Importance of Variance weighting in flashtpx

Wei Wang, Kushal K Dey

April 1, 2016

Contents

Overview

In this script, we look at the significance of variance weighting in `flashtpx` models. Note that in the general normal topic model (NTM), we assume that each gene has the same variance across samples. If I consider this assumption under the refined normal model on counts, that would look like

$$FLASH1: \qquad c_{ng} \sim N(\lambda_{ng}, \phi_g)$$

We may even make the assumption stricter and assume just a global variance ϕ instead of depending on the gene in ϕ_g.

$$FLASH2: \qquad c_{ng} \sim N(\lambda_{ng}, \phi)$$

We want to compare these two models of *flash* with the one originally proposed in the simulation runs.

$$c_{ng} \sim N(\lambda_{ng}, \lambda_{ng})$$

where the variance term λ_{ng} is replaced by the estimate obtained from topic model $\lambda^{\star}_{ng}$ and then the *flash* model is performed for the mean λ_{ng} with variance $\lambda^{\star}_{ng}$.

Since the third model is more informative and closely resembles the Poisson model that is known to fit counts data well, we feel that it would do better than the first and second models. But is the performance comparable from the loadings and the factors we get? That is the question we try to answer using a simulation example and then running both these methods on that simulation data.

We use the same model as in Simulation Expt 1.

Simulation Design

We load the packages and the functions we need to perform the model.

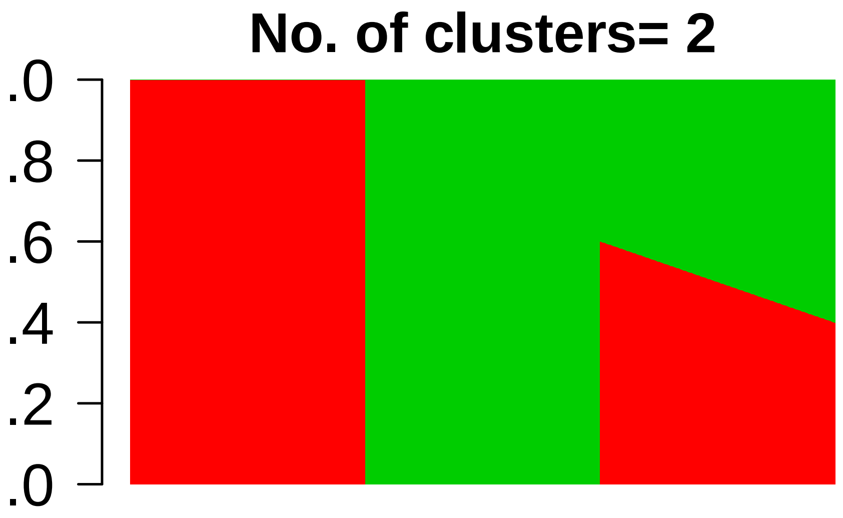
```
library(ashr)
library(irlba)
library(PMA)
source("../R/flash.R")
```

Next we determine the omega matrix determined in batches of 200, the first 200 batches of samples coming from one cluster completely, the next 200 coming from a second cluster and the final 200 being proportionally assigned to the two clusters. (Check barplot for more clarity).

```
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)
```



So we have two clusters. How do these clusters look? Assume we have 100 genes.

```
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 ...
```

So the first cluster has high expression at first 2 genes and low expression at the other 98 genes.

```
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
```

We first perform a topic model with $K = 2$.

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

```
##
## Estimating on a 600 document collection.
## Fitting the 2 topic model.
## log posterior increase: 7492.8, 34.5, 0.3, done.
```

```
omega <- topic.fit$omega
theta <- topic.fit$theta
```

We obtain an estimate of the mean λ_{ng} from the cluster membership probability (omega) matrix and the cluster distribution (theta) matrix. We then take the residuals of the counts from the mean matrix λ.

```
lambda <- 1000 * (omega %*% t(theta))
res_counts <- counts - lambda
```

