Factor analysis to replicate Topic model results

Kushal K Dey

April 9, 2016

Contents

Overview

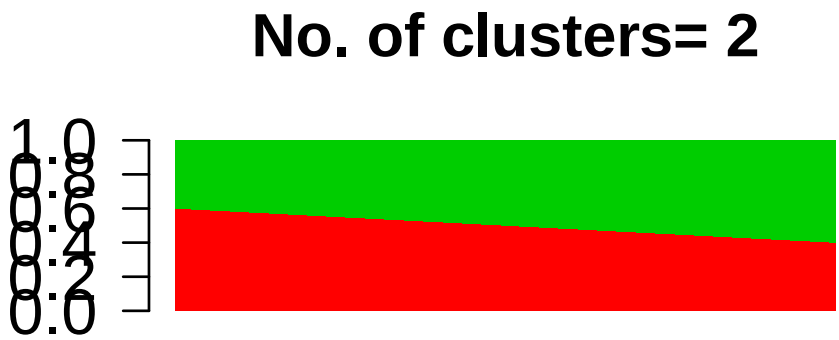
The package PMA due to Witten and Tibshirani is a very popular matrix decomposition package. In this script, we check if we use a variance scaling transform using Cholesky decomposition that sort of tries to pool the Poisson model features.

Simulation experiment 1

```
n.out <- 800
omega_sim <- cbind(seq(0.6, 0.4, length.out = n.out),
    1 - seq(0.6, 0.4, length.out = n.out))
colSums(omega_sim)
```

```
## [1] 400 400
```

```
K <- dim(omega_sim)[2]
barplot(t(omega_sim), col = 2:(K + 1), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```



```
freq <- rbind(c(0.1, 0.2, rep(0.7/98, 98)), c(rep(0.7/98,
    98), 0.1, 0.2))

counts <- t(do.call(cbind, lapply(1:dim(omega_sim)[1],
    function(x) rmultinom(1, 1000, prob = omega_sim[x,
        ] %*% freq))))

lambda <- 1000 * (omega_sim %*% freq)
lambda[lambda == 0] <- 1e-04
dim(lambda)
```

```
## [1] 800 100
```

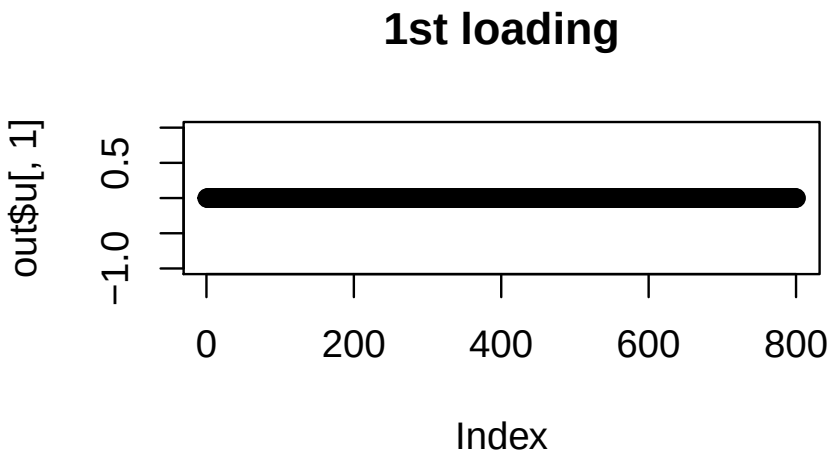
```
scaled_counts <- counts/sqrt(lambda)

require(PMA)
```

```
## Loading required package: PMA
## Loading required package: plyr
## Loading required package: impute
```

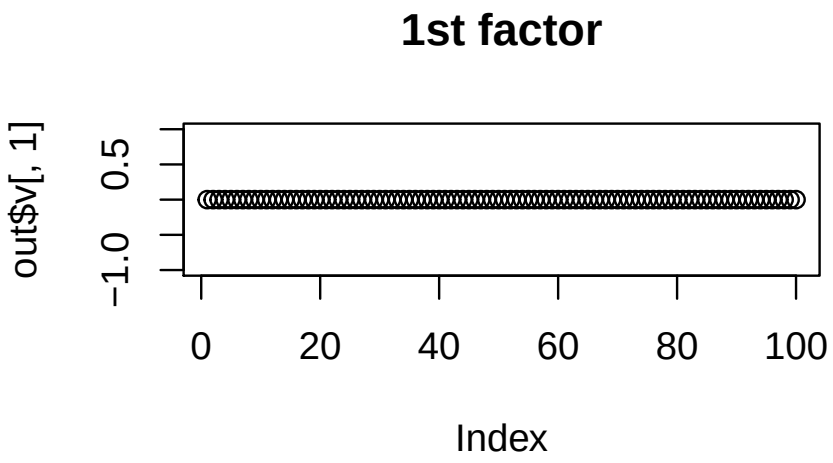
```
out <- PMD(scaled_counts, K = 4, upos = TRUE,
    vpos = TRUE, center = TRUE, sumabs = 1, niter = 20000,
    sumabsu = sqrt(600))
```

```
## 12
## 123456
## 12345678910111213141516171819202122232425262728293031323334353637383940414243444546474849505152535455 56
## 1234567
```

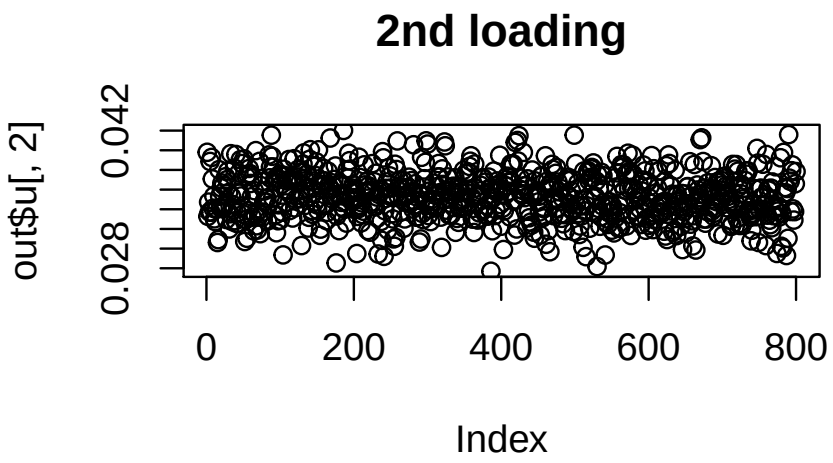
```
plot(out$u[, 1], main = "1st loading")
```



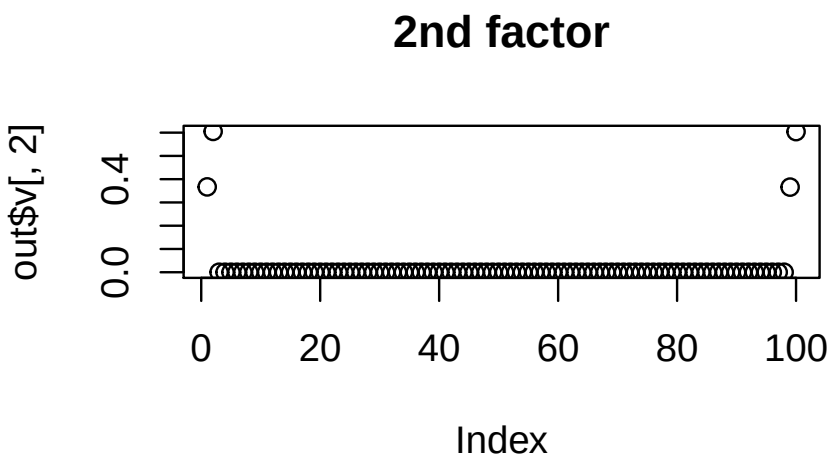
```r
plot(out$v[, 1], main = "1st factor")
```



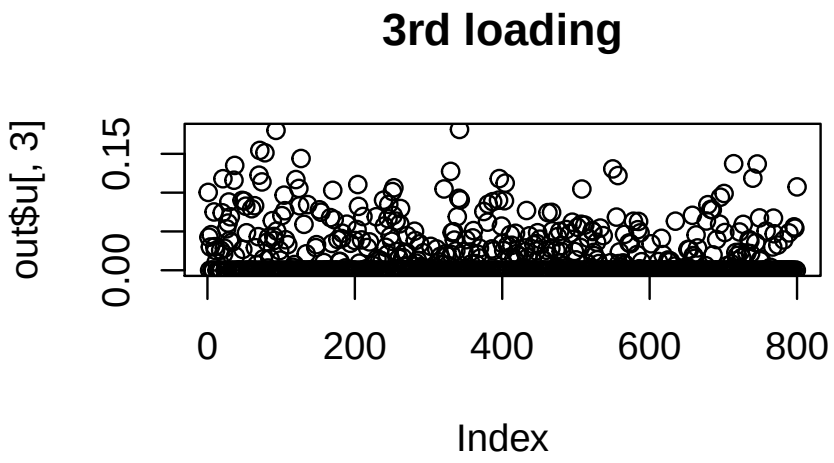

