jackstraw: Statistical Inference using Latent Variables

Neo Christopher Chung

January 19, 2018

1 Introduction

This is a vignette for the `jackstraw` package, which performs association tests between variables and estimated latent variables [1]. Latent variables are unobserved and must be estimated directly from the data, using various techniques such as principal component analysis, logistic factor analysis, and others. However, treating estimated latent variables as independent variables in a conventional regression framework would results in an anti-conservative bias (i.e., artificially inflated significance). The jackstraw account for this fact that latent variables are estimated from the data and to protect association tests from an anti-conservative bias.

2 Latent Variable Models

Latent variables (LVs) can be seen as underlying sources of variation, that are hidden and/or undefined. In some cases, underlying sources of variation may be directly measured, such as disease status or sex, and conventional association tests (e.g., F-test) can be utilized to assess statistical significance of association between variables and such measurements. However, underlying sources of variation may be impossible to measure or poorly defined, such as cancer subtypes and population structure. For example, cancer subtypes based on histological tumor classification are highly imprecise.

When we have a high-dimensional data where a large number of variables are measured for each observation, LVs may be estimated from the data itself. For example, principal component analysis (PCA) is a popular technique to estimate LVs from a family of continuous data. After applying PCA to the observed data, one may like to identify variables associated with a set of principal components (PCs). We must be very careful when "re-using" PCs (or other estimated latent variables) to examine the observed data. In other words, an unadjusted association test between variables and PCs results in anti-conservative p-values. The jackstraw method adjust this over-fitting problem and produces valid association p-values.

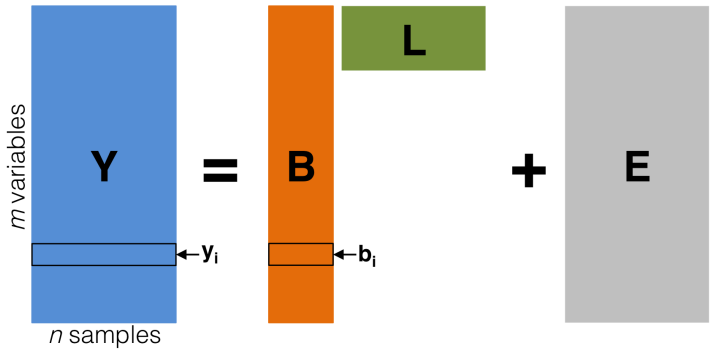


Figure 1: *Diagram of the latent variable model. The latent variable basis* $\boldsymbol{L}$ *is not observable, but may be estimated from* $\boldsymbol{Y}$*. The noise term* $\boldsymbol{E}$ *is independent random variation. We are interested in performing statistical hypothesis tests on* $\boldsymbol{b}_i$ $(i = 1, \ldots, m)$*, which quantifies the relationship between* $\boldsymbol{L}$ *and* $\boldsymbol{y}_i$ $(i = 1, \ldots, m)$*.* This figure is reprinted from [1]

3 Example of Continuous Data

3.1 Simulation of Gene Expression Data

We simulate a simple gene expression data with $m = 1000$ genes (or variables) and $n = 20$ samples (or observations). The following simulation code generates a dichonomous mean shift between the first set of 10 samples and the second set of 10 samples (e.g., this may represent a case-control study). This mean shift affects 10% of $m = 1000$ genes:

```
library(jackstraw)
library(corpcor)

set.seed(1)
B = c(runif(100, min = 0.1, max = 1), rep(0, 900))
L = c(rep(1, 10), rep(-1, 10))
L = L/sd(L)
E = matrix(rnorm(1000 * 20), nrow = 1000)
Y = B %*% t(L) + E

dim(Y)

## [1] 1000   20

Y[1:5, 1:5]

##            [,1]       [,2]        [,3]       [,4]
```

```
## [1,]  0.7284811  0.3723739  1.95390743 -0.7057011
## [2,] -0.1881271  0.6473216 -1.16625940  2.7916857
## [3,]  0.9411012 -0.4104836 -0.01000972  1.1611355
## [4,] -0.2352048  3.2953804  0.58087712  0.4482182
## [5,]  1.7074094  1.0763474 -0.64774412 -0.6600309
##             [,5]
## [1,]  0.46568781
## [2,] -0.03370393
## [3,]  0.55156033
## [4,]  0.70186429
## [5,] -0.74646792
```

