

code.R

luca

2019-11-19

```
## R code accompanying the paper "Projection pursuit based on Gaussian
## mixtures and evolutionary algorithms" by 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")
#
# The last example requires the following step to install "multtest" package:
#
# if(!requireNamespace("BiocManager", quietly = TRUE))
#   install.packages("BiocManager")
# BiocManager::install("Biobase")
# BiocManager::install("multtest")

# To reproduce exactly the results of the paper you must 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", output_format = "pdf_document")
```

```
library(ppgmmga)
```

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

```
library(mclust)
```

```
## Package 'mclust' version 5.4.5
```

```
## 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)
```

```

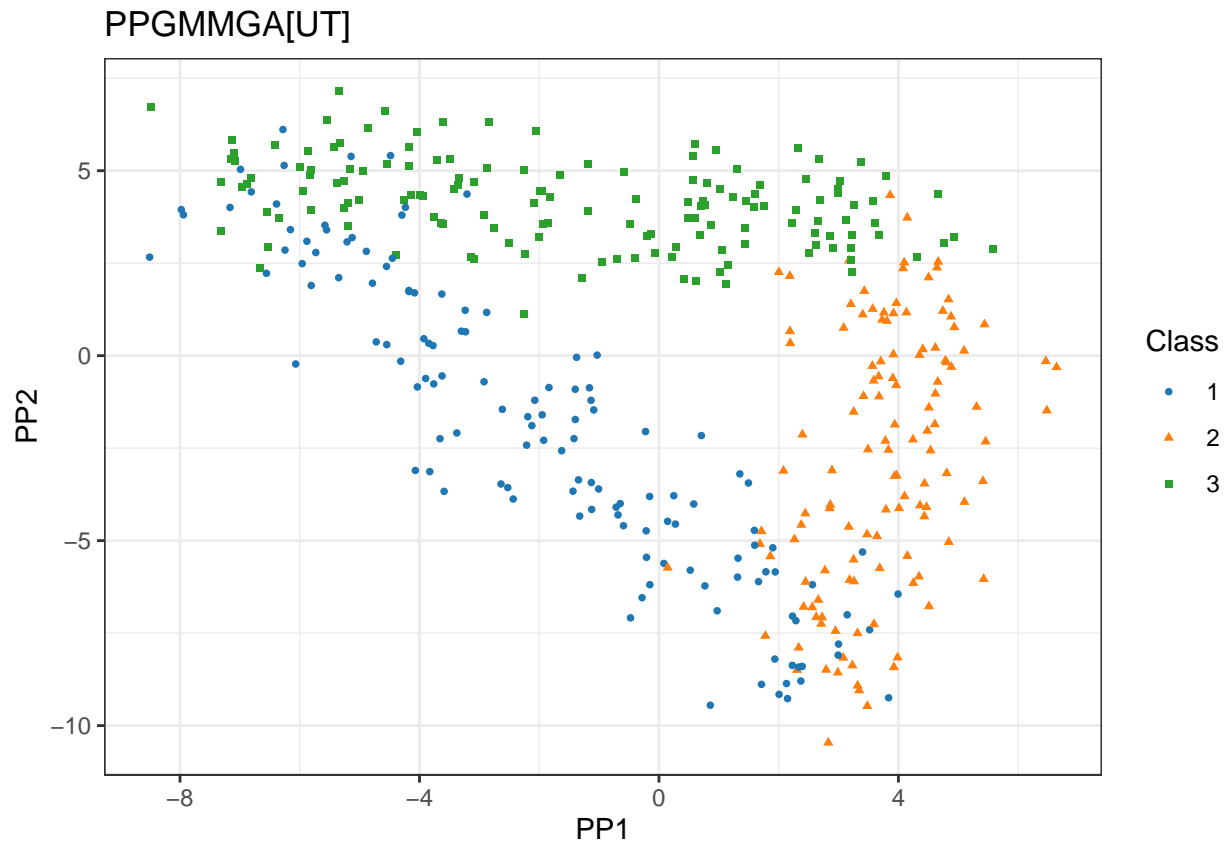
# to save time just fit once the GMM on centred variables
X <- scale(X, center = TRUE, scale = FALSE)
GMM <- densityMclust(X)

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

## -- ppgmga -----
##
## 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      x8
## [1,] 4.947688 1.564279 1.65297 1.772846 1.860352 1.978782 1.925953 1.962031
##      x9      x10     ...      x39      x40
## [1,] 1.772991 1.479144      2.475268 2.603082
##
## Estimated projection basis:
##      PP1      PP2
## [1,] 0.000000000 0.002666118
## [2,] 0.000000000 -0.103204090
## [3,] 0.000000000 -0.172811039
## [4,] 0.000000000 -0.256013398
## [5,] 0.000000000 -0.349488824
## [6,] 0.000000000 -0.358219294
## [7,] 0.000000000 -0.369150665
## [8,] 0.226104378 -0.215361103
## [9,] 0.369239385 -0.032090208
## [10,] 0.366293977 0.113903772
## [11,] 0.395802477 0.280543534
## [12,] 0.221566333 0.318361534
## [13,] 0.066745885 0.340140324
## [14,] -0.149497948 0.274138495
## [15,] -0.307079923 0.224324715
## [16,] -0.298667291 0.141228648
## [17,] -0.371244177 0.071374418
## [18,] -0.278767828 0.018010793
## [19,] -0.199996055 0.010327484
## [20,] -0.082079509 -0.002928313
## [21,] 0.006517717 -0.010401488
##
## 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]")
```

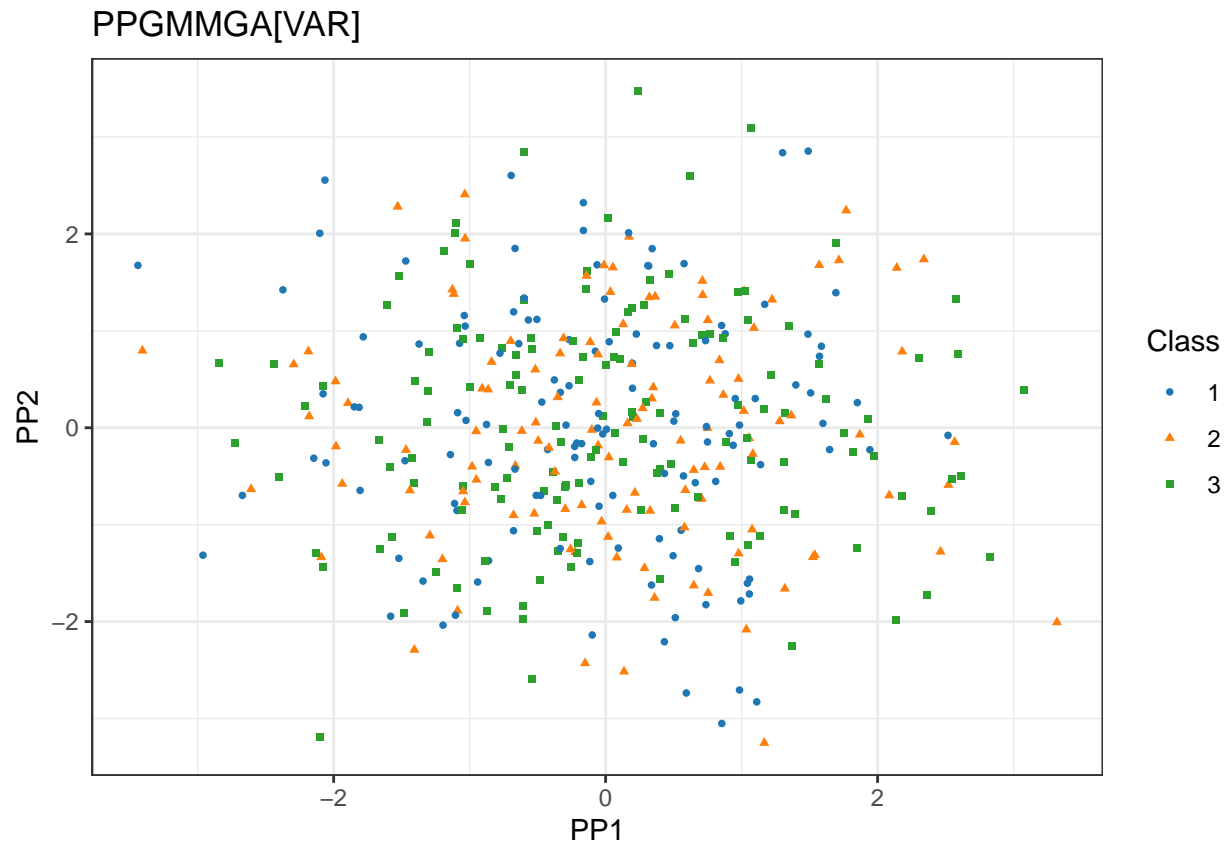


```
PPGMMGA2 <- ppgmmga(data = X, d = 2, gmm = GMM, scale = FALSE, approx = "VAR", seed = 2)
summary(PPGMMGA2, check = TRUE)
```

```
## -- 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,] -0.270833866 -0.42518990
## [2,]  0.094251343 -0.03268568
## [3,] -0.229349782  0.27242248
## [4,] -0.198356128  0.02240203
## [5,]  0.061709750 -0.12523085
```

```
## [6,] -0.368555402 0.14576832
## [7,] 0.185125139 0.26449236
## [8,] -0.099677218 0.08366799
## [9,] 0.393010661 0.14386953
## [10,] -0.139771012 -0.12533682
## [11,] -0.004814228 -0.11045200
## [12,] -0.281930538 0.13654905
## [13,] -0.159823971 0.21422205
## [14,] 0.209850939 0.13482083
## [15,] -0.148375928 0.14911403
## [16,] 0.104203801 0.26235397
## [17,] -0.034460982 -0.09770594
## [18,] -0.009333570 -0.30941206
## [19,] 0.137885030 0.36274157
## [20,] -0.111663447 -0.28902061
## [21,] -0.503248660 0.29454983
##
## Monte Carlo Negentropy approximation check:
##          VAR
## Approx Negentropy 0.233254344
## MC Negentropy    0.238876730
## MC se            0.003149867
## Relative accuracy 0.976463235
```

