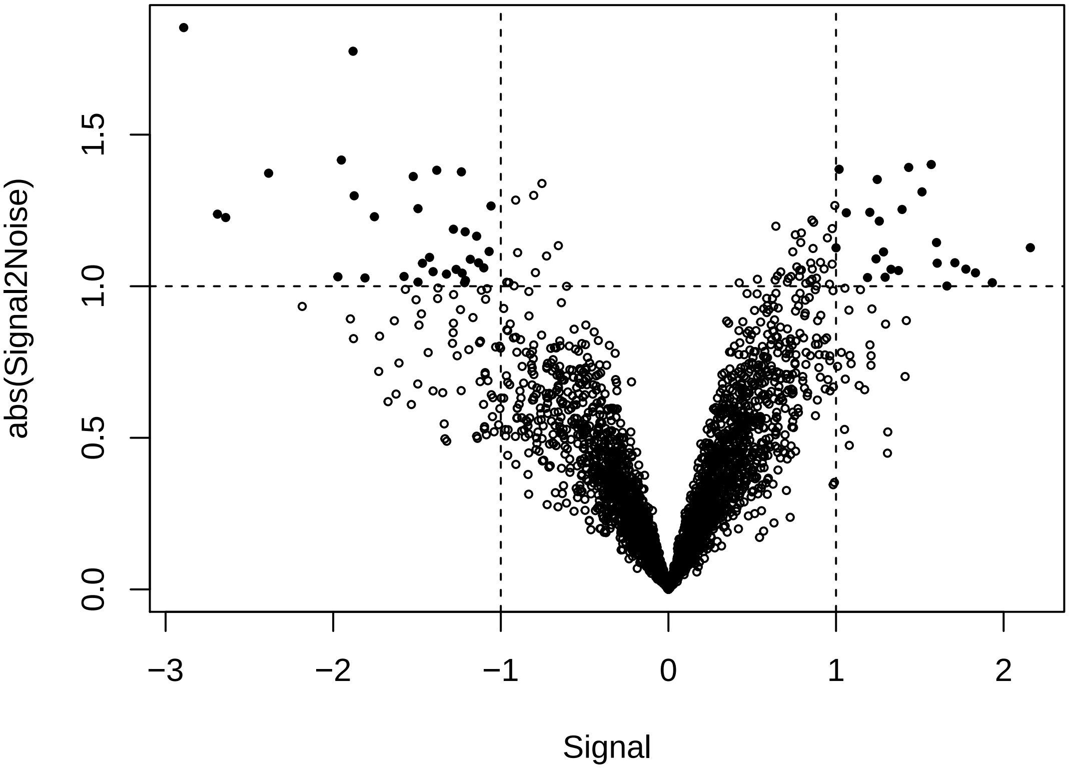
```
Class <- factor(golub.cl, levels = 0:1, labels = c("ALL", "AML"))
table(Class)
```

```
## Class
## ALL AML
##  27  11
```

```
GMM <- densityMclust(X, modelNames = c("EII", "VII", "EEI", "EVI", "VEI", "VVI"))
summary(GMM)
```

```
## -------------------------------------------------------
## Density estimation via Gaussian finite mixture modeling
## -------------------------------------------------------
##
## Mclust VVI (diagonal, varying volume and shape) model with 2 components:
##
##  log-likelihood  n    df       BIC       ICL
##       -71403.95 38 12205 -187204.6 -187204.6
```

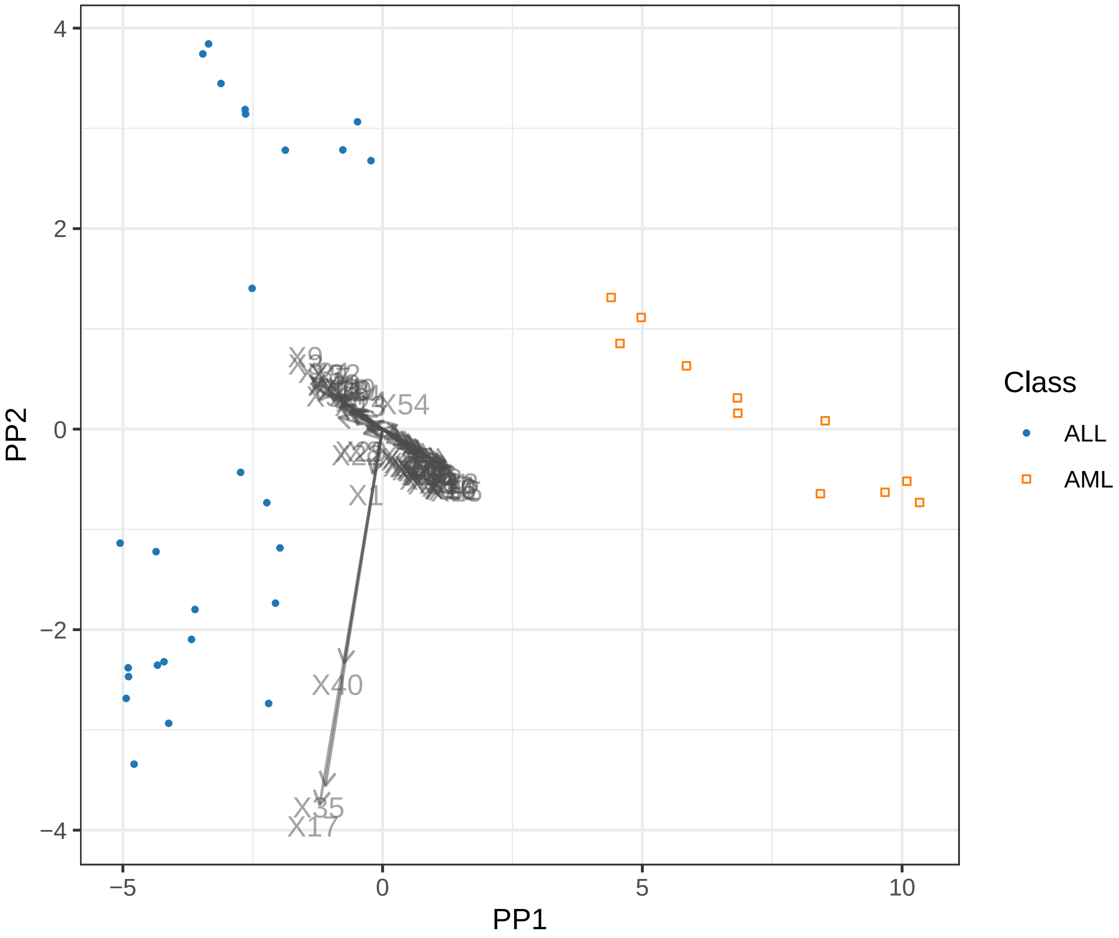
```
m <- GMM$parameters$mean
s <- sqrt(apply(GMM$parameters$variance$sigma, 3, diag))
Signal <- (m[,1]-m[,2])
Signal2Noise <- (m[,1]-m[,2])/(s[,1]+s[,2])
expr <- (abs(Signal) > 1 & abs(Signal2Noise) > 1)
plot(Signal, abs(Signal2Noise),
     cex = 0.5, pch = ifelse(expr, 19, 1))
abline(h = c(-1,1), v = c(-1,1), lty = 2)
```



```
g <- which(expr)

GMM2 <- densityMclust(X[,g], modelNames = c("EII", "VII", "EEI", "EVI", "VEI", "VVI"))
PPGMMGA <- ppgmmga(data = X[,g], d = 2, approx = "UT", seed = 1,
                   GMM = GMM2, scale = FALSE,
                   options = ppgmmga.options(maxiter = 2000))
summary(PPGMMGA, check = TRUE)
```

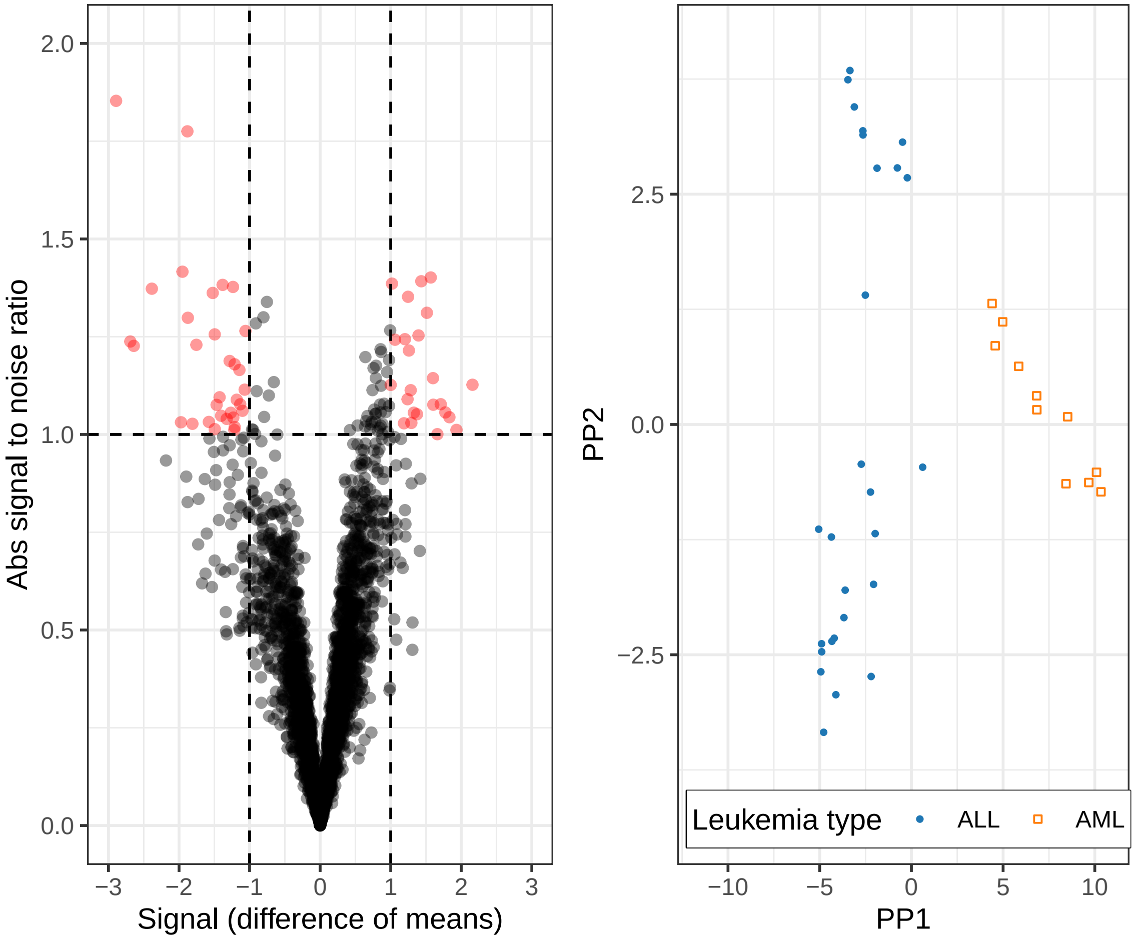
```
