Package 'DiscreteQvalue'

May 7, 2026

Title Improved q-Values for Discrete Uniform and Homogeneous Tests

Version 1.1

Maintainer Marta Cousido Rocha <martacousido@uvigo.es>

Description We consider a multiple testing procedure used in many modern applications which is the q-value method proposed by Storey and Tibshirani (2003), <doi:10.1073/pnas.1530509100>. The q-value method is based on the false discovery rate (FDR), hence versions of the q-value method can be defined depending on which estimator of the proportion of true null hypotheses, p0, is plugged in the FDR estimator. We implement the q-value method based on two classical pi0 estimators, and furthermore, we propose and implement three versions of the q-value method for homogeneous discrete uniform P-values based on pi0 estimators which take into account the discrete distribution of the P-values.

License GPL-2

Suggests coin, exactRankTests

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

NeedsCompilation no

Author Marta Cousido Rocha [aut, cre],
José Carlos Soage González [ctr],
Jacobo de Uña Álvarez [aut, ths],
Sebastian Döhler [aut]

Repository CRAN

Date/Publication 2020-04-01 17:10:03 UTC

Contents

DiscreteQvalue-package

Improved q-values for discrete uniform and homogeneous tests

Description

This package implements five different versions of the q-value multiple testing procedure proposed by Storey and Tibshirani (2003). The q-value method is based on the false discovery rate (FDR); different versions of the q-value method can be defined depending on the particular estimator used for the proportion of true null hypotheses, π_0, which is plugged in the FDR formula. The first version of the q-value uses the π_0 estimator in Storey (2002), with tunning parameter $\lambda = 0.5$; whereas the second version uses the π_0 estimator in Storey and Tibshirani (2003), which is based on an automatic method to select the tunning parameter λ. These two methods are only appropriate when the P-values follow a continuous uniform distribution under the global null hypothesis. This package also provides three other versions of the q-value for homogeneous discrete uniform P-values, which often appear in practice. The first discrete version of the q-value uses the π_0 estimator proposed in Liang (2016). The second discrete q-value method uses the estimator of π_0 proposed in Chen et al. (2014), when simplified for the special case of homogeneous discrete P-values. The third discrete version of the q-value employs a standard procedure but applied on randomized P-values. Once the estimated q-values are computed, the q-value method rejects the null hypotheses whose q-values are less than or equal to the nominal FDR level. All the versions of the q-value method explained above can be seen in Cousido-Rocha et al. (2019).

Details

- Package: 'DiscreteQvalue'
- Version: 1.0
- Maintainer: Marta Cousido Rocha <martacousido@uvigo.es>
- License: GPL-2

Value

- 'DQ'

Acknowledgements

This work has received financial support of the Call 2015 Grants for PhD contracts for training of doctors of the Ministry of Economy and Competitiveness, co-financed by the European Social Fund (Ref. BES-2015-074958). The authors acknowledge support from MTM2014-55966-P project, Ministry of Economy and Competitiveness, and MTM2017-89422-P project, Ministry of Economy, Industry and Competitiveness, State Research Agency, and Regional Development Fund, UE. The authors also acknowledge the financial support provided by the SiDOR research group through the grant Competitive Reference Group, 2016-2019 (ED431C 2016/040), funded by the "Consellería de Cultura, Educación e Ordenación Universitaria. Xunta de Galicia".

Author(s)

- Marta Cousido Rocha.
- José Carlos Soage González.
- Jacobo de Uña Álvarez.
- Sebastian Döhler.

References

- Chen, X., R. W. Doerge, and J. F. Heyse (2014). Methodology Multiple testing with discrete data: proportion of true null hypotheses and two adaptive FDR procedures. arXiv:1410.4274v2.
- Cousido-Rocha, M., J. de Uña-Álvarez, and S. Döhler (2019). Multiple testing methods for homogeneous discrete uniform P-values. Preprint.
- Liang, K. (2016). False discovery rate estimation for large-scale homogeneous discrete p-values. Biometrics 72, 639-648.
- Storey, J. D. (2002). A direct approach to false discovery rates. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 64 (3), 479-498.
- Storey, J. and R. Tibshirani (2003). Statistical significance for genomewide studies. Proceedings of National Academy of Science 100, 9440-9445.

