The dks package

Jeffrey Leek
Department of Biostatistics
Johns Hopkins Bloomberg School of Public Health
email: jleek@jhsph.edu

October 24, 2025

Contents

1 Overview

The dks package contains functions for calculating Bayesian and Frequentist diagnostics for multiple testing p-values [1].
There are a large number of multiple testing procedures that have been proposed or are under development. However, there has not been a standard criteria for determining whether a multiple testing procedure produces "correct" p-values. [1] proposed a new joint criterion for the null p-values from multiple tests.

Definition Joint Null Criterion *A set of null p-values is correctly specified if the distribution of each ordered p-value is equal to the distribution of the corresponding order statistic from an independent sample of the same size from the U(0,1) distribution. If the null p-values are denoted* $p_1, \ldots, p_{m_0}$ *then*

$$\mathbf{Pr}(p_{(i)} < \alpha) = \mathbf{Pr}(U_{(i)} < \alpha) \tag{1}$$

where $p_{(i)}$ *is the ith ordered p-value and* $U_{(i)}$ *is the ith order statistic of a sample of size* m_0 *from the uniform distribution.*

The Joint Null Criterion (JNC) specifies the joint behavior of a set of null p-values from a multiple testing study. This joint behavior is critical, since error estimates and significance calculation are performed on the single set of p-values obtained in any given study.

This document provides a tutorial for using the `dks` package to evaluate multiple testing procedures. The package consists of the functions: `dks` for calculating both the Frequentist and Bayesian diagnostic tests proposed by [1], including diagnostic plots, `dks.pvalue` for computing the double Kolmogorov-Smirnov test that the null p-values are uniformly distributed, `pprob.uniform` for calculating the posterior probability a set of p-values are uniformly distributed, `pprob.dist` for calculating the posterior probability distribution of the Beta hyperparameters, and `cred.set` for calculating a credible set of the hyperparameters of the Beta distribution. As with any R package, detailed information on functions, their arguments, and values can be obtained from the help files. For instance, to view the help file for the function `dks` within R, type `? dks`. Here we will perform the set of diagnostic tests on a previously simulated null p-values.

2 Suggestions for P-value Simulation

We will use simulated null p-values to demonstrate the `dks`. In practice, null p-values should be simulated using the multiple testing method that is being evaluated. The simulated studies should contain tests that are both null and alternative, resulting in both null and alternative p-values. The simulated data sets should mimic the expected behavior of real data sets as closely as possible. Multiple simulated studies should be performed under random variation in the simulated parameters to give an idea of the range of behavior of null p-values from the multiple testing method. Only the null p-values should be passed to the functions in the `dks` package.

3 Simulated P-values

We will demonstrate the functions using a simulated set of null p-values. The simulated null p-values come from 100 different simulated studies, and for each study there are 200 null p-values. The p-values are placed in a 200×100 matrix where each column corresponds to the set of null p-values for a single simulated study.
To load the data set type `data(dksdata)`, and to view a description of this data type `? dksdata`.

```
> library(dks)
> library(cubature)
> data(dksdata)
> dim(P)

[1] 200 100
```

4 The dks function

The `dks` computes both the Frequentist and Bayesian diagnostic tests proposed in [1]. The function accepts a matrix of simulated null p-values where each column corresponds to the simulated null p-values from a single study. For the Bayesian diagnostic criteria, it is possible to specify the range of the hyperparameters of the beta distribution.

```
> dks1 <- dks(P,plot=TRUE)

Computing DKS P-value...Computing Posterior Probabilities...
```