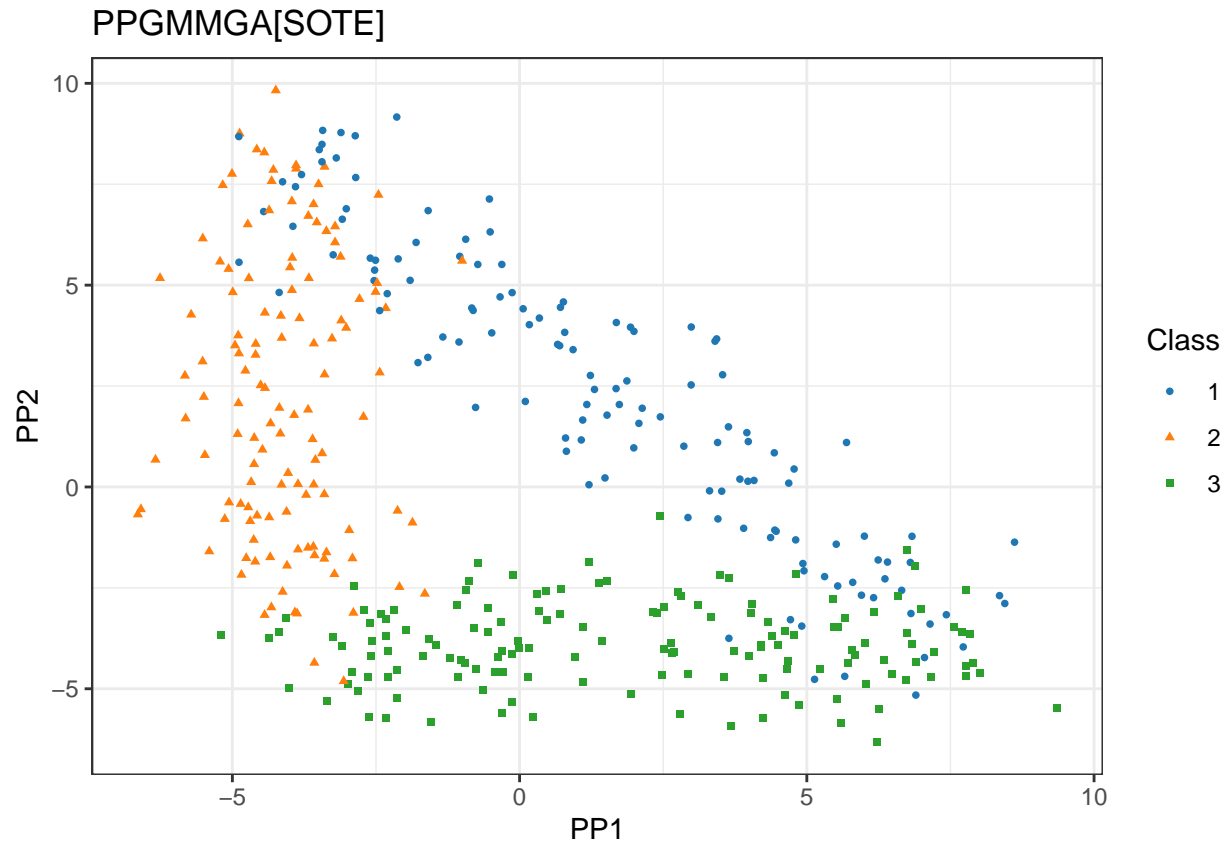
```
plot(PPGMMGA2, Class, drawAxis = FALSE) + ggtitle("PPGMMGA[VAR]")
```



```
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      x8
## [1,] 2.308029 1.563483 1.652205 1.769854 1.856781 1.983334 1.943112 1.998647
##      x9      x10     ...      x39      x40
## [1,] 1.827142 1.544723      2.553373 3.141593
##
## Estimated projection basis:
##      PP1      PP2
## [1,] 0.000000000 -2.421837e-17
## [2,] 0.000000000 0.000000e+00
## [3,] 0.000000000 1.760548e-01
## [4,] 0.000000000 2.639578e-01
## [5,] 0.000000000 3.601405e-01
## [6,] 0.000000000 3.588297e-01
## [7,] -0.164490596 3.478786e-01
## [8,] -0.250685174 1.855117e-01
## [9,] -0.367334286 -1.562337e-02
## [10,] -0.343462578 -1.605137e-01
## [11,] -0.349761669 -3.311279e-01
## [12,] -0.173457014 -3.452100e-01
## [13,] -0.018352260 -3.468925e-01
## [14,] 0.184516385 -2.544383e-01
## [15,] 0.331865281 -1.835839e-01
## [16,] 0.312360485 -1.025645e-01
## [17,] 0.375796576 -2.201596e-02
## [18,] 0.275608258 1.831515e-02
## [19,] 0.197085284 1.498987e-02
## [20,] 0.081316673 1.295551e-02
## [21,] -0.007312926 1.021088e-02
##
## Monte Carlo Negentropy approximation check:
##      SOTE
## Approx Negentropy 1.00581240
## MC Negentropy     1.02226057
## MC se             0.00312136
## Relative accuracy 0.98391000

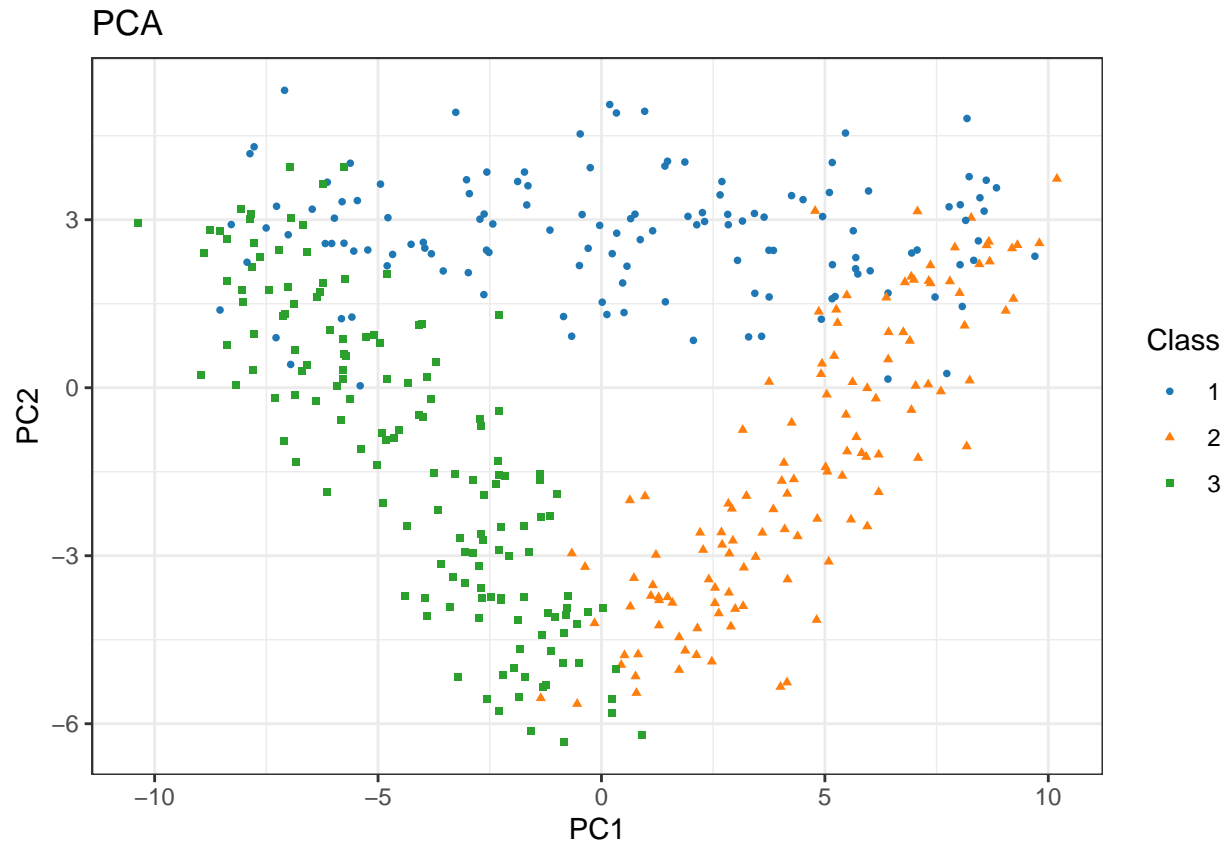
plot(PPGMMGA3, Class, drawAxis = FALSE) + ggtitle("PPGMMGA[SOTE]")
```



```
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(PPGMGA1)
```

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

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

```
## $Negentropy
```

```
## [1] 1.027086
```

```
##
```

```
## $se
```

```
## [1] 0.003138164
```

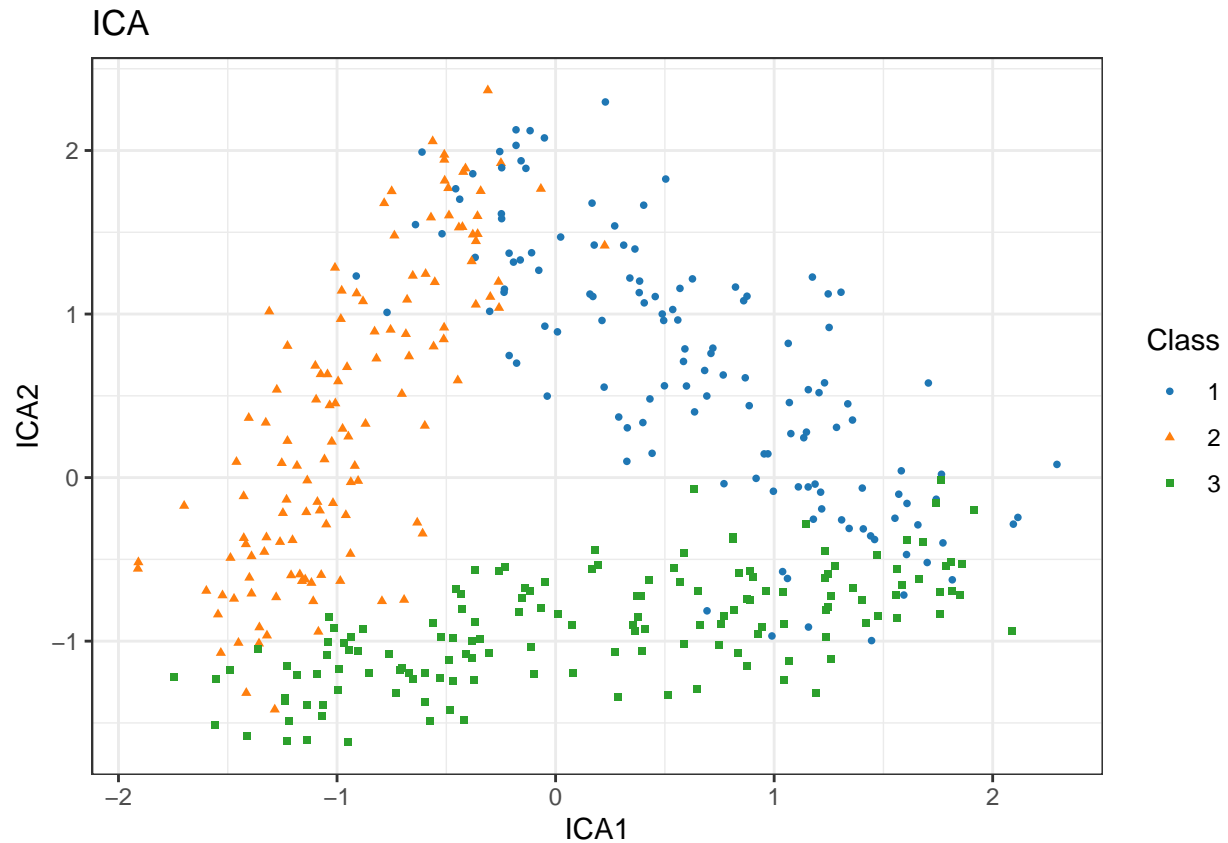
```
# trick for plotting
```

```
PPGMICA <- PPGMGA1; 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 89.01658  8.142409  9.034730 9.034730e+00
## VAR  NA  0.00000  89.754845 89.490559 8.949056e+01
## SOTE NA    NA  0.000000  6.032357 6.032357e+00
## PCA  NA    NA    NA  0.000000 3.146623e-13
## ICA  NA    NA    NA    NA  0.000000e+00
```

