

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

Luca Scrucchi 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)

# 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$x
Class <- factor(x$class)
```

```

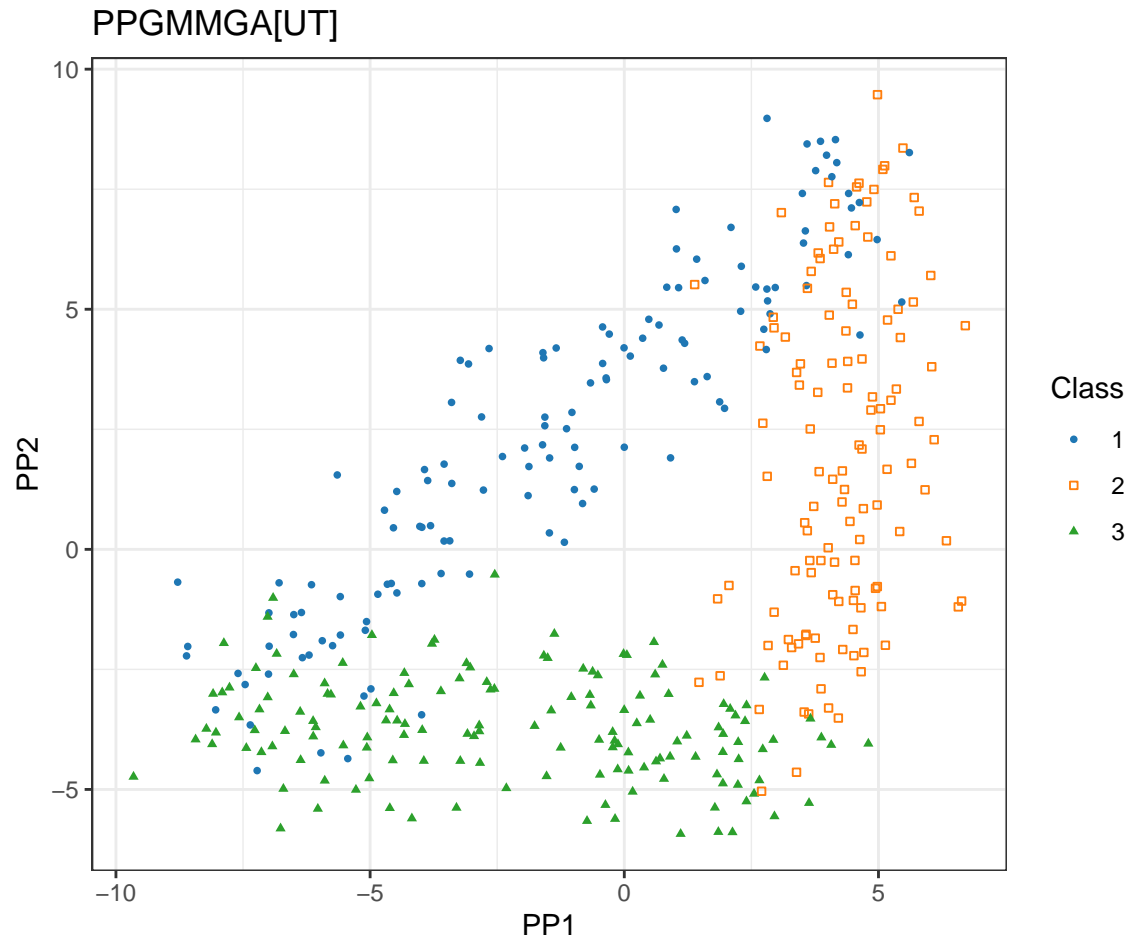
X <- scale(X, center = TRUE, scale = FALSE)
GMM <- densityMclust(X)

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    = 1.008164
## GA encoded basis solution:
##           x1      x2      x3      x4      x5      x6      x7
## [1,] 3.221607 1.562614 1.650441 1.767264 1.853727 1.982588 1.950518
##           x8      x9      x10    ...      x39 x40
## [1,] 2.016394 1.85695 1.583207      0.585661 0
##
## Estimated projection basis:
##           PP1      PP2
## [1,] 0.0000000 0.0000000
## [2,] 0.0000000 0.0000000
## [3,] 0.0000000 0.1761731
## [4,] 0.0000000 0.2656050
## [5,] 0.0000000 0.3632529
## [6,] 0.0791306 0.3516874
## [7,] 0.1908600 0.3341662
## [8,] 0.2643602 0.1655340
## [9,] 0.3638997 -0.0443326
## [10,] 0.3289603 -0.1867601
## [11,] 0.3222879 -0.3570488
## [12,] 0.1457335 -0.3578555
## [13,] -0.0085839 -0.3473052
## [14,] -0.2035061 -0.2387673
## [15,] -0.3443655 -0.1570678
## [16,] -0.3188717 -0.0775474
## [17,] -0.3757370 0.0074584
## [18,] -0.2729234 0.0395782
## [19,] -0.1945813 0.0300985
## [20,] -0.0795574 0.0198462
## [21,] 0.0081818 0.0092843
##
## Monte Carlo Negentropy approximation check:
##           UT
## Approx Negentropy 1.008164074
## MC Negentropy    1.027117785
## MC se            0.003117037
## Relative accuracy 0.981546702

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)
```

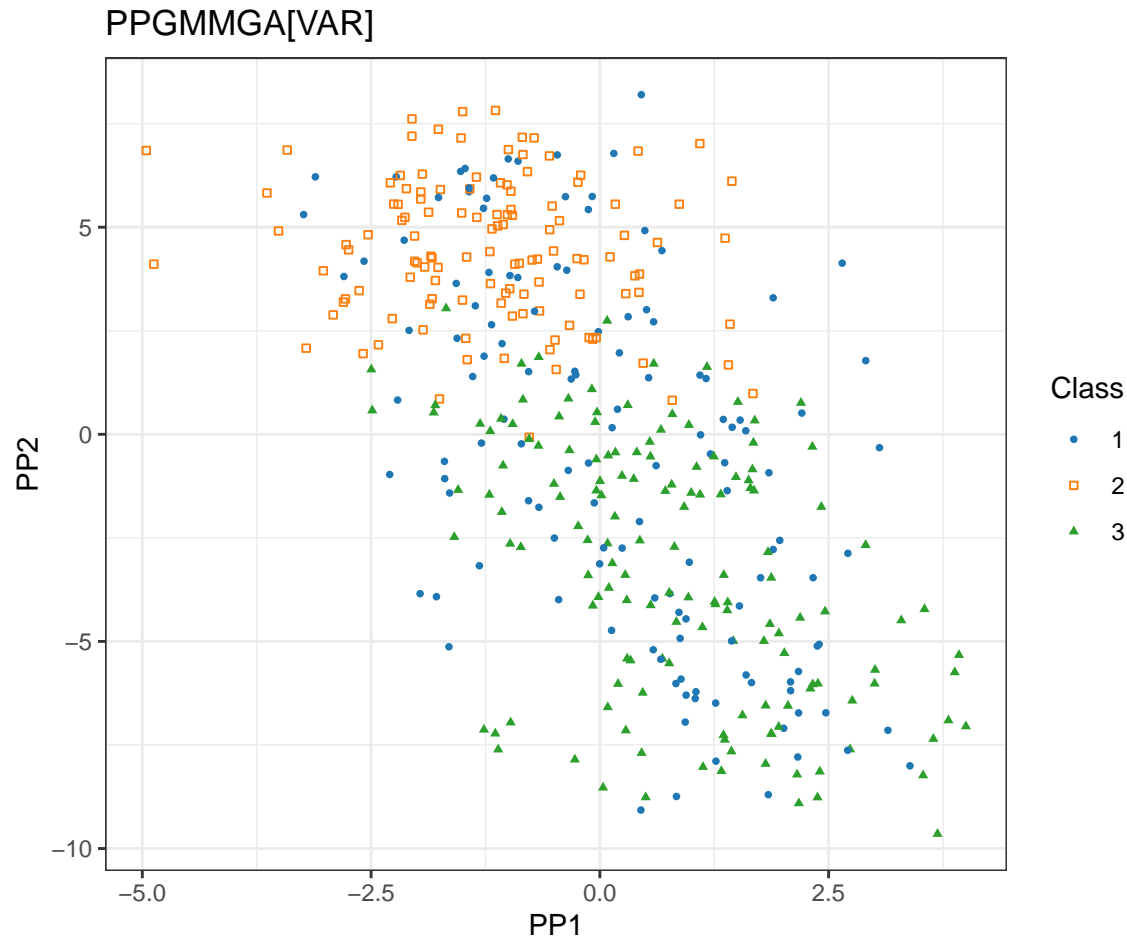
```
## -- ppgmmga -----
##
## Data dimensions           = 400 x 21
## Data transformation       = center
## Projection subspace dimension = 2
## GMM density estimate      = (EII,5)
## Negentropy approximation  = VAR
## GA optimal negentropy     = 0.3876899
## GA encoded basis solution:
##      x1      x2      x3      x4      x5      x6      x7
## [1,] 0.9946846 2.12249 1.474708 1.821916 1.355892 1.608575 1.84216
##      x8      x9      x10 ...      x39      x40
## [1,] 1.960242 1.635682 1.998009      0.887938 0.0838241
##
## Estimated projection basis:
##      PP1      PP2
## [1,] -0.1373078 0.0133209
## [2,] -0.0891986 -0.0041198
## [3,] 0.3364386 0.1142210
## [4,] 0.1335292 0.1058045
## [5,] -0.1514649 0.0768084
```

```

## [6,] 0.3041334 0.2362478
## [7,] 0.0111259 0.2490463
## [8,] 0.0657290 0.2858303
## [9,] -0.2417006 0.2743277
## [10,] -0.0548193 0.2132065
## [11,] -0.0949439 0.0855194
## [12,] 0.2700553 0.1181032
## [13,] 0.2954168 -0.0748888
## [14,] 0.0463277 -0.3394907
## [15,] 0.2932328 -0.3380552
## [16,] 0.2148813 -0.3308517
## [17,] 0.0303004 -0.3164006
## [18,] -0.1751073 -0.2824537
## [19,] 0.2106460 -0.1605167
## [20,] -0.0817067 -0.1882867
## [21,] 0.5241303 0.2060123
##
## Monte Carlo Negentropy approximation check:
##                               VAR
## Approx Negentropy 0.387689933
## MC Negentropy      0.486076949
## MC se              0.002955566
## Relative accuracy 0.797589628