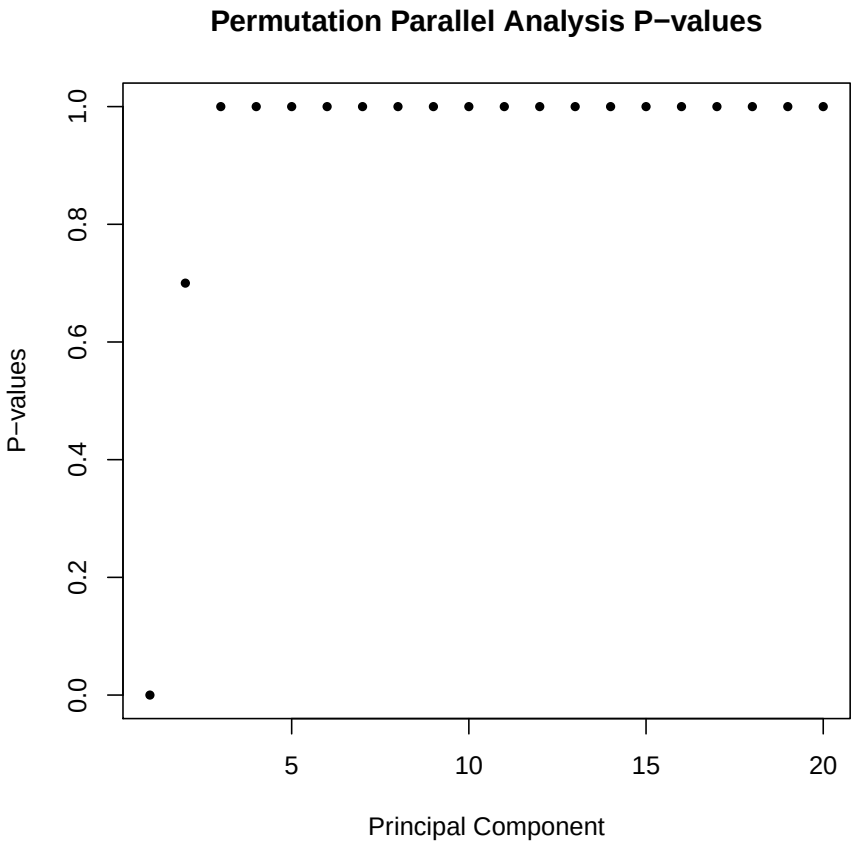
3.2 Application of the Jackstraw using PCA

When we have a dataset, we first need to understand the type(s) of latent variables (e.g., continuous, categorical, ordinal) and to decide on a method to estimate latent variables. Let's use principal component analysis (PCA).

In practice, we have to rely on a scree plot or other graphical and statistical means to estimate r. One particular method is called parallel analysis from Buja and Eyuboglu (1992), which is implemented in the `jackstraw` package. The `permutationPA` function compares the observed percent variance explained (PVE) for each PC to their "null" counterparts computed from a randomly permuted dataset.[1]

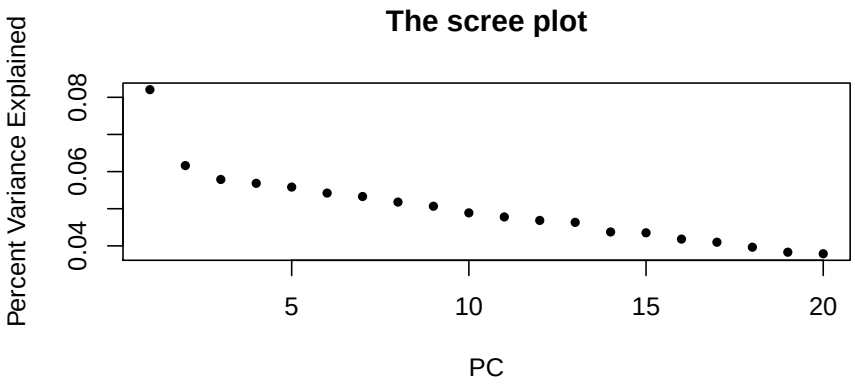
```
PA = permutationPA(Y, B = 10, threshold = 0.05)

## Estimating a number of significant principal component:

## 1  2  3  4  5  6  7  8  9  10

plot(PA$p, pch = 20, main = "Permutation Parallel Analysis P-values",
    ylab = "P-values", xlab = "Principal Component")
```

