Package 'manta'

October 16, 2019

Version 1.30.0

Date 2012-03-15

Title Microbial Assemblage Normalized Transcript Analysis

Author Ginger Armbrust, Adrian Marchetti

Maintainer Chris Berthiaume <chrisbee@uw.edu>, Adrian Marchetti <amarchetti@unc.edu>

Depends R (>= 1.8.0), methods, edgeR (>= 2.5.13)

Imports Hmisc, caroline(>= 0.6.6)

Suggests RSQLite, plotrix

Description Tools for robust comparative metatranscriptomics.

License Artistic-2.0

URL http://manta.ocean.washington.edu/

biocViews ImmunoOncology, DifferentialExpression, RNASeq, Genetics, GeneExpression, Sequencing, QualityControl, DataImport, Visualization

git_url https://git.bioconductor.org/packages/manta

git_branch RELEASE_3_9

git_last_commit 282c99b

git_last_commit_date 2019-05-02

Date/Publication 2019-10-15

R topics documented:

`cmdArgsToVariables` *Create R variables from command line parameters*

Description

Take the names of the variables specified on the R CMD BATCH call and turn them into R variables Assign the values after '=' to these variables.

Usage

```
cmdArgsToVariables()
```

Value

variable(s) stored in memory

Examples

```
## Not run:
## R CMD BATCH --some.variable=testing
cmdArgsToVariables()
print(some.variable)
#> 'testing'

## End(Not run)
```

`collapseRepliCounts` *Collapse multiple technical replicate count columns into two columns for plotting*

Description

.

Usage

```
collapseRepliCounts(x, pair=nv(levels(x$samples$group)[1:2] , c('ref','obs')))
```

Arguments

x	a MANTA object.
pair	the pairs indicating which columns to collapse

Value

a collapsed, two-column count table

See Also

DGEList, manta

Examples

```
cts <- matrix(data=rnbinom(28,2,.4), ncol=4, nrow=7)

colnames(cts) <- apply(expand.grid(c('a','b'),1:2), 1, paste, collapse='_')

x <- manta(cts, makeSampleDF(cts, group=rep(c('a','b'),2)))
collapseRepliCounts(x, pair=c('a','b'))
```

---

compbiasPlot *plot the compositional bias for the sub-taxinomic rank*

Description

.

Usage

```
compbiasPlot(x, pair=nv(levels(x$samples$group)[1:2] , c('ref','obs')), meta.lev='phylum', meta.l
```

Arguments

x	a MANTA object.
pair	named vector of the pair of conditions of interest
meta.lev	the taxinomic rank to plot.
meta.lev.lim	the taxinomic rank for diversity measure.
breaks	breaks for the histogram.
xlim	x axis limits.
type	type of plot.
col	colors
...	additional params.

Value

a compositional bias plot

See Also

DGEList, manta

Examples

```
manta.path <- system.file("extdata","PapaGO-BLAST.results-diatoms.Rdata", package="manta")
load(manta.path)

compbiasPlot(x, meta.lev='genus_sp')
```

---

compbiasTest *compositional bias test*

Description

.

Usage

```
compbiasTest(x, pair=nv(levels(x$samples$group)[1:2] , c('ref','obs')), meta.lev='phylum', meta.l
```

Arguments

x	a MANTA object.
pair	named vector of the pair of conditions of interest
meta.lev	which taxinomic level should this test be run at
meta.lev.lim	how many underlying taxinomic levels should the analysis be limited to

Value

A DGEList object.

See Also

DGEList, manta

Examples

```
manta.path <- system.file("extdata","PapaGO-BLAST.results-diatoms.Rdata", package="manta")
load(manta.path)

compbiasTest(x, meta.lev='genus_sp')
```

generateWeights *Generate Weights*

Description

Because the manta plot uses integer count data, many of the points overlap and hide a large portion of the data. This function allows one to apply a weighting scheme to jitter points out from under each other both to show the density and expose the content of their pies (if applicable).

Usage

```
generateWeights(x, w.clmn,  agg.clmn, cond.clmn, ct.clmns=NULL)
```

Arguments

x	an alignment dataframe
w.clmn	a string corresponding to a column with weight data
agg.clmn	a string corresponding to a column with an aggregation index.
cond.clmn	a string corresponding to a column with condition factor
ct.clmns	a string corresponding to a column with count data

Value

a 2 by n weight matrix

See Also

manta

Examples

```
align.path <- system.file("extdata","PapaGO-BLAST.results-diatoms.tab", package="manta")
a <- read.delim(align.path, stringsAsFactors=FALSE)
w <- generateWeights(a, 'what_e_value', 'what_def', 'treatment')
```

in2manta *Convert a count or alignment table into a MANTA object*

Description

This function converts a table of alignment data (BLAST for example) where each record is a read and each column is some parameter of the blast(s). The function will perform a cross-tabulation of the annotated reads into count data using (at the very least) an aggregation index column and a condition column. Alternatively, the input can be pre-tabulated counts and a corresponding annotation table. The two tableMeta helper functions (called by the x2manta functions) are also documented here.