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

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

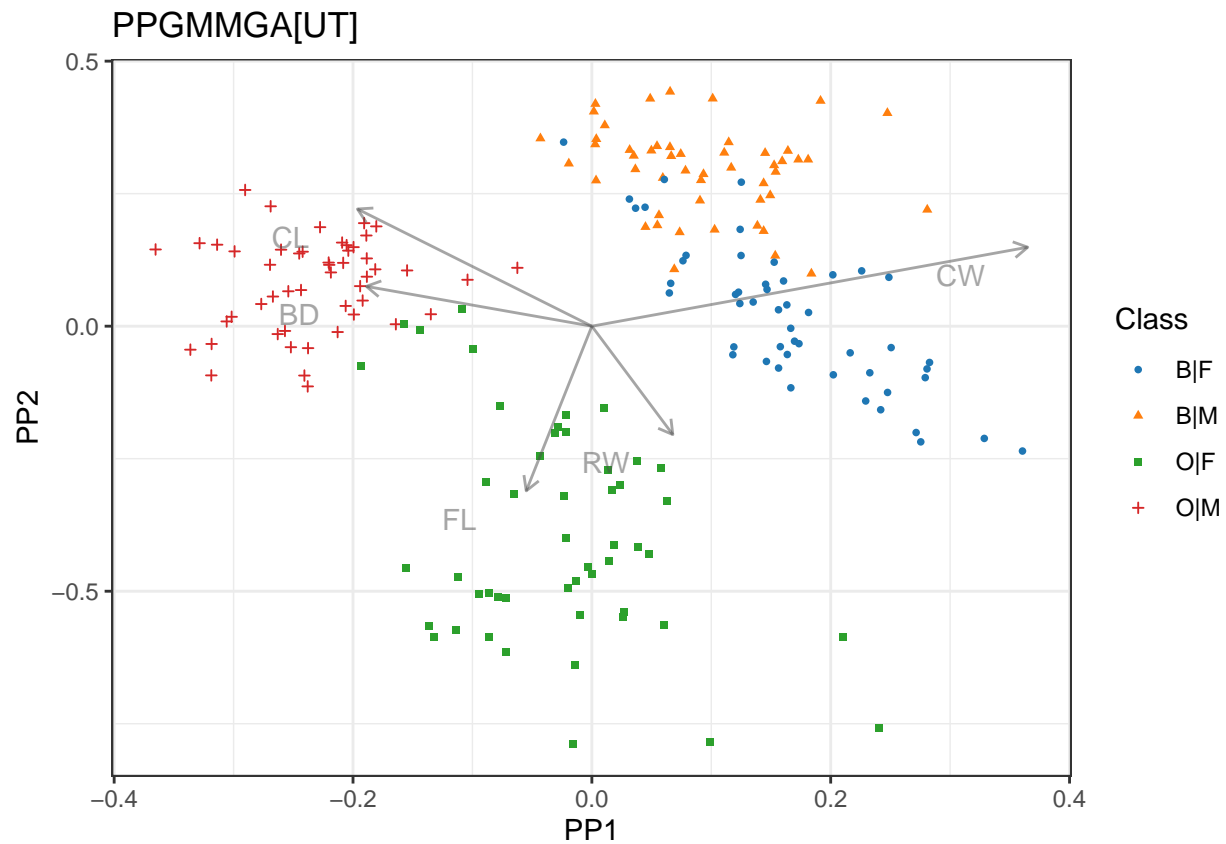
PPGMMGA1 <- ppgmmga(data = X, d = 2, approx = "UT", seed = 1)
summary(PPGMMGA1, check = TRUE)
```

```
## -- 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          x8
## [1,] 5.602055 1.990215 0.5319162 2.723246 1.010578 1.694094 1.978759 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]")
```

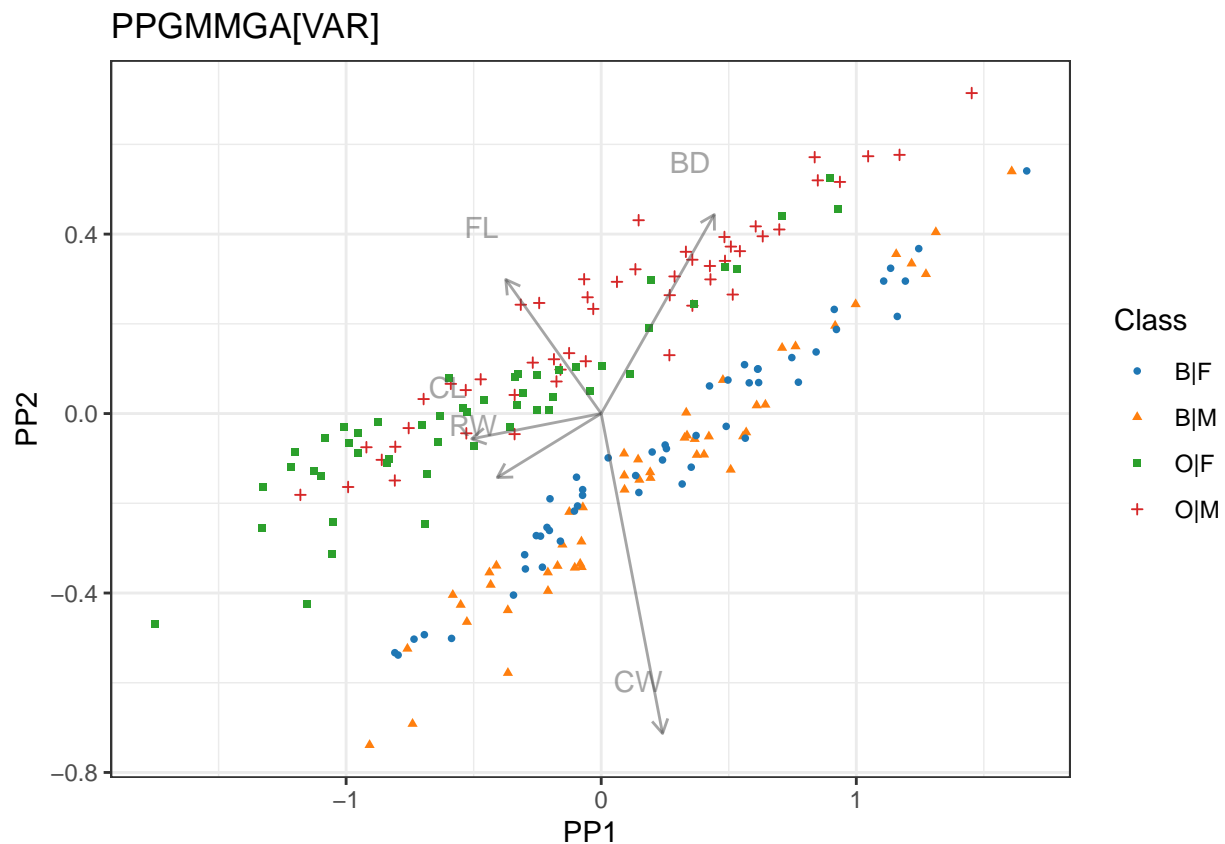


```
PPGMMGA2 <- ppgmmga(data = X, d = 2, approx = "VAR", seed = 2)
summary(PPGMMGA2, check = TRUE)
```

```
## -- 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      x8
## [1,] 3.884605 1.058383 1.260722 2.313448 3.332061 0.8057327 2.049418 2.367513
##
## Estimated projection basis:
##      PP1      PP2
## FL -0.4136463 0.33043049
## RW -0.4502869 -0.15755301
## CL -0.5612821 -0.06234149
## CW 0.2659398 -0.78843911
## BD 0.4902824 0.49037745
##
## Monte Carlo Negentropy approximation check:
##      VAR
## Approx Negentropy 0.271632884
## MC Negentropy     0.462154683
## MC se             0.002837045
## Relative accuracy 0.587753179
```

```
plot(PPGMGA2, Class) + ggtitle("PPGMMGA[VAR]")
```

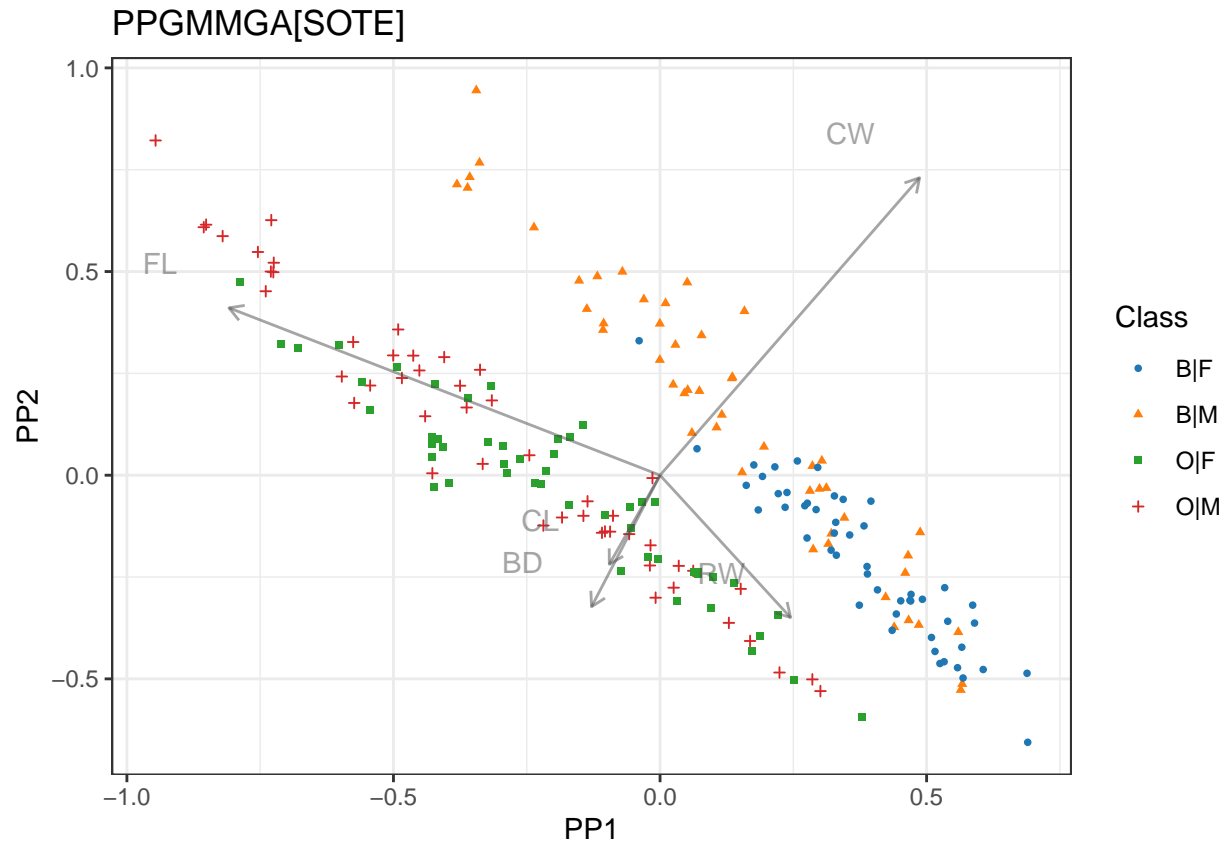


```
PPGMMGA3 <- ppgmga(data = X, d = 2, approx = "SOTE", seed = 3)
summary(PPGMMGA3, check = TRUE)
```

```

## -- 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 0.7000118
##
## Estimated projection basis:
##      PP1      PP2
## FL -0.81834537 0.4158548
## RW  0.24779744 -0.3536123
## CL -0.09627435 -0.2213935
## CW  0.49267164 0.7391007
## BD -0.13005070 -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]")