```r
plot(out$u[, 2], main = "2nd loading")
```



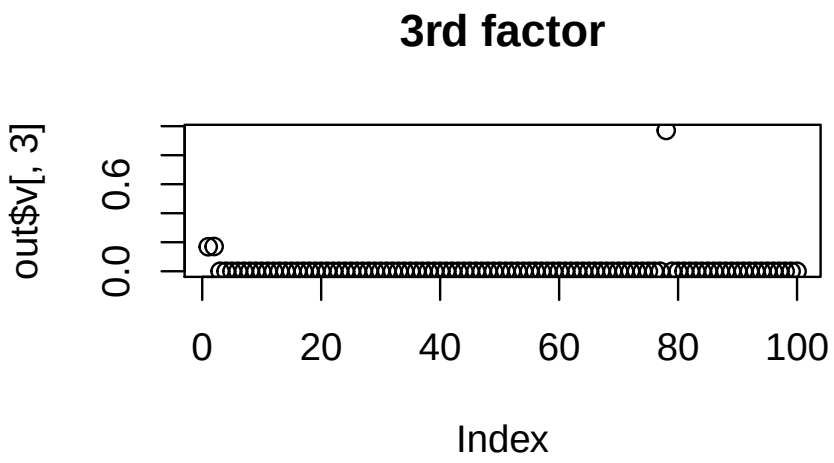

```r
plot(out$v[, 2], main = "2nd factor")
```



```
plot(out$u[, 3], main = "3rd loading")
```

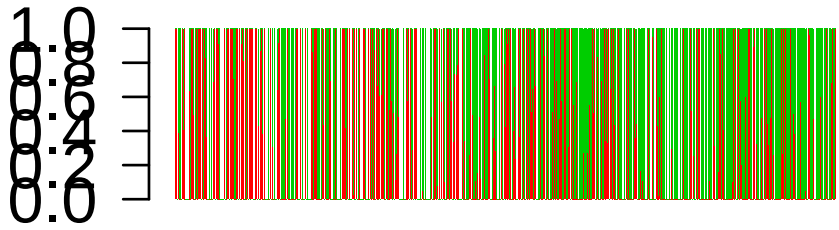



```
plot(out$v[, 3], main = "3rd factor")
```

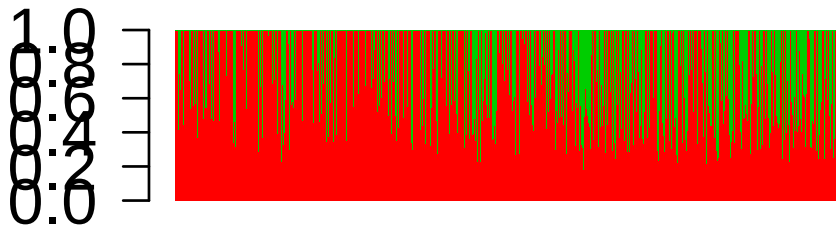



```
omega1 <- maptpx::normalize(cbind(out$u[, 2],
    out$u[, 3]), byrow = TRUE)

barplot(t(omega1), col = 2:(K + 1), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```

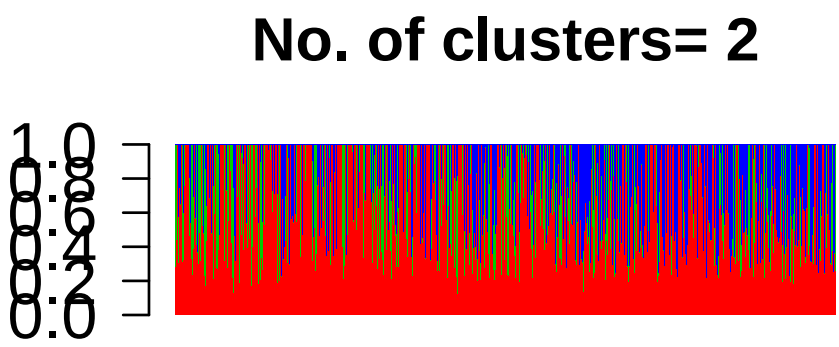

```
omega2 <- maptpx::normalize(cbind(out$u[, 3],
    out$u[, 4]), byrow = TRUE)

barplot(t(omega2), col = 2:(K + 1), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```



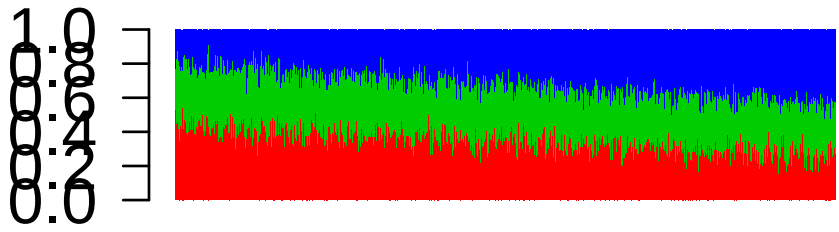

```
omega3 <- maptpx::normalize(cbind(out$u[, 2],
    out$u[, 4]), byrow = TRUE)

barplot(t(omega3), col = 2:(K + 1), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```



```
omega4 <- maptpx::normalize(cbind(out$u[, 2],
    out$u[, 3], out$u[, 4]), byrow = TRUE)

barplot(t(omega4), col = 2:(K + 2), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```

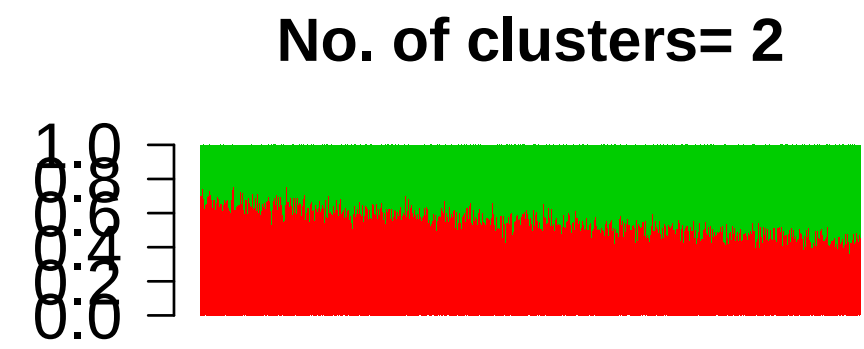



```
tpx.fit <- maptpx::topics(counts, K = 3)

##
## Estimating on a 800 document collection.
## Fitting the 3 topic model.
## log posterior increase: 322, 13.7, 9.2, 16.9, 58.9, 24.1, 11.2, 5.1, 2, 1, 0.5, 0.3, 0.3, 0.3, 0.2, 0.2

barplot(t(tpx.fit$omega), col = 2:(K + 2), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```




```
tpx.fit <- maptpx::topics(counts, K = 2)

##
## Estimating on a 800 document collection.
## Fitting the 2 topic model.
## log posterior increase: 89.8, 6.2, 2.3, 2.6, 3, 3.6, 4.5, 5.8, 7.7, 10.1, 12.8, 14.7, 14.9, 8.8, 4.9, 2
```

```
barplot(t(tpx.fit$omega), col = 2:(K + 2), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```



Simulation experiment 2

