

```

# xy = matrix(runif(10000), ncol = 2)
# xy = xy[xy[,1] < 0.1 | xy[,2] < 0.1,]
# xy = xy %%% matrix(c(1, 0.4, 0.4, 1), ncol = 2)
# xy = xy[xy[,1] <= 1 & xy[,2] <= 1,]

xyc = read.csv("synthetic_data.csv")
xy = xyc[,1:2]

set.seed(1234)
subset = sample.int(nrow(xy), 300)

library(fastICA)
library(NMF)

## Loading required package: methods
## Loading required package: pkgmaker
## Loading required package: registry
## Loading required package: rngtools
## Loading required package: cluster
## NMF - BioConductor layer [OK] | Shared memory capabilities [OK]
| Cores 2/2

fit.pca = prcomp(xy, center = TRUE, scale = FALSE)

temp = replicate(1000, fastICA(xy, 2, method = "C"), simplify = FALSE)
temp2 = sapply(temp, function(x) shapiro.test(x$S)$statistic)
fit.ica = temp[[which.max(temp2)]]

fit.nmf = nmf(t(xy[subset,]), rank = 2, nrun = 20, method = "snmf/r")

library(NMF)

## Loading required package: methods
## Loading required package: pkgmaker
## Loading required package: registry
## Loading required package: rngtools
## Loading required package: cluster
## NMF - BioConductor layer [OK] | Shared memory capabilities [OK]
| Cores 3/4

library(RColorBrewer)
library(ggplot2)
library(grid)

pal = brewer.pal(3, "Set2")[c(2, 3, 1)]

```

```

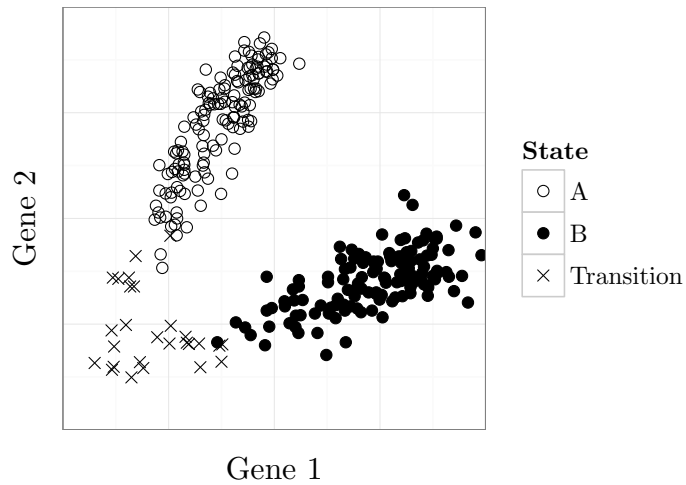
pal = sapply(pal, function(col) do.call(rgb, c(as.list(col2rgb(col)/255), alpha = 0.5)))
syms = c(19, 4, 21)
col = pal[xyc[,3]]
pch = syms[xyc[,3]]

# plot(0 ~ 0, type = "n", xlab = "Gene 1", ylab = "Gene 2", xlim = c(0, 1), ylim = c(0, 1),
# # arrows(
# #   x0 = c(0, 0),
# #   y0 = c(0, 0),
# #   x1 = c(0.5, 1),
# #   y1 = c(1, 0.45),
# #   col = "lightgrey", lwd = 5)
# points(xy[subset,1], xy[subset,2], col = "black", pch = pch[subset])
# legend("topright", legend = c("State A", "State B", "Transition State"), pch = syms[c(3, 1, 2)])

# for (spec in list(list(fit.pca$rotation, colMeans(xy)), list(fit.ica$A, colMeans(xy)), list(fit.ica$B, colMeans(xy))))
# {
#   plot(xy[subset,1], xy[subset,2], col = rgb(0, 0, 0, 0.5), pch = pch[subset], xlab = "", ylab = "")
#   temp = spec[[1]]
#   temp = t(t(temp) / sqrt(colSums(temp^2)) / 2)
#   arrows(
#     x0 = spec[[2]][1],
#     y0 = spec[[2]][2],
#     x1 = temp[1,] + spec[[2]][1],
#     y1 = temp[2,] + spec[[2]][2],
#     col = "black", lwd = 5)
# }

ggplot(data.frame(x = xyc[,1], y = xyc[,2], State = as.factor(c("B", "Transition", "A")[xyc[,3]])) +
  geom_point() +
  coord_fixed(xlim = c(0, 1), ylim = c(0, 1), ratio = 1) +
  theme_bw() +
  xlab("Gene 1") + ylab("Gene 2") +
  theme(axis.ticks = element_blank(), axis.text.x = element_blank(), axis.text.y = element_blank()) +
  scale_shape_manual(values = c(21, 19, 4))

```



```
baseplot = ggplot(data.frame(x = xyc[,1], y = xyc[,2], State = as.factor(c("B", "Transition"))),
  geom_point(col = "black", alpha = 0.7) +
  coord_fixed(xlim = c(0, 1), ylim = c(0, 1), ratio = 1) +
  theme_bw() +
  theme(axis.ticks = element_blank(), axis.text.x = element_blank(), axis.text.y = element_blank()) +
  scale_shape_manual(values = c(21, 19, 4)) +
  theme(legend.position="none")

for (spec in list(list(fit.pca$rotation, colMeans(xy)), list(fit.ica$A, colMeans(xy)), list(fit.ica$B, colMeans(xy)))) {
  temp = spec[[1]]
  temp = t(t(temp) / sqrt(colSums(temp^2)) / 2)
  thisplot = baseplot +
    geom_segment(x = spec[[2]][1], xend = temp[1,1] + spec[[2]][1], y = spec[[2]][1], yend = temp[1,2] + spec[[2]][1],
      arrow = arrow(length = unit(0.5, "cm")), size = 1.5, colour = "black") +
    geom_segment(x = spec[[2]][1], xend = temp[1,2] + spec[[2]][1], y = spec[[2]][1], yend = temp[1,2] + spec[[2]][1],
      arrow = arrow(length = unit(0.5, "cm")), size = 1.5, colour = "black")
  print(thisplot)
}
```