FLASH model 1 fitting

flash then fits the model

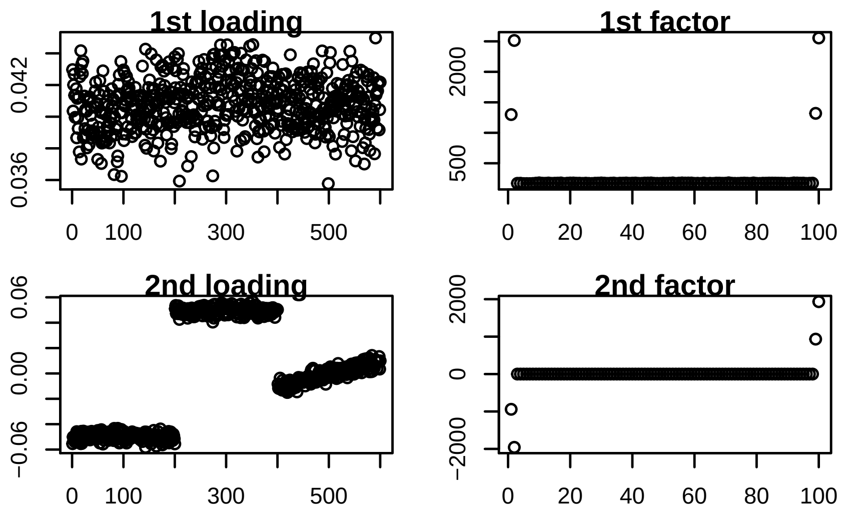
$$c_{ng} = \sum_{k=1}^{K} l_{nk} f_{kg} + e_{ng} \qquad e_{ng} \sim N(0, \phi_g)$$

We first determine the ϕ_g using the residual counts obtained after fitting topic model above and then computing the variance for each gene.

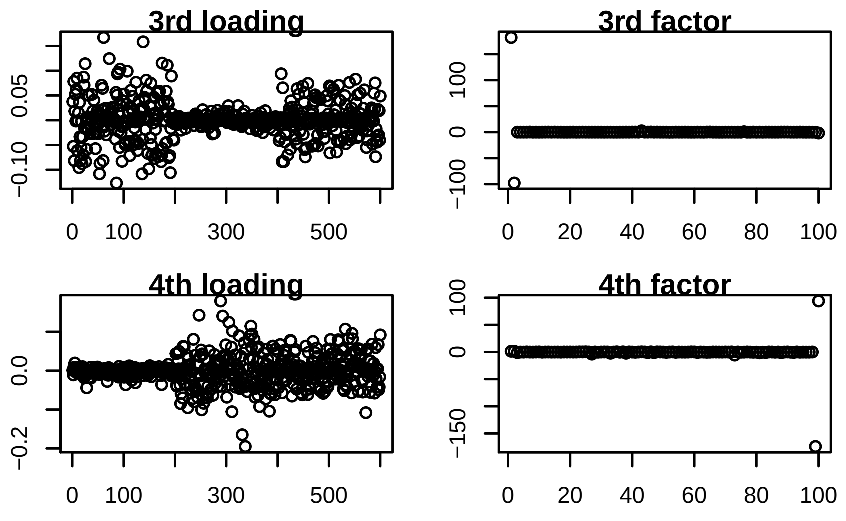
```r
var_genes <- apply(res_counts, 2, var)

rep.row <- function(x, n) {
    matrix(rep(x, each = n), nrow = n)
}

phi <- rep.row(var_genes, dim(res_counts)[1])
phi[phi == 0] <- 1e-04
dim(phi)
```