```
n.out <- 200
omega_sim <- rbind(cbind(rep(1, n.out), rep(0,
    n.out)), cbind(rep(0, n.out), rep(1, n.out)),
    cbind(seq(0.6, 0.4, length.out = n.out), 1 -
        seq(0.6, 0.4, length.out = n.out)))
dim(omega_sim)
```

```
## [1] 600   2
```

```
K <- dim(omega_sim)[2]

par(mar = c(2, 2, 2, 2))
barplot(t(omega_sim), col = 2:(K + 1), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```


```
freq <- rbind(c(0.1, 0.2, rep(0.7/98, 98)), c(rep(0.7/98,
    98), 0.1, 0.2))
str(freq)
```

```
##  num [1:2, 1:100] 0.1 0.00714 0.2 0.00714 0.00714 ...
```

```
counts <- t(do.call(cbind, lapply(1:dim(omega_sim)[1],
    function(x) rmultinom(1, 1000, prob = omega_sim[x,
        ] %*% freq))))
dim(counts)
```

```
## [1] 600 100
```

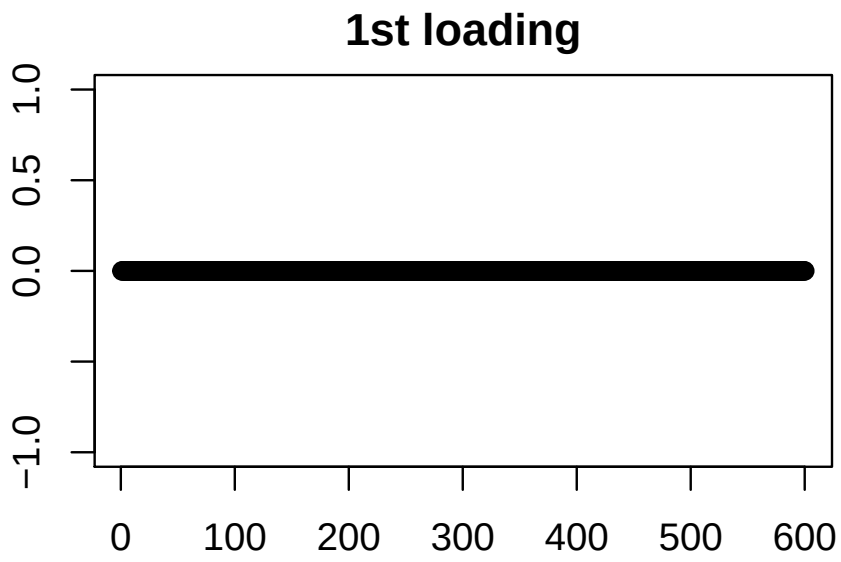
```
lambda <- 1000 * (omega_sim %*% freq)
lambda[lambda == 0] <- 1e-04
dim(lambda)
```

```
## [1] 600 100
```

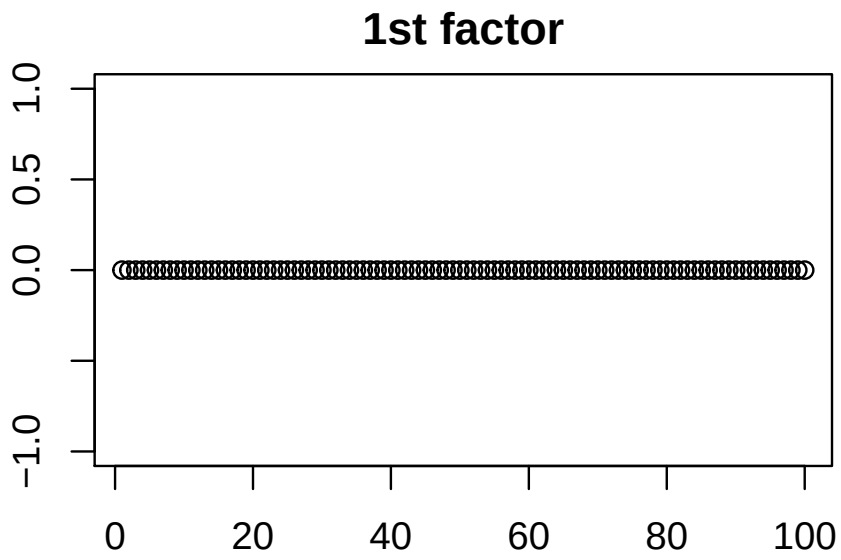
```
scaled_counts <- counts/sqrt(lambda)

require(PMA)
out <- PMD(scaled_counts, K = 4, upos = TRUE,
    vpos = TRUE, center = TRUE, sumabs = 1, niter = 20000,
    sumabsu = sqrt(600))
```

```
## 12
## 12345678910111213141516171819202122232425262728293031323 3
## 12345678
## 123456789101112131415
```

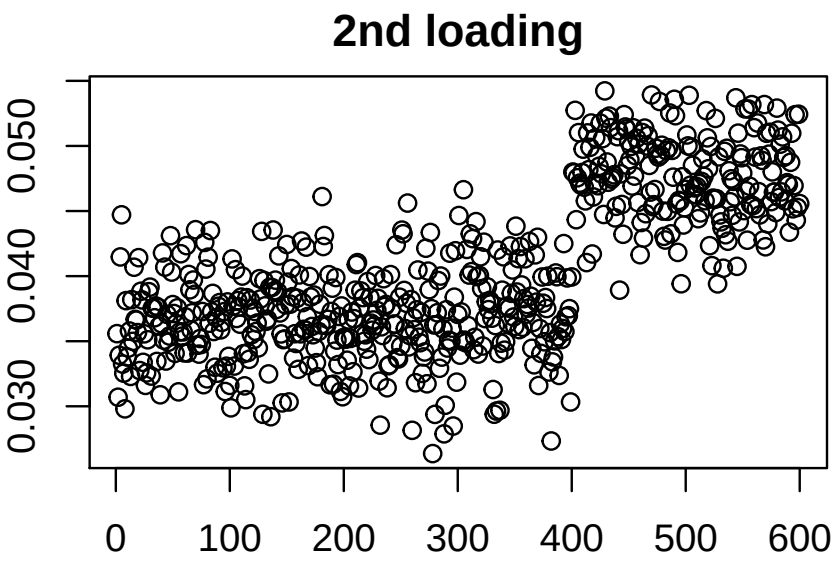
```
plot(out$u[, 1], main = "1st loading")
```



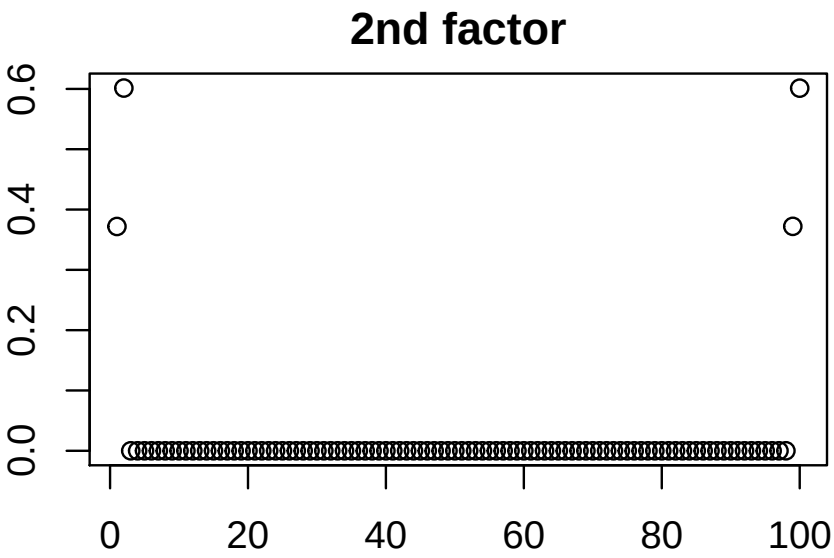

```
plot(out$v[, 1], main = "1st factor")
```



