Package ‘QuasiSeq’

February 19, 2015

Version 1.0-6
Date 2015-02-02
Title Analyzing RNA Sequence Count Tables Using Quasi-likelihood
Author Steve Lund <lundsp@gmail.com> and Long Qu <long.qu@wright.edu>
Maintainer Steve Lund <lundsp@gmail.com>
Depends R (>= 3.1.2)
Imports edgeR, Matrix, mgcv

Description Identify differentially expressed genes in RNA-seq count data using quasi-
Poisson or quasi-negative binomial models with QL, QLShrink and QLSpline meth-
License GPL (>= 2)
NeedsCompilation no
Repository CRAN
Date/Publication 2015-02-03 06:38:31

R topics documented:

NBDev .................................................. 2
PoisDev .................................................. 3
QL.fit ................................................... 4
QL.results ............................................ 7

Index 11
NBDev: Fit a negative binomial GLM for given design matrix

Description
A function called within QL.fit to fit a negative binomial GLM of each gene for given design matrix

Usage
```R
NBDev(counts, design, log.offset, nb.disp, print.progress=TRUE)
```

Arguments
- `design`: A single element from the `design.list` argument given to `QL.fit`.
- `log.offset`: A vector of log-scale, additive factors used to adjust estimated log-scale means for differences in library sizes across samples. Commonly used offsets include `log.offset=log(colSums(counts))` or `log.offset=log(apply(counts, 2, quantile, .75))`. The default setting in `QLfit` makes no adjustment for library sizes (i.e. `log.offset=0`).
- `nb.disp`: estimated negative binomial dispersion parameters obtained from either `estimateGLMTrendedDisp` or `estimateGLMCommonDisp` in package `edgeR`. These estimates are treated as known and are used to compute deviances.
- `print.progress`: logical. If TRUE, the function will provide an update on what gene (row number) is being analyzed. Updates occur frequently to start then eventually occur every 5000 genes.

Value
- list containing:
  - `dev`: vector containing the deviance for each gene under a negative binomial model fit to design matrix specified by `design`. This vector is passed along within the `QL.fit` function.
  - `means`: matrix of fitted mean values for each gene
  - `parms`: matrix of estimated coefficients for each gene. Note that these are given on the log scale. (i.e. intercept coefficient reports log(average count) and non-intercept coefficients report estimated log fold-changes.) Genes with at least one zero count and initial absolute coefficient estimates greater than three undergo the bias correction of Kosmidis & Firth (2009) to moderate extreme coefficient estimates.
**Author(s)**

Steve Lund <lundsp@gmail.com>

**References**

Kosmidis and Firth (2009) "Bias reduction in exponential family nonlinear models" *Biometrika, 96*, 793–804.


---

PoisDev

Compute Poisson deviances (up to a constant) for given design matrix

**Description**

A function called within QL.fit to compute Poisson deviances of each gene for given design matrix

**Usage**

PoisDev(counts, design, log.offset, print.progress=TRUE)

**Arguments**

- **counts**: RNA-seq data matrix of integer expression counts. Each row contains observations from a single gene. Each column contains observations from a single experimental unit.
- **design**: A single element from the design.list argument given to QL.fit.
- **log.offset**: A vector of log-scale, additive factors used to adjust estimated log-scale means for differences in library sizes across samples. Commonly used offsets include,\(\text{log.offset} = \text{log}(\text{colSums(counts)})\) or \(\text{log.offset} = \text{log} \left( \text{apply(counts,2,quantile,.75)} \right) \). The default setting in QL.fit makes no adjustment for library sizes (i.e. \text{log.offset}=0).
- **print.progress**: logical. If TRUE, the function will provide an update on what gene number is being analyzed. Updates occur frequently to start then eventually occur every 5000 genes. Note that updates will not occur for one-factor designs, for which closed form solutions are available.

**Value**

list containing:

- **dev**: vector containing the deviance for each gene under a Poisson model fit to design matrix (or vector, for one-factor experiments) specified by design. This vector is passed along within the QL.fit function.
- **means**: matrix of fitted mean values for each gene
- **parms**: matrix of estimated coefficients for each gene
**Author(s)**

Steve Lund <lundsp@gmail.com>

---

**QL.fit**

*Fit quasi-likelihood models to matrix of RNA-seq expression count data*

---

**Description**

Fit a quasi-likelihood model to RNA-seq expression count data using the methods detailed in Lund, Nettleton, McCarthy, and Smyth (2012).