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

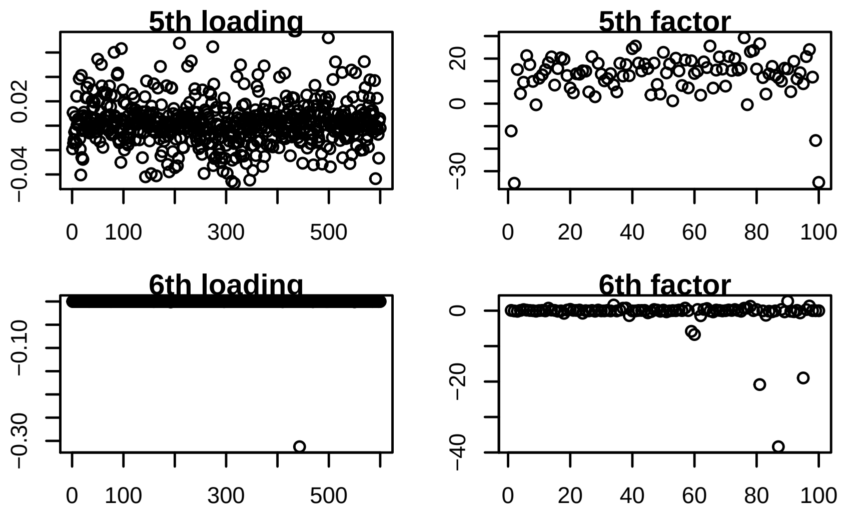
```r
g1 = suppressMessages(flash(counts, sigmae2_true = phi))
f = g1$f
l = g1$l
res = counts - l %*% t(f)
# g_new = flash(res,nonnegative =
# TRUE,sigmae2_true = lambda)
g2 = suppressMessages(flash(res, sigmae2_true = phi))
l = g2$l
f = g2$f
res = res - l %*% t(f)
g3 = suppressMessages(flash(res, sigmae2_true = phi))
l = g3$l
f = g3$f
res = res - l %*% t(f)
g4 = suppressMessages(flash(res, sigmae2_true = phi))
l = g4$l
f = g4$f
res = res - l %*% t(f)
g5 = suppressMessages(flash(res, sigmae2_true = phi))
l = g5$l
f = g5$f
res = res - l %*% t(f)
g6 = suppressMessages(flash(res, sigmae2_true = phi))
l = g6$l
f = g6$f
par(mfrow = c(2, 2))
par(cex = 0.6)
par(mar = c(3, 3, 0.8, 0.8), oma = c(1, 1, 1,
    1))
plot(g1$l, main = "1st loading")
plot(g1$f, main = "1st factor")
plot(g2$l, main = "2nd loading")
plot(g2$f, main = "2nd factor")
```



```r
plot(g3$l, main = "3rd loading")
plot(g3$f, main = "3rd factor")
plot(g4$l, main = "4th loading")
plot(g4$f, main = "4th factor")
```



```r
plot(g5$l, main = "5th loading")
plot(g5$f, main = "5th factor")
plot(g6$l, main = "6th loading")
plot(g6$f, main = "6th factor")
```



FLASH model 2 fitting

flash then fits the model

$$c_{ng} = \sum_{k=1}^{K} l_{nk} f_{kg} + e_{ng} \qquad e_{ng} \sim N(0, \phi)$$

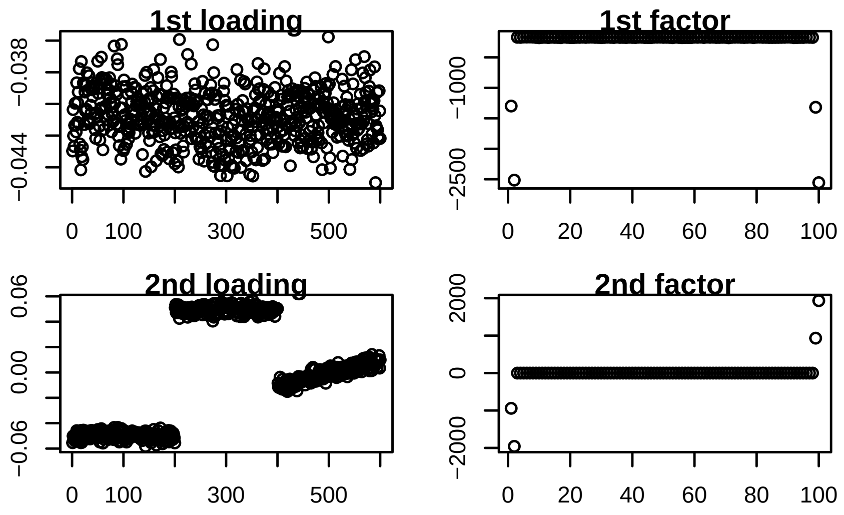
We first determine the ϕ using the residual counts obtained after fitting topic model above and then computing the variance of all the residuals.

```
var_total <- var(as.vector(res_counts))

phi <- matrix(var_total, nrow = dim(res_counts)[1],
    ncol = dim(res_counts)[2])
phi[phi == 0] <- 1e-04
dim(phi)
```

