Package 'Gimp'

January 31, 2016

Type Package

Title Good-Turing and empirical Bayes pseudocounts for sparse count matrices

Version 0.5

Date 2016-01-29

Maintainer Who to complain to <liyang.diao@gmail.com>

Description 16S rRNA sequence data is oftentimes extremely sparse, making it difficult to normalize properly for differential abundance analyses with current RNA-Seq methods, such as DESeq, edgeR, and metagenomeSeq. The count adjustment methods included here can be used to ameliorate these effects. The Good-Turing adjustment is derived from from Good-Turing frequency estimation methods. Code to generate simulated test data according to the Dirichlet-multinomial model from two different sets of parameters is also included. Further information on the methods and data can be found *IN THIS PAPER*.

Imports HMP

Suggests dirmult

License GPL-3

RoxygenNote 5.0.1

LazyData true

NeedsCompilation no

Author Liyang Diao [aut, cre], Glen Satten [aut]

R topics documented:

`avg.transform` *Averaging transformation*

Description

Averaging transformation for a vector of feature counts. For more details, see "Good-Turing Smoothing Without Tears", by William A. Gale.

Usage

```
avg.transform(x, log.transform = TRUE)
```

Arguments

`x`	vector of feature counts
`log.transform`	whether or not log transform should be applied

`dm.fit.eso` *Esophageal microbiome data*

Description

A list with 5 objects, including `gamma`, the parameter vector for a Dirichlet distribution. Estimated from the esophageal data set described in Pei Z et al.: Bacterial biota in the human distal esophagus. Proc Natl Acad Sci U S A 2004, 101:4250-4255. Estimated using package `dirmult`.

Format

A list with five objects

Details

- gamma. Parameter vector for Dirichlet distribution
- pi. Mean vector for Dirichlet distribution
- loglik. Final log-likelihood value
- ite. Number of iterations used
- theta. Estimated theta-value

dm.fit.lung *Lung microbiome data*

Description

A list with 5 objects, including gamma, the parameter vector for a Dirichlet distribution. Estimated from the (non-smoker) lung data set described in Charlson ES et al.: Topographical continuity of bacterial populations in the healthy human respiratory tract. Am J Respir Crit Care Med 2011, 184:957-963. Estimated using package dirmult.

Format

A list with five objects

Details

- gamma. Parameter vector for Dirichlet distribution
- pi. Mean vector for Dirichlet distribution
- loglik. Final log-likelihood value
- ite. Number of iterations used
- theta. Estimated theta-value

eb.pseudo *empirical Bayes adjustment of sparse counts matrix*

Description

eb.pseudo performs empirical Bayes adjustment for sparse count matrix "counts", where each sample is independently adjusted.

Usage

```
eb.pseudo(counts)
```

Arguments

counts	Integer count matrix with features in rows and samples in columns

Details

Essentially a wrapper funtion for eb.pseudo.sample, which performs empirical Bayes adjustment for a feature counts vector for a single sample.

See Also

eb.pseudo.sample for adjustment of a single sample. See sim.groups for a simple example.

eb.pseudo.sample *empirical Bayes adjustment of sparse counts vector*

Description

Primarily a helper function for eb.pseudo, but can be used on its own to normalize a single sample

Usage

```
eb.pseudo.sample(x)
```

Arguments

x	Feature count vector

Value

List with objects z, psi, and mu

See Also

eb.pseudo

fac.moment *calculates the first two factorial moments of a numeric vector*

Description

Helper function for eb.pseudo.sample

Usage

```
fac.moment(x, return.scaled = TRUE)
```

Arguments

x	Integer count matrix with features in rows and samples in columns
return.scaled	Whether or not to return the scaled factorial moments

See Also

eb.pseudo.sample

gt.fit.lm — *Linear model for average transformed counts*

Description

Low-level function for smoothed Good-Turing frequency estimation. Fits a linear model to counts subjected to "averaging transformation" in order to obtain better frequency estimates for samples where no 1s are observed.