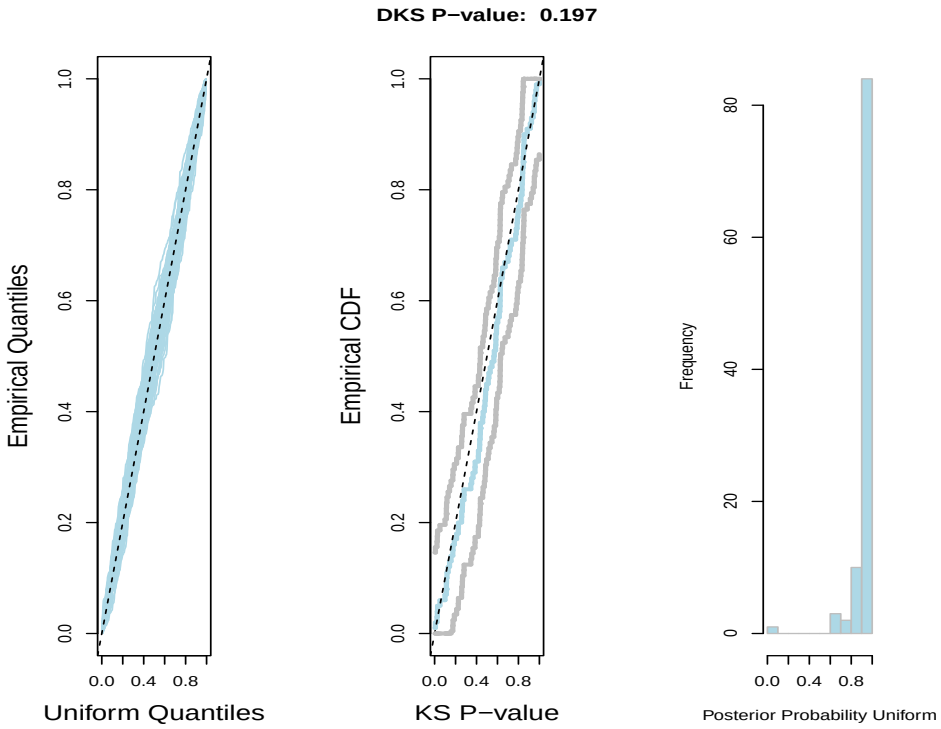

Figure 1: The diagnostic plots from the `dks` function. From left to right: (a) a quantile-quantile plot of the p-values for each study versus the uniform quantiles, these lines should be close to the 45° line, (b) the empirical distribution function of the first level KS test p-values (blue) with confidence bands (grey), again the line should fall along the 45° line, and (c) a histogram of the posterior probabilities that each set of p-values is uniform these values should be near one.

The double KS p-value is a nested KS-test against the uniform. First each study specific set of p-values is tested against the uniform and then the KS test p-values are tested against the uniform. If the double KS p-value is large then the p-values produced by the multiple testing procedure appear to be uniform across simulated studies. For each study, the posterior probability that the p-values are uniform is also calculated. For uniform p-values these posterior probabilities should be near one.
The diagnostic plots are, from left to right: (a) a quantile-quantile plot of the p-values for each study versus the uniform quantiles, these lines should be close to the 45° line, (b) the empirical distribution function of the first level KS test p-values (blue) with confidence bands (grey), again the line should fall along the 45° line, and (c) a histogram of the posterior probabilities that each set of p-values is uniform these values should be near one.

5 The `pprob.dist` function

The `pprob.dist` calculates the posterior distribution for the hyperparameters of the Beta distribution given the observed set of p-values. If the p-values are approximately uniform, then the posterior distribution should be concentrated at the values of (1,1). Figure 2 shows examples of the distribution functions for a range of values of (α, β). The parameter values $(\alpha, \beta) = (1, 1)$ correspond to the uniform distribution (black), values of $\beta > 1$ and $\alpha < 1$ correspond to distributions that are stochastically smaller than the uniform (green), values of $\beta < 1$ and $\alpha > 1$ correspond to distributions that are stochastically larger than the uniform (blue), values of $\beta > 1$ and $\alpha > 1$ correspond to distributions in the center of the interval (purple), and values of $\beta < 1$ and $\alpha < 1$ correspond to distributions on the extremes of the interval (red).