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)
```

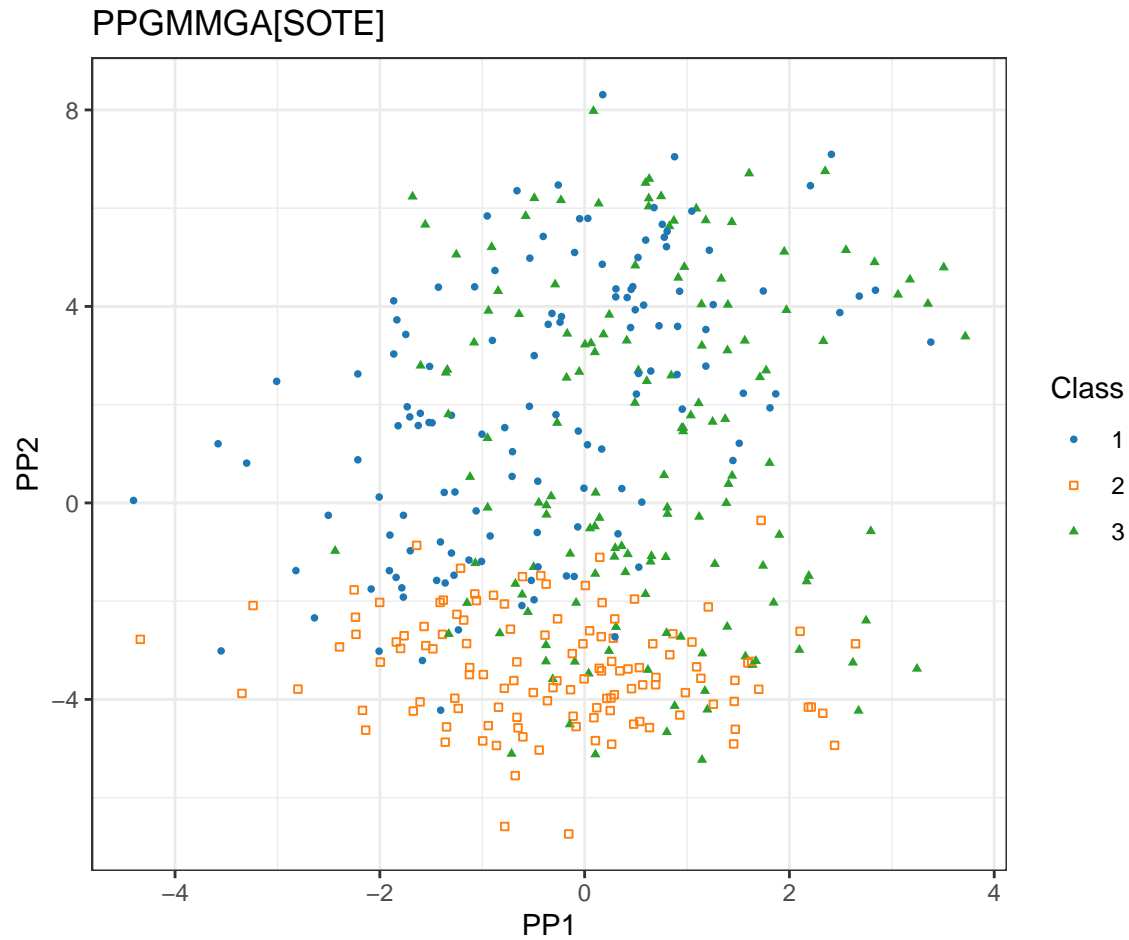
```
## -- ppgmmga -----
##
## Data dimensions           = 400 x 21
## Data transformation       = center
## Projection subspace dimension = 2
## GMM density estimate      = (EII,5)
## Negentropy approximation   = SOTE
## GA optimal negentropy     = 0.4207243
## GA encoded basis solution:
##      x1      x2      x3      x4      x5      x6      x7
## [1,] 4.090231 2.174023 1.561098 1.775365 1.299332 1.455583 1.731476
##      x8      x9      x10     ...      x39      x40
## [1,] 1.726346 1.485217 1.898199      2.236906 1.45377
##
## Estimated projection basis:
##      PP1      PP2
## [1,] 0.0000000 0.0208209
## [2,] 0.0000000 -0.0922720
## [3,] -0.3625121 -0.0387516
## [4,] -0.2321652 0.0571599
## [5,] 0.0008816 0.0957178
```

```

## [6,] -0.4237511 -0.0362567
## [7,] -0.0842197 -0.1564190
## [8,] -0.1345155 -0.2459679
## [9,]  0.2234381 -0.3444830
## [10,] 0.0279066 -0.3839229
## [11,] 0.0398374 -0.4946242
## [12,] -0.2482152 -0.1762226
## [13,] -0.2411028 -0.1206079
## [14,] 0.0643191 0.0237031
## [15,] -0.1180026 0.2179558
## [16,] -0.1234514 0.1848872
## [17,] 0.0892967 0.4094404
## [18,] 0.2162036 0.2458063
## [19,] -0.1672839 0.1234050
## [20,] 0.0079868 -0.0362507
## [21,] -0.5673024 0.0875544
##
## Monte Carlo Negentropy approximation check:
##                               SOTE
## Approx Negentropy 0.420724315
## MC Negentropy      0.444695733
## MC se              0.002886773
## Relative accuracy 0.946094789

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

```



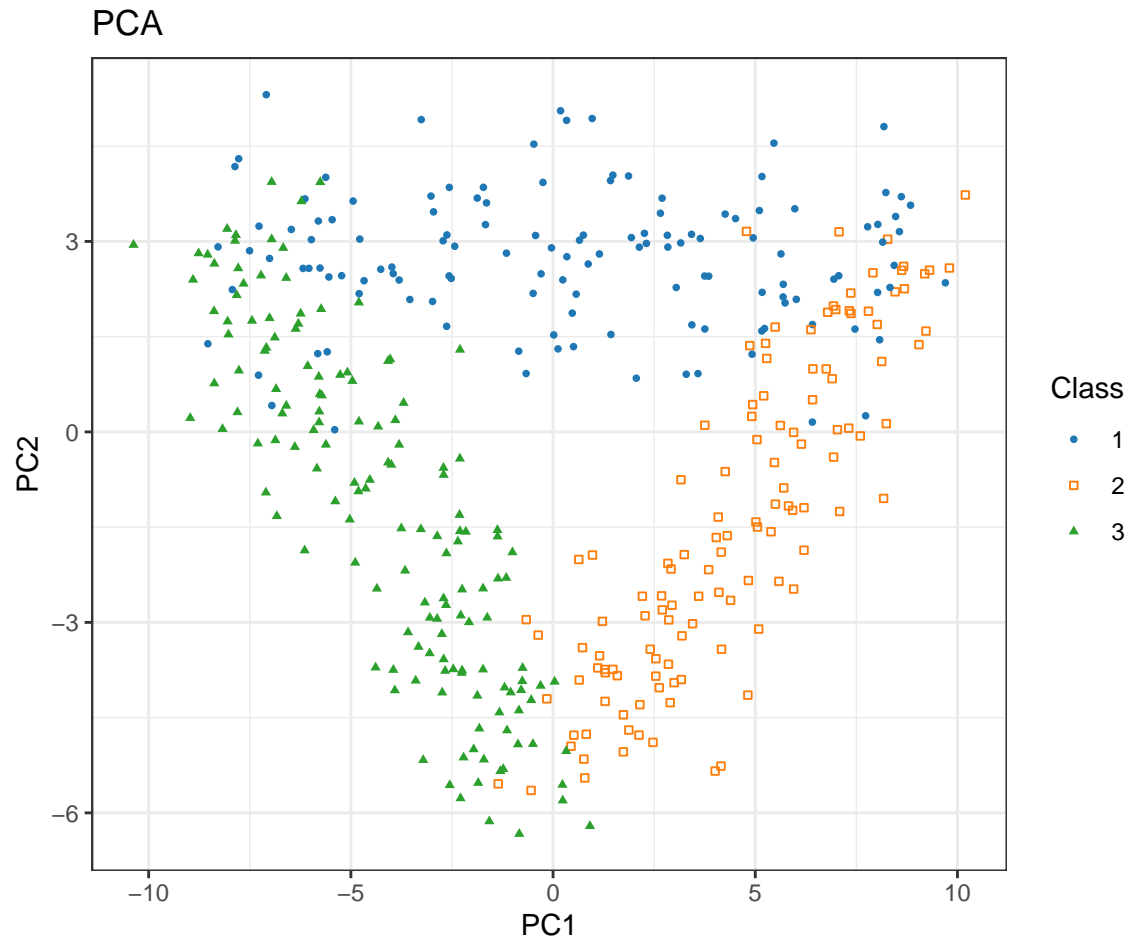
```

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

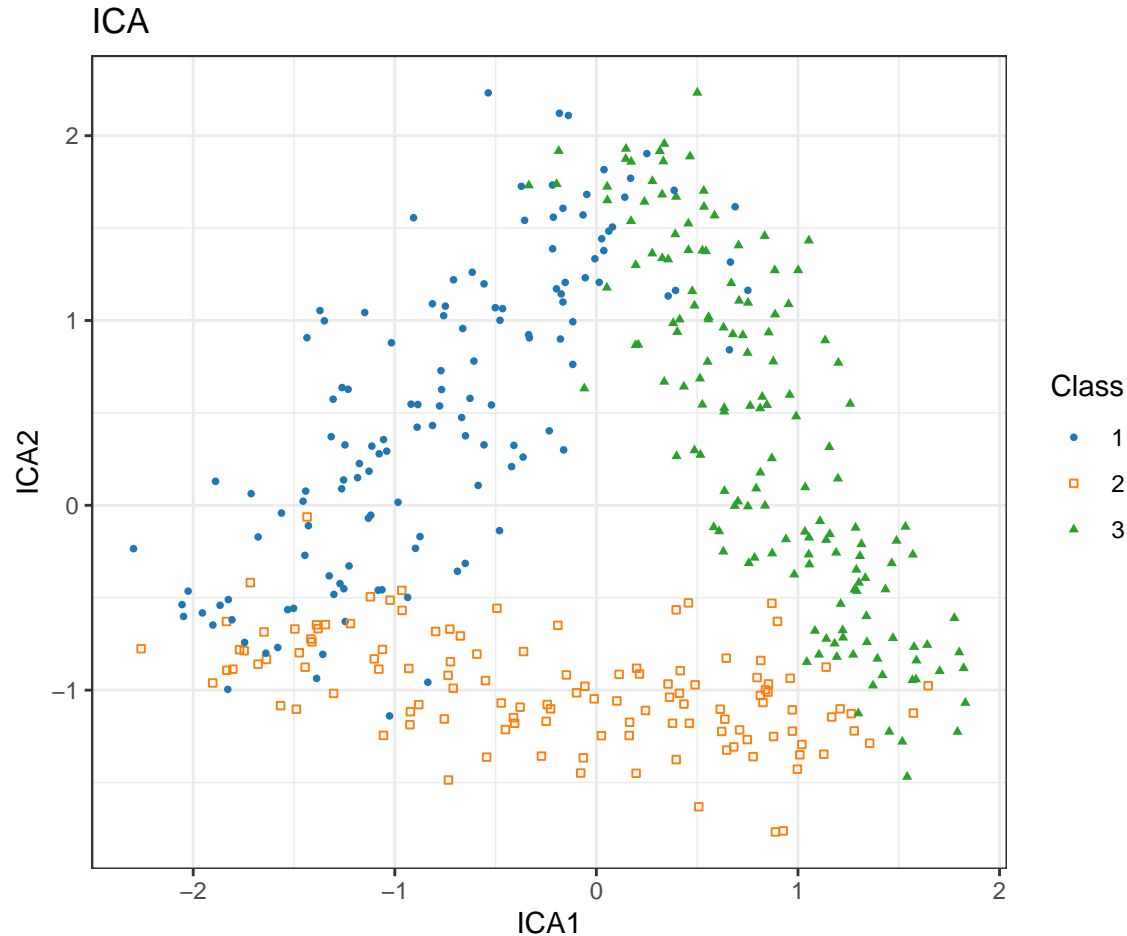
## $Negentropy
## [1] 1.029051
##
## $se
## [1] 0.003107752

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

```



```
ICA <- NegentropyFASTICA(PGMMGA1)
## Loading required package: fastICA
ICA[c("Negentropy", "se")]
## $Negentropy
## [1] 1.02659
##
## $se
## [1] 0.003134187
# trick for plotting
PPGMICA <- PGMMGA1; PPGMICA$approx <- "ICA"
PPGMICA$basis <- ICA$basis
PPGMICA$Z <- ICA$Z
plot(PPGMICA, Class, drawAxis = FALSE) + ggtitle("ICA")
```

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

```
##      UT      VAR      SOTE      PCA      ICA
## UT    0 88.39142 80.40689  5.935955  5.9359552257446752677
## VAR  NA  0.00000 45.94790 88.845326 88.8453258173464632819
## SOTE NA      NA  0.00000 80.407538 80.4075382793979116514
## PCA  NA      NA      NA  0.000000  0.00000000000003091358
## ICA  NA      NA      NA      NA  0.00000000000000000000
```