`DQ` *Improved q-values for discrete uniform and homogeneous tests*

Description

Performs the five versions of the q-value method considered in Cousido-Rocha et al. (2019). The q-value method is based on the false discovery rate (FDR), and the versions differ in the estimator of the proportion of true null hypotheses, π_0, which is plugged in the FDR estimator. Specifically, we consider as possible estimators for π_0: two usual estimators for continuous and possibly heterogeneous P-values; an estimator for discrete P-values defined in two steps: firstly the P-values are randomized, and then the usual π_0 estimator for continuous P-values is applied; and the estimators recently proposed for discrete P-values by Liang (2016) and Chen et al. (2014).

Usage

```
DQ(
  pv,
  ss = NULL,
  ss_inf = FALSE,
  method = c("ST", "SS", "Liang", "Chen", "Rand"),
  lambda = seq(0.05, 0.95, 0.05)
)
```

Arguments

pv	A vector of P-values.
ss	Support of the discrete distribution of the P-values. Only required for "Liang", "Chen" and "Rand" methods which are specifically proposed for discrete P-values. If the P-values are continuous the methods "ST" and "SS" do not need this argument, hence "ss=NULL" by default.
ss_inf	Logical. Default is FALSE. A variable indicating whether the support of the discrete distribution of the P-values is finite or infinite. See details.
method	The q-value method. By default the "Chen" method is computed.
lambda	The value of the tuning parameter to estimate π_0 when the method is "ST". See details.

Details

The function implements the five different versions of the q-value method in Cousido-Rocha et al. (2019). Three versions are novel adaptions for the case of homogeneous discrete uniform P-values, whereas the other two are classical versions of the q-value method for P-values which follow a continuous uniform distribution under the global null hypothesis. The classical versions are the q-value method based on the π_0 estimator proposed in Storey (2002) with tunning parameter λ = 0.5, and the q-value procedure which uses the π_0 estimator in Storey and Tibshirani (2003) who proposed an automatic method to estimate π_0. We refer to these methods as "SS" and "ST", respectively. On the other hand the three adaptations of the q-value method for homogeneous discrete uniform P-values are: "Liang" which considers the π_0 estimator proposed in Liang (2016); "Chen" which uses a simplification for homogeneous discrete P-values of the algorithm for the estimation of π_0 proposed in Chen et al. (2014); and "Rand" which employs the standard q-value procedure but applied to randomized P-values. For details of the different q-value versions, in particular for the novel adaptations for homogeneous discrete uniform P-values, see Cousido-Rocha et al. (2019).

As we mentioned above the novel adaptations of the q-value method are developed for homogeneous discrete uniform P-values. Specifically, suppose that we test a large number of null hypothesis, p, and that the P-values $\{pv_1, \ldots, pv_p\}$ are observations of the random variables $PV_i, i = 1, \ldots, p$. Homogeneous means that all the P-values share an identical support S with $0 < t_1 < \cdots < t_s < t_{s+1} \equiv 1$. On the other hand, making an abuse of language, we say that the P-values follow a discrete uniform distribution if it holds $Pr(PV_i \leq t) = t$ for $t \in S, i = 1, \ldots, p$. The classical discrete uniform distribution is a particular case.

The argument "lambda" must be a sequence of values in $[0, 1)$, for details of this parameter see Storey (2002) or Storey and Tibshirani (2003). The latter paper recommends the default value "lambda=seq(0.05,0.95,0.05)".

The support of the discrete distribution of the P-values can be finite or infinte. Hence the parameter "ss inf" must be "FALSE" if the support is finite and "TRUE" if the support is infinite. See examples where a poisson setting is considered.

Finally, it is relevant to mention that Cousido-Rocha et al. (2019) verified (via simulations) that the results of the different q-values methods for dependent P-values are very similar to the ones corresponding to the independent setting.

