Package 'RSPS'

May 29, 2015

Type Package

Title RNA-Seq Power Simulation

Version 1.0

Date 2015-05-20

Author Milan Bimali, Joseph Usset, Brooke L. Fridley

Maintainer Milan Bimali <mbimali@kumc.edu>

Description Provides functions for estimating power or sample size for RNA-Seq studies. Empirical approach is used and the data is assumed to be count in nature. The underlying distribution of data is assumed to be Poisson or negative binomial. The package contains 6 function; 4 functions provide estimates of sample size or power for Poisson and Negative Binomial distribution; 2 functions provide plots of power for given sample size or sample size for given power.

License GPL-2

Imports lattice,plyr,gridExtra

LazyData yes

NeedsCompilation no

Repository CRAN

Date/Publication 2015-05-29 08:40:48

R topics documented:

RSPS-package *Provides estimates of power or sample size using empirical approach.*

Description

The R package RSPS uses an efficient simulation algorithm to empirically determine statistical power and necessary sample size for RNA-Seq studies. The data are simulated from Poisson (no overdispersion) or Negative Binomial distribution (overdispersion). The package allows one to monitor the progress of the function when the power is being computed. There are two functions for providing plots of the estimated power for given sample size and estimated sample size to achieve desired power.

Details

Package:	RSPS
Type:	Package
Version:	1.0
Date:	2015-05-19
License:	GPL-2

Author(s)

Milan Bimali Maintainer: Milan Bimali <mbimali@kumc.edu>

References

If necessary, users are referred to power and sample sizes computation found in introductory statistics textbooks.

See Also

rpoiss, rnbinom

Examples

```
power.poiss <- poiss.pow(n=c(5,10,15),lambda1=c(3),k=c(2,2.5,3),
alpha=0.05,seed = 20,numsim=100,monitor=TRUE)
power.poiss
power.plot(power.poiss,cutoff=c(0.8,0.9))
# Another example (takes longer to run)
#power.poiss <- poiss.pow(n=c(3,5,10,15),lambda1=c(3),k=c(2,2.5,3),
#alpha=0.001,seed = 20,numsim=500,monitor=TRUE)
#power.poiss
#power.plot(power.poiss,cutoff=c(0.8,0.9))
```

`negbin.pow` *Estimates power for given sample size using simulation from Negative Binomial distribution*

Description

The function provides estimate of power for given sample size when there is over-dispersion. The data is simulated from Negative Binomial distribution.

Usage

```
negbin.pow(n, lambda1, k, disp, alpha, seed, numsim, monitor, sig)
```

Arguments

`n`	A vector of positive integers representing the sample size
`lambda1`	Mean count under the null distribution. It can be a vector.
`k`	Fold change desired under the alternative distribution. It can be a vector.
`disp`	The over-dispersion parameter. 1 represent no pver-dispersion and values above one represent over-dispersion.
`alpha`	Type I error rate: a value between 0 and 1. It can be a vector.
`seed`	Value of seed to ensure reproducibility of results.
`numsim`	Number of simulations. 1000 is recommended.
`monitor`	If TRUE, it allows us to view the progress of the function.
`sig`	Number of significant digits after decimal.

Details

The test statistic used is the scaled difference. Please contact the authors for more details on algorithm.

Value

`Mean.Null`	Mean Count under Null distribution.
`Effect.Size`	Fold Change Under the alternate hypothesis.
`Disp.Par`	Over-dispersion parameter.
`Power`	Estimated Power.
`Std.Err`	Standard Error.

Note

The alternative ecological parameterization is used for Negative binomial distribution.

Author(s)

Milan Bimali

References

None

See Also

rnbinom

Examples

```
power.negbin <- negbin.pow(n=c(5,10),lambda1=c(3,5),
k=c(2,3),disp=2,alpha=0.001,seed = 20,
numsim=100,monitor=TRUE)
power.plot(power.negbin)
# Another example (takes longer to run)
#power.negbin <- negbin.pow(n=c(3,5,10,15),lambda1=c(3,5),
#k=c(2,2.5,3),disp=2,alpha=0.001,seed = 20,
#numsim=1000,monitor=TRUE)
#head(power.negbin)
```

`negbin.samp` *Estimates sample sizes for desired power using simulation from Negative Binomial distribution*

Description

The function provides estimate of sample size for given power when there is over-dispersion. The data is simulated from Negative Binomial distribution.