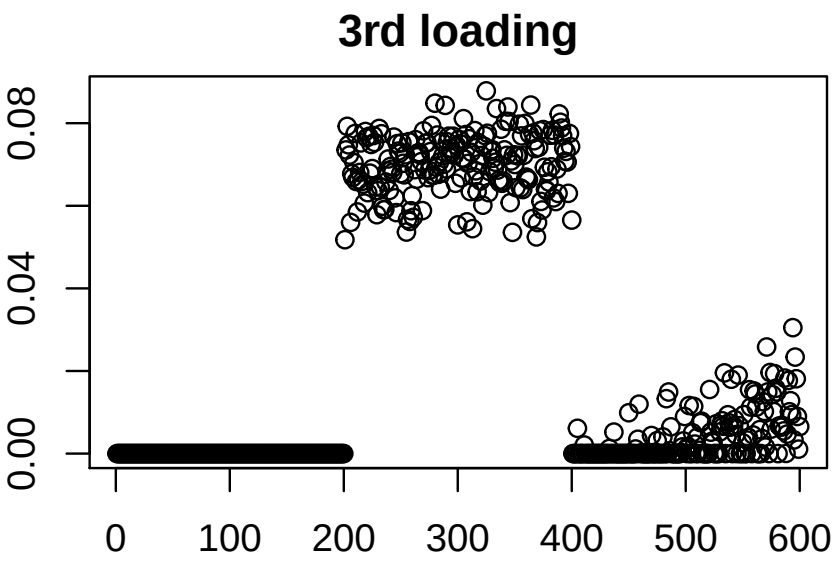

```
plot(out$u[, 2], main = "2nd loading")
```



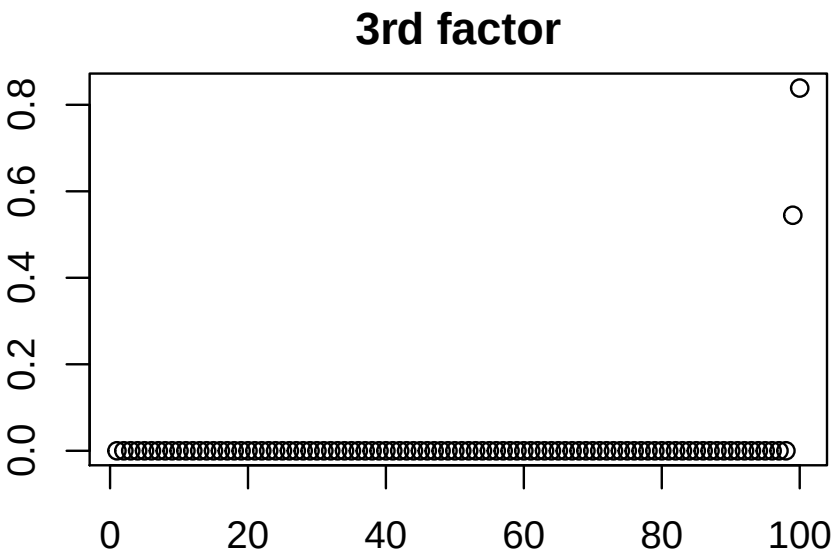

```
plot(out$v[, 2], main = "2nd factor")
```




```
plot(out$u[, 3], main = "3rd loading")
```




```
plot(out$v[, 3], main = "3rd factor")
```

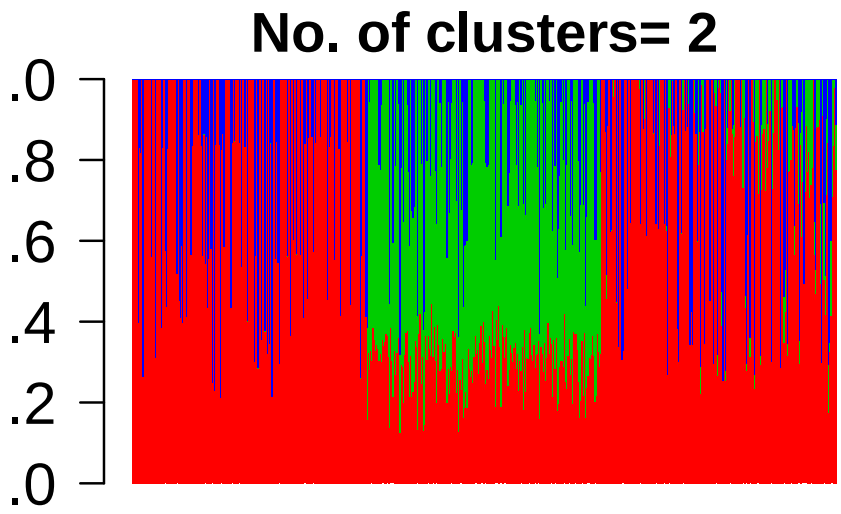



```
omega1 <- maptpx::normalize(cbind(out$u[, 2],
    out$u[, 3]), byrow = TRUE)

barplot(t(omega1), col = 2:(K + 1), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```


```
omega2 <- maptpx::normalize(cbind(out$u[, 3],
    out$u[, 4]), byrow = TRUE)

barplot(t(omega2), col = 2:(K + 1), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```


```
omega3 <- maptpx::normalize(cbind(out$u[, 2],
    out$u[, 4]), byrow = TRUE)

barplot(t(omega3), col = 2:(K + 1), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```

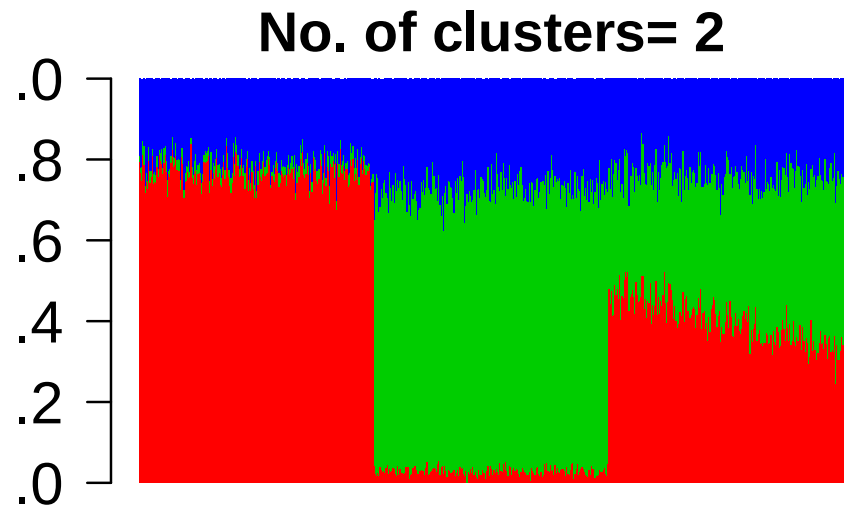

```
omega4 <- maptpx::normalize(cbind(out$u[, 2],
    out$u[, 3], out$u[, 4]), byrow = TRUE)

barplot(t(omega4), col = 2:(K + 2), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```




```
tpx.fit <- maptpx::topics(counts, K = 3)
```

```
##
## Estimating on a 600 document collection.
## Fitting the 3 topic model.
## log posterior increase: 5946.7, 11.8, 1.8, 1.1, 0.8, 1, 9.9, 0.3, 0.3, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.
```

```
barplot(t(tpx.fit$omega), col = 2:(K + 2), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```




```
tpx.fit <- maptpx::topics(counts, K = 2)
```

```
##
## Estimating on a 600 document collection.
## Fitting the 2 topic model.
## log posterior increase: 6152.1, 35.2, 0.3, done.
```

```
barplot(t(tpx.fit$omega), col = 2:(K + 1), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```

