Package 'NewmanOmics'

August 5, 2019

Type Package

LazyData true

Title Extending the Newman Studentized Range Statistic to Transcriptomics

Version 1.0.4

Date 2019-08-05

Author Zachary Abrams, Greg Gershkowitz, Anoushka Joglekar, Chao Liu, Kevin R. Coombes

Maintainer Kevin R. Coombes <krc@silicovore.com>

Description Extends the classical Newman studentized range statistic in various ways that can be applied to genome-scale transcriptomic or other expression data.

License Apache License (== 2.0)

Depends R (>= 3.5.0)

Imports methods, stats, graphics, grDevices, oompaBase

VignetteBuilder knitr

Suggests knitr, rmarkdown

URL http://oompa.r-forge.r-project.org/

NeedsCompilation no

Repository CRAN

Date/Publication 2019-08-05 17:30:03 UTC

R topics documented:

bankStat *Newman Banked Statistic*

Description

The Newman Banked Statistic is used to compare an individual sample to a cohort of similar samples.

Usage

```
bankStat(bankObj, testSet, bankMatrix)
createBank(bankMatrix)
```

Arguments

bankObj	Compressed representation of the cohort being compared to.
testSet	Matrix containing data from one or more individual samples to be compared to the bank.
bankMatrix	Data for the bank of "normal" or "untreated" or "baseline" control samples.

Value

A list containing two matrices: the `nu.statistics` and the `p.values`.

Examples

```
data(GSE6631)
HN <- as.matrix(log2(1 + GSE6631))
bankMatrix <- HN[,seq(1, ncol(HN), 2)] # odd columns are normal
testSet <- HN[, seq(2, 6, 2)]   # evn columns are tumor
bs <- bankStat(testSet = testSet, bankMatrix = bankMatrix)
summary(bs$nu.statistics)
summary(bs$p.values)
```

fitMix3 *Compute FDR from Three-Component Beta Mixture*

Description

Provides functions to fit a beta-mixture model to a set of p-values that has peaks at both zero and one, and to estimate false discovery rates.

Usage

```
fitMix3(datavec, forever=100, epsilon=0.001, print.level=0)
computeFDR(object, alpha)
computeCutoff(object, fdr)
```

Arguments

datavec	A numeric vector containing p-values.
forever	An integer; maximum number of iterations while fitting the mixture model.
epsilon	A real number; change in the log likelihood that should be used to terminate the model-fitting loop.
print.level	An integer; how much detail should nlm print while fitting the model.
object	An object of the MixOf3Beta class.
alpha	A real number between 0 and 1; the cutoff on the nominal p-value where the FDR should be computed.
fdr	A real number beteen 0 and 1; the targeted FDR value.

Details

We have observed empirically that the set of p-values obtained when computing the Newman paired test statistic often has peaks both at zero (representing genes of interest) and at one (representing "boring" genes that change much less than expected). We attribute the latter phenomenon to the fact that we use locally smoothed instead of gene-by-gene estimates of the standard deviation; genes whose SD is increased by the smoothing process contribute to the boring peak near one.

To estimate p-values in this context, we fit a three-component beta mixture model, combining (1) a right-peaked distribution Beta(L,1), (2) a left-peaked dfistribution Beta(1,M), and (3) a uniform distribution. Specfically, we look for models of the form

$$alpha * Beta(L, 1) + beta * Beta(1, M) + gamma * Beta(1, 1)$$

.

Model-fitting uses an expectation-maximization (EM) algorithm. In addition to the parameters mle=c(L,M) and psi=c(alpha,beta,gamma), we introduce a matrix Z of latent variables that indicate which distribution each point is likely to arise form. Z has three columns (one for each mixture component) and one row for each p-value; the entries in each row are nonegative and sum to one. The M-step of the algorithm uses the nlm optimization function to compute the maximum-likelihood mle values given psi and Z. The E-step first updates psi from the Z-matrix, and then updates the values of Z based on the current mle.

We are able to use the mixture distribution to compute the relationship between a cutoff on the nominal p-values and the false discovery rate (FDR).

Value

The model-fitting function, fitMix3, returns an object of the MixOf3Beta class.

The computeFDR function returns a real number in [0,1], the false discovery rate assiociated with the nominal cutoff.

The computeCutoff function returns a real number in [0,1], the cutoff required to achieve the desired FDR.