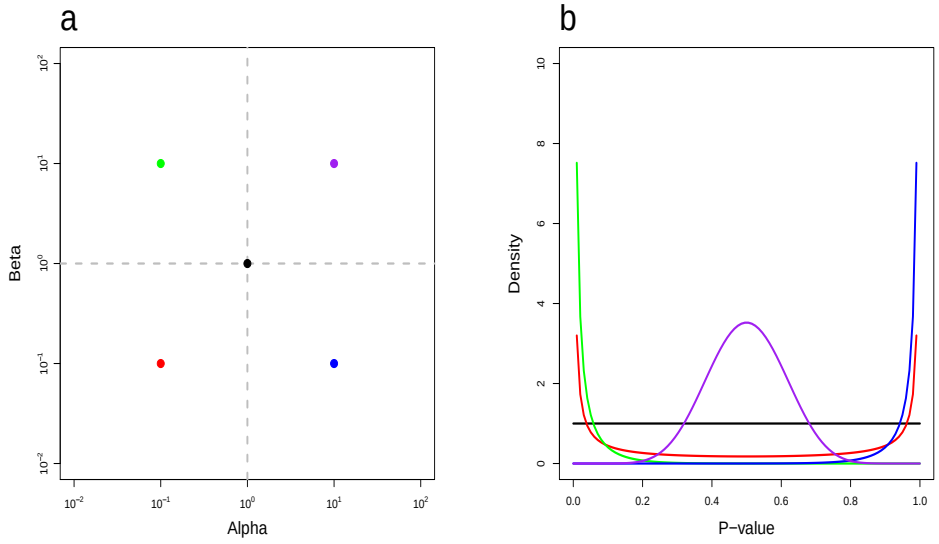

Figure 2: **a.** The parameter space for the beta distribution, with different parameter combinations highlighted. **b.** The beta density for the values of (α, β) highlighted in **a**. The densities mimic the behavior of p-values observed in high-dimensional multiple testing experiments.

We use the code to calculate the posterior distribution for the p-values from the first study (the first column of P). In practice since each study may correspond to a different distribution of null

p-values, it may be appropriate to calculate the posterior for each study. If it is expected that the p-values exhibit extreme behavior in one of the directions illustrated in Figure 2, then the range of the parameters α and β should be extended.
The function returns the posterior values at the grid points defined by (alpha[1] + i×delta, beta[1] + j×delta). We can plot the posterior distribution using the image command in R and put a dot at (1,1), which indicates the uniform distribution.

```
>  ## Calculate the posterior distribution
>  delta <- 0.1
>   dist1 <- pprob.dist(P[,1])
>   ## Plot the posterior distribution
>   alpha <- seq(0.1,10,by=delta)
>   beta <- seq(0.1,10,by=delta)
>   pprobImage = image(log10(alpha),log10(beta),dist1,xaxt="n",yaxt="n",xlab="Alpha",ylab="Beta"
>   axis(1,at=c(-2,-1,0,1,2),labels=c("10^-2","10^-1","10^0","10^1","10^2"))
>   axis(2,at=c(-2,-1,0,1,2),labels=c("10^-2","10^-1","10^0","10^1","10^2"))
>   points(0,0,col="blue",cex=1,pch=19)
```