Value

A list containing the following components:

`pi0`	An estimate of the proportion of null P-values.
`q.values`	A vector of the estimated q-values (the main quantities of interest).

Author(s)

- Marta Cousido-Rocha
- José Carlos Soage González
- Jacobo de Uña-Álvarez
- Sebastian Döhler

References

- Chen, X., R. W. Doerge, and J. F. Heyse (2014). Methodology Multiple testing with discrete data: proportion of true null hypotheses and two adaptive FDR procedures. arXiv:1410.4274v2.
- Cousido-Rocha, M., J. de Uña-Álvarez, and S. Döhler (2019). Multiple testing methods for homogeneous discrete uniform P-values. Preprint.
- Gibbons, J. D. and S. Chakraborti (1992). Nonparametric Statistical Inference. Third Edition. Marcel Dekker, Inc, New York.
- Liang, K. (2016). False discovery rate estimation for large-scale homogeneous discrete p-values. Biometrics 72, 639-648.
- Storey, J. D. (2002). A direct approach to false discovery rates. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 64 (3), 479-498.
- Storey, J. and R. Tibshirani (2003). Statistical significance for genomewide studies. Proceedings of National Academy of Science 100, 9440-9445.

Examples

```
# We consider a simple simulated data set to illustrate the use of the DQ function.
# We have simulated the following situation.
# We have two groups, for example, 5 patients with tumor 1 and 5 patients
# with tumor 2. For each patient 100 variables are measured, for example,
# gene expression levels. Besides, the distributions of 30 of the variables
# are different in the two groups, and the differences are in location.


# We consider a collection of densities {f1=N(0,1), f2=N(0,1/4), f3=N(2,1), f4=N(2,1/4)}. In
# the first group (tumor 1) the sample of each variable (gene) comes from one of
# the four densities with the same probability 1/4. On the other hand, in the second
# group the sample of each variable comes from the same density as in the first
# group except for 30 randomly selected variables for which the density changes
# producing location differences. Specifically, if the variable follows
# f1 in the first group, its density, in the second group, is f3  producing a
# change on its location parameter.The situation for the other cases is as follows:
# the density f2 (f3 or f4) in group 1 leads to density f4 (f1 or f2, respectively)
# in the second one.
```

```
set.seed(123)
p <- 100
n = m = 5
inds <- sample(1:4, p, replace = TRUE)
X <- matrix(rep(0, n * p), ncol = n)

for (j in 1:p){
  if (inds[j] == 1){
    X[j, ] <- rnorm(n)}
  if (inds[j] == 2){
    X[j, ] <- rnorm(n, sd = sqrt(1/4))
  }
  if (inds[j] == 3){
    X[j, ]<-rnorm(n, mean = 2)
  }
  if (inds[j] == 4){
    X[j, ]<-rnorm(n, mean = 2, sd = sqrt(1/4))
  }
}

rho <- 0.3

ind <- sample(1:p, rho * p)
li <- length(ind)
indsy <- inds

for (l in 1:li){
  if (indsy[ind[l]] == 1){indsy[ind[l]] = 3} else{
    if (indsy[ind[l]] == 2){indsy[ind[l]] = 4} else {
      if (indsy[ind[l]] == 3){indsy[ind[l]] = 1}
      else{indsy[ind[l]] = 2}}}}

Y <- matrix(rep(0, m * p), ncol = m)

for (j in 1:p){
  if (indsy[j] == 1){
    Y[j, ] <- rnorm(m)}
  if (indsy[j] == 2){
    Y[j, ] <- rnorm(m, sd = sqrt(1/4))
  }
  if (indsy[j] == 3){
    Y[j, ] <- rnorm(m, mean = 2)
  }
  if (indsy[j] == 4){
    Y[j, ]<- rnorm(m, mean = 2, sd = sqrt(1/4))
  }
}


# We can see which are the variables with different distributions in the two data sets.

dif <- which(inds != indsy)
```

```
# Cross table for (X,Y) density indexes:

table(inds,indsy)

# Our interest is to identify which variables have a different distribution in the two groups.
# Hence, since the differences between the distributions are in location, we applied
# Wilcoxon-Mann-Whitney test to verify for each variable the equality of distribution
# in the two groups.



library(exactRankTests)