Examples

```
set.seed(98765)
ds <- c(rbeta(3000, 20, 1),
        rbeta(1000, 1, 40),
        runif(6000))
fit <- fitMix3(ds)
computeFDR(fit, 0.01)
computeCutoff(fit, 0.01)
computeFDR(fit, 0.0016438)
computeCutoff(fit, 0.05)
computeFDR(fit, 0.00702114)
```

MixOf3Beta-class *Class "MixOf3Beta"*

Description

Represents the results of fitting a beta-mixture model to a set of p-values that has peaks at both zero and one.

Details

Given a set of p-values (or any data on the interval [0,1]) that has peaks at both ends of the interval, we fit a three-componet mixture model. One component is uniform, and represents the expected distribution under the null hypothesis that nothing interesting is happening anywhere. The second component has the distribution Beta(1,M); this has a peak at zero and represents the features of interest. The final component has the distribution Beta(L,1). In the context of the Newman paired statistic, this represents genes or features whose variabilirt is smaller than the locally smoothed estimate of the standard deviation; we can think of these as "extra boring".

Creating Objects

In practice, users will use the `fitMix3` function to construct an object of the `MixOf3Beta` class. Hand construction is strongly discouraged.

Slots

`input`: A numeric vector containing the input p-values.

`mle`: A numeric vactor of length 2 containing the bea parameters `L` and M (in that order).

`psi`: A numeric vector of length three containing the mixing parameters, in the order (right-peak component, left-peak component, and uniform-component).

`Z`: A matrix of size N (number of features) by 3. This contains the latent indicator matrix. Each row corresponds to a gene or feature, and the entries show the proabbiltiy that the feature arose from the right, left, or uniform comnponent.

Methods

plot(x, y, ...) Plot the decompositon of the data into thre pieces.

hist(x, lcol = "red", breaks=101, ...) Plot a histogram of the p-values along with the fitted model of the distribution.

image(x) Plot a (sorted) image of the latent variable Z-matrix.

Author(s)

Kevin R. Coombes <krc@silicovore.com>

References

Abrams ZB, Joglekar A, Gershkowitz GR, Sinicropi-yao S, Asiaee A, Carbone DP, Coombes KR. Personalized Transcriptomics: Selecting Drugs Based on Gene Expression Profiles. Preprint.

See Also

pairedStat, NewmanPaired-class

Examples

```
set.seed(98765)
ds <- c(rbeta(3000, 20, 1),
        rbeta(1000, 1, 40),
        runif(6000))
fit <- fitMix3(ds)
image(fit, col=topo.colors(64))
hist(fit, col="skyblue", lcol="blue")
plot(fit)
```

---

newman-data *Datasets to Illustrate the Newman Tests*

Description

These data sets contain paired normal and tumor samples usewd to ilustrate te Newman pairted test and the Newman bank test.

Usage

```
data(LungPair)
data(GSE6631)
```

Format

LungPair is a data matrix containing normalized second generation sequencing data from The Cancer Genome Atlas (TCGA), with 20,531 row (genes) and 2 columns (samples). The first column contains data for the normal sample and the second column contains data for the tumor sample from the patient with barcode TCGA.38.4625.

GSE6631 is a data matrix containing normalized Affymetrix microarray data from paired head-and-neck cancer samples in the Gene Expression Omnibus set GSE6631. The matrix contains 200 rows (a rando subset of genes) and 44 columns (samples). The odd numbered colukmns are derived from normal mucosa; te even numbered columns are derived from paired tumor samples from the same patient.

Source

The full squamous cell lung cancer (LUSC) data from TCGA was downloaded from http://firebrowse.org/, and the data for this pair were separated and saved as a binary R data file. The head-and-neck cancer data were downloaded from the Gene Expression Omnibus at http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE6631. A subset of 2000 genes was randomly selected before saving the binary R data file.

References

Kuriakose MA, Chen WT, He ZM, Sikora AG et al. Selection and validation of differentially expressed genes in head and neck cancer. Cell Mol Life Sci 2004 Jun;61(11):1372-83.

Examples

```
data(LungPair)
data(GSE6631)
```

NewmanPaired-class *Class "NewmanPaired"*

Description

Represents the reusults of computing the Newman Paired test statistic on one or more paired samples.

Creating Objects

In practice, users will use the pairedStat function to construct an object of the NewmanPaired class. Hand construction is strongly discouraged.

Slots

`pairedMean`: A matrix of size N (number of features) by S (number of sample pairs). The mean expression of each feature in each paired sample. Also called "A" in the M-versus-A plots of the microarray era.

`difference`: A matrix of size N (number of features) by S (number of sample pairs). The difference (perturbed - base) in expression of each feature in each paired sample. Also called "M" in the M-versus-A plots of the microarray era.