## -- ppgmmga ----------------------------
##
## Data dimensions               = 38 x 56
## Data transformation           = center
## Projection subspace dimension = 2
## GMM density estimate          = (EVI,3)
## Negentropy approximation      = UT
## GA optimal negentropy         = 3.052731
## GA encoded basis solution:
##            x1       x2       x3       x4       x5       x6       x7
## [1,] 2.961383 1.706882 1.427382 1.617797 1.663534 1.785555 1.710749
##            x8       x9      x10  ...     x109    x110
## [1,] 1.673226 1.688247 1.425894       1.51975 1.49349
##
## Estimated projection basis:
##              PP1        PP2
## [1,]  -0.0318658 -0.0738131
## [2,]   0.1749085 -0.0535659
## [3,]  -0.2254643  0.0729786
## [4,]  -0.0997140  0.0336894
## [5,]  -0.1557879  0.0556949
## [6,]   0.0978273 -0.0336522
## [7,]  -0.1370882  0.0509615
## [8,]  -0.0365645 -0.0012691
## [9,]  -0.2267972  0.0853027
## [10,]  0.1359959 -0.0516087
## [11,]  0.0834733 -0.0260896
## [12,]  0.1975309 -0.0643125
## [13,]  0.2036362 -0.0545833
## [14,] -0.0467671  0.0110721
## [15,]  0.2158926 -0.0677566
## [16,]  0.1946148 -0.0619737
## [17,] -0.2026186 -0.6269986
## [18,] -0.1036298  0.0310602
## [19,] -0.1061229  0.0281186
## [20,]  0.0552799 -0.0156133
##  ...
## [55,] -0.1416022  0.0196443
## [56,]  0.1356656 -0.0412995
##
## Monte Carlo Negentropy approximation check:
##                            UT
## Approx Negentropy 3.052731483
## MC Negentropy     3.052821017
## MC se             0.003639312
## Relative accuracy 0.999970672
```