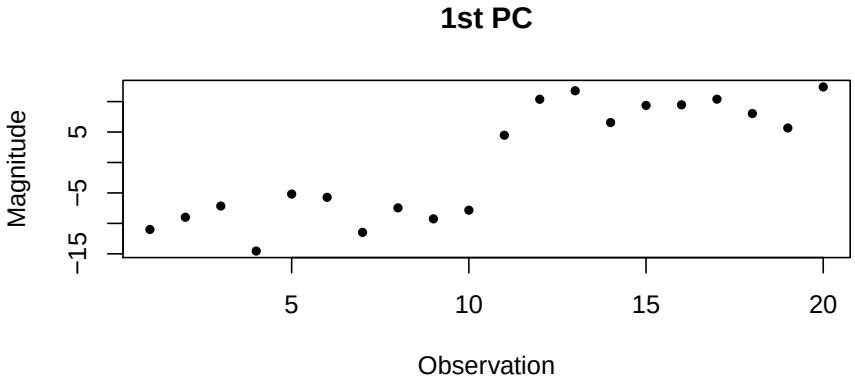
[1] Determining the number of "statistically significant" PCs is an active area of research, and defining a number of significant PCs depends on the data structure and the context. Refer to Anderson (1963), Tracy and Widom (1996), Johnstone (2001), Leek (2010). We do not advocate the blind use of parallel analysis to obtain $\widehat{r}$.



The permutation parallel analysis suggests $r = 1$, which is corroborated by the scree plot (a scatter plot of ordered PVE). Furthermore, we can visualize the first principal component (PC).

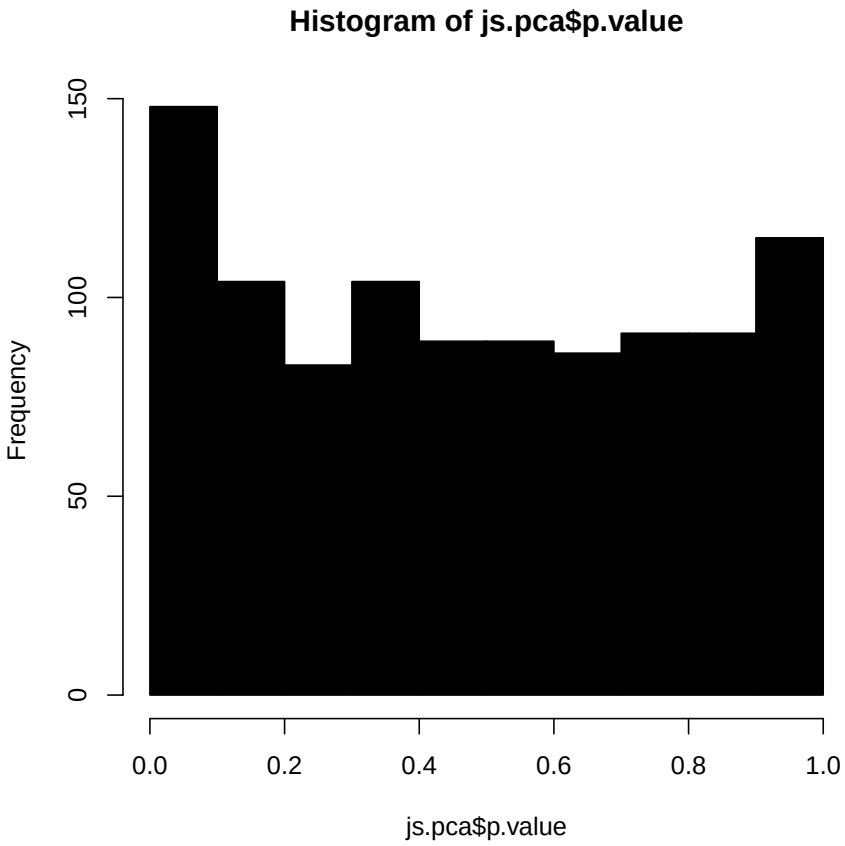
```
svd.out = fast.svd(Y)

par(mfrow = c(2, 1))
plot(svd.out$d^2/sum(svd.out$d^2), pch = 20, main = "The scree plot",
    xlab = "PC", ylab = "Percent Variance Explained")
plot(svd.out$d[1] * svd.out$v[, 1], pch = 20, main = "1st PC",
    xlab = "Observation", ylab = "Magnitude")
```