`smoothSD`: A matrix of size N (number of features) by S (number of sample pairs). The results of fitting a loess smooth to the relationship between the `PairedMean` and the observed estimate of standard deviation (i.e., `abs(difference)/sqrt(2)`).

`nuStatistics`: A matrix of size N (number of features) by S (number of sample pairs). The Newman paired statistics, nu.

`pValues`: A matrix of size N (number of features) by S (number of sample pairs). Empirical p-values for the Newman statistics.

Methods

x[i,j] Select a subset of features or sample pairs.

dim(x) The dimension, N by S, of the object.

plot(x, y, which = NULL, ask = NULL, high = 0.99, low = 0.01, ...) Plot the results of the analysis of one sample pair.

hist(x, breaks=101, xlab="P-value", ...) Plot a histogram of the p-values for one sample-pair.

Author(s)

Kevin R. Coombes <krc@silicovore.com>

References

Abrams ZB, Joglekar A, Gershkowitz GR, Sinicropi-yao S, Asiaee A, Carbone DP, Coombes KR. Personalized Transcriptomics: Selecting Drugs Based on Gene Expression Profiles. Preprint.

See Also

`pairedStat`, `bankStat`

Examples

```
showClass("NewmanPaired")
```

`pairedStat` *Paired Newman Statistic*

Description

The Paired Newman Statistic is used for one-to-one comparison of paired individual samples. Commonly used to find differential expression between tumor-normal pairs or before-after treatment pairs.

Usage

```
pairedStat(baseData, perturbedData = NULL, pairing = NULL)
```

Arguments

`baseData`	Either a list or a matrix. May contain data for just the base condition (for example, normal samples or samples before treatment) or for both the base condition and the perturbed condition (for example, tumor samples or samples after treatment). See details.
`perturbedData`	An optional matrix containing data for the "perturbed" samples. May be NULL if the `baseData` argument is a list or a matrix containing all the data.
`pairing`	An optional vector indicating the pairing between base and perturbed samples. Entries must be integers. Positive integers indicate perturbed samples and negative integers with the same absolute value indicate the paired base samples. See details.

Details

In the simplest case, we have gene expression data on one "base" sample and one "perturbed" sample, and the goal is to identify genes whose expression changes between the two states. Our primary assumption is that the standard deviation (SD) of gene expression varies as a smooth function of the mean; fitting such a curve allows us to detect individual genes whose difference is large compared to the smoothed SD.

Note that this assumption is most useful on the log-transformed scale (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4120293/). If your data is on a raw scale, then we recommend transforming it before computing the Newman paired statistic.

The input arguments to the `pairedStats` function are moderately complicated in order to allow users to choose a convenient method for supplying data when they have multiple paired samples. The first posssibility is to have all the base samples in one matrix and all the perturbed samples in a second matrix. In this case, we assume (without checking) that the columns in the two matrices correspond to the paired samples, and that the genes-rows are in the same order.

The second possibility is to put the data for both the base samples and the perturbed samples in the same matrix. In this case, the user must supply a `pairing` vector to explain how the samples should be matched. If the column order is ("base1", "perturbed1", "base2", "perturbed2", ...), then the pairiing vector should be written as `c(-1,1,-2,2,-3,3,...)`.

The third possibility is to provide the paired samples in a list, each of whose entries is a matrix with two columns,with the first column being the base state and the second column being the perturbed state.

This flexibility means that there are three equivalent ways to input the data even if you have only one base sample (with data in the one-column matrix B) and one perturbed sample (with data in the one-column matrix P). If we let `BP <-cbind(B,P)` , then we can choose (1) `pairedStats(B,P)`, or (2) `pairedStats(list(BP))`, or (3) `pairedStats(BP,pairing = c(-1,1))`.

Value

A list containing two marices: the `nu.statistics` and the `p.values`.

Examples

```
data(GSE6631)
Normal <- GSE6631[, c(1,3)]
Tumor <- GSE6631[, c(2,4)]

### input two separate matrices
ps1 <- pairedStat(Normal, Tumor)
summary(ps1@nu.statistics)
summary(ps1@p.values)

### input one combined matrix and a pairing vector
ps2 <- pairedStat(GSE6631, pairing=c(-1, 1, -2, 2))
summary(ps2@nu.statistics)
summary(ps2@p.values)

### input a list of matrix-pairs
ps3 <- pairedStat(list(One = GSE6631[, 1:2],
                       Two = GSE6631[, 3:4]))
summary(ps3@nu.statistics)
summary(ps3@p.values)
```

Index