```
# 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)
```

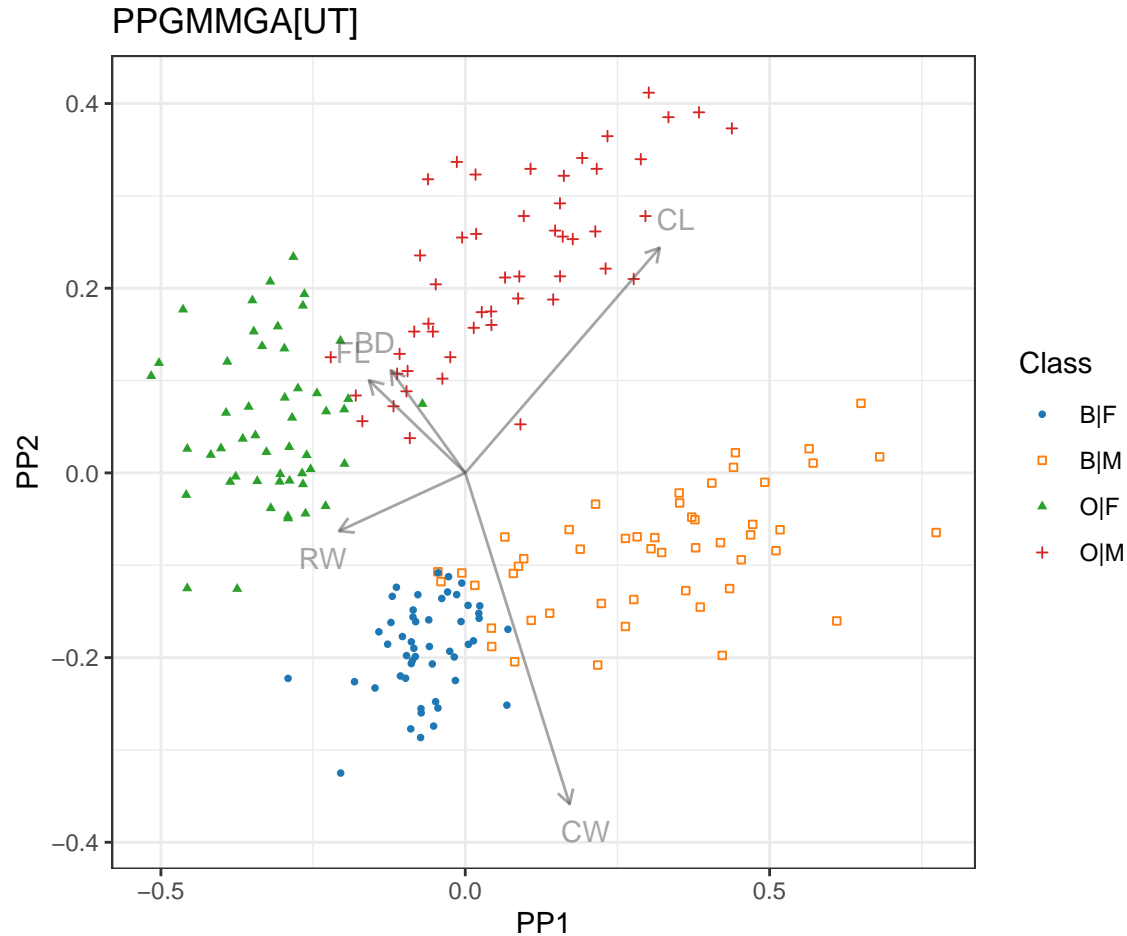
```
## -- ppgmmga -----
##
## Data dimensions      = 200 x 5
## Data transformation  = center & scale
```

```

## Projection subspace dimension = 2
## GMM density estimate          = (VEE,6)
## Negentropy approximation      = UT
## GA optimal negentropy         = 0.5522694
## GA encoded basis solution:
##           x1          x2          x3          x4          x5          x6          x7
## [1,] 0.651438 1.303515 1.964807 2.456677 5.453973 1.790279 0.707448
##           x8
## [1,] 2.728327
##
## Estimated projection basis:
##           PP1          PP2
## FL -0.3416030  0.2163801
## RW -0.4480185 -0.1360339
## CL  0.6897371  0.5264884
## CW  0.3702636 -0.7745130
## BD -0.2641106  0.2400291
##
## Monte Carlo Negentropy approximation check:
##                               UT
## Approx Negentropy 0.552269355
## MC Negentropy     0.577244094
## MC se              0.003014059
## Relative accuracy 0.956734527

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

```

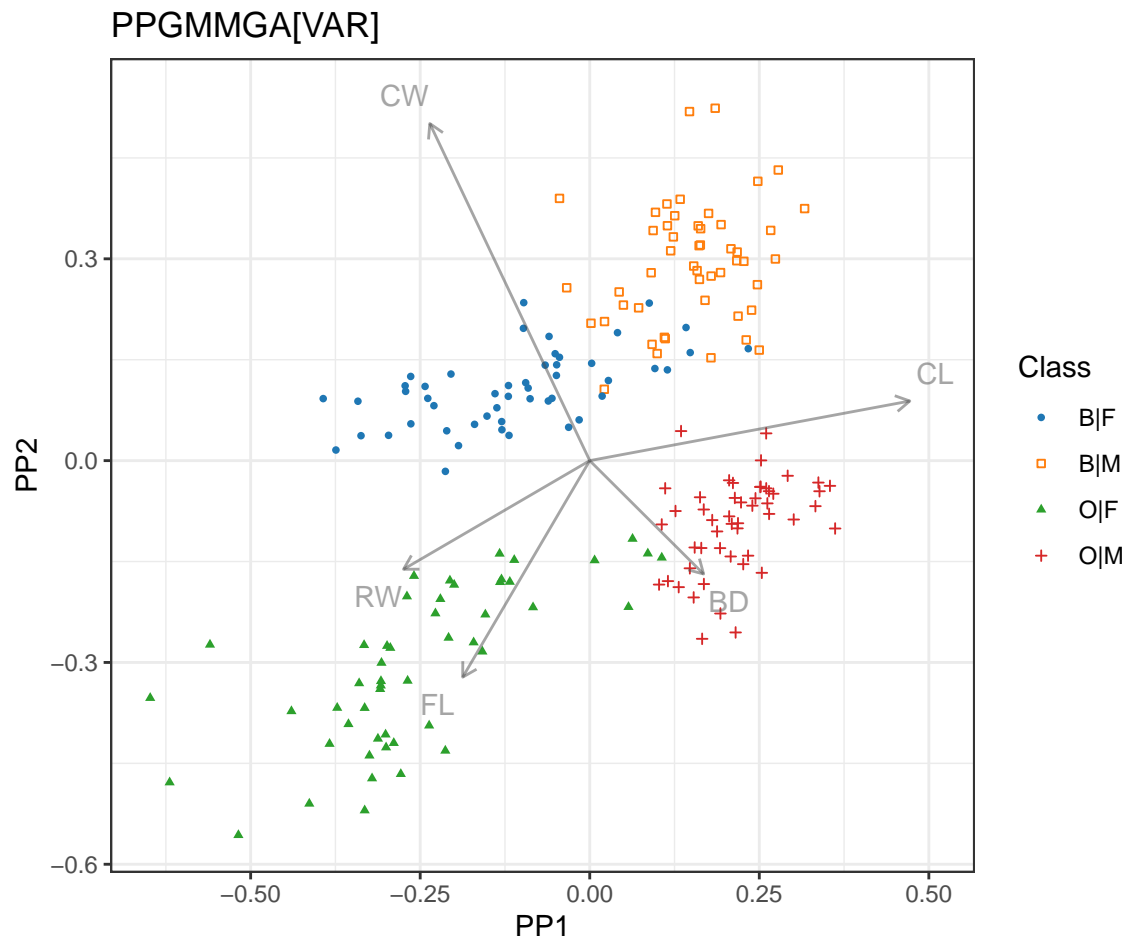


```
PPGMMGA2 <- ppgmmga(data = X, d = 2, GMM = GMM, approx = "VAR", seed = 2)
summary(PPGMMGA2, check = TRUE)
```

```
## -- 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    = 0.427159
## GA encoded basis solution:
##      x1      x2      x3      x4      x5      x6      x7
## [1,] 3.740322 1.307811 1.958632 0.613622 4.201613 1.803534 0.720979
##      x8
## [1,] 1.22604
##
## Estimated projection basis:
##      PP1      PP2
## FL -0.2901027 -0.4983814
## RW -0.4252002 -0.2501020
## CL  0.7308252  0.1372216
## CW -0.3651826  0.7759788
## BD  0.2599644 -0.2609445
```

```
##
## Monte Carlo Negentropy approximation check:
##          VAR
## Approx Negentropy 0.427158984
## MC Negentropy    0.585491721
## MC se            0.002933649
## Relative accuracy 0.729573056

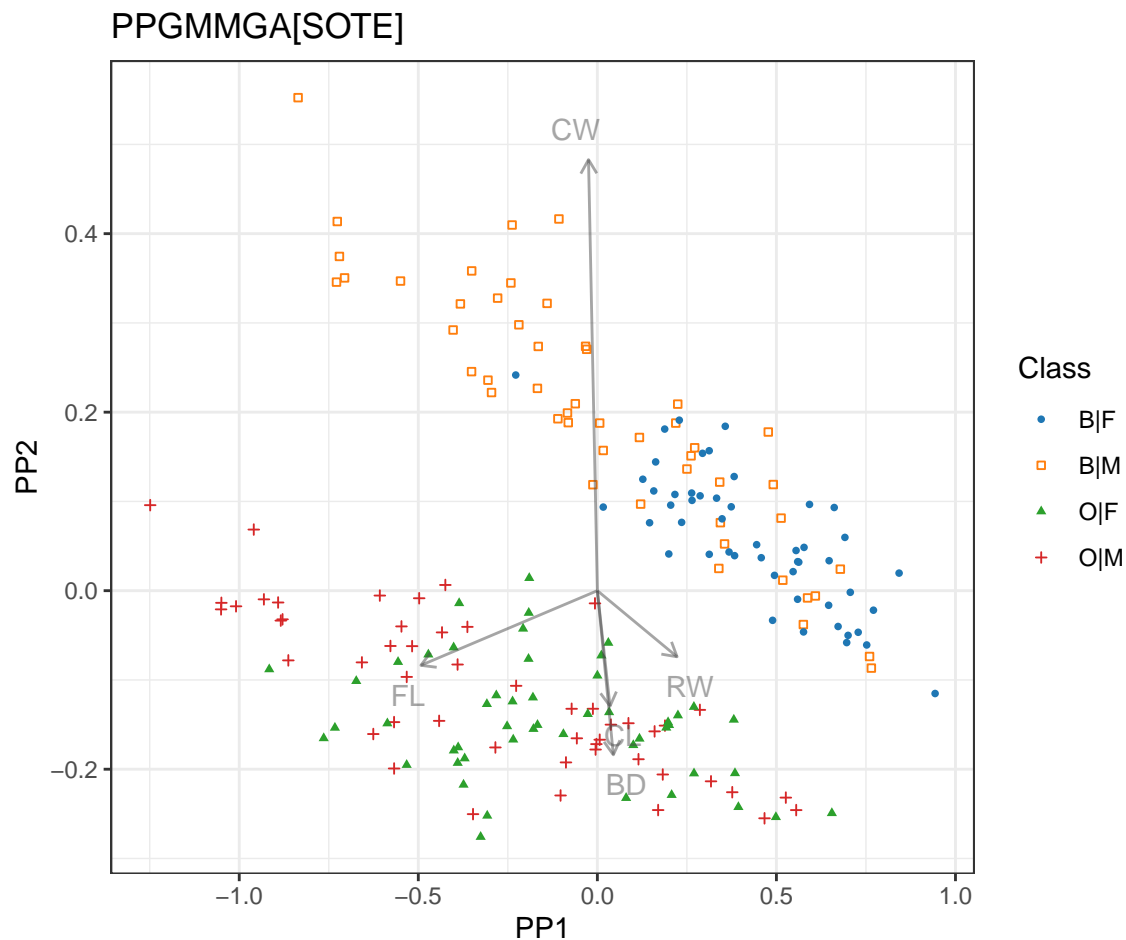
plot(PPGMMGA2, Class) + ggtitle("PPGMMGA[VAR]")
```