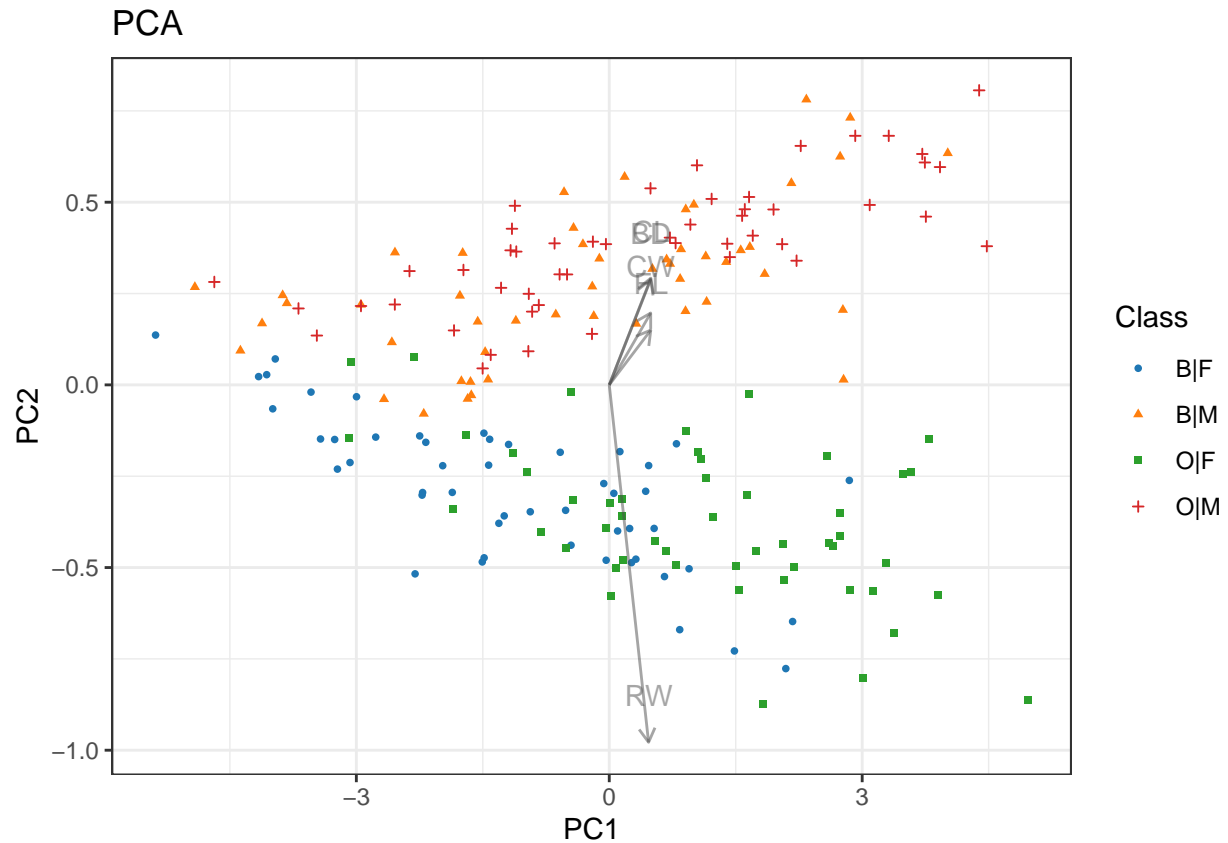
```



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

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

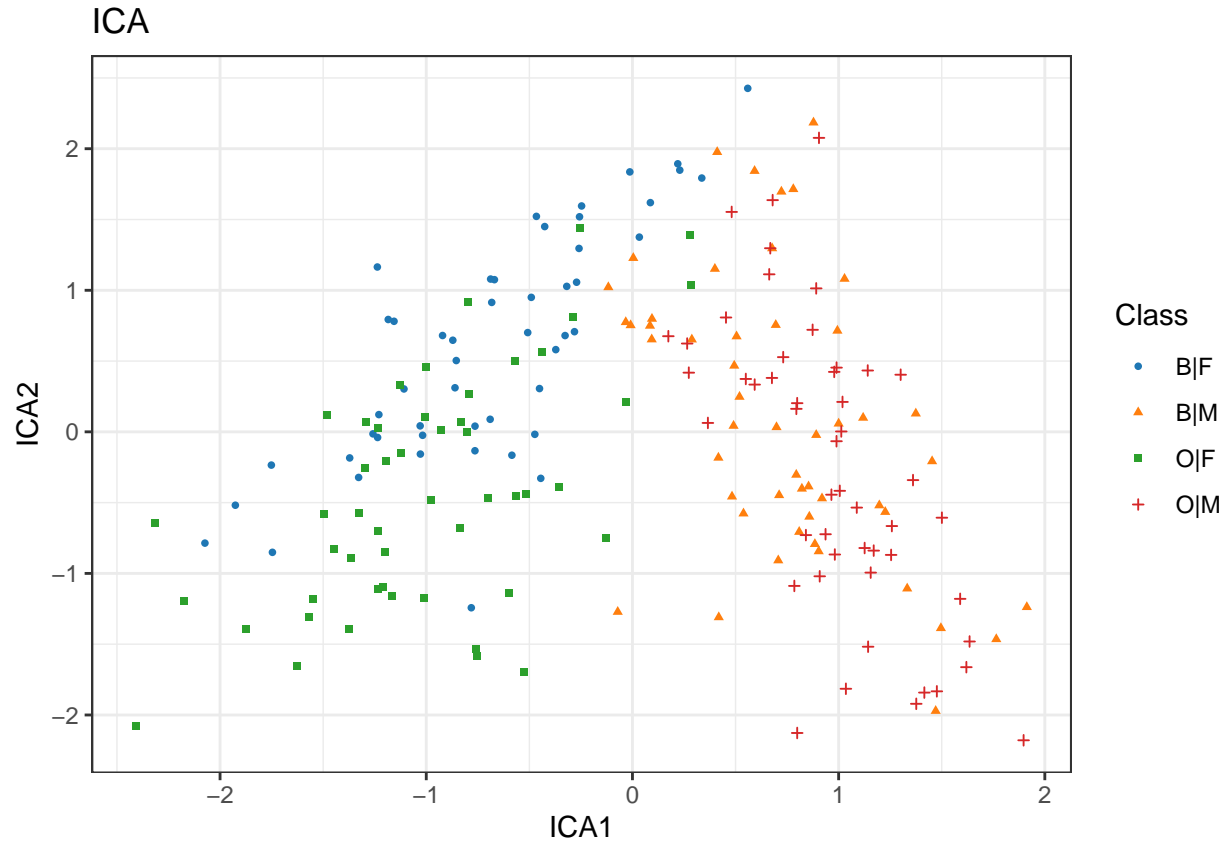
```
PPGMMPCA <- PPGMMGA1; PPGMMPCA$approx <- "PCA"
PPGMMPCA$basis <- PCA$basis
PPGMMPCA$Z <- PCA$Z
plot(PPGMMPCA, Class) + ggtitle("PCA")
```



```
ICA <- NegentropyFASTICA(PGMMGA1)
ICA[c("Negentropy", "se")]
```

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

```
# trick for plotting
PPGMICA <- PPGMGA1; 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 76.23147 66.29325 88.11079 8.811079e+01
## VAR  NA  0.00000 85.69386 88.20356 8.820356e+01
## SOTE NA      NA  0.00000 88.34848 8.834848e+01
## PCA  NA      NA      NA  0.00000 2.323089e-13
## 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"))

PPGMMGA1 <- ppgmmga(data = X, d = 1, 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
```

```

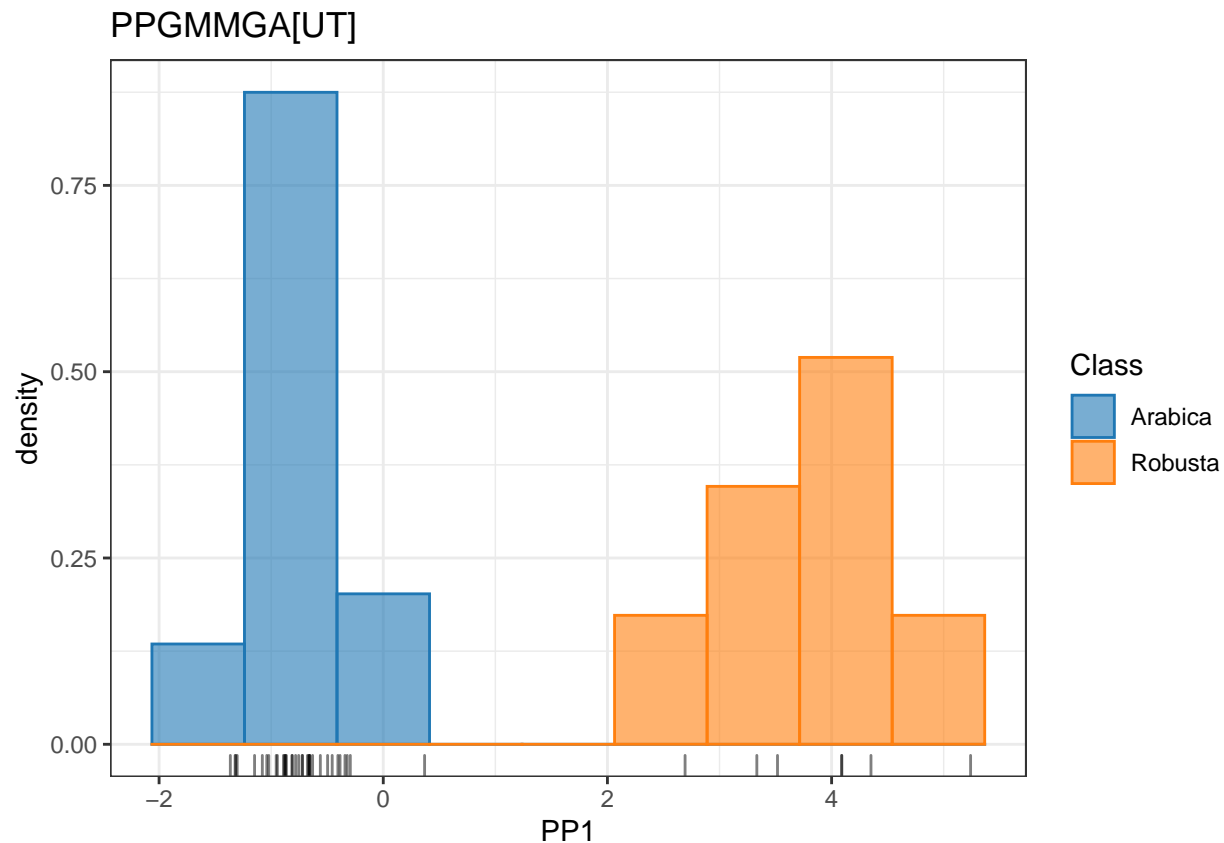
## GA encoded basis solution:
##      x1      x2      x3      x4      x5      x6      x7      x8
## [1,] 3.974492 1.336151 1.470531 1.514307 1.715505 0.9849445 2.792978 1.478995
##      x9      x10     x11
## [1,] 0.252026 0.9978684 1.750353
##
## Estimated projection basis:
##      PP1
## Water      -0.04133779
## Bean Weight -0.03758600
## Extract Yield -0.01014117
## ph Value     0.03663144
## Free Acid    0.26242157
## Mineral Content 0.02494652
## Fat         -0.74871707
## Caffeine     0.52862017
## Trigonelline -0.13932588
## Chlorogenic Acid 0.05463602
## Neochlorogenic Acid 0.09735438
## Isochlorogenic Acid 0.23249763
##
## 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]")

```



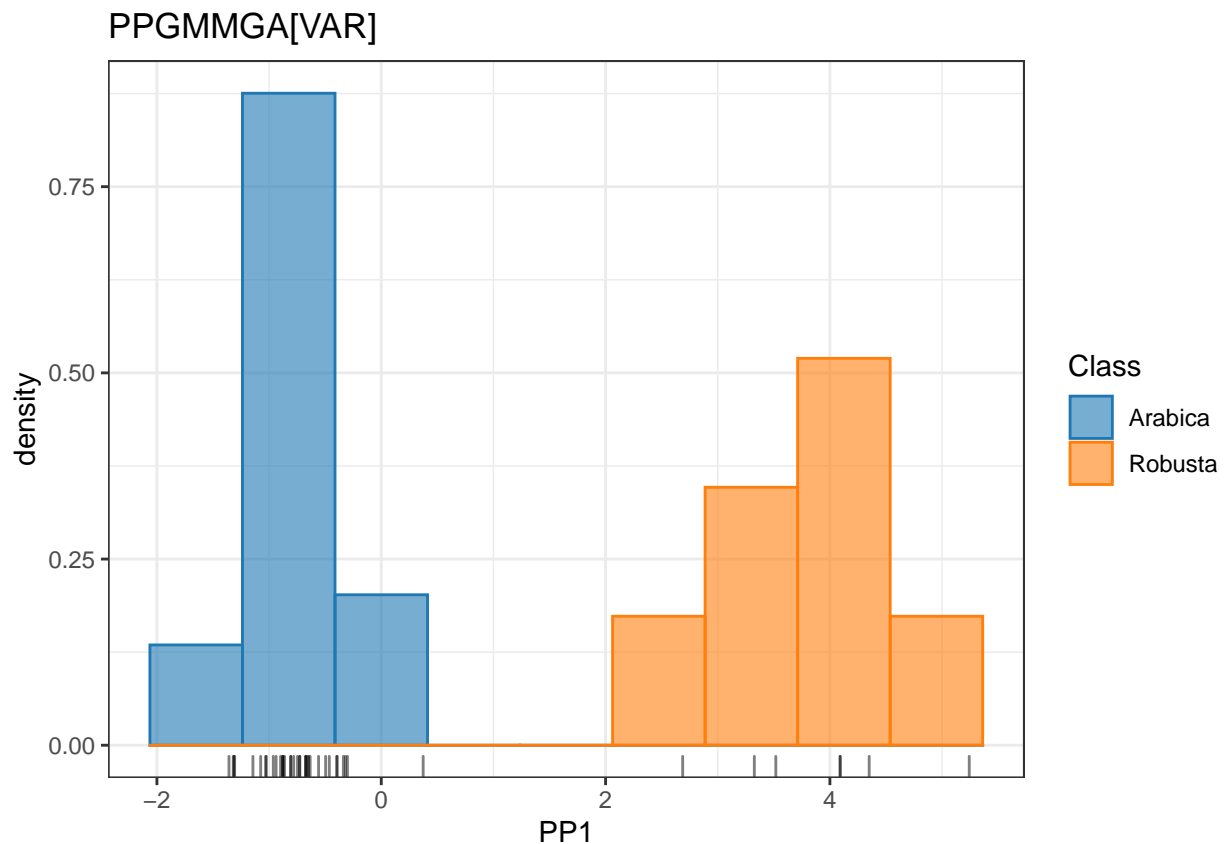
```
PPGMMGA2 <- ppgmga(data = X, d = 1, approx = "VAR", seed = 2)
summary(PPGMMGA2, check = TRUE)
```