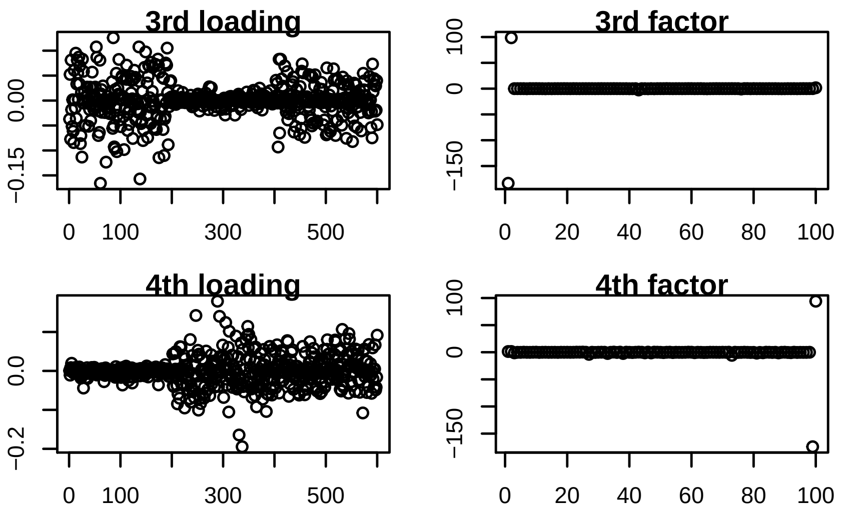
```
## [1] 600 100
```

```
g1 = suppressMessages(flash(counts, sigmae2_true = phi))
f = g1$f
l = g1$l
res = counts - l %*% t(f)
# g_new = flash(res,nonnegative =
# TRUE,sigmae2_true = lambda)
g2 = suppressMessages(flash(res, sigmae2_true = phi))
l = g2$l
f = g2$f
res = res - l %*% t(f)
g3 = suppressMessages(flash(res, sigmae2_true = phi))
```

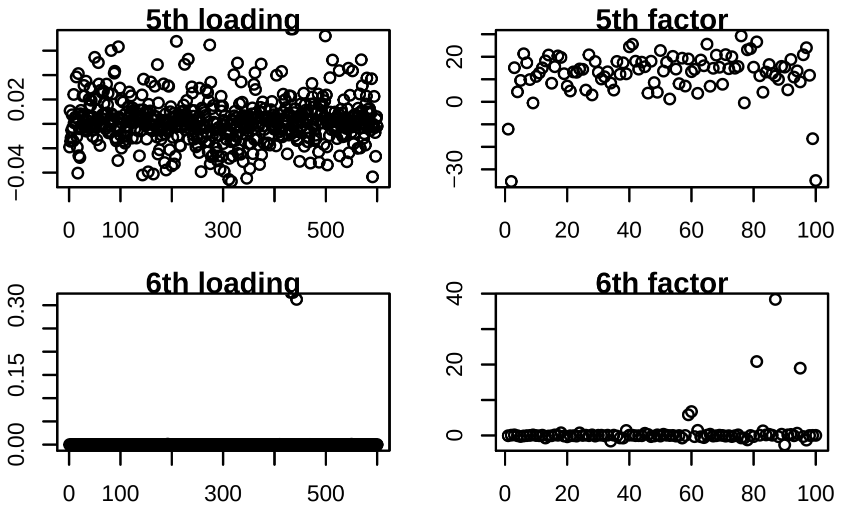
```
l = g3$l
f = g3$f
res = res - l %*% t(f)
g4 = suppressMessages(flash(res, sigmae2_true = phi))
l = g4$l
f = g4$f
res = res - l %*% t(f)
g5 = suppressMessages(flash(res, sigmae2_true = phi))
l = g5$l
f = g5$f
res = res - l %*% t(f)
g6 = suppressMessages(flash(res, sigmae2_true = phi))
l = g6$l
f = g6$f
par(mfrow = c(2, 2))
par(cex = 0.6)
par(mar = c(3, 3, 0.8, 0.8), oma = c(1, 1, 1,
    1))
plot(g1$l, main = "1st loading")
plot(g1$f, main = "1st factor")
plot(g2$l, main = "2nd loading")
plot(g2$f, main = "2nd factor")
```



```
plot(g3$l, main = "3rd loading")
plot(g3$f, main = "3rd factor")
plot(g4$l, main = "4th loading")
plot(g4$f, main = "4th factor")
```



```
plot(g5$l, main = "5th loading")
plot(g5$f, main = "5th factor")
plot(g6$l, main = "6th loading")
plot(g6$f, main = "6th factor")
```