```
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      = 0.5684239
## GA encoded basis solution:
##      x1      x2      x3      x4      x5      x6      x7
## [1,] 5.136615 1.488665 1.616421 1.505593 3.358836 1.923403 0.335616
```

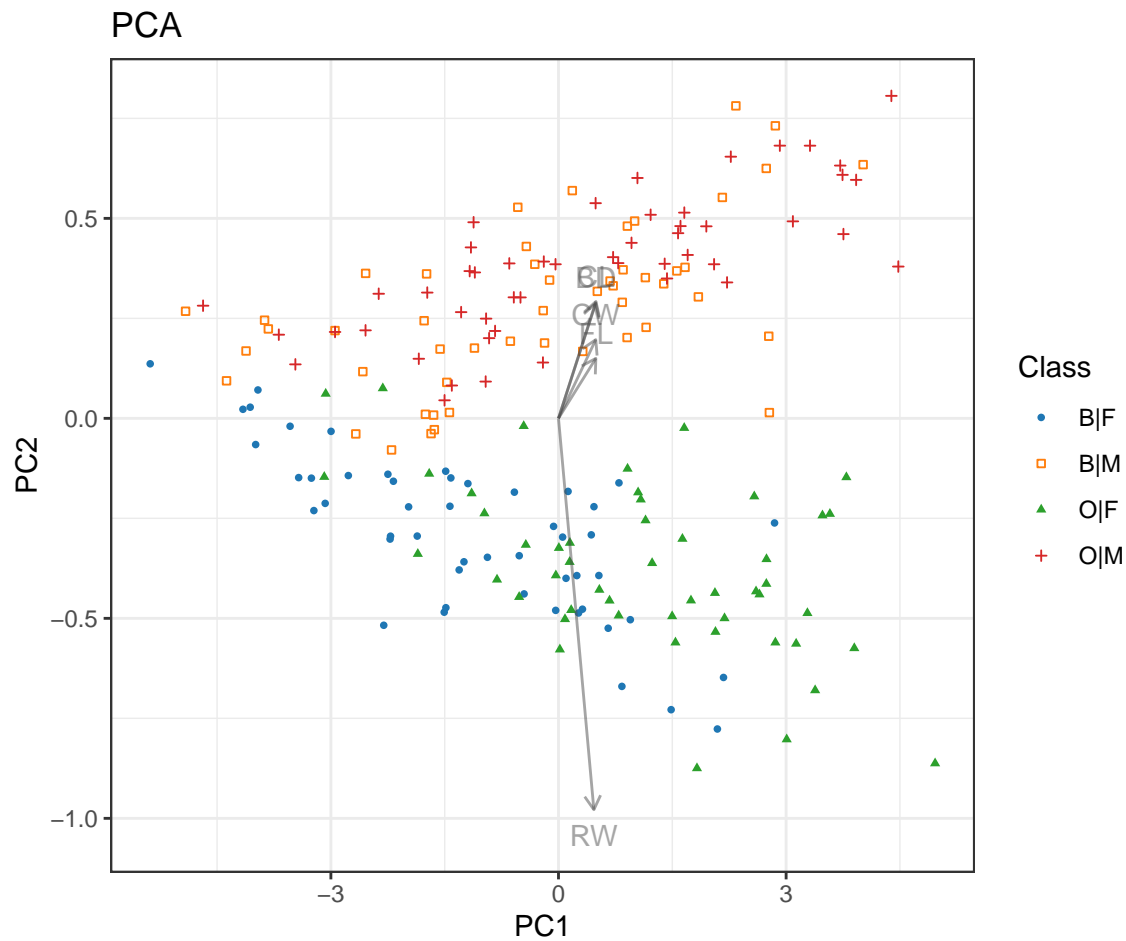
```
##          x8
## [1,] 2.480046
##
## Estimated projection basis:
##      PP1      PP2
## FL -0.9054123 -0.1536440
## RW  0.4089303 -0.1360782
## CL  0.0648695 -0.2377028
## CW -0.0454554  0.8872751
## BD  0.0820388 -0.3378114
##
## Monte Carlo Negentropy approximation check:
##          SOTE
## Approx Negentropy 0.568423861
## MC Negentropy     0.490694953
## MC se             0.002842244
## Relative accuracy 1.158405761
plot(PPGMMGA3, Class) + ggtitle("PPGMMGA[SOTE]")
```



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

```
## [1] 0.2008818
##
## $se
## [1] 0.002603359

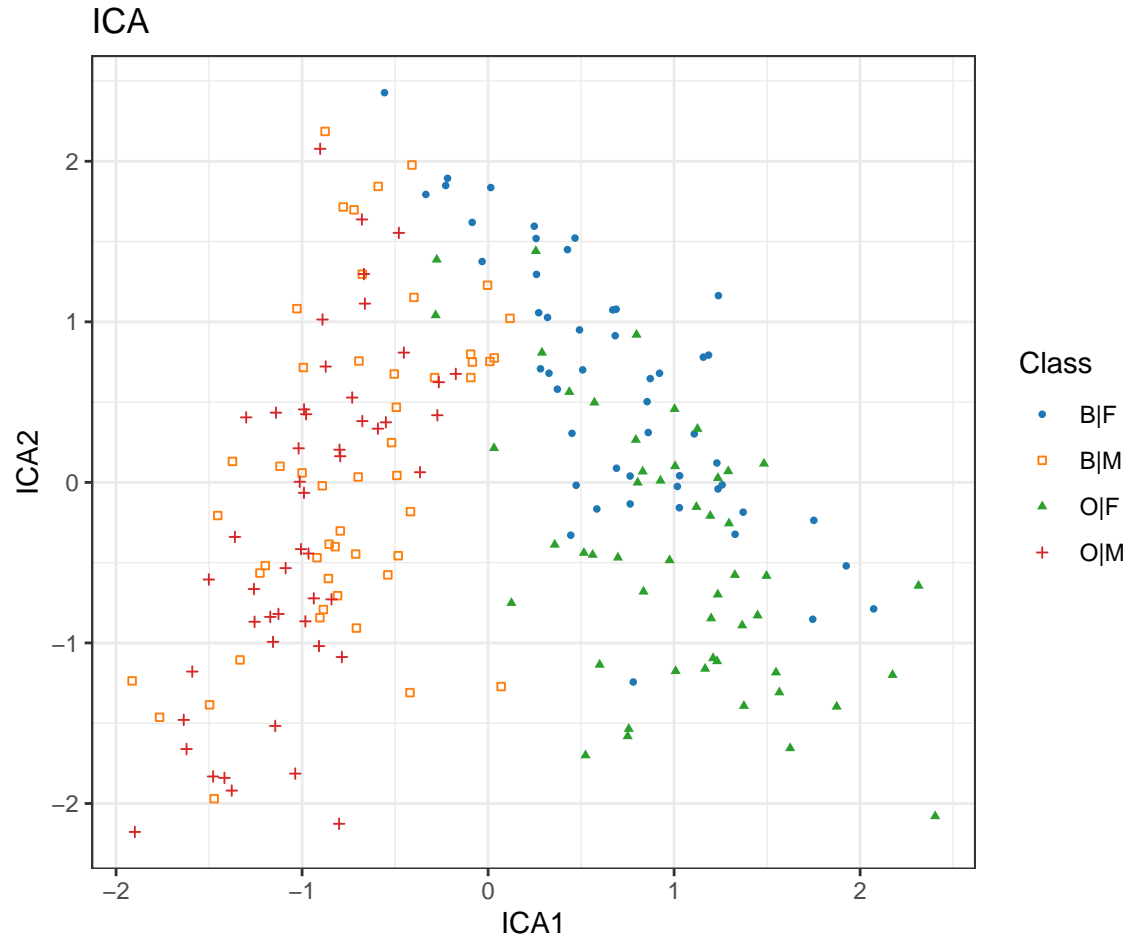
PPGMMPCA <- PPGMMGA1; PPGMMPCA$approx <- "PCA"
PPGMMPCA$basis <- PCA$basis
PPGMMPCA$Z <- PCA$Z
plot(PPGMMPCA, Class) + ggtitle("PCA")
```



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

## $Negentropy
## [1] 0.1963885
##
## $se
## [1] 0.002625278

# 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 22.47418 86.53113 88.33718 8.833718e+01
## VAR NA  0.00000 72.86603 88.79454 8.879454e+01
## SOTE NA      NA  0.00000 88.34951 8.834951e+01
## PCA  NA      NA      NA  0.00000 8.505309e-14
## ICA  NA      NA      NA      NA  0.000000e+00
```

```
# 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)
```

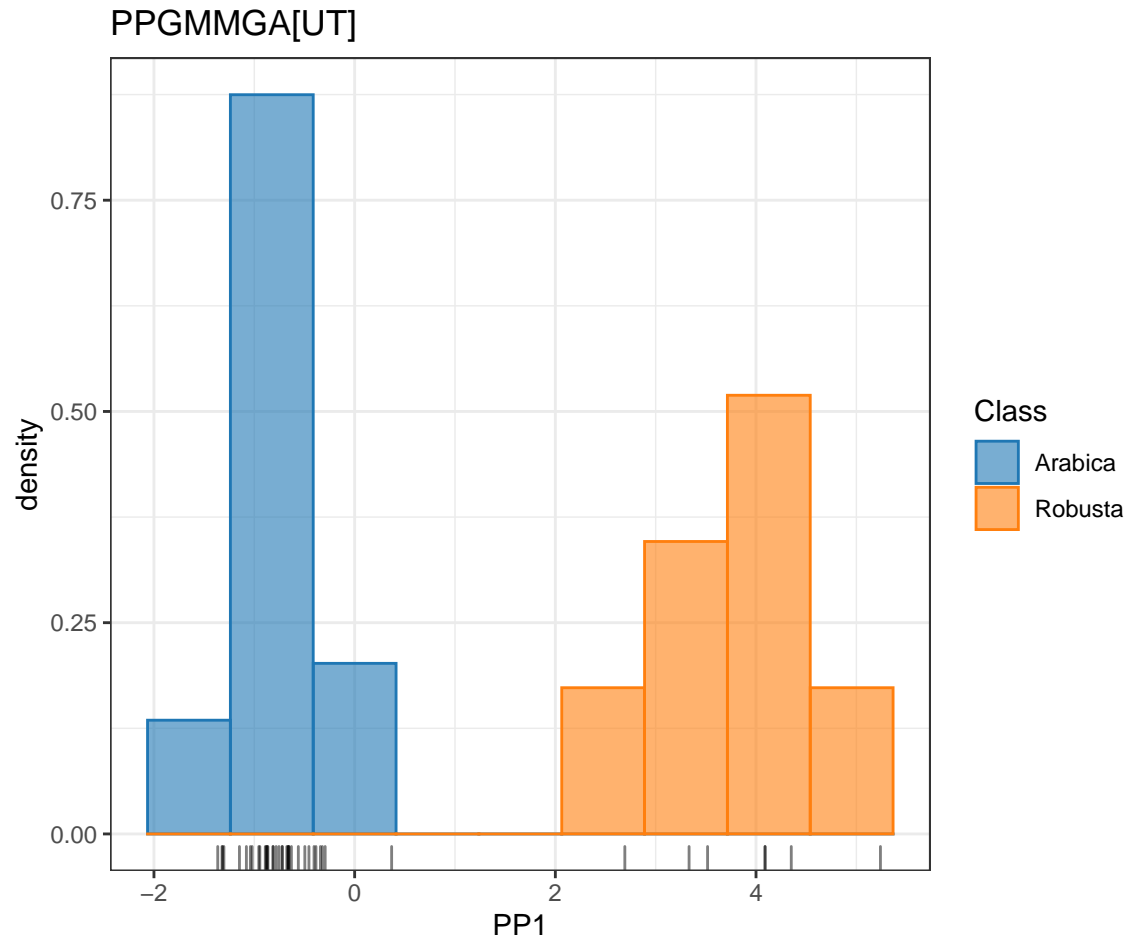
```
## -- ppgmmga -----
##
## Data dimensions      = 43 x 12
```

```

## 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,] 0.8326641 1.805441 1.671062 1.627284 1.426087 2.156648 0.3486215
##          x8          x9          x10         x11
## [1,] 1.662591 2.889578 2.144143 1.39098
##
## Estimated projection basis:
##          PP1
## Water          -0.0413148
## Bean Weight    -0.0375828
## Extract Yield  -0.0101527
## ph Value       0.0366543
## Free Acid      0.2624278
## Mineral Content 0.0249453
## Fat           -0.7487155
## Caffeine       0.5286198
## Trigonelline   -0.1393264
## Chlorogenic Acid 0.0546349
## Neochlorogenic Acid 0.0973545
## Isochlorogenic Acid 0.2324969
##
## Monte Carlo Negentropy approximation check:
##          UT
## Approx Negentropy 1.073235747
## MC Negentropy    1.072795970
## MC se           0.003233716
## Relative accuracy 1.000409935

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