library(coin)

# We compute the P-values

p <- nrow(X)
pv <- 1:p
for (i in 1:p){
  pv[i] <- wilcox.exact(X[i, ], Y[i, ])$p.value
}

# When the sample size is small, in this case n=m=5, the distribution of
# the Wilcoxon's statistic is calibrated using an exact permutation test. Hence,
# the P-values are homogeneous discrete uniform distributed with support points
# of such distribution:

ss <- c(1, 2, 4, 7, 12, 19, 28, 39, 53,69, 87, 106, 126) / 126

# When the number of P-values is large enough "ss" is equal to:
sort(unique(pv))

# For details about the Wilcoxon-Mann-Whitney test and its exact distribution
# see Section 9.2 of Gibbons and Chakraborti (1992).

# We apply Chen method:

R <- DQ(pv, ss = ss, method = "Chen")

# The estimate of the proportion of null P-values:

R$pi0

# Summary of the vector of the estimated q-values:

summary(R$q.values)

# How many null hypotheses are rejected?
alpha <- 0.05
sum(R$q.values < alpha)

# Which variables correspond to such null hypotheses?
```

```
which(R$q.values < alpha)

# Classification table (Decision at nominal level alpha vs. reality):

table(R$q.values < alpha,inds != indsy)

# The conclusion from the previous table is that Chen method reports
# 21 true positives and 9 false negatives.


# We can also apply Liang and SS methods as follows.
RLiang <- DQ(pv, ss = ss, method = "Liang")
RSS <- DQ(pv, ss = ss, method = "SS")

# The next graphic help us to see that Liang method (for discrete P-values)
# is more powerful than SS method (only suitable for continuous P-values).

plot(RSS$q.values,RLiang$q.values)
abline(a = 0, b = 1, col = 2, lty = 2)

# We consider a poisson setting to show a case where the support of the discrete
# distribution of the P-values is infinte.
# We generate 100 values of a poisson distribution with event rate 10.
# Then we compute the probability that each of the values come from a
# a poisson distribution with event rate 10. This set of probabilities
# are considered as our set of P-values.

p <- 100
N <- rpois(p, lambda = 10)
pv <- 1:p
for(i in 1:p){
  pv[i] <- ppois(N[i], lambda = 10)
}

# It is well know that the support of a poisson distribution is infinite
# and equal to the natural numbers. Hence to know the support of the P-values
# defined above, we compute for 1,..., 50 their corresponding P-value.
# We only considered 50 values because for large values than 50 the P-value is 1 again.

nn_0 <- 50
ss <- 1:(nn_0 + 1)
for (i in 1:(nn_0 + 1)){
 ss[i] <- ppois(i - 1, lambda = 10)
}

# We eliminate repeated values.
ss <- unique(ss)
# For Chen method the relevant support points are only the values below tau[100] = 0.5.
# We define the support ss as such values. Then, we can apply Chen method. Of
# course s_inf = TRUE.

indicator <- which(ss <= 0.5)
```

```
ssi <- ss[indicator]
R <- DQ(pv, ss = ssi, ss_inf = "TRUE", method = "Chen")
# For Liang method the relevant support values are also the values below 0.5, hence
# ss defined above is suitable.
R <- DQ(pv, ss = ssi, ss_inf = "TRUE", method = "Liang")

# For Rand method the relevant support values are those ones with match with
# the P-values, and also the largest support points smaller than each one of such P-values.
# Hence, ss only includes the relevant points, as we can see below.
pv_unique <- unique(pv)
p_u <- length(pv_unique)
ind <- 1:p_u
for(i in 1:p_u){
  ind[i] <- which(ss == pv_unique[i])
}
p_u == length(ind)
ind_minus <- ind - 1
ind_final <- unique(sort(c(ind, ind_minus)))
ss <- ss[ind_final]
# Now, we can apply Rand method.
R <- DQ(pv, ss = ss, ss_inf = "TRUE", method = "Rand")
```

Index