**Usage**

QL.fit(counts, design.list, test.mat=NULL, log.offset=NULL, Model="NegBin", print.progress=TRUE, NBdisp="trend", ...)

**Arguments**

- **counts**: RNA-seq data matrix of integer expression counts. Each row contains observations from a single gene. Each column contains observations from a single experimental unit.

- **design.list**: List of design matrices for the full model and reduced model(s). The first element of design.list must describe the overall full model, as this design is used to compute deviance residuals for estimating dispersion. One-factor designs may be specified as vectors. The number of rows in each design matrix (or the length of each design vector) must be ncol(counts). Means are modeled with a log link function.

- **test.mat**: T by 2 matrix dictating which designs are to be compared, where T is the total number of desired hypothesis tests for each gene. Each row contains two integers, which provide the indices within design.list of the two designs to be compared. If test.mat is not specified, the default is compare the first design (the full model) to each of the other designs provided in design.list in order (i.e. first design compared to second design, first design compared to third design, first design compared to fourth design, etc.).

- **log.offset**: A vector of log-scale, additive factors used to adjust estimated log-scale means for differences in library sizes across samples. Commonly used offsets include log.offset=log(colSums(counts)) and log.offset=log(apply(counts[rowSums(counts)!=0],),). The default setting makes no adjustment for library sizes (i.e. log.offset=0).

- **Model**: Must be one of "Poisson" or "NegBin", specifying use of a quasi-Poisson or quasi-negative binomial model, respectively.

- **print.progress**: logical. If TRUE, updates are provided regard what gene (row number) is being analyzed. Updates occur frequently to start then eventually occur every 5000 genes.
**NBdisp**

Used only when `Model="NegBin"`. Must be one of "trend", "common" or a vector of non-negative real numbers with length equal to `nrow(counts)`. Specifying `NBdisp="trend"` or `NBdisp="common"` will use `estimateGLMTrendedDisp` or `estimateGLMCommonDisp`, respectively, from the package edger to estimate negative binomial dispersion parameters for each gene. Estimates obtained from other sources can be used by entering `NBdisp` as a vector containing the negative binomial dispersion value to use for each gene when fitting quasi-likelihood model.

... arguments to be passed to the function `estimateGLMTrendedDisp` or `estimateGLMCommonDisp` from the package edger.

**Value**

List containing:

- **"LRT"** matrix providing unadjusted likelihood ratio test statistics. Each column contains statistics from a single hypothesis test, applied separately to each gene.
- **"phi.hat.dev"** vector providing unshrunken, deviance-based estimates of quasi-dispersion (phi) for each gene.
- **"phi.hat.pearson"** vector providing unshrunken, Pearson estimates of quasi-dispersion (phi) for each gene.
- **"mn.cnt"** vector providing average count for each gene.
- **"den.df"** denominator degrees of freedom. Equal to the number of samples minus the number of fitted parameters in the full model, which is specified by the first element of `design.list`.
- **"num.df"** vector of numerator degrees of freedom for each test, computed as the difference in the number of fitted parameters between the full and reduced models for each test.
- **"Model"** Either "Poisson" or "NegBin", specifying which model (quasi-Poisson or quasi-negative binomial, respectively) was used.
- **"nb.disp"** Only appears when `Model="NegBin"`. Vector providing negative binomial dispersion parameter estimate used during fitting of quasi-negative binomial model for each gene.

**fitted.values** matrix of fitted mean values

**coefficients** matrix of estimated coefficients for each gene. Note that these are given on the log scale. (i.e. intercept coefficient reports log(average count) and non-intercept coefficients report estimated log fold-changes.) Genes with at least one zero count and initial absolute coefficient estimates greater than three undergo the bias correction of Kosmidis & Firth (2009) to moderate extreme coefficient estimates.