```

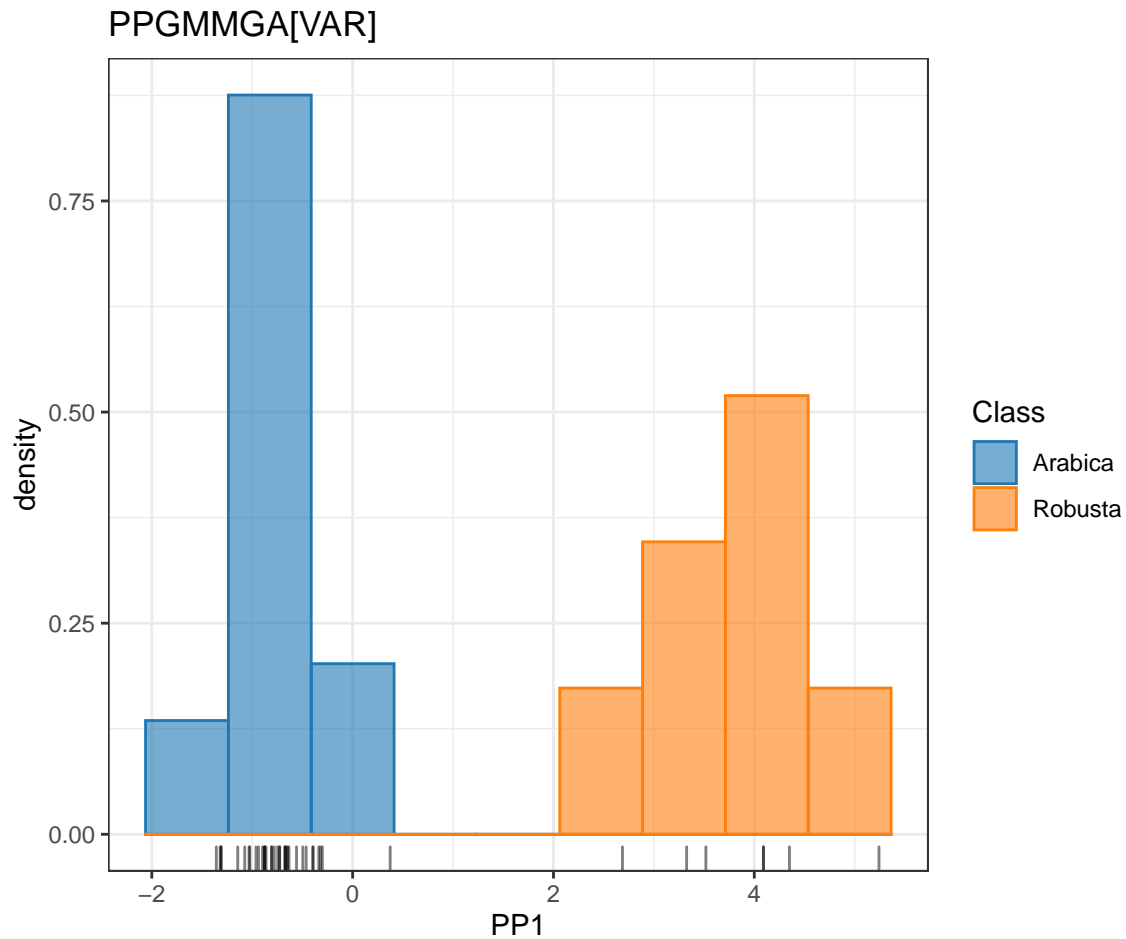



```
PPGMMGA2 <- ppgmmga(data = X, d = 1, GMM = GMM, approx = "VAR", seed = 2)
summary(PPGMMGA2, check = TRUE)
```

```
## -- 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      x8
## [1,] 3.985616 1.335038 1.470809 1.5147 1.715589 0.983017 2.795794 1.479646
##      x9      x10     x11
## [1,] 0.260509 0.9744 1.746787
##
## Estimated projection basis:
##      PP1
## Water      -0.0420993
## Bean Weight -0.0374314
## Extract Yield -0.0100178
## ph Value     0.0388425
## Free Acid    0.2594324
```

```
## Mineral Content      0.0245412
## Fat                  -0.7483489
## Caffeine             0.5300336
## Trigonelline         -0.1393753
## Chlorogenic Acid     0.0542438
## Neochlorogenic Acid  0.0970599
## Isochlorogenic Acid  0.2335804
##
## Monte Carlo Negentropy approximation check:
##                      VAR
## Approx Negentropy 1.073043193
## MC Negentropy    1.075246812
## MC se            0.003224378
## Relative accuracy 0.997950592
```

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



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

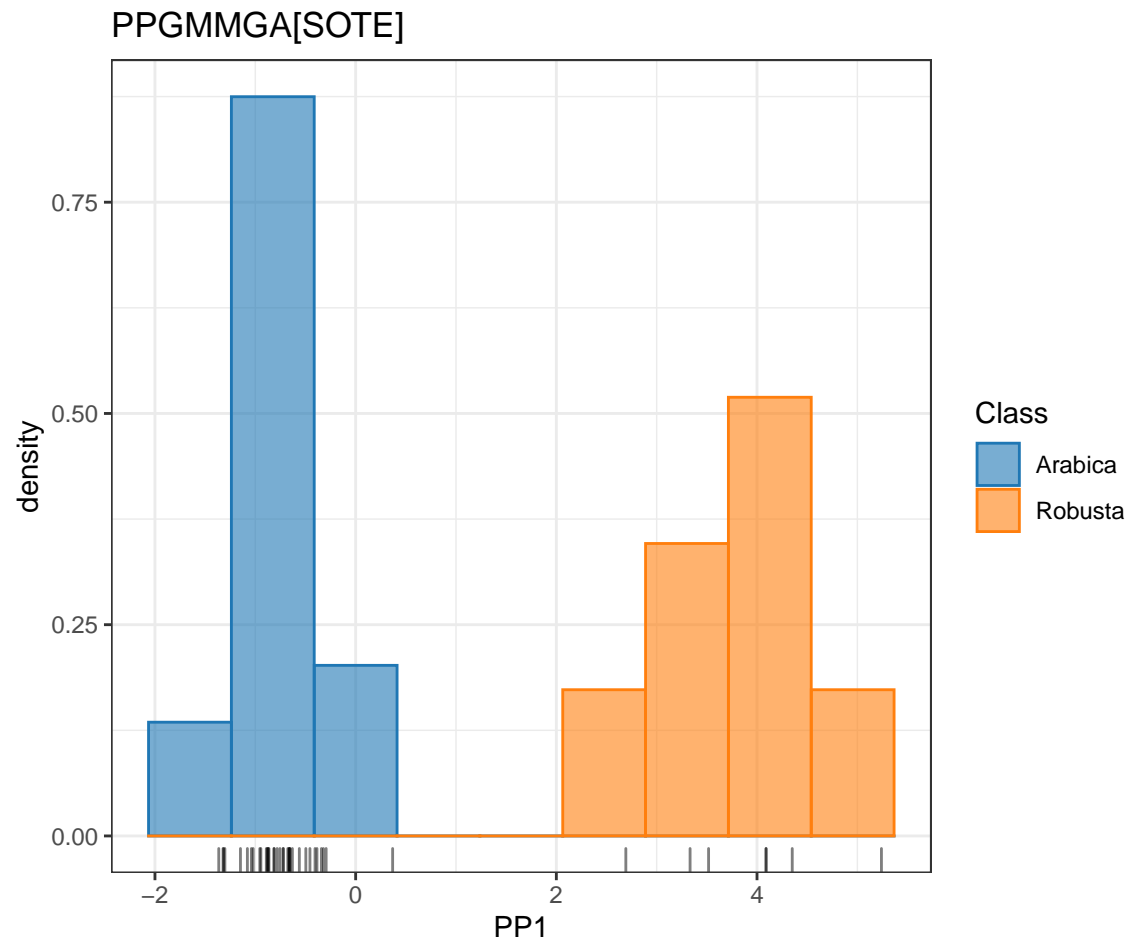
```
## -- ppgmmga -----
##
## Data dimensions      = 43 x 12
## Data transformation  = center & scale
```

```

## 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,] 0.8325712 1.805456 1.671057 1.627278 1.426087 2.156669 0.3485883
##      x8      x9      x10     x11
## [1,] 1.662573 2.88958 2.143406 1.391774
##
## Estimated projection basis:
##      PP1
## Water      -0.0413323
## Bean Weight -0.0376058
## Extract Yield -0.0101120
## ph Value    0.0366083
## Free Acid   0.2623999
## Mineral Content 0.0249378
## Fat         -0.7487118
## Caffeine    0.5286355
## Trigonelline -0.1393254
## Chlorogenic Acid 0.0546292
## Neochlorogenic Acid 0.0973498
## Isochlorogenic Acid 0.2325116
##
## Monte Carlo Negentropy approximation check:
##      SOTE
## Approx Negentropy 1.073229544
## MC Negentropy    1.074855480
## MC se           0.003234009
## Relative accuracy 0.998487298

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

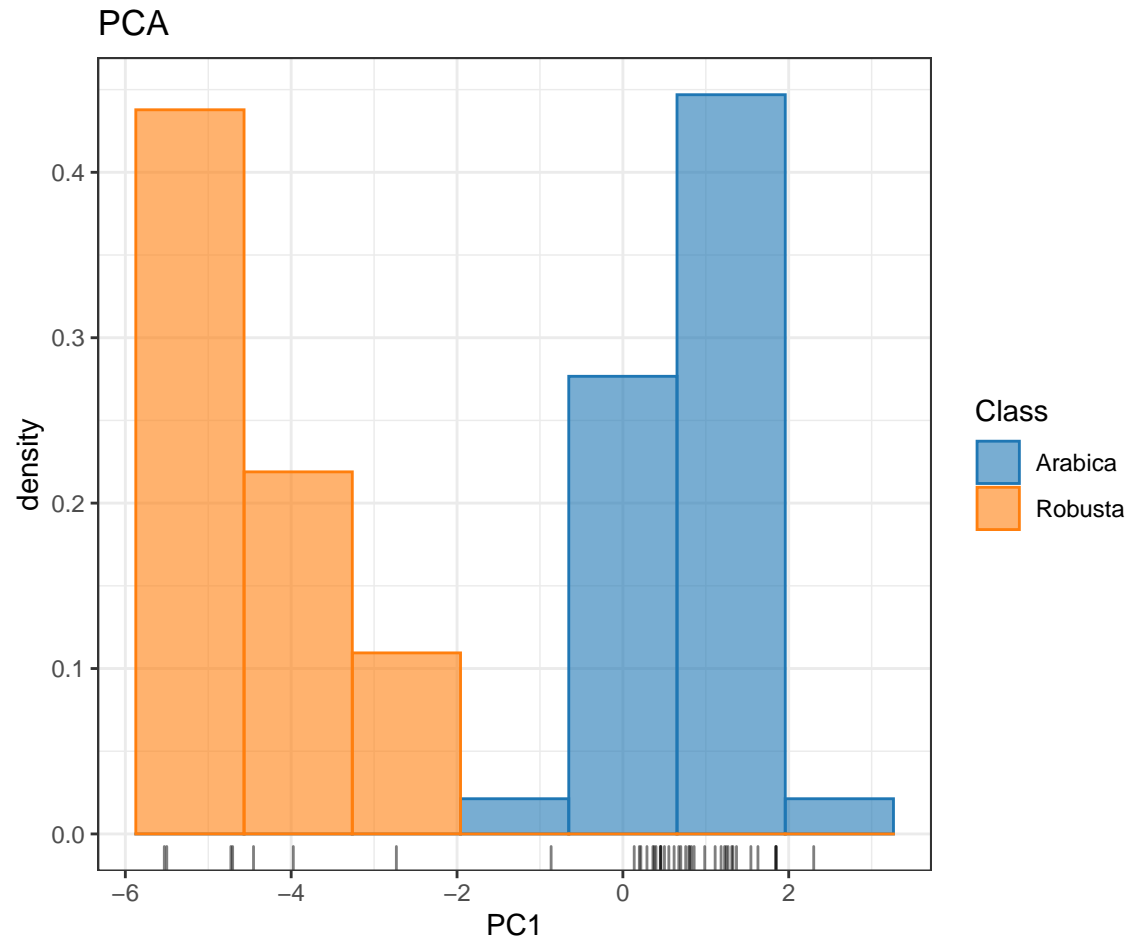
```