```
## -- ppgmga -----
##
## 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,] 3.985399 1.335038 1.470806 1.514711 1.715593 0.9829951 2.795832 1.479609
##      x9      x10     x11
## [1,] 0.2606084 0.9750563 1.747732
##
## Estimated projection basis:
##      PP1
## Water      -0.04211335
## Bean Weight -0.03746027
## Extract Yield -0.01007805
## ph Value     0.03881464
## Free Acid    0.25939371
## Mineral Content 0.02454818
## Fat         -0.74834825
```



```
## Caffeine          0.53005071
## Trigonelline      -0.13937917
## Chlorogenic Acid   0.05423270
## Neochlorogenic Acid 0.09706206
## Isochlorogenic Acid 0.23358019
##
## 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]")
```



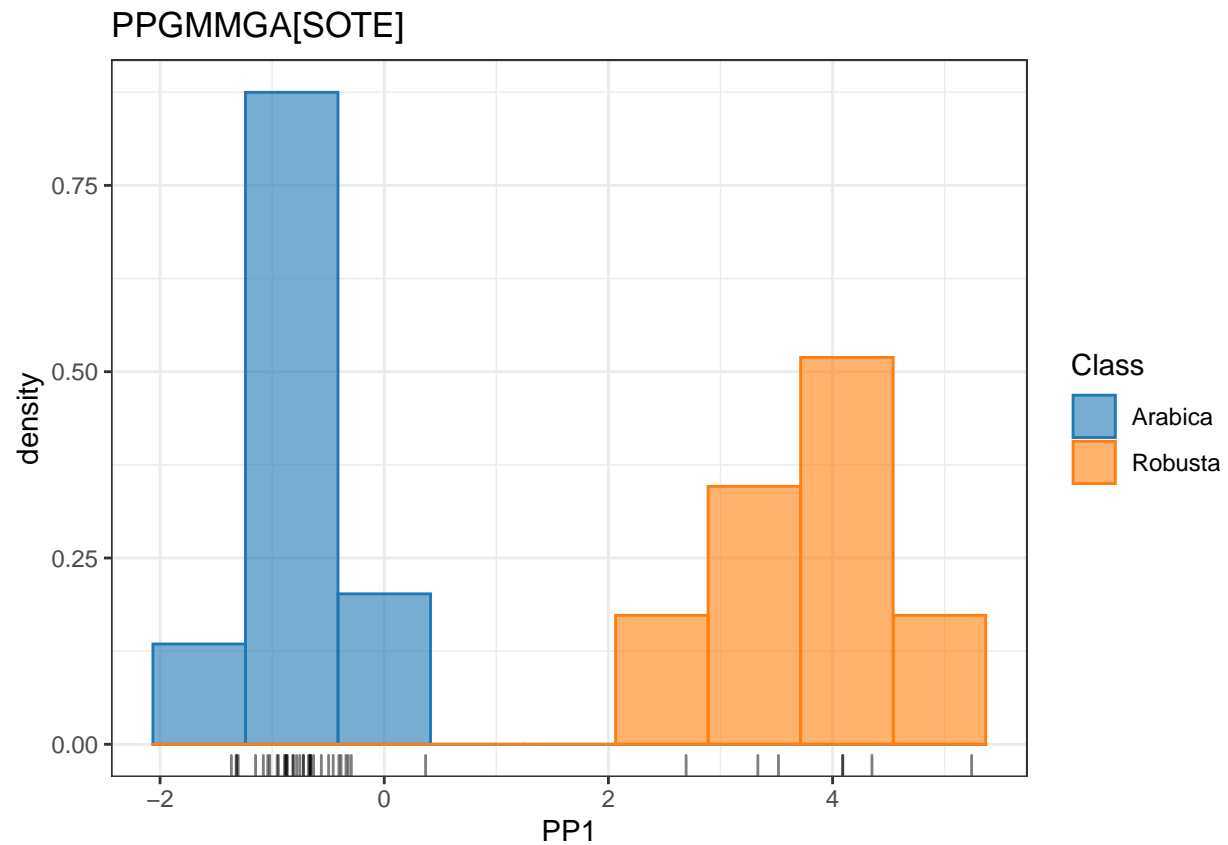
```
PPGMMGA3 <- ppgmmga(data = X, d = 1, approx = "SOTE", seed = 3)
summary(PPGMMGA3, check = TRUE)
```

```
## -- 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           x8
## [1,] 0.8329046 1.805445 1.671062 1.627284 1.426087 2.156657 0.3485631 1.662614
##           x9           x10          x11
## [1,] 2.889511 2.143055 1.390861
##
## Estimated projection basis:
##                               PP1
## Water                        -0.04135581
## Bean Weight                  -0.03760198
## Extract Yield                -0.01016740
## ph Value                     0.03659589
## Free Acid                    0.26237868
## Mineral Content              0.02494752
## Fat                          -0.74872622
## Caffeine                     0.52862642
## Trigonelline                 -0.13932556
## Chlorogenic Acid             0.05463441
## Neochlorogenic Acid          0.09735435
## Isochlorogenic Acid          0.23250154
##
## 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]")