FLASH model 3 fitting

flash then fits the model

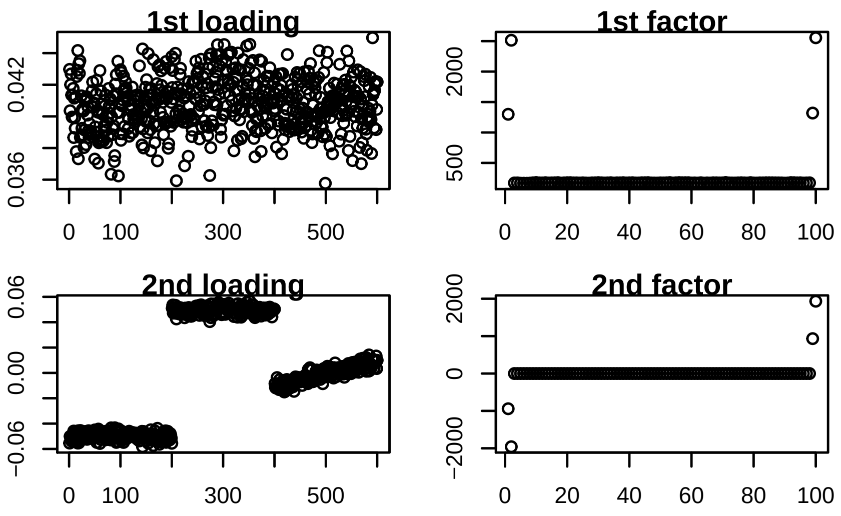
$$c_{ng} = \sum_{k=1}^{K} l_{nk} f_{kg} + e_{ng} \qquad e_{ng} \sim N(0, \lambda^{\star}_{ng})$$

where we ontain $\lambda^{\star}_{ng}$ as follows

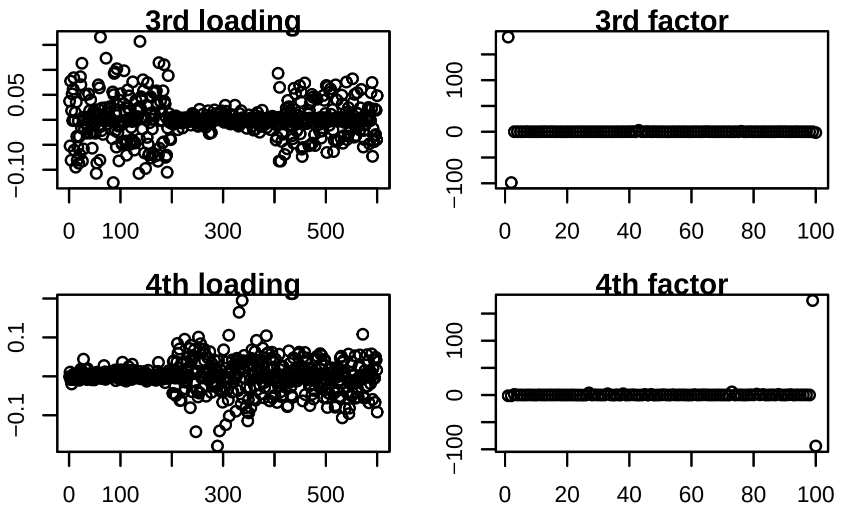
```
lambda_star <- 1000 * (omega %*% t(theta))
lambda_star[lambda_star == 0] <- 1e-04
dim(lambda_star)
```