We are now ready to apply the jackstraw to our data, in order to assess statistical significance of association between variables and the first PC:

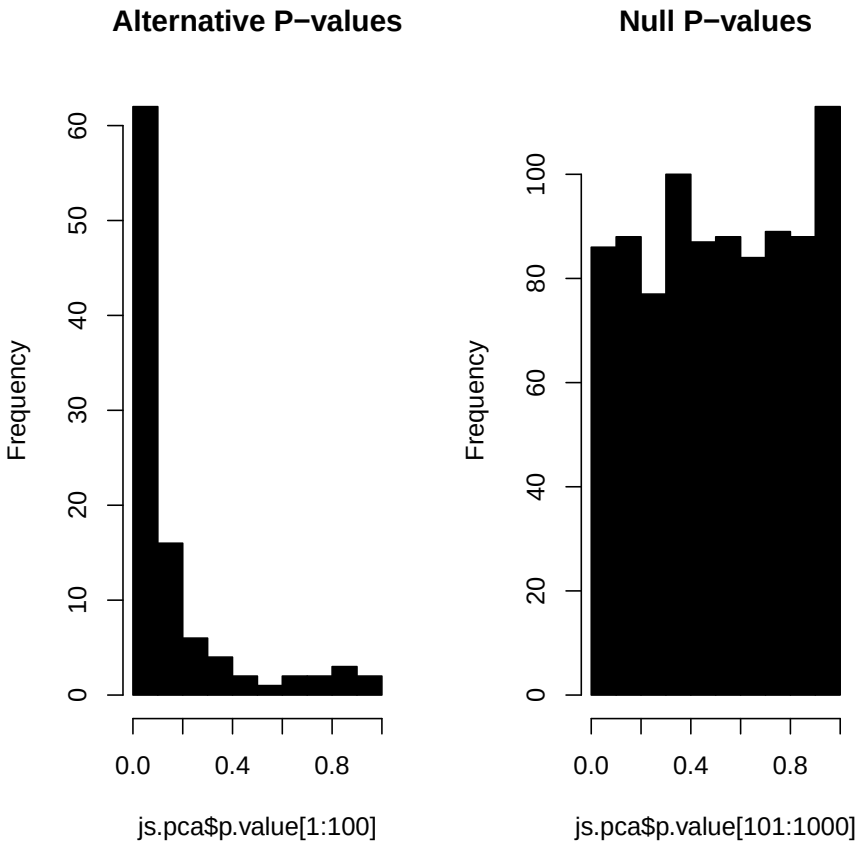
```
js.pca = jackstraw_pca(Y, r = 1, s = 100, B = 100,
    verbose = FALSE)

hist(js.pca$p.value, 10, col = "black")
```



By setting `method="PCA"`, the jackstraw uses a corresponding method to estimate latent variables. In this instance, this wrapper function passes along all the arguments to `jackstraw.PCA` to perform association tests between variables and principal components.