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

## $Negentropy
## [1] 0.777914
##
## $se
## [1] 0.003155727

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

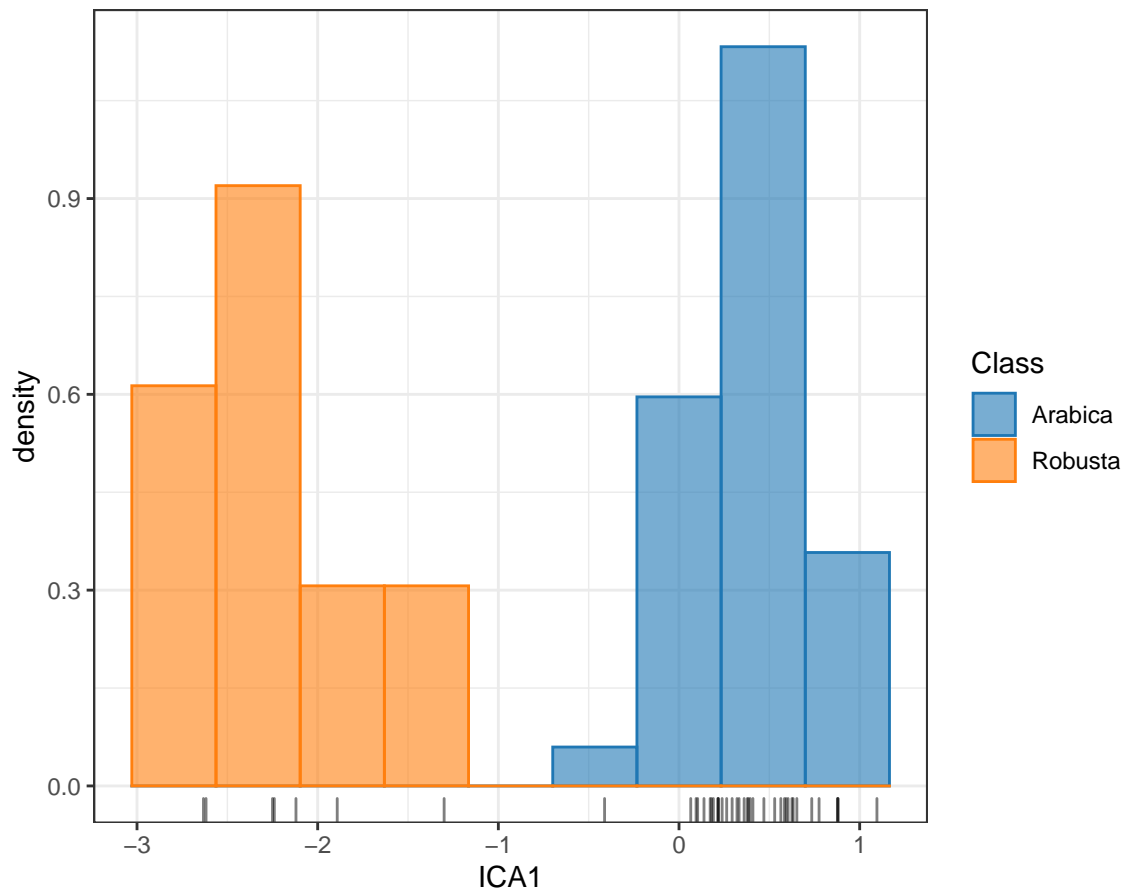


```
ICA <- NegentropyFASTICA(PPGMMGA1)
ICA[c("Negentropy", "se")]

## $Negentropy
## [1] 0.7765708
##
## $se
## [1] 0.00316075

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

ICA



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

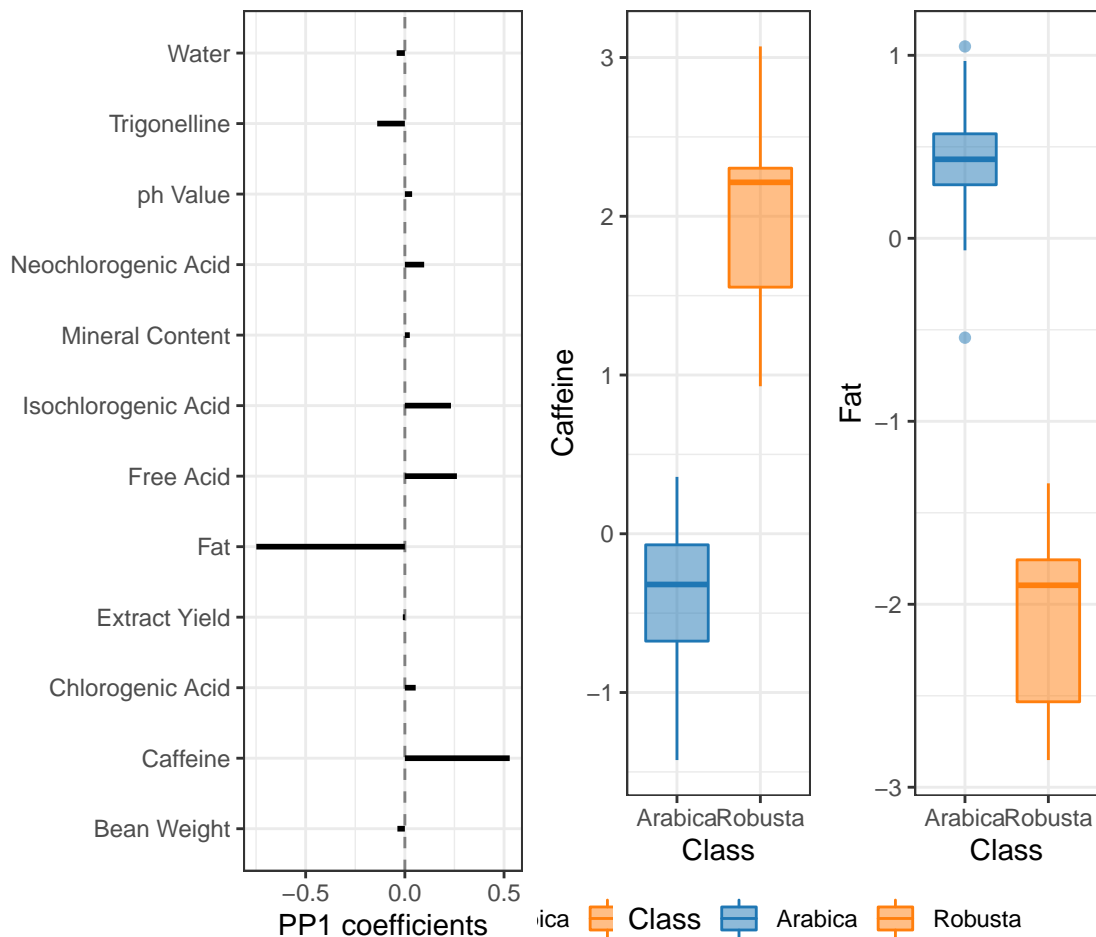
```
##      UT      VAR      SOTE      PCA      ICA
## UT    0 0.2439603 0.004425442 34.12113 3.412113e+01
## VAR  NA 0.0000000 0.243395125 34.16566 3.416566e+01
## SOTE NA      NA 0.000000000 34.12113 3.412113e+01
## PCA  NA      NA      NA 0.00000 5.565848e-14
## ICA  NA      NA      NA      NA 0.000000e+00
```

```
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      x8
## [1,] 2.578947 1.71223 1.57052 1.084437 1.532506 1.574246 1.506409 1.552572
##      x9      x10
## [1,] 1.512311 1.401587
##
## Estimated projection basis:
##      PP1
## Ht   -0.4580404
## Wt    0.7263082
## LBM  -0.1466988
## RCC  -0.0510054
## WCC  -0.0159045
## Hc   -0.0562725
## Hg    0.0030172
## Ferr -0.0335036
## BMI  -0.4627437
## SSF  -0.0002740
## Bfat  0.1409622
##
## Monte Carlo Negentropy approximation check:
##      UT
## Approx Negentropy 0.271642503
## MC Negentropy     0.224658574
## MC se              0.001828253
## Relative accuracy 1.209134809

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    = 0.2283455
## GA encoded basis solution:
##      x1      x2      x3      x4      x5      x6      x7
## [1,] 2.501606 1.248873 1.744959 0.9429893 1.575927 1.631489 1.550005
##      x8      x9      x10
## [1,] 1.597356 1.616326 1.759836
##
## Estimated projection basis:
##      PP1
## Ht   -0.4419706
## Wt    0.5936294
## LBM   0.1415970
## RCC   0.0343314

```



```

## WCC    0.0200386
## Hc     -0.0156905
## Hg      0.0458626
## Ferr   0.0038790
## BMI    -0.5487679
## SSF     0.1643818
## Bfat   -0.3163921
##
## Monte Carlo Negentropy approximation check:
##                               VAR
## Approx Negentropy 0.228345539
## MC Negentropy    0.306087586
## MC se            0.002418108
## Relative accuracy 0.746013722

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      = 0.3843009
## GA encoded basis solution:
##      x1      x2      x3      x4      x5      x6      x7      x8
## [1,]  0 1.261321 2.334781 1.561184 1.587058 1.474591 1.589836 1.539495
##           x9      x10
## [1,] 1.613303 2.313061
##
## Estimated projection basis:
##           PP1
## Ht      0.0000000
## Wt     -0.5036257
## LBM     0.4619512
## RCC     0.0290666
## WCC    -0.0214180
## Hc      0.0130314
## Hg     -0.0660545
## Ferr   0.0111831
## BMI    -0.0066110
## SSF     0.6589392
## Bfat   -0.3045587
##
## Monte Carlo Negentropy approximation check:
##                               SOTE
## Approx Negentropy 0.384300923
## MC Negentropy    0.117444763
## MC se            0.002591381
## Relative accuracy 3.272184422

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

```

```

## $Negentropy
## [1] 0.1300477
##
## $se
## [1] 0.00191437

PPGMMPCA <- PPGMGA1; PPGMMPCA$approx <- "PCA"
PPGMMPCA$basis <- PCA$basis
PPGMMPCA$Z <- PCA$Z