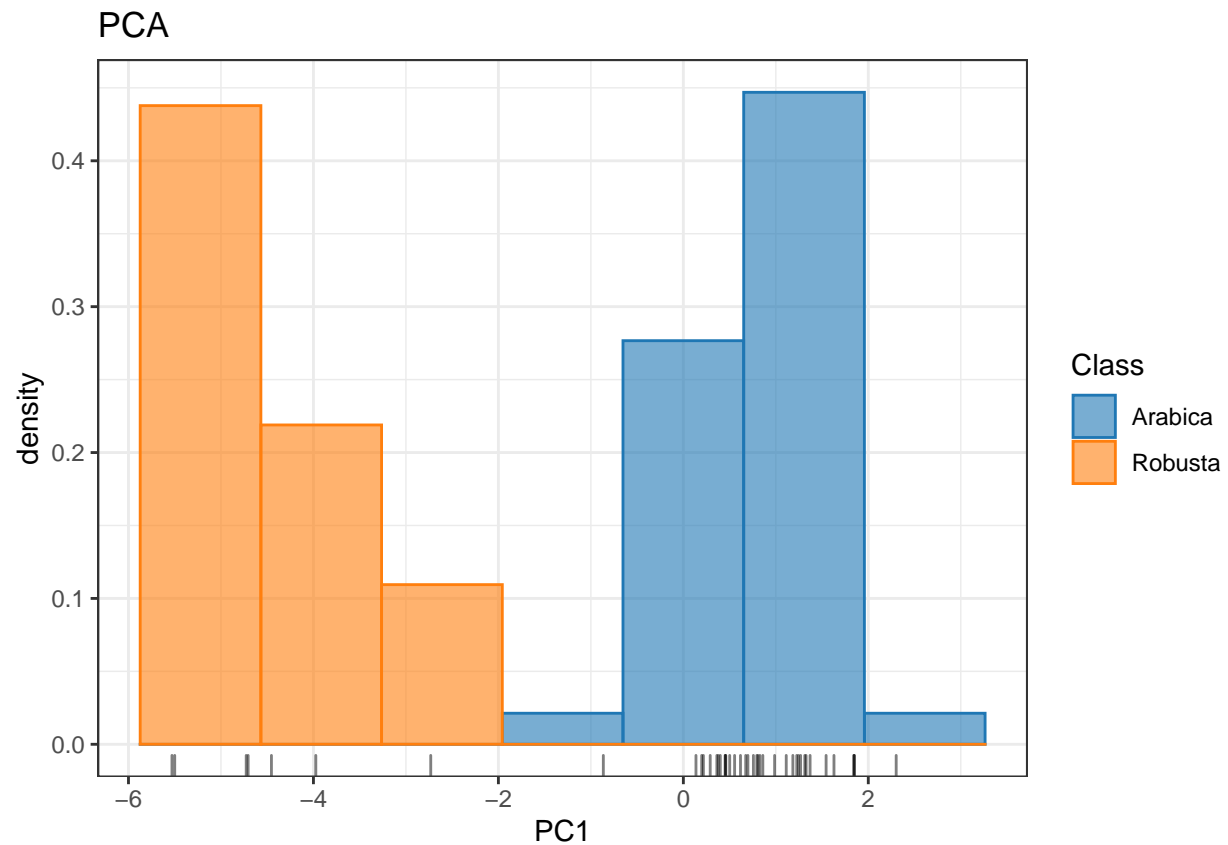
```



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

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

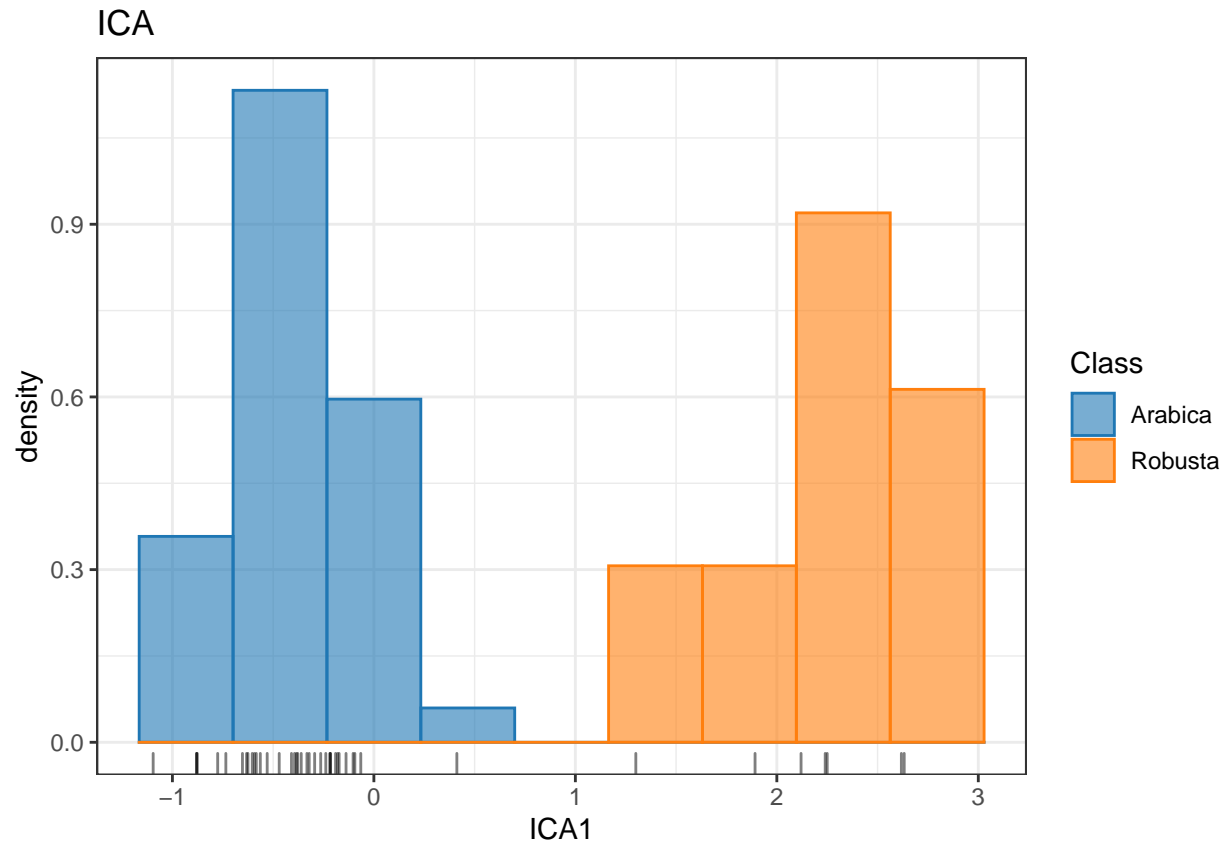
```
PPGMMPCA <- PPGMMGA1; PPGMMPCA$approx <- "PCA"
PPGMMPCA$basis <- PCA$basis
PPGMMPCA$Z <- PCA$Z
plot(PPGMMPCA, Class, nbins = 9) + ggtitle("PCA")
```



```
ICA <- NegentropyFASTICA(PPGMGA1)
ICA[c("Negentropy", "se")]

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

# trick for plotting
PPGMICA <- PPGMGA1; PPGMICA$approx <- "ICA"
PPGMICA$basis <- ICA$basis
PPGMICA$Z <- ICA$Z
plot(PPGMICA, Class, bins = 9) + ggtitle("ICA")
```

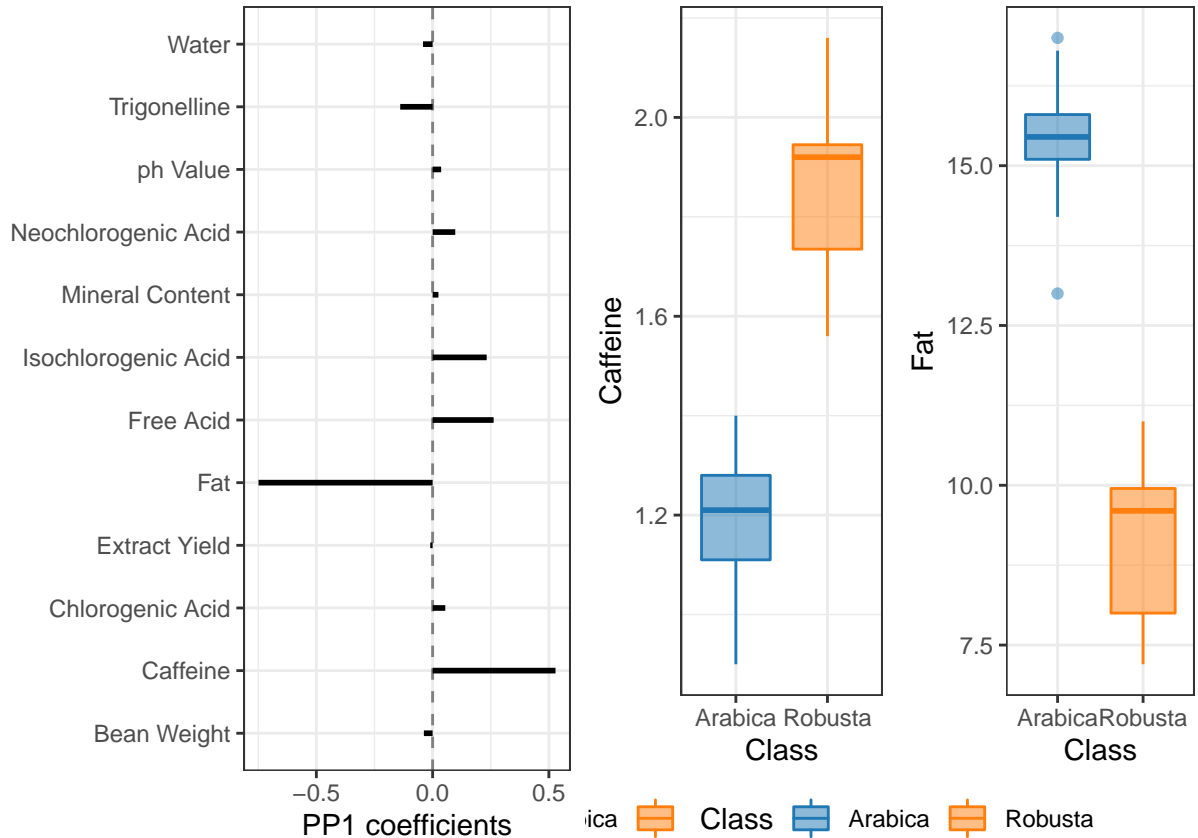


```
ppgmmga.table.angle(PPGMMGA1, PPGMMGA2, PPGMMGA3, PPGMMPCA, PPGMMICA)
```

```
##      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 1.950549e-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 <- data.frame(Caffeine = X$Caffeine, Fat = X$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"))

# d = 1
PPGMMGA1 <- ppgmmga(data = X, d = 1, 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,] 2.578943 1.71223 1.570522 1.084439 1.532505 1.574244 1.506406 1.552572
```

```

##           x9           x10
## [1,] 1.512312 1.401586
##
## Estimated projection basis:
##           PP1
## Ht   -0.4580430989
## Wt    0.7263071065
## LBM   -0.1466999874
## RCC   -0.0510050518
## WCC   -0.0159047878
## Hc    -0.0562744323
## Hg     0.0030150136
## Ferr  -0.0335047014
## BMI   -0.4627419594
## SSF   -0.0002720778
## Bfat   0.1409626924
##
## 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, 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           x8
## [1,] 2.50211 1.248956 1.744734 0.9432044 1.575921 1.631473 1.549986 1.597335
##           x9           x10
## [1,] 1.616311 1.758922
##
## Estimated projection basis:
##           PP1
## Ht   -0.441848179
## Wt    0.594089394
## LBM   0.140952643
## RCC   0.034327166
## WCC   0.020027317
## Hc    -0.015708249
## Hg     0.045860682
## Ferr  0.003875996
## BMI   -0.548642232
## SSF   0.164175721
## Bfat  -0.316313046
##

```

```

## 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, 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      x8
## [1,] 2.829624 1.458553 1.559532 1.273174 1.552948 1.577181 1.534831 1.570474
##      x9      x10
## [1,] 1.553346 1.080926
##
## Estimated projection basis:
##      PP1
## Ht   -0.2570311985
## Wt    0.7969975781
## LBM  -0.4465301163
## RCC  -0.0165622446
## WCC  -0.0003059067
## Hc   -0.0341522256
## Hg    0.0060642049
## Ferr -0.0169542366
## BMI  -0.2913841572
## SSF  -0.0111927153
## Bfat -0.1120074908
##
## 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, 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      x8
## [1,] 2.136126 1.587227 1.591097 0.9096454 1.577396 1.602194 1.590791 1.586196
##      x9      x10 ...      x19      x20
## [1,] 1.529604 2.209708      1.561785 1.395024
```