**Author(s)**

Steve Lund <lundsp@gmail.com>
References


Examples

```r
## Not run:
set.seed(234092)

n.genes<-100
n.de<-round(.5*n.genes)
trt<rep(1:2,each=4)
n.samp<-length(trt)
mu<-rgamma(n.genes,1.5,.01)

## specify gene specific negative binomial dispersions
size<-((log(mu+exp(1))-1)/mu) ## Var(Y)=E(Y)log(E(Y)+exp(1))

## add noise to gene specific negative binomial dispersions
size<-size*4.5/rchisq(n.genes,4.5)

sim.nn<-matrix(mu,n.genes,2)

## Simulate fold changes
B<-exp((2*rbinom(n.de,1,.5)-1)*(.25+rbeta(n.de,1,2)))

sim.nn[1:n.de,1]<-sim.nn[1:n.de,1]*B^(.5)+5
sim.nn[1:n.de,2]<-sim.nn[1:n.de,2]*B^(-.5)

## Simulate library size factors
sim.offset<-2^rnorm(n.samp,0,.15)

## Compute final means
sim.nn2<-t(t(sim.nn[,trt])*sim.offset)

## Simulate data
sim.dat<-matrix(rnbinom(n.samp*n.genes,mu=sim.nn2,size=1/size),n.genes,n.samp)

## Simulate estimated dispersions to save time

# THIS STEP SHOULD NOT BE PERFORMED WHEN ANALYZING REAL DATA#

est.nb.disp<-size*rchisq(n.genes,n.samp-2)/(n.samp-2)
est.nb.disp<est.nb.disp

## Keep genes with at least 10 total counts
```
est.nb.disp<-est.nb.disp[rowSums(simdat)>9]
simdat<-simdat[rowSums(simdat)>9,]

### Create list of designs describing model under null and alternative hypotheses
design.list<-vector("list",2)
design.list[[1]]<-model.matrix(~as.factor(trt)) #This also could have just been \`trt\'.
design.list[[2]]<-rep(1,length(trt))

log.offset<-log(apply(simdat,2,quantile,.75))

### Analyze using QL, QLShrink and QLSpline methods applied to quasi-MPoisson model
fit<-QL.fit(simdat, design.list, log.offset=log.offset, Model="Poisson")
results<-QL.results(fit)

### How many significant genes at FDR=.05 from QLSpline method?
apply(results$Q.values[[3]]<.05,2,sum)

### Indexes for Top 10 most significant genes from QLSpline method
head(order(results$P.values[[3]]), 10)

### Analyze using QL, QLShrink and QLSpline methods
### applied to quasi-negative binomial model
fit2<-QL.fit(simdat, design.list, log.offset=log.offset, nb.disp=est.nb.disp, Model="NegBin")

########################################################################
# Note: 'nb.disp' typically will not be specified when #
# calling QL.fit while analyzing real data. Providing #
# numeric values for 'nb.disp' prevents neg binomial  
# dispersions from being estimated from the data.  #
########################################################################

results2<-QL.results(fit2)

### How many significant genes at FDR=.05 for QLSpline method?
apply(results2$Q.values[[3]]<.05,2,sum)

### Indexes for Top 10 most significant genes from QLShrink method
head(order(results2$P.values[[2]]), 10)

### End(Not run)

---

QL.results | Obtain p-values and q-values using results from QL.fit

---

Description

Usage

```
QL.results(fit, Dispersion="Deviance", spline.df=NULL, Plot=TRUE)
```

Arguments

- **fit**
  - The list returned by the function QL.fit

- **Dispersion**
  - Must be one of "Deviance" or "Pearson", specifying which type of estimator should be used for estimating quasi-likelihood dispersion parameter.

- **spline.df**
  - Optional. User may specify the degrees of freedom to use when fitting a cubic spline to log-scale(estimated dispersion) versus the log(average count). Default uses cross-validation in sreg function to pick appropriate degrees of freedom.

- **Plot**
  - logical. If TRUE, the estimated dispersion versus the average count are plotted on a log-scale with the corresponding cubic spline fit overlaid.

Value

- list containing:
  - "P.values" list of matrices providing p-values for the QL, QLShrink and QLSpline methods, respectively. The i^th column of each element of pvals corresponds to the hypothesis test assigned in the i^th row of test.mat.
  - "Q.values" list of matrices providing q-values for the QL, QLShrink and QLSpline methods, respectively. The i^th column of each element of qvals corresponds to the hypothesis test assigned in the i^th row of test.mat. Q-values are computed using the methods of Nettleton et al. (2006) JABES 11, 337-356.
  - "F.stat" list of matrices providing F-statistics for the QL, QLShrink and QLSpline methods, respectively. The i^th column of each element of F.stat corresponds to the hypothesis test assigned in the i^th row of test.mat.
  - "m0" matrix providing estimated number of true null hypotheses for each test(arranged by row) under each of the three methods(arranged by column). m0 values are computed using the methods of Nettleton et al. (2006) JABES 11, 337-356.
  - "d0" vector containing estimated additional denominator degrees of freedom gained from shrinking dispersion estimates in the QLShrink and QLSpline procedures, respectively.