Usage

```
gt.fit.lm(x)
```

Arguments

x	vector of feature counts

See Also

avg.transform

gt.pseudo — *Good-Turing adjustment of sparse counts matrix*

Description

gt.pseudo performs Good-Turing adjustment for sparse count matrix "counts", where each sample is independently adjusted.

Usage

```
gt.pseudo(counts)
```

Arguments

counts	Integer count matrix with features in rows and samples in columns

Details

Essentially a wrapper funtion for gt.pseudo.sample, which performs Good-Turing adjustment for a feature counts vector for a single sample.

See Also

gt.pseudo.sample for adjustment of a single sample. See sim.groups for a simple example.

gt.pseudo.sample *Good-Turing adjustment of sparse counts vector*

Description

Primarily a helper function for `gt.pseudo`, but can be used on its own to normalize a single sample

Usage

```
gt.pseudo.sample(x)
```

Arguments

x	Feature count vector

See Also

gt.pseudo

pred.mass *Good-Turing frequency estimator*

Description

Estimates the mass which should be assigned to all features with count k in sample x.

Usage

```
pred.mass(x, k = 0)
```

Arguments

x	vector of feature counts
k	count value for which mass should be estimated

pred.mass.smooth *Simple Good-Turing frequency estimator, smoothed*

Description

Estimates the mass which should be assigned to all features with count k in sample x, smoothed by Gale's method (see "Good-Turing Smoothing Without Tears")

Usage

```
pred.mass.smooth(x, k = 0)
```

Arguments

x	vector of feature counts
k	count value for which mass should be estimated

sim.counts *Dirichlet-multinomial counts simulation*

Description

Simulates counts according to a Dirichlet-multinomial model, according to model parameters given by object dm.fit.

Usage

```
sim.counts(dm.fit, ls, remove.zero.rows = FALSE)
```

Arguments

dm.fit	dirmult object. Alternatively, a list object with at least gamma object, corresponding to parameters of a Dirichlet distribution
ls	vector of library sizes of samples to be generated
remove.zero.rows	whether or not to remove rows of all zeroes

Details

Simulates a single set of counts, generated according to the parameters in object dm.fit. Requires package dirmult to generate samples.

sim.groups *Dirichlet-multinomial case/control counts simulation*

Description

Simulates counts according to a Dirichlet-multinomial model, according to model parameters given by object dm.fit.

Usage

```
sim.groups(dm.fit, Nshuffle = 15, params = list(mu1 = 300, sd1 = 50, N1 =
  20, mu2 = 300, sd2 = 50, N2 = 20))
```

Arguments

dm.fit	dirmult object. Alternatively, a list object with at least gamma object, corresponding to parameters of a Dirichlet distribution
Nshuffle	number of features to shuffle
params	list of parameters describing the number the mean and standard deviation of library sizes for case and control samples, as well as the number of samples of each

Details

Simulates two sets of counts (case/control), where one set is generated according to the parameters in object `dm.fit`, and and the second set is generated such that `Nshuffle` number of high abundance features are shuffled with `Nshuffle` number of low abundance features, generating a final data set with a large number of differences between case and control samples for large enough `Nshuffle`. Requires package `dirmult` to generate samples.

Value

List with objects `counts`, `i.spike`, `groups`: a counts table, feature names which have been shuffled, and group membership (case/control), respectively

See Also

`sim.counts` for simulation of a single group of samples

Examples

```
library(HMP)
library(dirmult)
S <- sim.groups(dm.fit.eso)
S$eb <- eb.pseudo(S$counts)
S$gt <- gt.pseudo(S$counts)
round(head(S$eb),4)

S <- sim.groups(dm.fit.lung, Nshuffle = 50,
     params = list(mu1 = 2000, sd1 = 200, N1 = 20,
                   mu2 = 6000, sd2 = 600, N2 = 20))
S$eb <- eb.pseudo(S$counts)
S$gt <- gt.pseudo(S$counts)
round(head(S$gt),4)
```

Index