Usage

```
align2manta(x, cond.clmn, agg.clmn, gene.clmns, meta.clmns, weight.clmn=NULL, tag.clmn=NULL, ...)
counts2manta(x, annotation, a.merge.clmn, agg.clmn, gene.clmns=NULL, meta.clmns=NULL, ...)
tableMetaSums(x, meta.clmns, cond.clmn=NULL, count.clmns=NULL)
tableMetas(x, agg.clmn, meta.clmns, cond.clmn=NULL, count.clmns=NULL)
```

Arguments

x	The alignment or counts table.
cond.clmn	A string indicating which column contains the conditions. (only two different levels two allowed)
agg.clmn	A string indicating which column in the annotation table contains the aggregation index.
annotation	An annotation table with genes and/or meta information.
a.merge.clmn	A stringn indicating which column in the annotation table on which to merge. These should correspond to the row names of the counts table.
gene.clmns	A vector of strings indicating which column contains gene annotation information.
meta.clmns	A vector of strings indicating which column contains meta/taxinomic information.
weight.clmn	A string indicating which column contains weighting information.
count.clmns	A string indicating which column in a pre-cross-tabulated meta annotation table contains counts. These counts are multiplied by the result of tableMetas()'s cross-tabulated counts to genereate the meta tables.
tag.clmn	A string indicating which column corresponds to individual read names. Only necessary when a single read has multiple annotation records in the table.
...	additional parameters passed along to manta()

Value

A MANTA object

See Also

manta

Examples

```
align.path <- system.file("extdata","PapaGO-BLAST.results-diatoms.tab", package="manta")
a <- read.delim(align.path, stringsAsFactors=FALSE)
x <- align2manta(a, cond.clmn='treatment', agg.clmn='what_def',
gene.clmns=c('what_def','kid','pathway'),
meta.clmns=c('family','genus_sp'))



cts.path <- system.file("extdata","PapaGO-BWA.counts-diatoms.tab", package="manta")
cts <- read.delim(cts.path)
cts.annot.path <- system.file("extdata","PapaGO-BWA.annot-diatoms.tab", package="manta")
cts.annot <- read.delim(cts.annot.path, stringsAsFactors=FALSE)
```

```
x <- counts2manta(cts, annotation=cts.annot,
             a.merge.clmn='query_seq', agg.clmn='what_def', meta.clmns=c('family','genus_sp'),
                    gene.clmns=c('what_def','kid','pathway'))
```

`makeSampleDF` *Make a Sample Dataframe for use in Initializing a MANTA object*

Description

The sample dataframe contains a row for each sample with a factor indicating grouping and library sizes.

Usage

```
makeSampleDF(counts, group=factor(colnames(counts)), lib.size=colSums(counts))
```

Arguments

`counts`	first number
`group`	a factor specifying which of each of count columns belong to each of the two conditions.
`lib.size`	the sizes (cumulative counts) of each of the libraries.

Value

a sample dataframe

See Also

DGEList, manta, setLibrarySizes

Examples

```
cts.path <- system.file("extdata","PapaGO-BWA.counts-diatoms.tab", package="manta")
cts <- read.delim(cts.path)
sdf <- makeSampleDF(counts=cts)
```

manta *Create a MANTA object*

Description

The MANTA object contains counts, genes, library information just like a EdgeR's DGEList. Additionally, however, it contains 'meta' annotation (typically taxinomic classifications). This function converts all of listed component elements into a MANTA object.

Usage

```
manta(counts, samples=makeSampleDF(counts), genes=NULL, meta=NULL, meta.sum=NULL, weights=NULL, no
```

Arguments

counts	A numeric matrix containing the read counts. Rows should be named by a unique gene identifier.
samples	The experimental sample dataframe (nearly identical to the one in a DGEList object).
weights	A numberic matrix of count weights for each count. Should be the same dimentions as the count table.
genes	A dataframe of genes annotations that have corresponding count data.
meta	A taxinomic level list of gene lists of cross tabulations of taxinomic (meta) annotations for genes that have corresponding count data.
meta.sum	A list of aggregated counts for (one per taxinomic level).
norm	boolean specifiying if manta should automatically normalize using calcNormFactors().
disp	boolean specifying if manta should automatically estimate the common dispersion via estimateCommonDisp().
...	additional parameters passed to DGEList

Value

A MANTA object.