Author(s)

Steve Lund <lundsp@gmail.com>

References

Examples

```r
set.seed(234092)

n.genes<-100
n.de<-round(.5*n.genes)
trt<-rep(1:2,each=4)
n.samp<-length(trt)
mu<-rgamma(n.genes,1.5,.01)

## specify gene specific negative binomial dispersions
size<-(log(mu+exp(1))-1)/mu  ### Var(Y)=E(Y)log(E(Y)+exp(1))

## add noise to gene specific negative binomial dispersions
size<-size*4.5/rchisq(n.genes,4.5)

sim.mn<-matrix(mu,n.genes,2)

### Simulate fold changes
B<-exp((2*rbinom(n.de,1,.5)-1)*(0.25+rbeta(n.de,1,2)))
sim.mn[1:n.de,1]<-sim.mn[1:n.de,1]*B^(.5)+5
sim.mn[1:n.de,2]<-sim.mn[1:n.de,2]*B^(-.5)

### Simulate library size factors
sim.offset<-2*(rnorm(n.samp,0,.15))

### Compute final means
sim.mn2<-t(t(sim.mn[,trt])*sim.offset)

### Simulate data
simdat<-matrix(rbinom(n.samp*n.genes,mu=sim.mn2,size=1/size),n.genes,n.samp)

### Simulate estimated dispersions to save time

#### THIS STEP SHOULD NOT BE PERFORMED WHEN ANALYZING REAL DATA ####
### Compute final means
est.nb.disp<-size*rchisq(n.genes,n.samp-2)/(n.samp-2)
est.nb.disp<-est.nb.disp

### Keep genes with at least 10 total counts
est.nb.disp<-est.nb.disp[rowSums(simdat)>9]
simdat<-simdat[rowSums(simdat)>9]

### Create list of designs describing model under null and alternative hypotheses
design.list<-vector("list",2)
design.list[[1]]<-model.matrix(~as.factor(trt))  #This also could have just been `\`trt`. 
design.list[[2]]<-rep(1,length(trt))
log.offset<-log(apply(simdat,2,quantile,.75))
```
### Analyze using QL, QLShrink and QLSpline methods applied to quasi-Poisson model

```r
g.fit <- QL.fit(simdat, design.list, log.offset=log.offset, Model="Poisson")
```

```r
results <- QL.results(g.fit)
```

### How many significant genes at FDR=.05 from QLSpline method?

```r
apply(results$Q.values[[3]] < .05, 2, sum)
```

### Indexes for Top 10 most significant genes from QLSpline method

```r
head(order(results$P.values[[3]]), 10)
```

### Analyze using QL, QLShrink and QLSpline methods applied to quasi-negative binomial model

```r
g2.fit <- QL.fit(simdat, design.list, log.offset=log.offset, nb.disp=est.nb.disp, Model="NegBin")
```

### Note

```
nb.disp typically will not be specified when calling QL.fit while analyzing real data. Providing numeric values for 'nb.disp' prevents neg binomial dispersions from being estimated from the data.
```

```r
results2 <- QL.results(g2.fit)
```

### How many significant genes at FDR=.05 for QLSpline method?

```r
apply(results2$Q.values[[3]] < .05, 2, sum)
```

### Indexes for Top 10 most significant genes from QLShrink method

```r
head(order(results2$P.values[[2]]), 10)
```
Index

*Topic RNA-seq, quasi-likelihood, differential expression, Poisson
PoisDev, 3

*Topic RNA-seq, quasi-likelihood, differential expression, negative binomial
NBDev, 2

*Topic RNA-seq, quasi-likelihood, differential expression
QL.fit, 4
QL.results, 7

estimateGLMCommonDisp, 5
estimateGLMTrendedDisp, 5

NBDev, 2

PoisDev, 3

QL.fit, 4
QL.results, 7