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

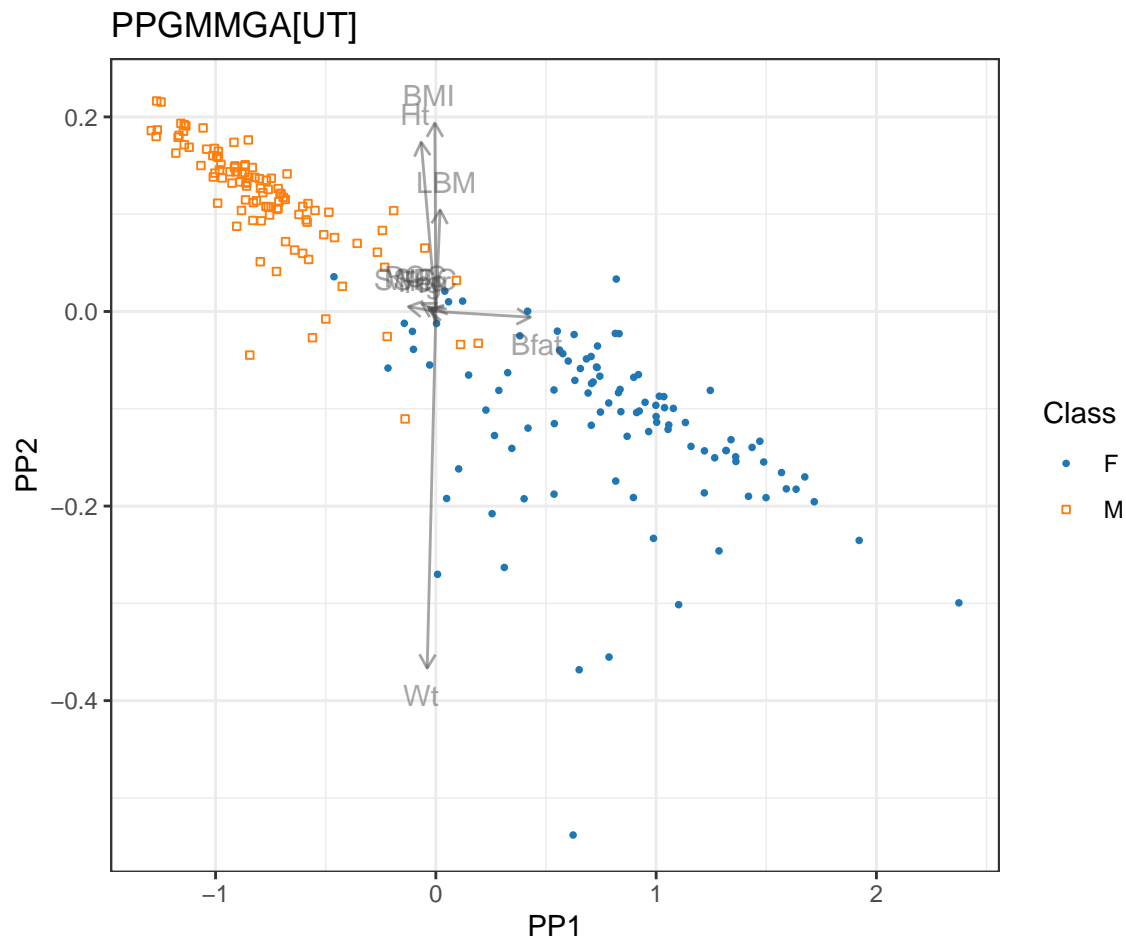
##      UT      VAR      SOTE      PCA
## UT      0 34.82077 61.57180 81.52611
## VAR NA   0.00000 88.40282 86.59055
## SOTE NA      NA   0.00000 87.65942
## PCA NA      NA      NA   0.00000

# 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     = 0.9178616
## GA encoded basis solution:
##      x1      x2      x3      x4      x5      x6      x7
## [1,] 4.178085 0.377279 2.395741 1.610094 1.769646 1.84858 1.680921
##      x8      x9      x10     ...      x19      x20
## [1,] 1.892426 2.250302 1.323756      1.489877 1.335914
##
## Estimated projection basis:
##      PP1      PP2
## Ht   -0.1441472 0.3768024
## Wt   -0.0852938 -0.7938715
## LBM   0.0422401 0.2260971
## RCC  -0.1395438 0.0179494
## WCC  -0.0739899 0.0099513
## Hc   -0.0258806 0.0080812
## Hg   -0.0671508 0.0031486
## Ferr -0.0493454 0.0071840
## BMI  -0.0098216 0.4193335
## SSF  -0.2705883 0.0107538
## Bfat  0.9296703 -0.0128037
##
## Monte Carlo Negentropy approximation check:
##      UT
## Approx Negentropy 0.917861613
## MC Negentropy     0.919371636
## MC se              0.003823592
## Relative accuracy 0.998357549

```

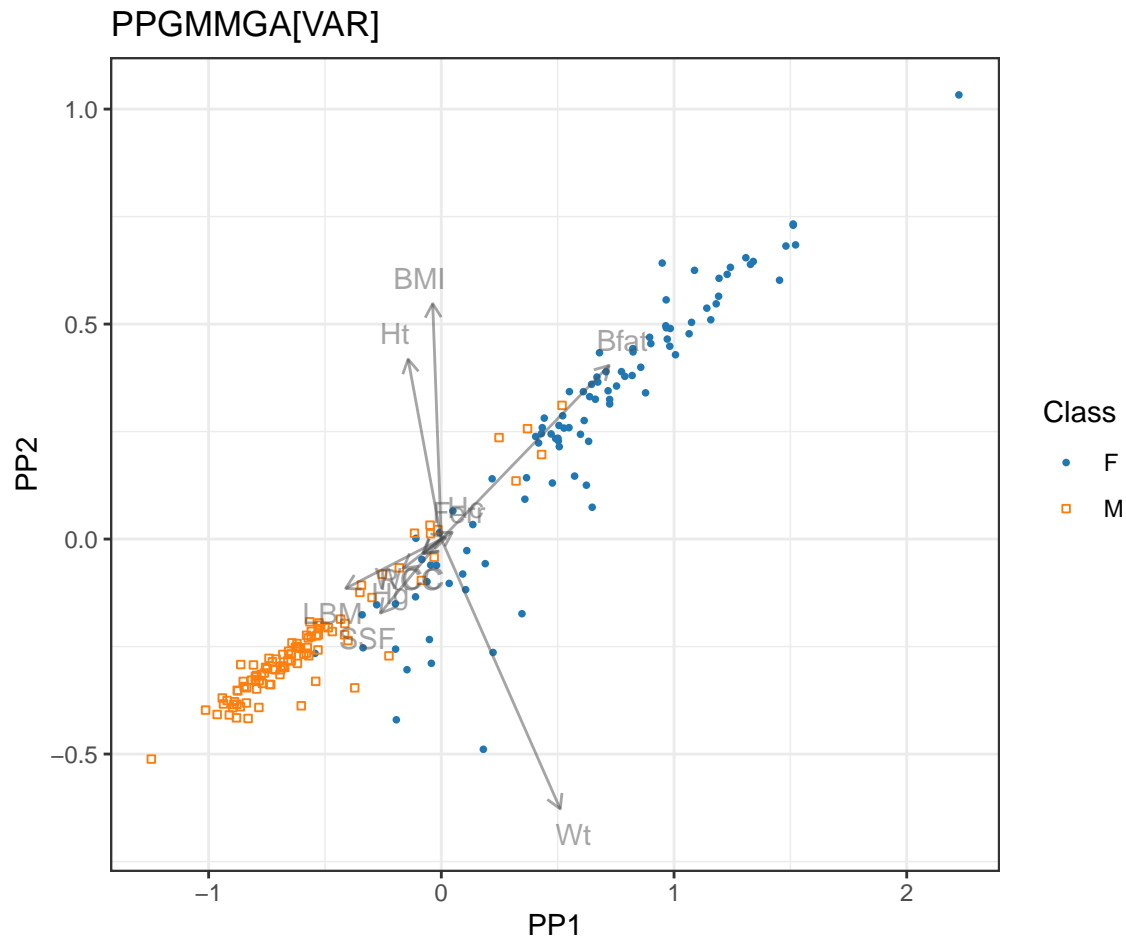
```
plot(PPGMMGA1, Class) + ggtitle("PPGMMGA[UT]")
```



```
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      = 0.6681225
## GA encoded basis solution:
##      x1      x2      x3      x4      x5      x6      x7
## [1,] 2.869292 2.337128 1.213752 1.517949 1.598576 1.336524 1.639846
##      x8      x9      x10 ...      x19      x20
## [1,] 1.452773 1.459103 0.9116159      1.415665 0.7566211
##
## Estimated projection basis:
##      PP1      PP2
## Ht    -0.1371829 0.4022067
## Wt     0.4912788 -0.6033797
## LBM   -0.3952004 -0.1103756
```

```
## RCC -0.0723723 -0.0331480
## WCC -0.0769908 -0.0317032
## Hc 0.0452199 0.0152109
## Hg -0.1564173 -0.0651868
## Ferr 0.0187231 0.0027219
## BMI -0.0356568 0.5264894
## SSF -0.2518056 -0.1655963
## Bfat 0.6934970 0.3882971
##
## Monte Carlo Negentropy approximation check:
## VAR
## Approx Negentropy 0.668122507
## MC Negentropy 0.855357462
## MC se 0.003885917
## Relative accuracy 0.781103265
plot(PPGMMGA2, Class) + ggtitle("PPGMMGA[VAR]")
```



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

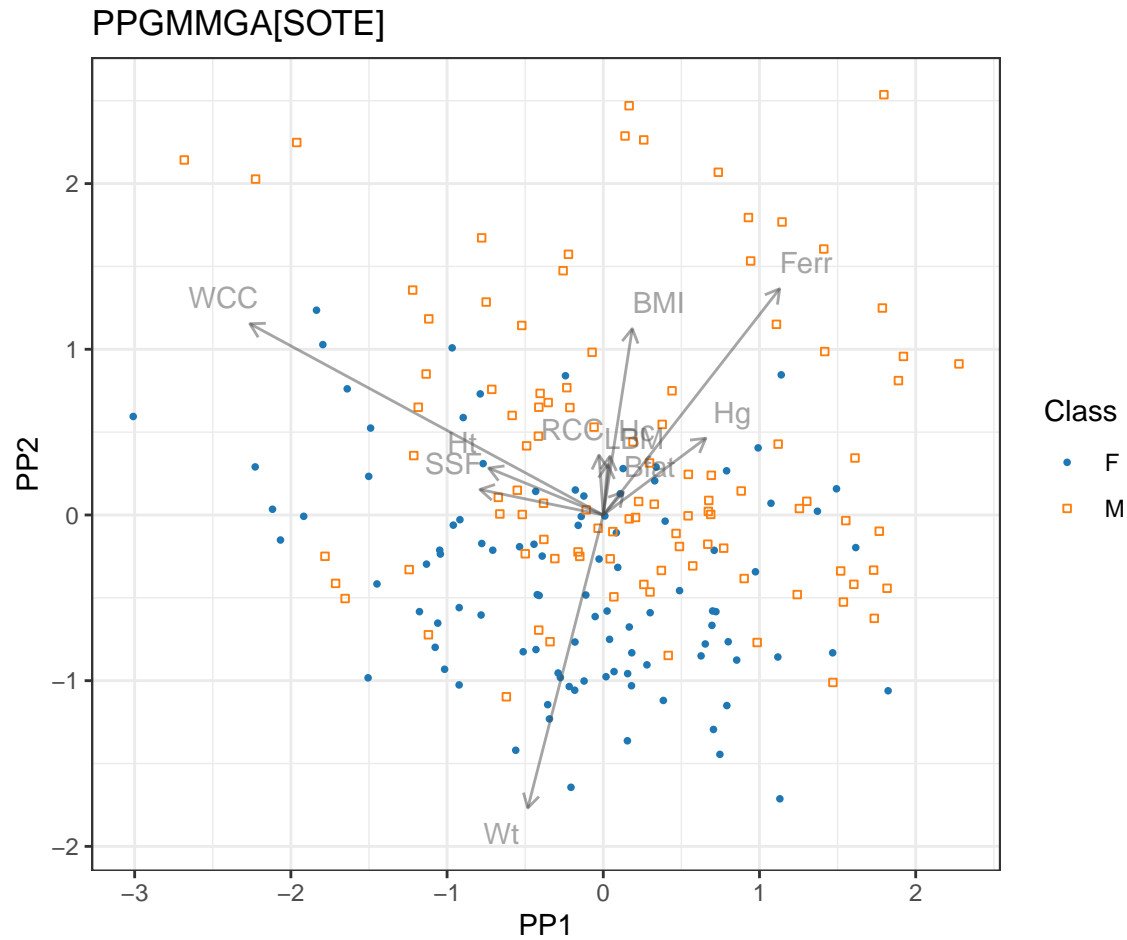
```
## -- ppgmmga -----
##
## Data dimensions          = 202 x 11
```

```

## Data transformation          = center & scale
## Projection subspace dimension = 2
## GMM density estimate        = (EVE,5)
## Negentropy approximation     = SOTE
## GA optimal negentropy        = 0.6887413
## GA encoded basis solution:
##           x1      x2      x3      x4      x5      x6      x7
## [1,] 0.9880969 1.615409 1.292303 1.638029 1.992781 1.834814 1.588476
##           x8      x9      x10    ...      x19      x20
## [1,] 0.3707612 1.539346 1.602909      1.345166 1.398721
##
## Estimated projection basis:
##           PP1      PP2
## Ht   -0.2550357  0.0983958
## Wt   -0.1680770 -0.6148345
## LBM   0.0098120  0.1058958
## RCC  -0.0096143  0.1255712
## WCC  -0.7865117  0.4012917
## Hc    0.0149203  0.1232018
## Hg    0.2281517  0.1610070
## Ferr  0.3925109  0.4745835
## BMI   0.0645295  0.3908672
## SSF  -0.2746341  0.0534389
## Bfat  0.0445974  0.0481189
##
## Monte Carlo Negentropy approximation check:
##                                     SOTE
## Approx Negentropy  0.6887412916
## MC Negentropy      0.0008819829
## MC se              0.0032159042
## Relative accuracy 780.9009746571