```
##
```

```
## Estimated projection basis:
```

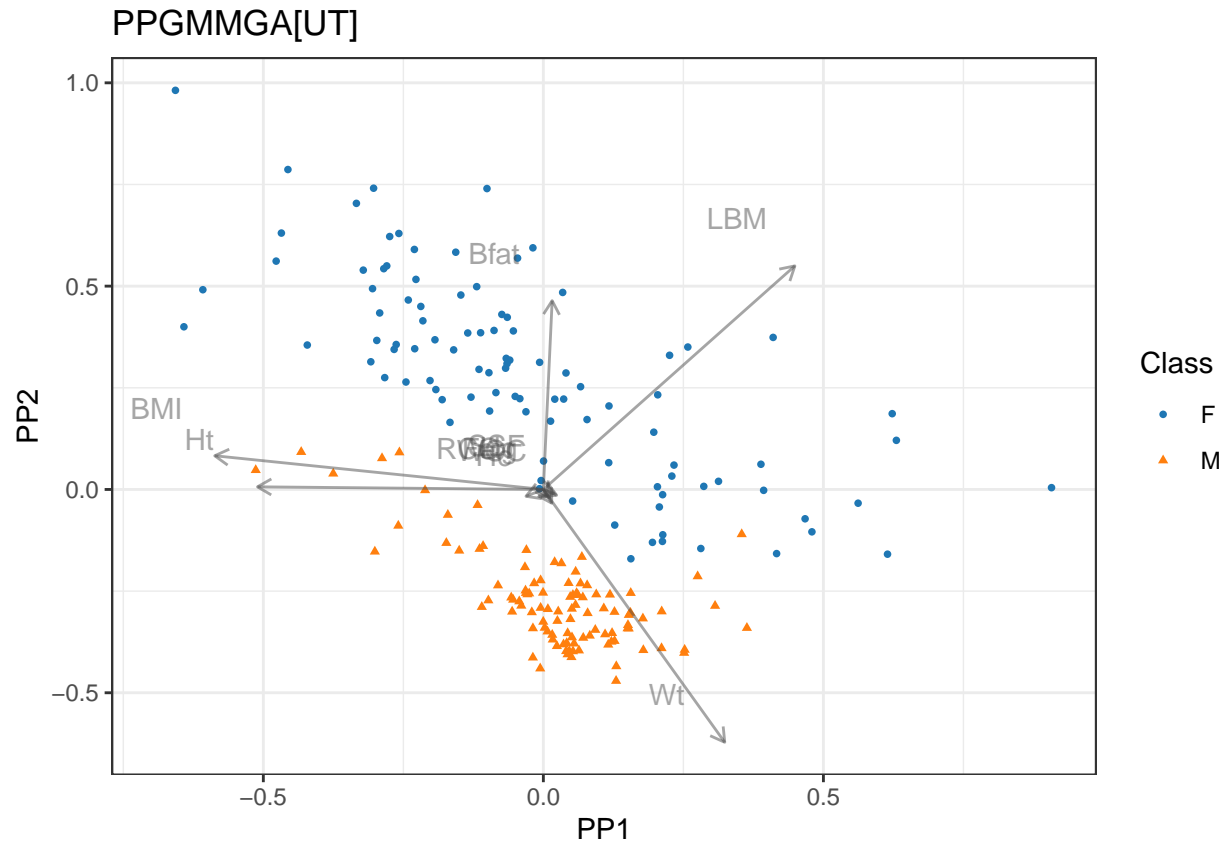
```
##      PP1      PP2
## Ht -0.533934574 0.006809153
## Wt  0.338727883 -0.650440783
## LBM 0.469717356 0.575110072
## RCC -0.032465113 -0.017550141
## WCC 0.012141141 -0.022754000
## Hc  0.015766981 -0.035444790
## Hg  0.024768302 -0.014873087
## Ferr 0.005207033 -0.017508383
## BMI -0.613816283 0.086723790
## SSF 0.020296520 -0.010037309
## Bfat 0.016430114 0.485692524
```

```
##
```

```
## 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] ")
```

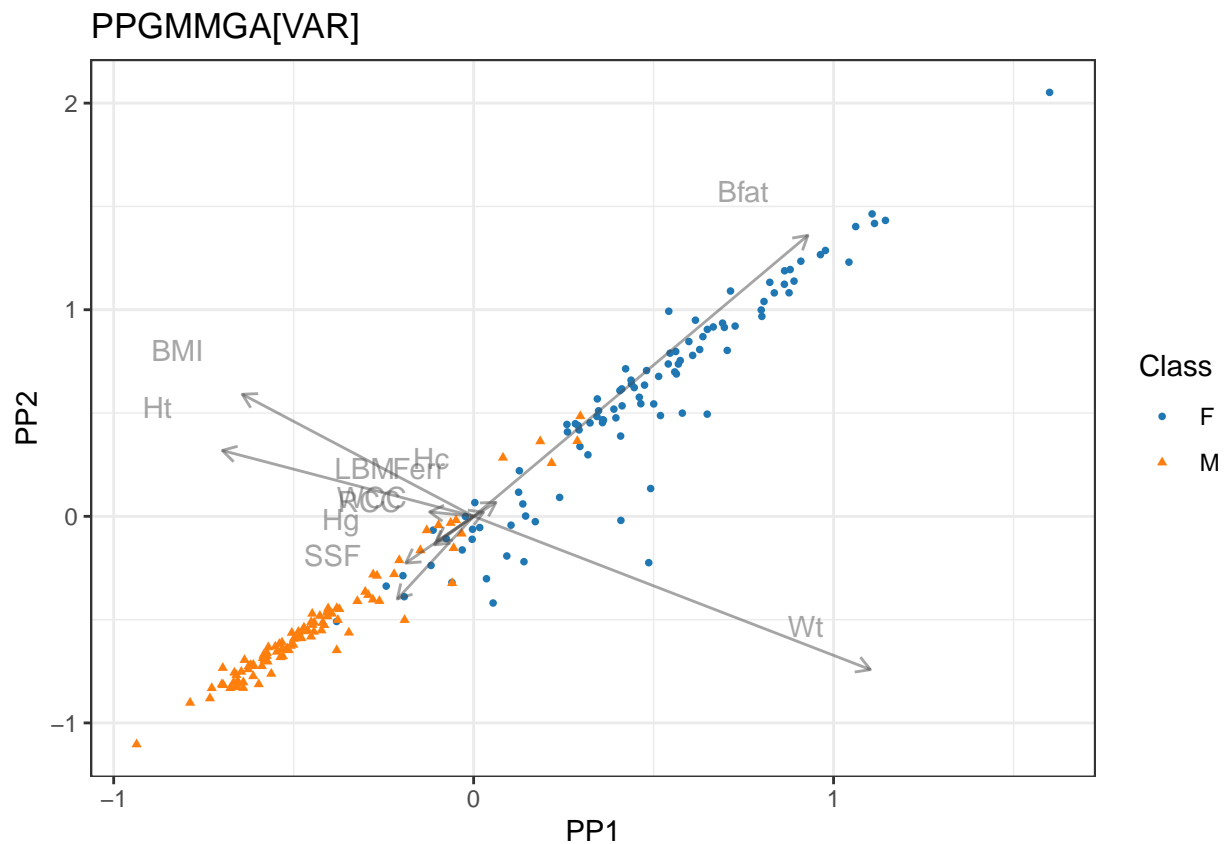


```
PPGMMGA2 <- ppgmga(data = X, d = 2, approx = "VAR", seed = 2)
summary(PPGMMGA2, check = TRUE)
```

```
## -- ppgmga -----
##
## 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      x8
## [1,] 2.576889 2.126278 1.428576 1.121862 1.591549 1.428802 1.617856 1.493145
##      x9      x10     ...      x19      x20
## [1,] 1.487974 1.477187      0.7094031 1.214405
##
## Estimated projection basis:
##      PP1      PP2
## Ht  -0.39657554 0.18103469
## Wt   0.62598534 -0.42121520
## LBM -0.06957117 0.01225575
## RCC -0.06178527 -0.07833838
## WCC -0.05811054 -0.06858472
## Hc   0.03527920 0.03844621
## Hg  -0.10720894 -0.12859893
```

```
## Ferr 0.01572355 0.01149466
## BMI -0.36502829 0.33551809
## SSF -0.12042972 -0.22720486
## Bfat 0.52735303 0.77231526
##
## Monte Carlo Negentropy approximation check:
##          VAR
## Approx Negentropy 0.664469205
## MC Negentropy    0.861798543
## MC se           0.003870203
## Relative accuracy 0.771026141
```

```
plot(PPGMGA2, Class) + ggtitle("PPGMMGA[VAR]")
```

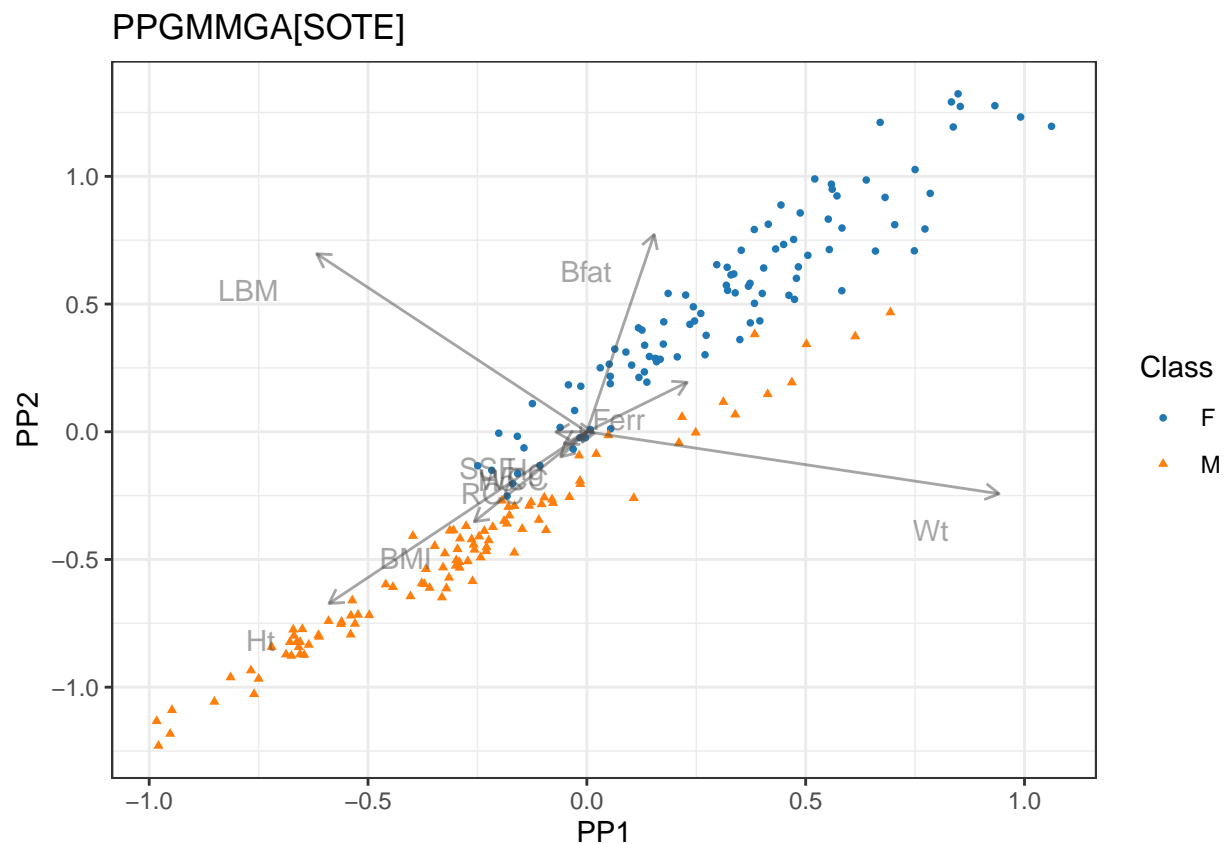


```
PPGMMGA3 <- ppgmmga(data = X, d = 2, approx = "SOTE", seed = 3)
summary(PPGMMGA3, check = TRUE)
```

```
## -- 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      x8
```

```
## [1,] 2.582654 1.686772 1.517114 1.374398 1.74885 1.582927 1.530416 1.562764
##          x9      x10    ...      x19      x20
## [1,] 1.523765 1.063616      1.658494 0.7380267
##
## Estimated projection basis:
##          PP1          PP2
## Ht   -0.442889575 -0.5055384348
## Wt    0.708088098 -0.1825067950
## LBM   -0.464083479  0.5240682520
## RCC   -0.044970177 -0.0739107522
## WCC   -0.007683044 -0.0309834018
## Hc    -0.038647257 -0.0359108172
## Hg     0.011613450 -0.0015633451
## Ferr  0.172293667  0.1456523483
## BMI   -0.193548529 -0.2649471029
## SSF   -0.053296590 -0.0005336044
## Bfat  0.115715591  0.5808166389
##
## Monte Carlo Negentropy approximation check:
##          SOTE
## Approx Negentropy 1.042831789
## MC Negentropy     0.235172122
## MC se              0.003039366
## Relative accuracy 4.434334221
```