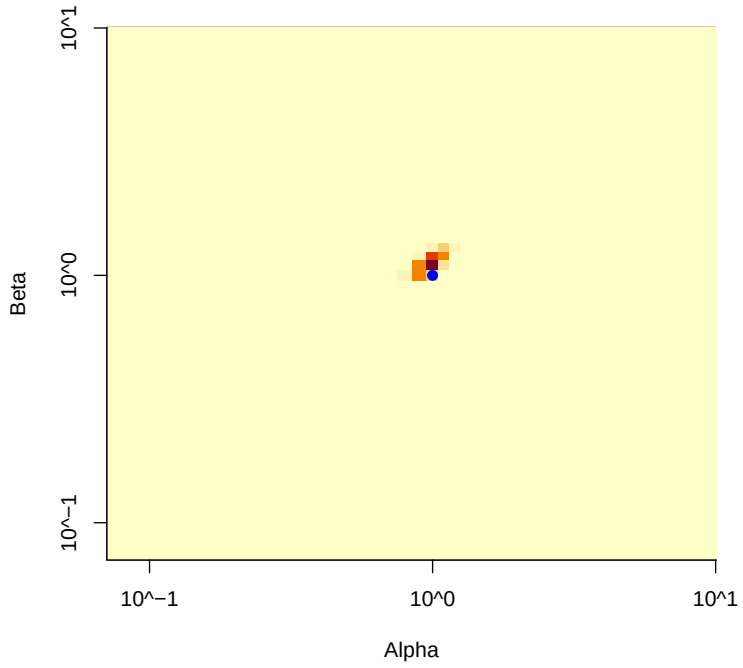

Figure 3: A plot of the posterior distribution from the observed example. The p-values are uniform and so the posterior distribution is centered at (1,1), indicated by the blue dot.

6 The `cred.set` function

The `cred.set` takes as input the computed distribution from the function `pprob.dist`. The grid size, delta, must be the same for both function calls. The user sets the level of the credible set, here we calculate a 80% credible interval and since the p-values are approximately uniform the 80% credible interval includes (1,1).

```
>  ## Calculate a 80% credible set
>   cred1 <- cred.set(dist1,delta=0.1, level=0.80)
>   ## Plot the posterior and the credible set
>
>   alpha <- seq(0.1,10,by=delta)
>   beta <- seq(0.1,10,by=delta)
>   credImage=image(log10(alpha),log10(beta),cred1$cred,xaxt="n",yaxt="n",xlab="Alpha",ylab="Bet
>   axis(1,at=c(-2,-1,0,1,2),labels=c("10^-2","10^-1","10^0","10^1","10^2"))
>   axis(2,at=c(-2,-1,0,1,2),labels=c("10^-2","10^-1","10^0","10^1","10^2"))
>   points(0,0,col="blue",cex=1,pch=19)
```

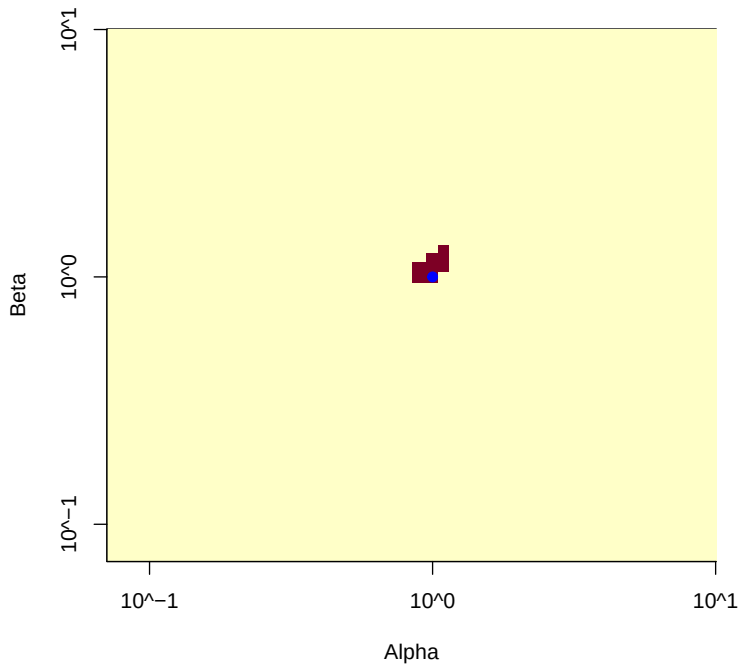

Figure 4: A plot of the 80% credible set for the interval. The p-values are uniform and so the credible set includes (1,1), indicated by the blue dot.

References

[1] Leek J.T. and Storey J.D. *The Joint Null Criterion for Multiple Hypothesis Tests*, SAGMB 10:28