See Also

DGEList

Examples

```
cts.path <- system.file("extdata","PapaGO-BWA.counts-diatoms.tab", package="manta")
cts <- read.delim(cts.path)
samples <- makeSampleDF(cts)

x <- manta(counts= cts, samples = samples)
```

manta-class *Microbial Assemblage Normalized Transcript Analysis - class*

Description

A simple list-based class for storing read counts from digital gene expression technologies and other important information for the analysis of (meta)transcriptomic data.

Slots/List Components

Objects of this class contain (at least) the following list components:

`counts`: numeric matrix containing the read counts.

`samples`: data.frame containing the library size and group labels.

Additionally the class should contain the following meta/taxinomic information list components:

`meta`: numeric matrix containing the read counts.

`meta.sum`: data.frame containing the library size and group labels.

Also, the list could also contain further optional information:

`genes`: data.frame containing further gene annotation information for each row in counts.

Methods

This class inherits directly from class `list` so any operation appropriate for lists will work on objects of this class. `manta` objects also have a `show` method.

See Also

`manta`

meta2counts *Convert a manta object's meta slot data into counts*

Description

This is a helper function for the mantaMethod

Usage

```
meta2counts(obj, meta.lev, rm.sum=TRUE, meta.subset=NULL)
```

Arguments

`obj`	A manta object.
`meta.lev`	The taxinomic rank level (phylum, genus, etc) from which to pull counts.
`rm.sum`	Whether or not to remove the overall 'sum' (over all conditions) column.
`meta.subset`	Which sub rank level (the subset) for which to estimate the normalization. By default is does the overall normalization

Value

a count matrix

See Also

manta, mantaMethod

Examples

```
manta.path <- system.file("extdata","PapaGO-BLAST.results-diatoms.Rdata", package="manta")
load(manta.path)

tab <- meta2counts(x, meta.lev='genus_sp', meta.subset='Pseudo-nitzschia granii')
```

---

metataxa2subcounts *create a new (sub) count table out of a subcomponent of the metatranscriptome*

Description

A simple and easy way to pull a subcomponent transcriptome out of the metatranscriptome. This one line function is useful for those wanting to just see the count data for one species or to create a new DGE or manta object on a subset of data.

Usage

```
metataxa2subcounts(x, meta.lev='species', taxa.filter)
```

Arguments

x	a manta object.
meta.lev	the name of the taxinomic rank on which to subset.
taxa.filter	the name of the taxonomy at the meta.lev rank on which to subset.

Value

a count table

See Also

manta

Examples

```
load(system.file('extdata','PapaGO-BLAST.results-diatoms.Rdata',package='manta'))
metataxa2subcounts(x, meta.lev='species', taxa.filter='Pseudo-nitzschia granii')
```

nf2nr *convert the normalization factors to a normalization line*

Description

.

Usage

```
nf2nr(x, pair, method='nf', absolute=TRUE)
```

Arguments

x	a MANTA object.
pair	The two dimentions (conditions) which you wish to convert.
method	.
absolute	.

Value

a scalar normalization ratio

See Also

DGEList, manta, calcNormFactors

Examples

```
manta.path <- system.file("extdata","PapaGO-BLAST.results-diatoms.Rdata", package="manta")
load(manta.path)

nf2nr(x)
```

normfact2absTMM *Convert a count or alignment table into a MANTA object*

Description

This function converts a table of alignment data (BLAST for example) where each record is a read and each column is some parameter of the blast(s). The function will perform a cross-tabulation of the annotated reads into count data using (at the very least) an aggregation index column and a condition column. Alternatively, the input can be pre-tabulated counts and a corresponding annotation table. The two tableMeta helper functions (called by the x2manta functions) are also documented here.

Usage

```
normfact2absTMM(x, pair, f=nv(x$samples, 'norm.factors'), sums=colSums(x$counts))
```

Arguments

x	The manta object (can be NULL if f and sums are specified independently).
pair	A named vector indicating the condition names and the corresponding the reference or observation status of each.
f	The Normalization factors. By default uses the norm.factors column of the sample dataframe in x.
sums	The column sums of the counts. By default uses the column sums of the x$counts table.

Value

a scalar normalization factor

See Also

manta, mantaMethod

Examples

```
conditions <- caroline::nv(factor(x=1:2, labels=c('ambient','plusFe')) ,c('ref','obs'))
manta.path <- system.file("extdata","PapaGO-BLAST.results-diatoms.Rdata", package="manta")
load(manta.path)

x$mm <- normfact2absTMM(x=x, pair=conditions)
```

---

nr — *Print out all the normalization ratios for each subset in a specified taxinomic rank of a manta object*

Description

.