Since the data was simulated, we can visualize the "null" p-values for the truly null variables and the "alternative" p-values for the truly alternative variables. Note that the null p-values is approximating the uniform distribution between 0 and 1:

```
par(mfrow = c(1, 2))
hist(js.pca$p.value[1:100], 10, col = "black", main = "Alternative P-values")
hist(js.pca$p.value[101:1000], 10, col = "black", main = "Null P-values")
```



4 Example of Categorical Data

4.1 Simulation of Genotype Data

We simulate a genotype matrix with $m = 5000$ loci (or variables) and $n = 100$ people (or observations). And a population structure was generated to affect only among 50% of loci. This is parametized by $\pi_0 = .5$. In other words, 50% of loci are independent of population structure:

```
library(jackstraw)
library(lfa)

set.seed(2)
m = 5000
n = 100
```

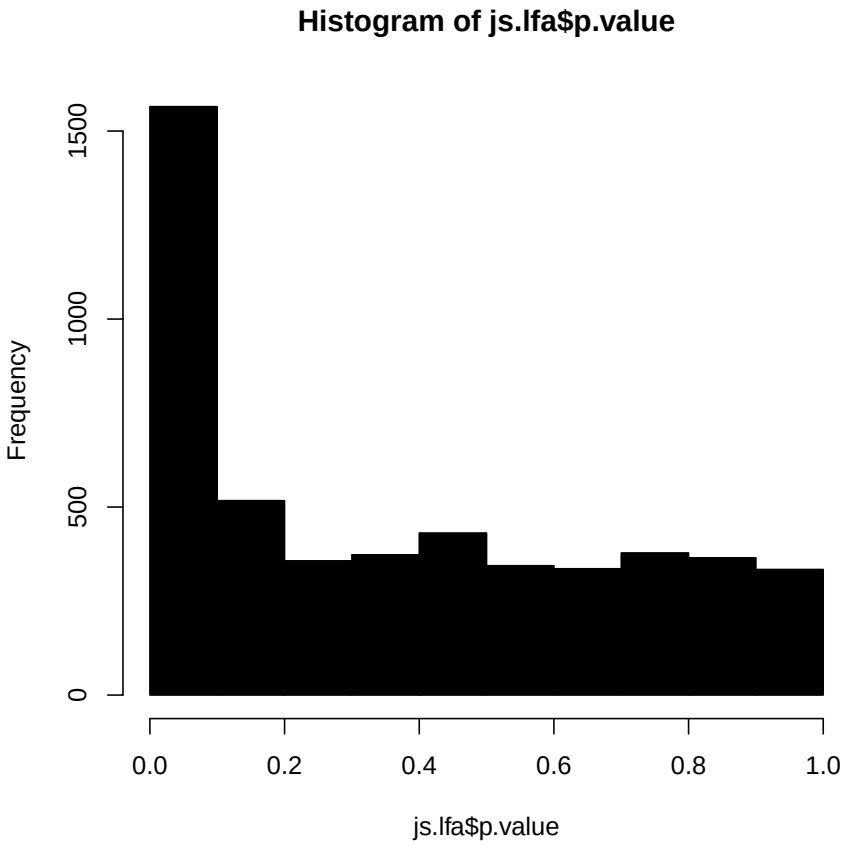
```
pi0 = 0.5
m0 = round(m * pi0)
m1 = m - round(m * pi0)
B = matrix(0, nrow = m, ncol = 1)
B[1:m1, ] = matrix(runif(m1 * n, min = -0.5, max = 0.5),
    nrow = m1, ncol = 1)
L = matrix(rnorm(n), nrow = 1, ncol = n)
BL = B %*% L
prob = exp(BL)/(1 + exp(BL))

dat = list(Y = matrix(rbinom(m * n, 2, as.numeric(prob)),
    m, n), H = c(rep(1, m1), rep(0, m0)))

dim(dat$Y)

## [1] 5000  100

dat$Y[1:5, 1:5]

##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    0    1    0    0
## [2,]    1    0    1    2    2
## [3,]    0    0    2    2    2
## [4,]    2    0    0    0    1
## [5,]    1    0    1    2    2
```

The output includes the genotype matrix `dat$Y`, the status of hypothesis `dat$H`, and the true probability from which the data is simulated `dat$prob`. See that the simulated genotype matrix is encoded in $0, 1, 2$.