```
plot(PPGMMGA, Class)
```



```
df <- data.frame(Signal, Signal2Noise, expr)
plot1 <- ggplot(df, aes(x = Signal, y = abs(Signal2Noise), color = expr)) +
  geom_point(alpha = 0.4) +
  scale_color_manual(values = c("black", "red")) +
  geom_hline(yintercept = 1, lty = 2) +
  geom_vline(xintercept = c(-1,1), lty = 2) +
  scale_x_continuous(breaks = seq(-3, 3, by = 1), limits = c(-3,3)) +
  xlab("Signal (difference of means)") +
  ylim(c(0, 2)) +
  ylab("Abs signal to noise ratio") +
  theme(legend.position = "none")

plot2 <-
plot(PPGMMGA, Class, drawAxis = FALSE) +
  scale_x_continuous(expand = expand_scale(c(0.5, 0.1))) +
  scale_y_continuous(expand = expand_scale(c(0.2, 0.1))) +
  guides(col = guide_legend(title = "Leukemia type"),
         pch = guide_legend(title = "Leukemia type")) +
  theme(legend.position = c(0.02, 0.02),
        legend.direction = "horizontal",
        legend.justification = c(0,0),
        legend.margin = margin(2, 2, 2, 2),
        legend.box.background = element_rect(colour = "black"))
```

```
plots <- grid.arrange(plot1, plot2, nrow = 1, ncol = 2)
```



```
# Restore default sample's behavior:
RNGkind(sample.kind = "default")
```