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

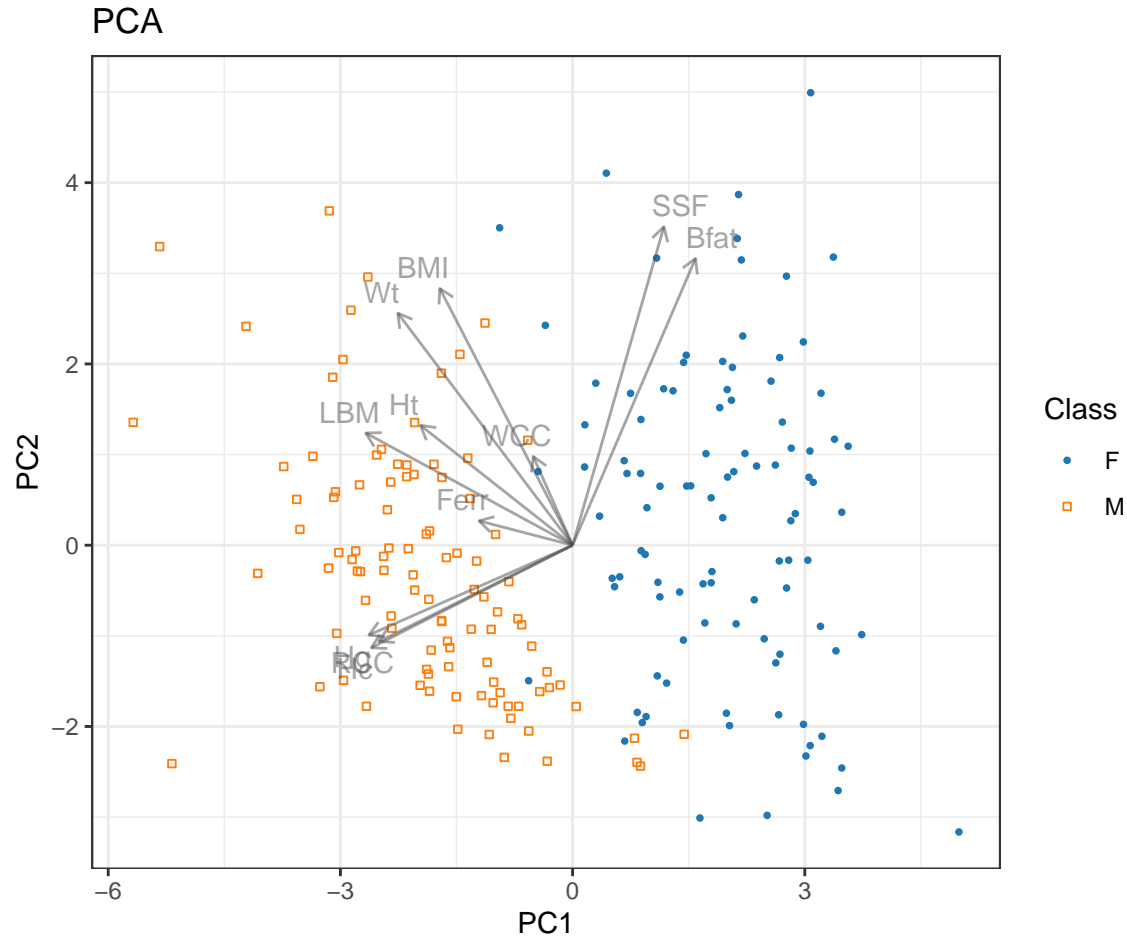
```



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

## $Negentropy
## [1] 0.2726372
##
## $se
## [1] 0.003172528

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 37.06684 78.94577 88.65538
## VAR  NA  0.00000 82.78473 88.53278
## SOTE NA      NA  0.00000 77.31532
## 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
```

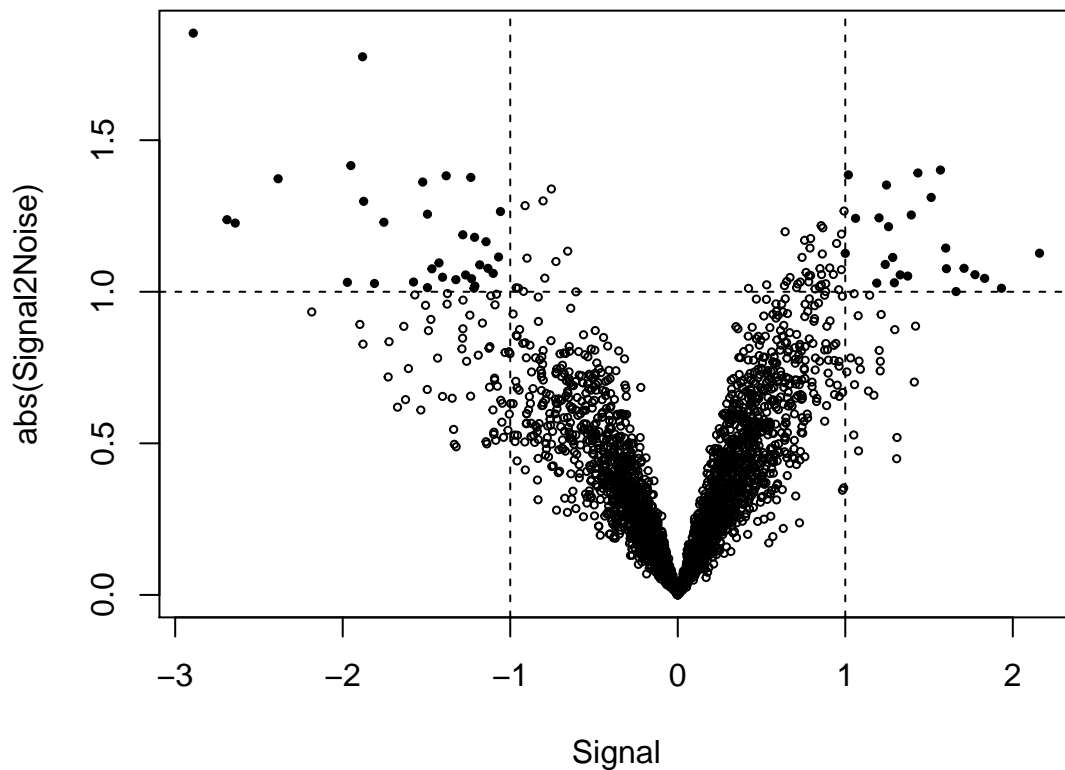
```
Class <- factor(golub.cl, levels = 0:1, labels = c("ALL", "AML"))
table(Class)
```

```
## 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))
```



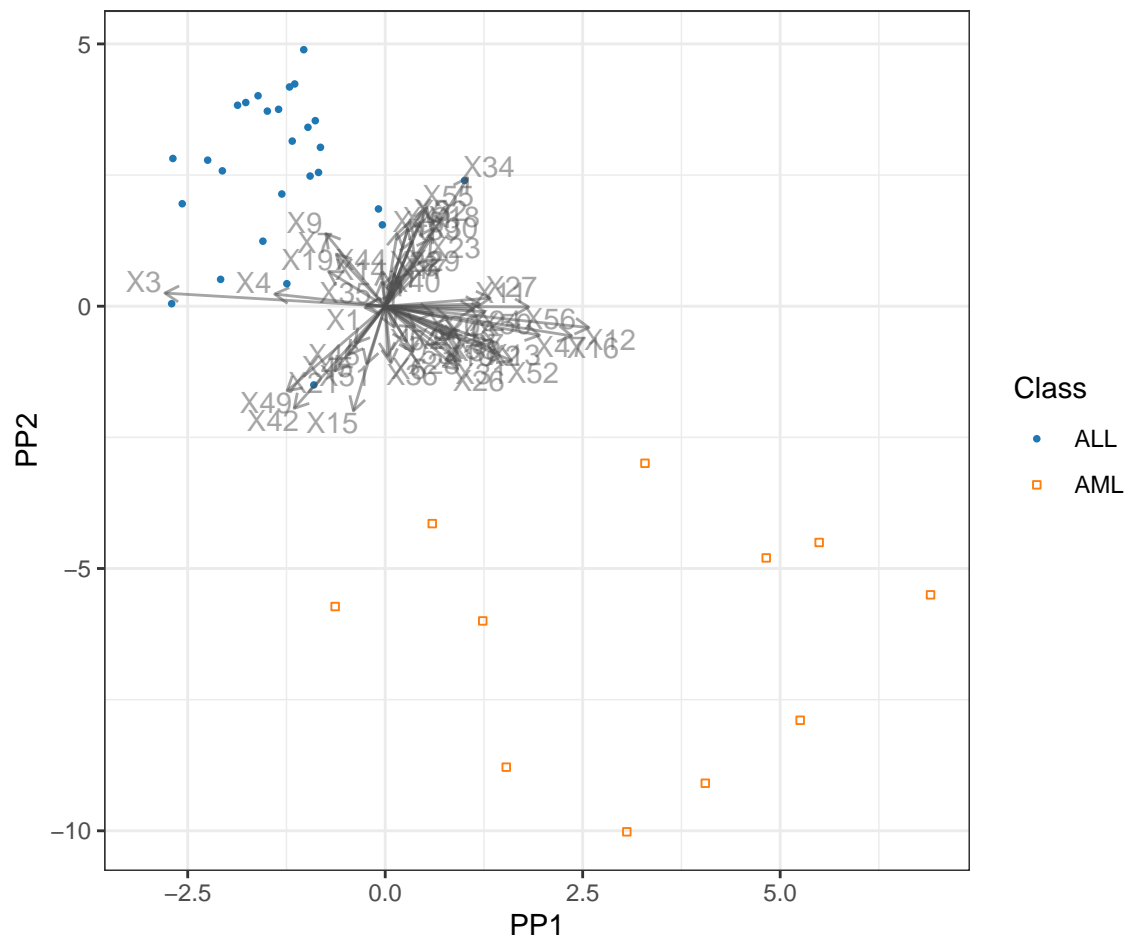
```

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     = 2.566196
## GA encoded basis solution:
##
##      x1      x2      x3      x4      x5      x6      x7      x8
## [1,] 2.934324 1.816842 1.643984 1.586751 1.60815 1.795067 1.53734 1.751092
##      x9      x10     ...      x109     x110
## [1,] 1.390211 1.593232      1.725517 2.297229
##
## Estimated projection basis:
##      PP1      PP2
## [1,] -0.0344708 -0.0035623
## [2,]  0.1639216 -0.0971614
## [3,] -0.3738132  0.0335813
## [4,] -0.1875198  0.0311018
## [5,]  0.0421182  0.2163993
## [6,]  0.0043695 -0.1281541
## [7,] -0.0829323  0.1331646
## [8,]  0.0253343  0.0741803
## [9,] -0.1008211  0.1854352
## [10,] 0.1043692 -0.0953775
## [11,] 0.1603914  0.0054578
## [12,] 0.3456837 -0.0546875
## [13,] 0.1834630 -0.0931680
## [14,] -0.0054127  0.0630182
## [15,] -0.0538920 -0.2672767
## [16,] 0.3166428 -0.0759974
## [17,] 0.0250350  0.0658775
## [18,] 0.0809384  0.1992905
## [19,] -0.0961525  0.0886790
## [20,] 0.0814943 -0.0187977
## ...
## [55,] 0.0709205  0.2522812
## [56,] 0.2435705 -0.0018754
##
## Monte Carlo Negentropy approximation check:
##      UT
## Approx Negentropy 2.566196212
## MC Negentropy     2.572299809
## MC se             0.003990328
## Relative accuracy 0.997627183

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(PPGMGA, Class, drawAxis = FALSE) +
  scale_x_continuous(expand = expand_scale(c(0.5, 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)
```