4.2 Application of the Jackstraw using LFA

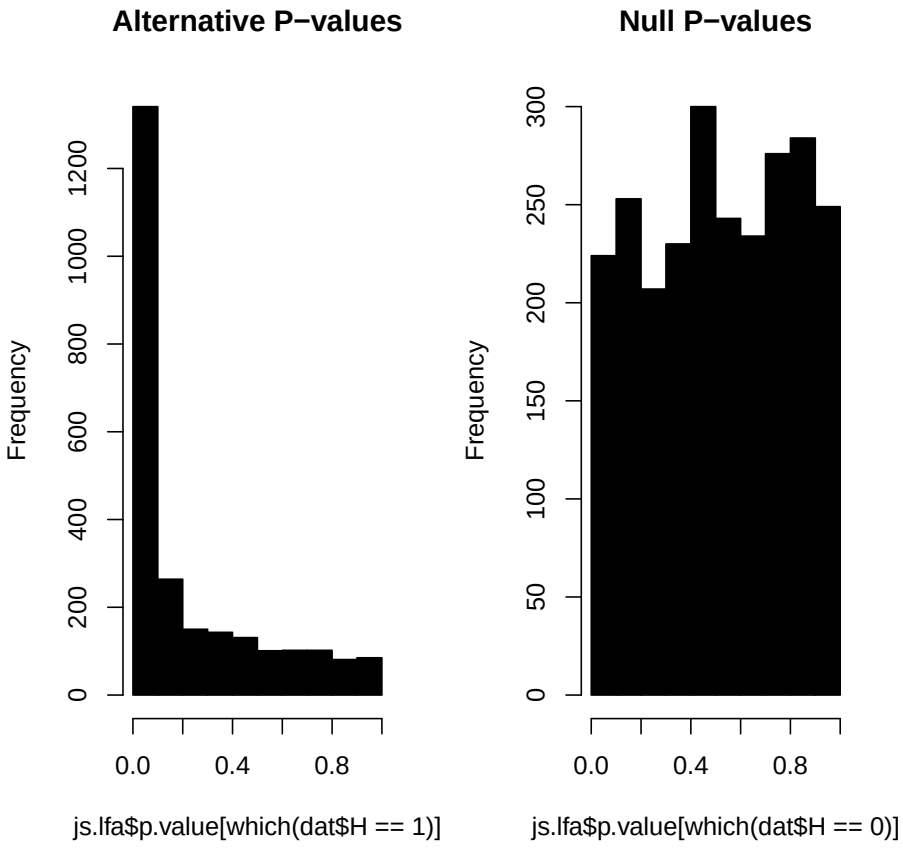
With a genotype matrix, we can use the Logistic Factor Analysis from [2]. We are interested in identifying loci which are truly associated with an underlying population structure (which we must directly estimated from the genotype matrix). LFA estimates this underlying population structure with "logistic factors (LFs)" and we will carry out association tests between loci and LFs.

```
js.lfa = jackstraw_lfa(dat$Y, r = 2, FUN = function(x) lfa.corpcor(x,
    2)[, , drop = FALSE], s = 200, B = 10, devR = TRUE)

##
## Computating null statistics (10 total iterations): 1  2  3  4  5  6  7  8  9  10

hist(js.lfa$p.value, 10, col = "black")
```



Since the data was simulated, we can look at p-values according to their true status. In this case, we see that the null p-values are again approximately distributed uniformly between 0 and 1:

```
par(mfrow = c(1, 2))
hist(js.lfa$p.value[which(dat$H == 1)], 10, col = "black",
    main = "Alternative P-values")
hist(js.lfa$p.value[which(dat$H == 0)], 10, col = "black",
    main = "Null P-values")
```


Alternative P−values
Frequency
0
200
400
600
800
1000
1200
0.0
0.4
0.8
js.lfa$p.value[which(dat$H == 1)]
Null P−values
Frequency
0
50
100
150
200
250
300
0.0
0.4
0.8
js.lfa$p.value[which(dat$H == 0)]

References

[1] N. C. Chung and J. D. Storey. Statistical significance of variables driving systematic variation in high-dimensional data. *Bioinformatics*, 31(4):545–554, 2015.

[2] W. Hao, M. Song, and J. D. Storey. Probabilistic models of genetic variation in structured populations applied to global human studies. *Bioinformatics*, Advance Access:10.1093/bioinformatics/btv641, 2015.