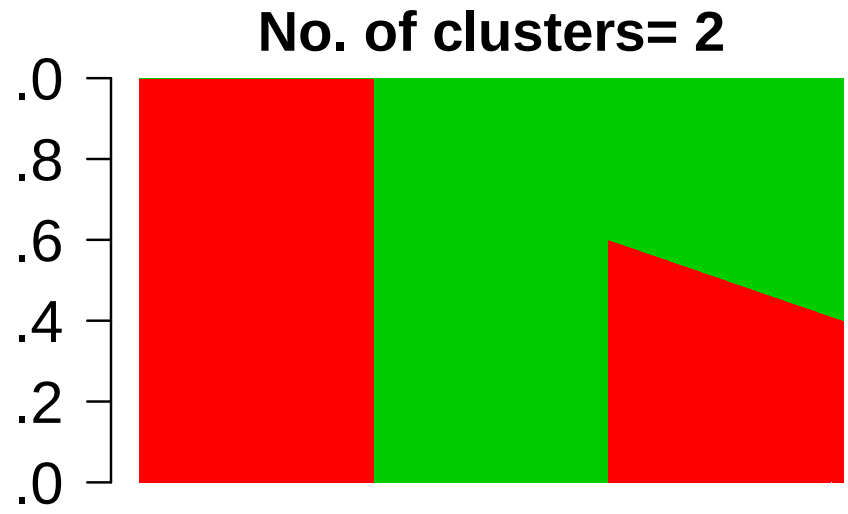
Simulation experiment 3

This experiment is similar to Simulation model 1 but with lot more samples and genes.

```
n.out <- 1000
omega_sim <- rbind(cbind(rep(1, n.out), rep(0,
    n.out)), cbind(rep(0, n.out), rep(1, n.out)),
    cbind(seq(0.6, 0.4, length.out = n.out), 1 -
        seq(0.6, 0.4, length.out = n.out)))
dim(omega_sim)
```

```
## [1] 3000    2
```

```
K <- dim(omega_sim)[2]

par(mar = c(2, 2, 2, 2))
barplot(t(omega_sim), col = 2:(K + 1), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```




```
freq <- rbind(c(0.1, 0.2, rep(0.7/998, 998)),
    c(rep(0.7/998, 998), 0.1, 0.2))
str(freq)
```

```
##  num [1:2, 1:1000] 0.1 0.000701 0.2 0.000701 0.000701 ...
```

```
counts <- t(do.call(cbind, lapply(1:dim(omega_sim)[1],
    function(x) rmultinom(1, 1000, prob = omega_sim[x,
        ] %*% freq))))
dim(counts)
```

```
## [1] 3000 1000
```

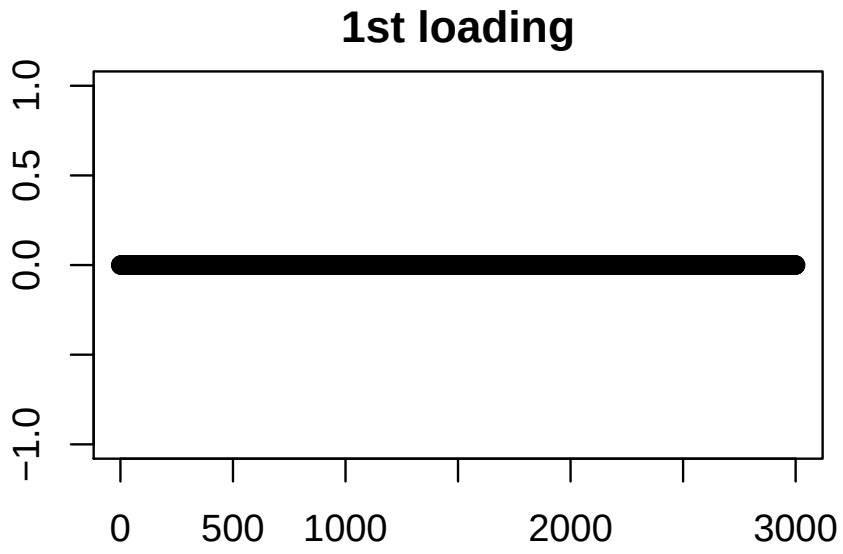
```
lambda <- 1000 * (omega_sim %*% freq)
lambda[lambda == 0] <- 1e-04
dim(lambda)
```

```
## [1] 3000 1000
```

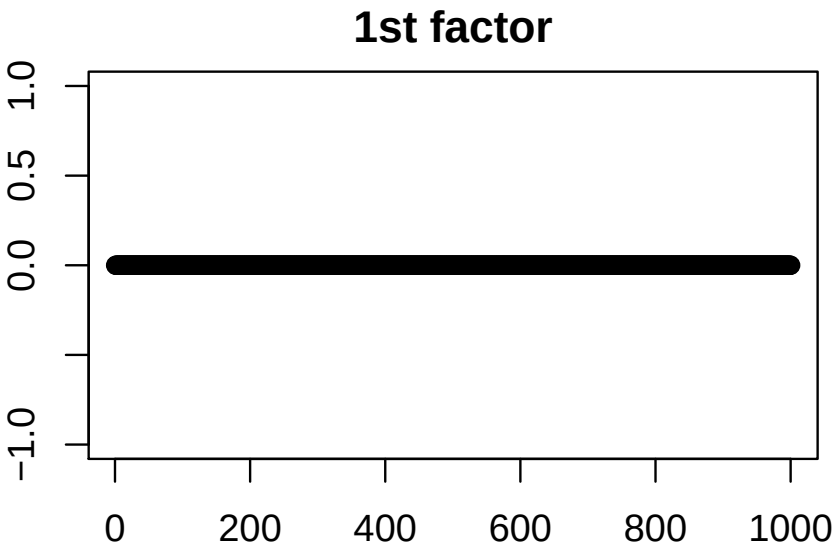
```
scaled_counts <- counts/sqrt(lambda)

require(PMA)
out <- PMD(scaled_counts, K = 4, upos = TRUE,
    vpos = TRUE, center = TRUE, sumabs = 1, niter = 20000,
    sumabsu = sqrt(600))
```

```
## 12
## 123456789101112131415161718192021222324252627 28
## 123456789
## 12345678910
```

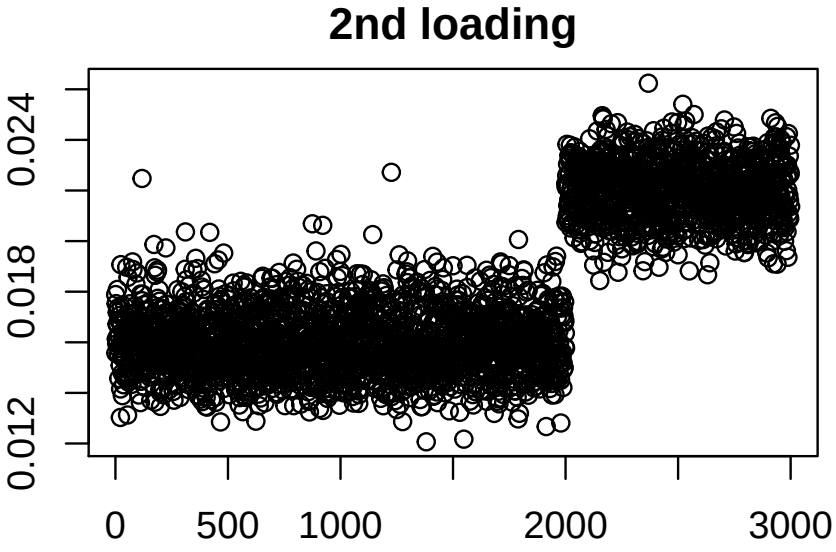
```
plot(out$u[, 1], main = "1st loading")
```



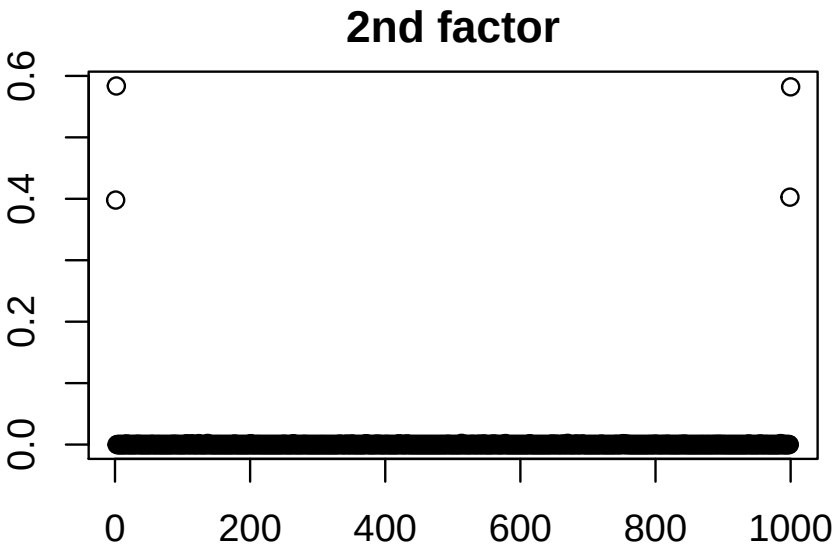