```
plot(PPGMGA3, Class) + ggtitle("PPGMMGA[SOTE]")
```



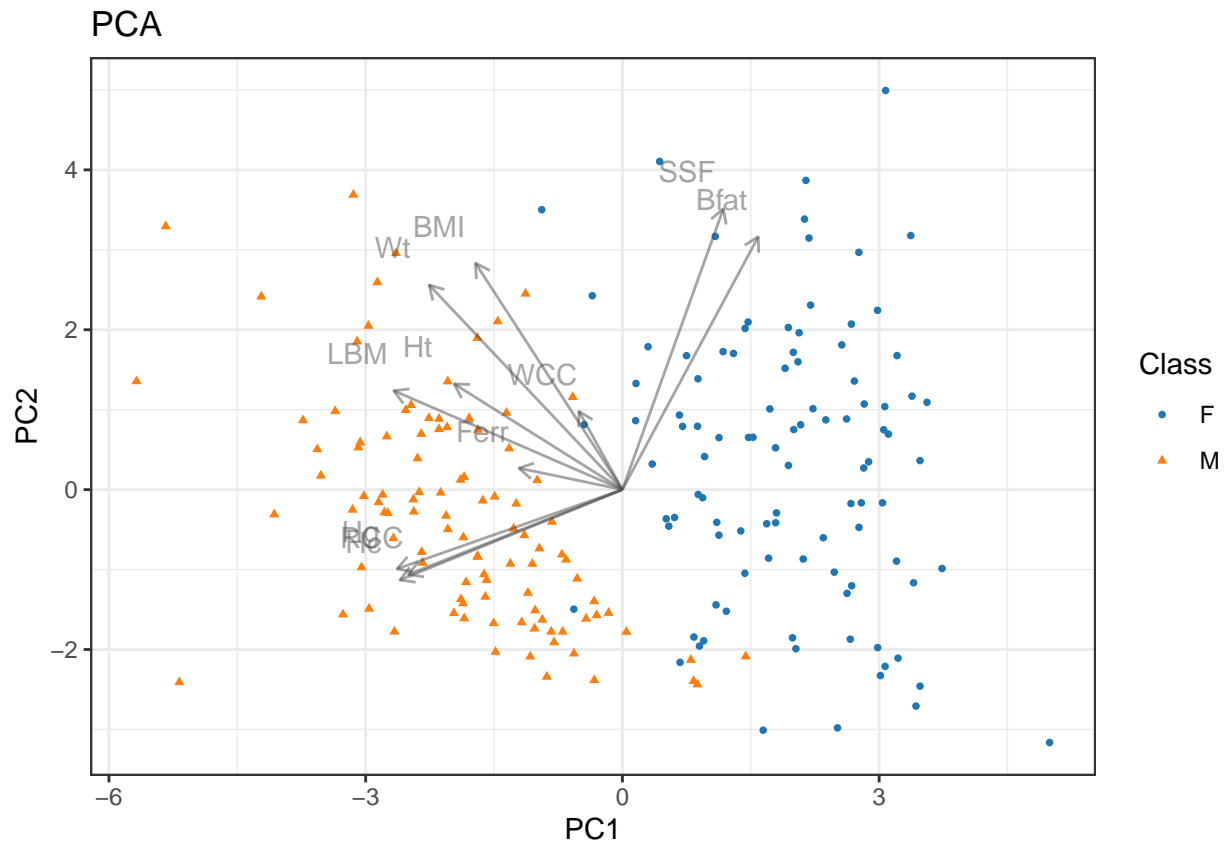
```

PCA <- NegentropyPCA(PPGMGA1)
PCA[c("Negentropy", "se")]

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

PPGMMPCA <- PPGMGA1; 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)

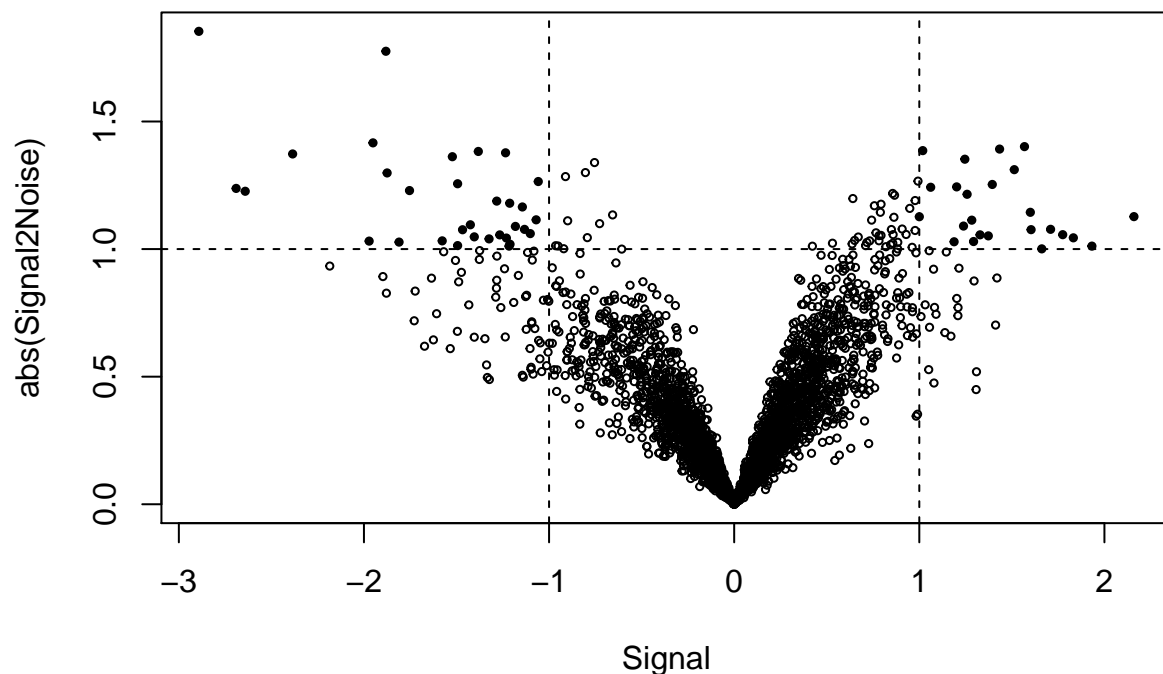
## Class
## ALL AML
## 27 11

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)

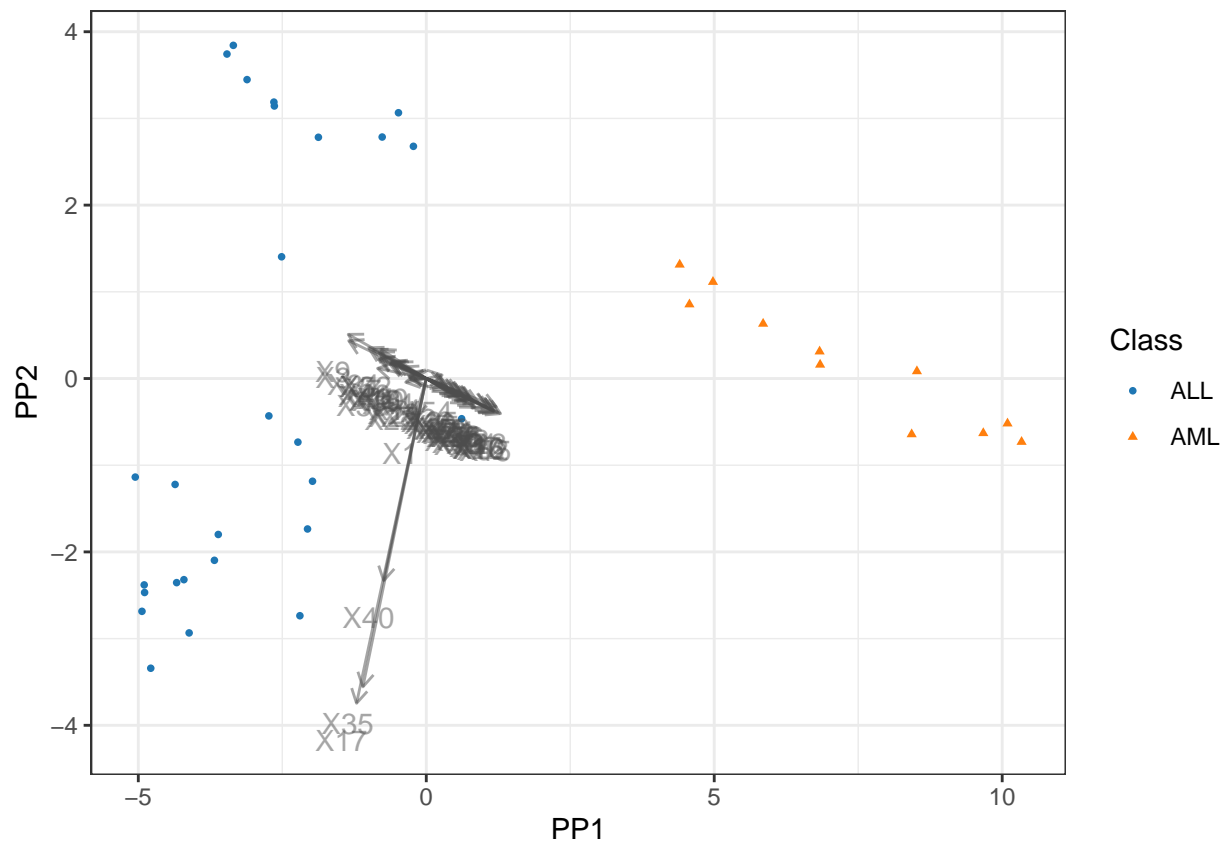
GMM2 <- densityMclust(X[,g], modelNames = c("EII", "VII", "EEI", "EVI", "VEI", "VVI"))
PPGMMGA <- ppgmnga(data = X[,g], d = 2, approx = "UT", seed = 1,
                   gmm = GMM2, scale = FALSE,
                   options = ppgmnga.options(maxiter = 2000))
summary(PPGMMGA, check = TRUE)
```

```
## -- ppgmnga -----
##
## 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      x8
## [1,] 2.961383 1.706882 1.427382 1.617797 1.663534 1.785555 1.710749 1.673226
##      x9      x10     ...      x109     x110
## [1,] 1.688247 1.425894      1.51975 1.49349
##
## Estimated projection basis:
##      PP1      PP2
## [1,] -0.03186584 -0.073813075
## [2,]  0.17490850 -0.053565852
```

```

## [3,] -0.22546431 0.072978554
## [4,] -0.09971397 0.033689424
## [5,] -0.15578786 0.055694931
## [6,] 0.09782733 -0.033652219
## [7,] -0.13708824 0.050961503
## [8,] -0.03656448 -0.001269107
## [9,] -0.22679716 0.085302730
## [10,] 0.13599589 -0.051608732
## [11,] 0.08347333 -0.026089592
## [12,] 0.19753093 -0.064312454
## [13,] 0.20363619 -0.054583305
## [14,] -0.04676711 0.011072147
## [15,] 0.21589262 -0.067756563
## [16,] 0.19461478 -0.061973720
## [17,] -0.20261864 -0.626998628
## [18,] -0.10362983 0.031060211
## [19,] -0.10612293 0.028118561
## [20,] 0.05527987 -0.015613317
## ...
## [55,] -0.14160222 0.019644267
## [56,] 0.13566559 -0.041299539
##
## Monte Carlo Negentropy approximation check:
##                               UT
## Approx Negentropy 3.052731483
## MC Negentropy     3.052821017
## MC se             0.003639312
## Relative accuracy 0.999970672
plot(PPGMGA, 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) +
  xlim(c(-6,11)) + ylim(c(-4,4)) +
  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)
```