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

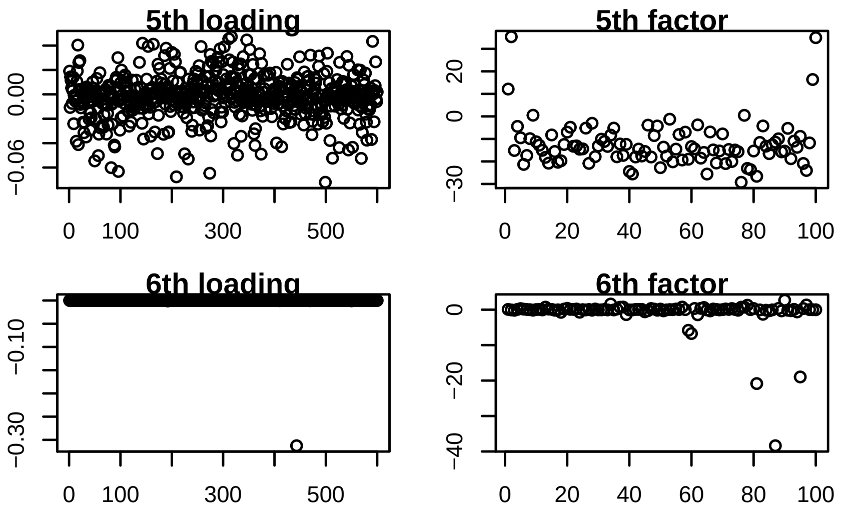
```
g1 = suppressMessages(flash(counts, sigmae2_true = lambda_star))
f = g1$f
l = g1$l
res = counts - l %*% t(f)
# g_new = flash(res,nonnegative =
# TRUE,sigmae2_true = lambda)
g2 = suppressMessages(flash(res, sigmae2_true = lambda_star))
l = g2$l
f = g2$f
res = res - l %*% t(f)
g3 = suppressMessages(flash(res, sigmae2_true = lambda_star))
l = g3$l
f = g3$f
res = res - l %*% t(f)
g4 = suppressMessages(flash(res, sigmae2_true = lambda_star))
l = g4$l
f = g4$f
res = res - l %*% t(f)
g5 = suppressMessages(flash(res, sigmae2_true = lambda_star))
l = g5$l
f = g5$f
res = res - l %*% t(f)
g6 = suppressMessages(flash(res, sigmae2_true = lambda_star))
l = g6$l
f = g6$f
par(mfrow = c(2, 2))
par(cex = 0.6)
par(mar = c(3, 3, 0.8, 0.8), oma = c(1, 1, 1,
    1))
plot(g1$l, main = "1st loading")
plot(g1$f, main = "1st factor")
plot(g2$l, main = "2nd loading")
plot(g2$f, main = "2nd factor")
```



```r
plot(g3$l, main = "3rd loading")
plot(g3$f, main = "3rd factor")
plot(g4$l, main = "4th loading")
plot(g4$f, main = "4th factor")
```



```r
plot(g5$l, main = "5th loading")
plot(g5$f, main = "5th factor")
plot(g6$l, main = "6th loading")
plot(g6$f, main = "6th factor")
```



```r
sessionInfo()
```

```
## R version 3.2.4 (2016-03-10)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: OS X 10.10.5 (Yosemite)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils
## [5] datasets  methods   base
##
## other attached packages:
## [1] Rcpp_0.12.4   PMA_1.0.9     impute_1.44.0
## [4] plyr_1.8.3    irlba_2.0.0   Matrix_1.2-4
## [7] ashr_1.0.8
##
## loaded via a namespace (and not attached):
##  [1] knitr_1.12.3      magrittr_1.5
##  [3] MASS_7.3-43       doParallel_1.0.10
##  [5] pscl_1.4.9        SQUAREM_2014.8-1
##  [7] lattice_0.20-33   foreach_1.4.3
##  [9] stringr_1.0.0     tools_3.2.4
## [11] parallel_3.2.4    grid_3.2.4
## [13] maptpx_1.9-2      htmltools_0.3
## [15] iterators_1.0.8   assertthat_0.1
## [17] yaml_2.1.13       digest_0.6.9
## [19] formatR_1.2.1     codetools_0.2-14
## [21] slam_0.1-32       evaluate_0.8
## [23] rmarkdown_0.9.2   stringi_1.0-1
## [25] truncnorm_1.0-7   tufte_0.2
```