```
plot(out$v[, 1], main = "1st factor")
```



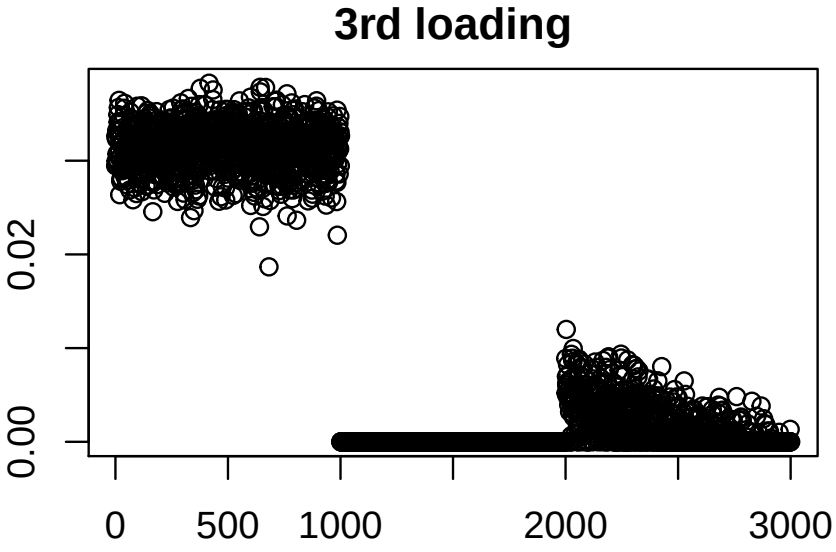

```
plot(out$u[, 2], main = "2nd loading")
```



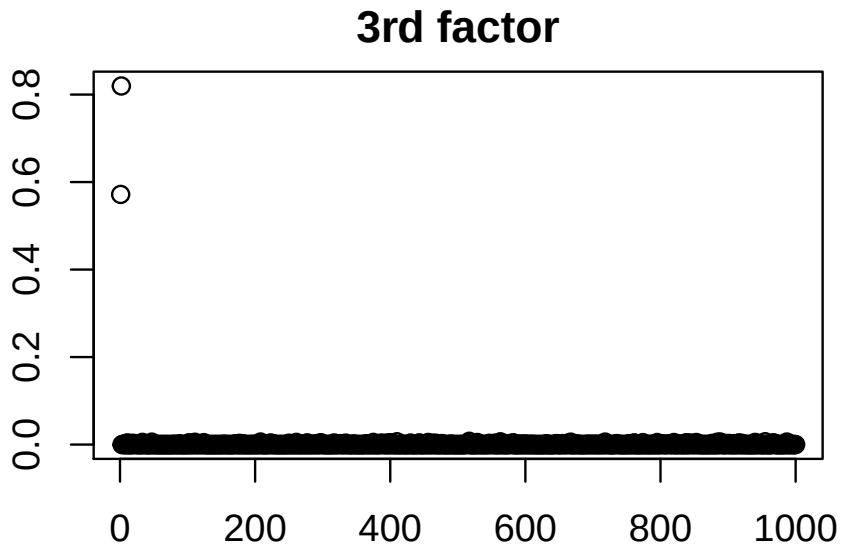

```
plot(out$v[, 2], main = "2nd factor")
```




```
plot(out$u[, 3], main = "3rd loading")
```




```
plot(out$v[, 3], main = "3rd factor")
```



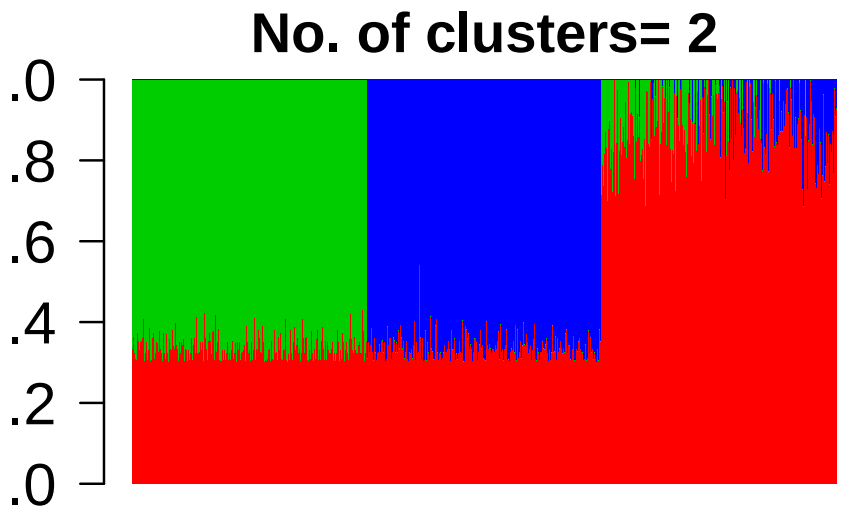

```
omega1 <- maptpx::normalize(cbind(out$u[, 2],
    out$u[, 3]), byrow = TRUE)

barplot(t(omega1), col = 2:(K + 1), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```


```
omega2 <- maptpx::normalize(cbind(out$u[, 3],
    out$u[, 4]), byrow = TRUE)

barplot(t(omega2), col = 2:(K + 1), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```

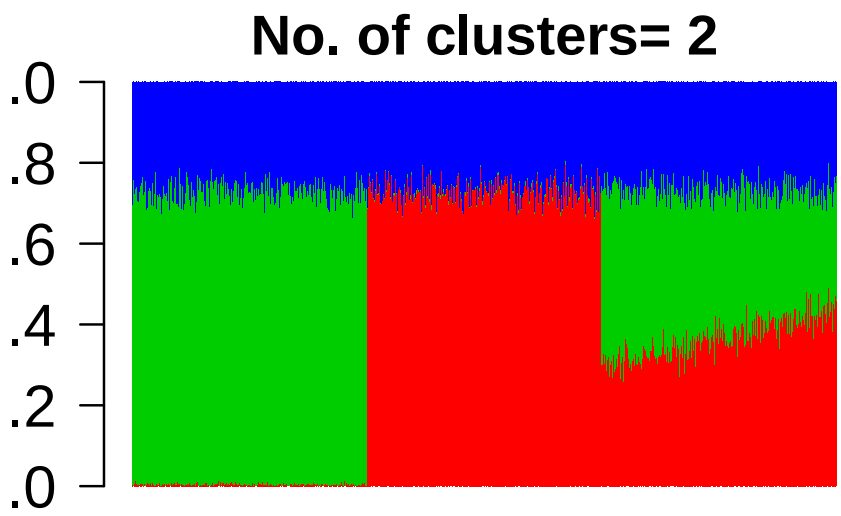

```
omega3 <- maptpx::normalize(cbind(out$u[, 2],
    out$u[, 4]), byrow = TRUE)

barplot(t(omega3), col = 2:(K + 1), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```


```
omega4 <- maptpx::normalize(cbind(out$u[, 2],
    out$u[, 3], out$u[, 4]), byrow = TRUE)

barplot(t(omega4), col = 2:(K + 2), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
```

```
    cex.main = 1.4)
```




```
# tpx.fit <- maptpx::topics(counts, K=3)
# save(tpx.fit,
# file='../rdas/pma_tpx_compare_1.rda')

tpx.fit <- get(load(file = "../rdas/pma_tpx_compare_1.rda"))
barplot(t(tpx.fit$omega), col = 2:(K + 2), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```




```
tpx.fit <- maptpx::topics(counts, K = 2)