Usage

```
negbin.samp(power, lambda1, k, disp, alpha, seed, numsim, sig)
```

Arguments

`power`	A vector of values between 0 and 1 representing desired power.
`lambda1`	Mean count under the null distribution. It can be a vector.
`k`	Fold change desired under the alternative distribution. It can be a vector.
`disp`	The over-dispersion parameter. 1 represent no pver-dispersion and values above one represent over-dispersion.
`alpha`	Type I error rate: a value between 0 and 1. It can be a vector.
`seed`	Value of seed to ensure reproducibility of results.
`numsim`	Number of simulations. 1000 is recommended.
`sig`	Number of significant digits after decimal.

Details

The test statistic used is the scaled difference. Please contact the authors for more details on algorithm.

Value

`Power.Expected`	Desired Power.
`Mean.Null`	Mean Count under Null distribution.
`Effect.Size`	Fold Change Under the alternate hypothesis.
`Disp.Par`	Over-dispersion parameter.
`N.est`	Estimated sample size.
`Power.est`	Estimated Power.
`Std.Err`	Standard Error.

Note

The alternative ecological parameterization is used for Negative binomial distribution.

Author(s)

Milan Bimali

References

None

See Also

rnbinom

Examples

```
#-------------------------------------------------
power = c(0.7,0.8);lambda1=3;k=c(2,3);
disp=2;alpha=0.1;seed = 20;numsim=100
sample.negbin <- negbin.samp(power,lambda1,k,disp,alpha,seed,numsim)
head(sample.negbin)
# Another example (takes longer to run)
#power = seq(0.7,0.95,0.05);lambda1=3;k=c(2,2.5,3);
#disp=2;alpha=0.005;seed = 20;numsim=1000
#sample.negbin <- negbin.samp(power,lambda1,k,disp,alpha,seed,numsim)
#head(sample.negbin)
```

poiss.pow *Estimates power for given sample size using simulation from Poisson distribution*

Description

The function provides estimate of power for given sample size when there is no over-dispersion. The data is simulated from Poisson distribution.

Usage

```
poiss.pow(n, lambda1, k, alpha = 0.05, seed = 20, numsim = 2000, monitor = TRUE, sig = 3)
```

Arguments

n	A vector of positive integers representing the sample size
lambda1	Mean count under the null distribution. It can be a vector.
k	Fold change desired under the alternative distribution. It can be a vector.
alpha	Type I error rate: a value between 0 and 1. It can be a vector.
seed	Value of seed to ensure reproducibility of results.
numsim	Number of simulations. 1000 is recommended.
monitor	If TRUE, it allows us to view the progress of the function.
sig	Number of significant digits after decimal.

Details

The test statistic used is the scaled difference. Please contact the authors for more details on algorithm.

Value

Mean.Null	Mean Count under Null distribution.
Effect.Size	Fold Change Under the alternate hypothesis.
Power	Estimated Power.
Std.Err	Standard Error.

Note

None

Author(s)

Milan Bimali

References

None

See Also

rpois

Examples

```
#-------------------------------------------------
power.poiss <- poiss.pow(n=c(5,10,15),lambda1=c(3),k=c(2,3),
alpha=0.001,seed = 20,numsim=100,monitor=TRUE)
power.poiss
# Another example (takes longer to run)
#power.poiss <- poiss.pow(n=c(3,5,10,15),lambda1=c(3),k=c(2,2.5,3),
#alpha=0.001,seed = 20,numsim=1000,monitor=TRUE)
#power.poiss
```

---

poiss.samp *Estimates sample sizes for desired power using simulation from Poisson distribution*

Description

The function provides estimate of sample size for given power when there is over-dispersion. The data is simulated from Poisson distribution.

Usage

```
poiss.samp(power, lambda1, k, alpha, seed, numsim, sig)
```

Arguments

power	A vector of values between 0 and 1 representing desired power.
lambda1	Mean count under the null distribution. It can be a vector.
k	Fold change desired under the alternative distribution. It can be a vector.
alpha	Type I error rate: a value between 0 and 1. It can be a vector.
seed	Value of seed to ensure reproducibility of results.
numsim	Number of simulations. 1000 is recommended.
sig	Number of significant digits after decimal.