##
## Estimating on a 3000 document collection.
```

```
## Fitting the 2 topic model.
## log posterior increase: 1037.8, 26, 18.4, 15.5, 16.6, 29.5, 154.9, 25811.5, 8410.4, 26.7, 1.7, 0.4, 0.1
```

```
barplot(t(tpx.fit$omega), col = 2:(K + 1), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```

Simulation Experiment 4

```
n.out <- 200
omega_sim <- rbind(cbind(rep(1, n.out), rep(0,
    n.out), rep(0, n.out)), cbind(rep(0, n.out),
    rep(1, n.out), rep(0, n.out)), cbind(rep(0,
    n.out), rep(0, n.out), rep(1, n.out)), cbind(seq(0.3,
    0.4, length.out = n.out), seq(0.4, 0.2, length.out = n.out),
    1 - seq(0.3, 0.4, length.out = n.out) - seq(0.4,
        0.2, length.out = n.out)))

K <- dim(omega_sim)[2]

par(mar = c(2, 2, 2, 2))
barplot(t(omega_sim), col = 2:(K + 1), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```


```
freq <- rbind(c(0.1, 0.2, rep(0.7/98, 98)), c(rep(0.7/98,
    98), 0.1, 0.2), c(rep(0.4/49, 49), 0.1, 0.2,
    rep(0.3/49, 49)))

str(freq)
```

```
##  num [1:3, 1:100] 0.1 0.00714 0.00816 0.2 0.00714 ...
```

```
counts <- t(do.call(cbind, lapply(1:dim(omega_sim)[1],
    function(x) rmultinom(1, 1000, prob = omega_sim[x,
        ] %*% freq))))
dim(counts)
```

```
## [1] 800 100
```

```
lambda <- 1000 * (omega_sim %*% freq)
lambda[lambda == 0] <- 1e-04
dim(lambda)
```

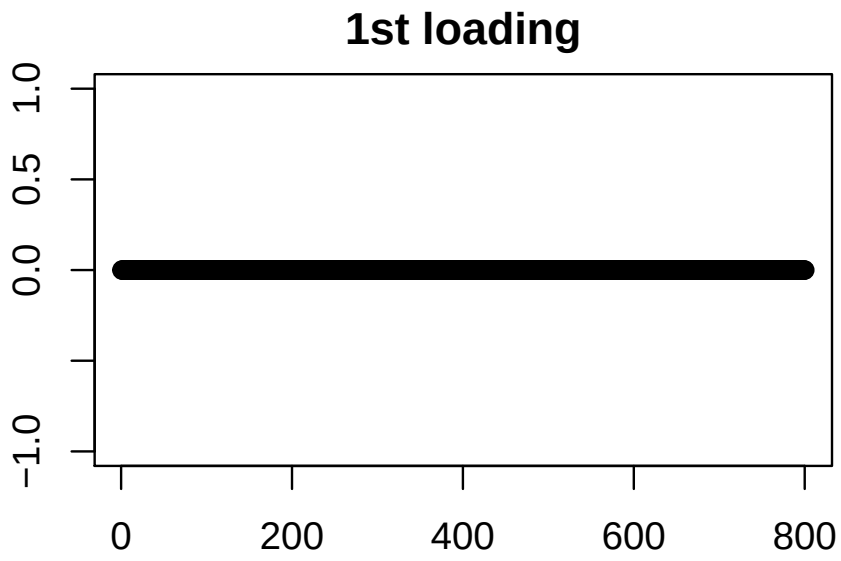
```
## [1] 800 100
```

```
scaled_counts <- counts/sqrt(lambda)

require(PMA)
out <- PMD(scaled_counts, K = 5, upos = TRUE,
    vpos = TRUE, center = TRUE, sumabs = 1, niter = 20000,
    sumabsu = sqrt(600))
```

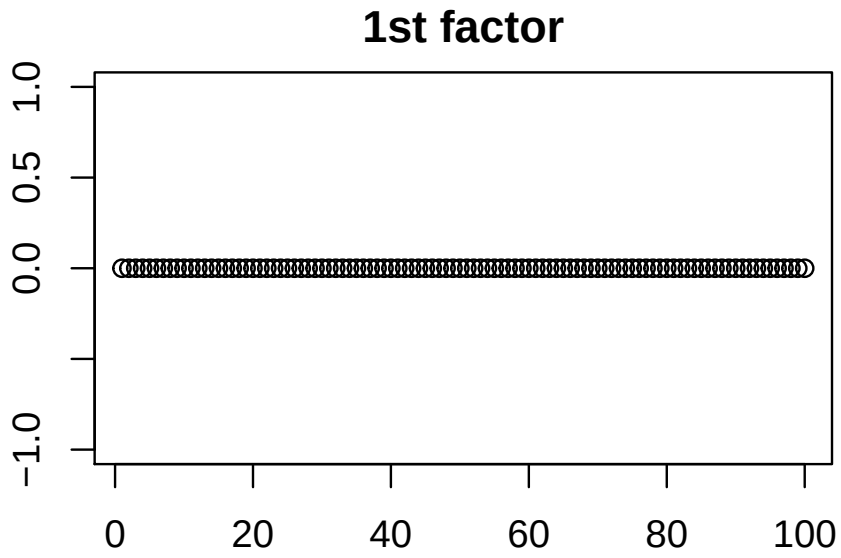
```
## 12
## 12345678910111213141516171819202122232425262728293031323334353637383940414 2
```

```
## 12345
## 123456
## 12345678
```

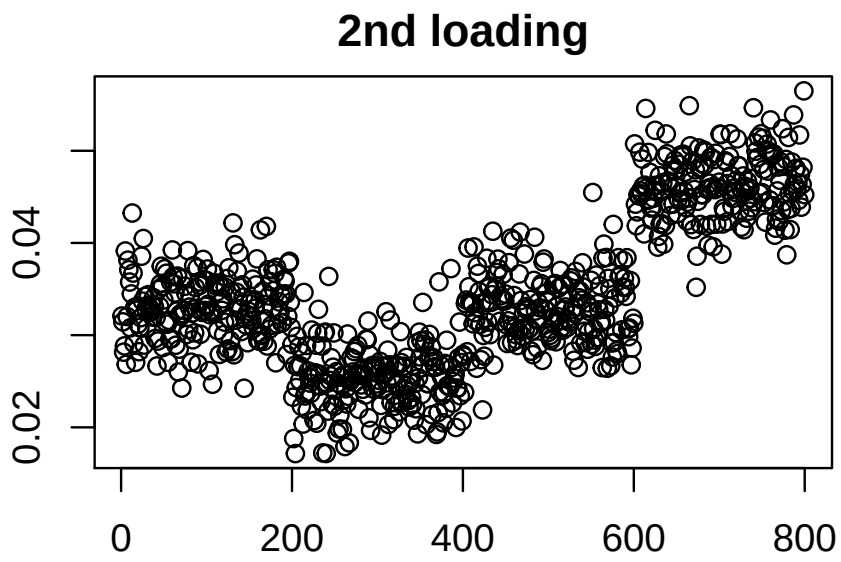
```
plot(out$u[, 1], main = "1st loading")
```



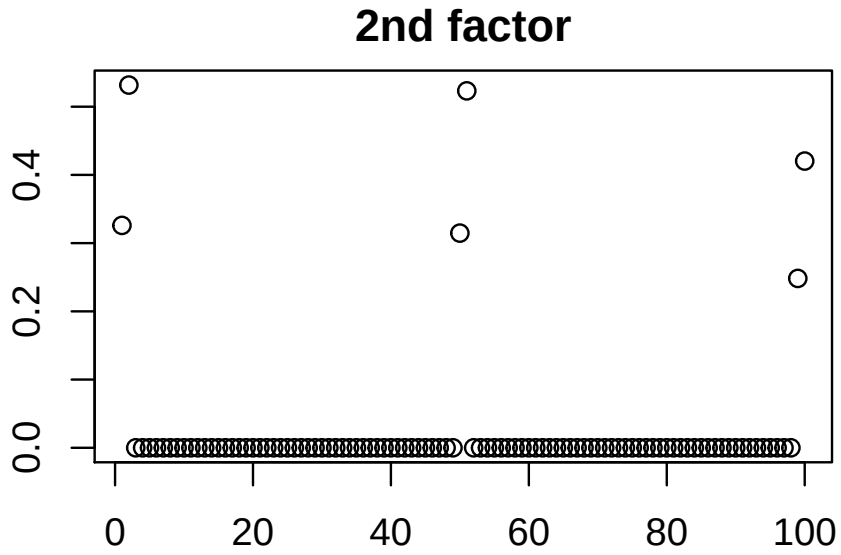

```
plot(out$v[, 1], main = "1st factor")
```



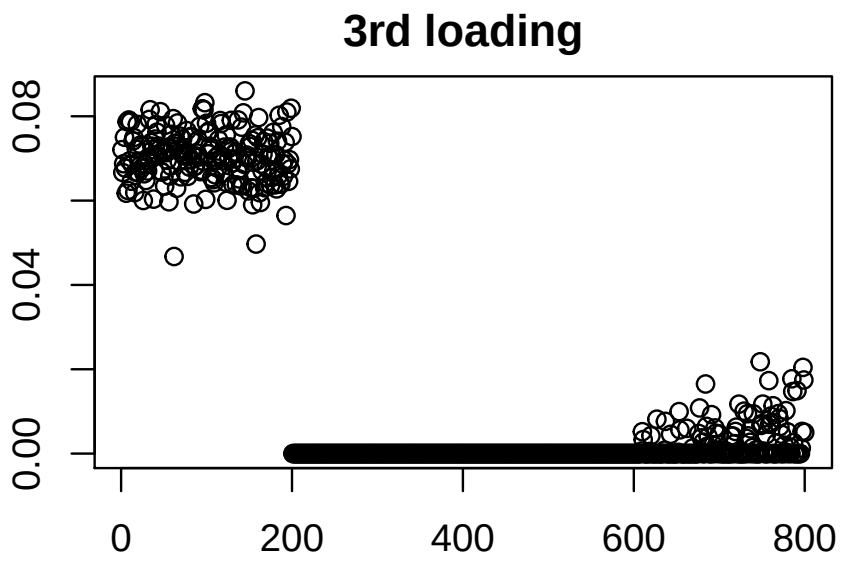

```
plot(out$u[, 2], main = "2nd loading")
```



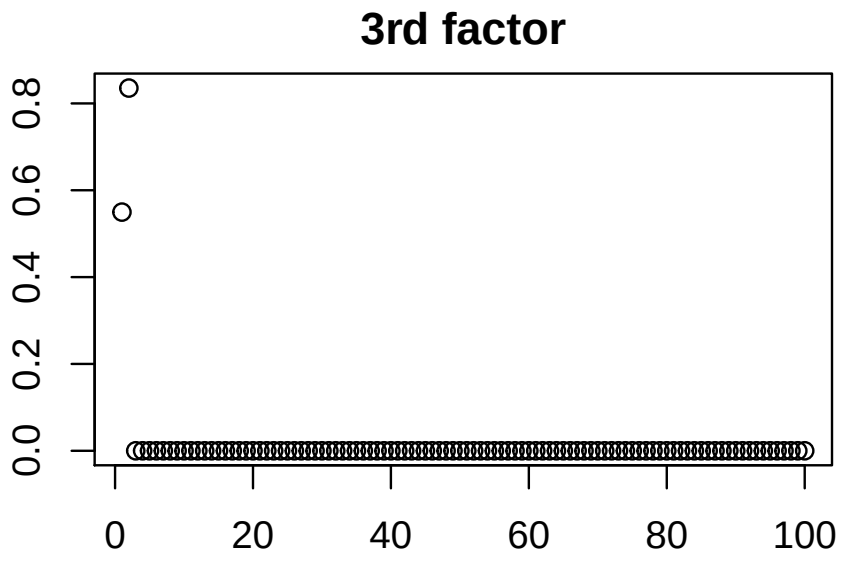

```
plot(out$v[, 2], main = "2nd factor")
```



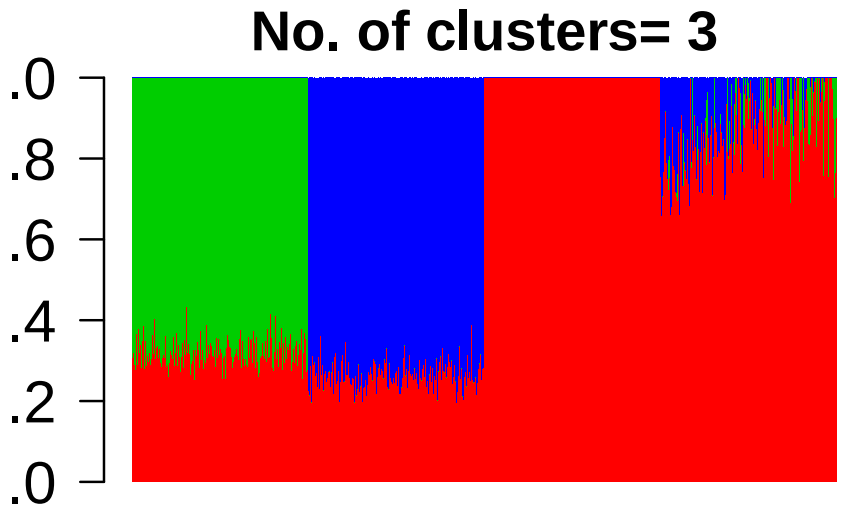

```
plot(out$u[, 3], main = "3rd loading")
```




```
plot(out$v[, 3], main = "3rd factor")
```

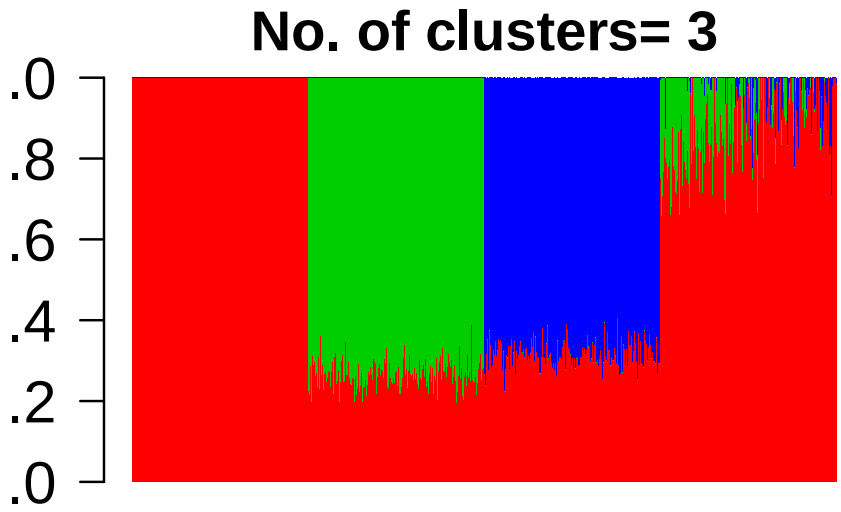



```
K <- 3
omega1 <- maptpx::normalize(cbind(out$u[, 2],
    out$u[, 3], out$u[, 4]), byrow = TRUE)

barplot(t(omega1), col = 2:(K + 1), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```




```
omega2 <- maptpx::normalize(cbind(out$u[, 3],
    out$u[, 4], out$u[, 5]), byrow = TRUE)

barplot(t(omega2), col = 2:(K + 1), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```


```
omega3 <- maptpx::normalize(cbind(out$u[, 2],
    out$u[, 4], out$u[, 5]), byrow = TRUE)

barplot(t(omega3), col = 2:(K + 1), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```




```
tpx.fit <- maptpx::topics(counts, K = 3)
```

```
##
## Estimating on a 800 document collection.
## Fitting the 3 topic model.
## log posterior increase: 58203.2, 11.2, 1.1, 0.4, done.
```

```
barplot(t(tpx.fit$omega), col = 2:(K + 2), axisnames = F,
    space = 0, border = NA, main = paste("No. of clusters=",
        K), las = 1, ylim = c(0, 1), cex.axis = 1.5,
    cex.main = 1.4)
```