Details

The test statistic used is the scaled difference. Please contact the authors for more details on algorithm.

Value

`Power.Expected`	Desired Power.
`Mean.Null`	Mean Count under Null distribution.
`Effect.Size`	Fold Change Under the alternate hypothesis.
`N.est`	Estimated sample size.
`Power.est`	Estimated Power.
`Std.Err`	Standard Error.

Note

None

Author(s)

Milan Bimali

References

None

See Also

rpois

Examples

```
power = c(0.7,0.8);lambda1=3;k=seq(2,3,0.5);
alpha=0.01;seed = 20;numsim=100
sample.poiss <- poiss.samp(power,lambda1,k,alpha,seed,numsim)
sample.poiss
# Another example (takes longer to run)
#power = seq(0.7,0.9,0.05);lambda1=3;k=seq(2,3,0.5);
#alpha=0.005;seed = 20;numsim=1000
#sample.poiss <- poiss.samp(power,lambda1,k,alpha,seed,numsim)
#sample.poiss
```

---

power.plot *Provides plot of Estimated Power vs Sample Size*

Description

This function plots the estimate power vs sample size computed using function poiss.pow or negbin.pow for different fold changes. Please see example on how to use it.

Usage

```
power.plot(fit, ylim = c(-0.05, 1.05), cutoff = 0.8)
```

Arguments

fit	Function used to estimate the power. See example.
ylim	Limits of y-axis
cutoff	Cutoff value for power

Details

This function can only be run if the function poiss.pow or negbin.pow has been used. If a vector of type I error, null count, and overdispersion (for negbin.pow) is provided the plot is based on the smalles of these values.

Note

None

Author(s)

Milan Bimali

References

None

See Also

xyplot

Examples

```
#-------------------------------------------------
power.poiss <- poiss.pow(n=c(5,10,15),lambda1=c(3),k=c(2,2.5,3),
alpha=0.05,seed = 20,numsim=100,monitor=TRUE)
power.poiss
power.plot(power.poiss,cutoff=c(0.8,0.9))
# Another example (takes longer to run)
#power.poiss <- poiss.pow(n=c(3,5,10,15,20),lambda1=c(3),k=c(1.5,2,2.5,3),
#alpha=0.001,seed = 20,numsim=500,monitor=TRUE)
#power.plot(power.poiss,cutoff=c(0.8,0.9))
#power.negbin <- negbin.pow(n=c(3,5,10,15,20),lambda1=c(3,5),k=c(1.5,2,2.5,3,3.5),
#disp=2,alpha=0.001,seed = 20,numsim=1000,monitor=TRUE)
#power.plot(power.negbin,cutoff=0.8)
```

`sample.plot` *Provides plot of Estimated sample size vs power*

Description

This function plots the Estimated sample size vs power computed using function poiss.pow or negbin.pow for different fold changes. Please see example on how to use it.

Usage

```
sample.plot(fit, ylim = c(-0.05, 1.05), cutoff = 0.8)
```

Arguments

`fit`	Function used to estimate the sample size. See example.
`ylim`	Limits of y-axis
`cutoff`	Cutoff value for power

Details

This function can only be run if the function poiss.samp or negbin.samp has been used. If a vector of type I error, null count, and overdispersion (for negbin.pow) is provided the plot is based on the smalles of these values.

Note

None

Author(s)

Milan Bimali

References

None

See Also

xyplot

Examples

```
power = seq(0.5,0.9,0.05);lambda1=3;k=seq(2,3,1);
alpha=0.05;seed = 20;numsim=100
sample.poiss <- poiss.samp(power,lambda1,k,alpha,seed,numsim)
sample.plot(sample.poiss,cutoff=c(0.8,0.9))
# Another example (takes longer to run)
#power = seq(0.5,0.9,0.05);lambda1=3;k=seq(2,3,0.5);
```

```
#alpha=0.005;seed = 20;numsim=1000
#sample.poiss <- poiss.samp(power,lambda1,k,alpha,seed,numsim)
#sample.plot(sample.poiss,cutoff=c(0.8,0.9))
```

Index