Usage

```
nr(obj, meta.lev, pair)
```

Arguments

obj	A manta object.
meta.lev	which taxinomic rank to use for subset normalization estimation.
pair	A named vector of conditions factors

Value

table of normalization ratios

See Also

manta, meta2manta

Examples

```
manta.path <- system.file("extdata","PapaGO-BLAST.results-diatoms.Rdata", package="manta")
load(manta.path)
x <- calcNormFactors(x)

conditions <- factor(x=1:2, labels=c('ambient','plusFe')); names(conditions) <- c('ref','obs')

nr(x, meta.lev='genus_sp', pair=conditions)
```

---

outGenes *find the most significant or highest fold change outlier genes*

Description

.

Usage

```
outGenes(x, n=50, p=.05, FC=1, A.pct=.05, uk.filter=NULL, method='BH', verbose=TRUE)
```

Arguments

x	a MANTA object.
n	number of genes to return.
p	adjusted p-value cut-off.
FC	fold change cut-off.
A.pct	percentage of genes that should be right most outliers.
uk.filter	a character vector of unknown gene name paterns to filter out (eg 'lcl').
method	multiple testing correction method from p.adjust
verbose	should the function print these results (or just return the table)

Value

a table of the outlier genes

See Also

topTags

Examples

```
manta.path <- system.file("extdata","PapaGO-BLAST.results-diatoms.Rdata", package="manta")
load(manta.path)

de <- exactTest(x)
outGenes(de)
```

`plot.manta` *Plot a MAnTA object*

Description

A MANTA RAy plot is designed for visualizing comparative meta-transcriptomics, but can be used in any case where fold change data assessment calls for displaying additional meta information.

Usage

```
manta.ra(x, uniques=4, pair=nv(levels(x$samples$group)[1:2] , c('ref','obs')),
                       nr=0, alpha = 0.01, normalize=FALSE,
                meta.level=names(x$meta.sum)[1], meta.lgnd.lim=6, lgd.pos='topright', lgd.cex=.75,
                       annot=NULL, vrb.axlabs=TRUE, jitter=.43, border='black',
                       rex=2, flat=FALSE, tail=.5, arms=.5, spine=1, ...)
```

Arguments

`x`	a MAnTA object.
`uniques`	a boolean specifying whether or not to plot the library-unique genes (those with zero counts in one or the other library).
`pair`	a named vector indicating the condition names and the corresponding the reference or observation status of each.
`nr`	a numeric value indicating the asymptotic normalization ratio line.
`alpha`	a statistical significance value.
`normalize`	A boolean specifying whether or not to normalize the counts into proportions.
`meta.level`	a number or string specifies which taxonomic level (which meta list element) should be used to generate the pie charts. 0 or FALSE indicates no pies should be drawn.
`meta.lgnd.lim`	an integer specifying the top n most common taxonomic levels to use in the pies and legend.
`lgd.pos`	a string specifying the general position of the legend (see the legend function documentation).
`lgd.trunk`	a boolean specifying if the legend should trunkate (for example) Genus species to G.species
`lgd.cex`	a numeric value specifying the character expansion for the legend text.
`pie.lwd`	the line thickness of the pie polygon border.

annot	a named vector of strings that match which points should be annotated in the plot. names indicate the colors of the text.
vrb.axlabs	a boolean indicating if verbose axis labels should be use to spell out exactly how the axes are calculated.
jitter	whether or not or how much to jitter the counts into surrounding, non-overlapping space.
rex	a numeric value specifying the radial expansion of the plot points.
flat	a boolean for the radial expansion of points as a function of both R and A axes.
tail	a numeric or boolean value indicating the line thickness of the two trailing curved significance lines of the RAy.
arms	a numeric or boolean value indicating the line thickness of the two leading straight separator lines of the RAy.
spine	a numeric or boolean value indicating the line thickness of the normalization line (whose y position is specified by mm).
border	a vector of strings used to color the borders of the points.
...	other parameters passed to plot.

Value

A MAnTA RAy plot.

See Also

maPlot, plotMA, raPlot, pies

Examples

```
manta.path <- system.file("extdata","PapaGO-BLAST.results-diatoms.Rdata", package="manta")
load(manta.path)
plot(x, meta.lev='genus_sp')
```

---

pplacer2manta *convert a pplacer taxinomic placement repository to a MANTA object*

Description

This function creates a single manta object by traversing a directory of directories of pplacer SQLite taxonomy databases (where each database called "taxtable.db" resides in a sub folder named by a single gene locus).

Usage

```
pplacer2manta(dir, group.pattern='_([[:alpha:]]+)_',
  groups=c('coastal','costal','DCM','surface','upwelling'),
uk.name='unknown', ...)
```

Arguments

dir	directory of gene directories each with a single gene taxonomic read placement database.
group.pattern	the regular expression used to parse out the name of the different conditions/groups from the read names.
groups	the actual list of groups.
uk.name	the rowname to appear in each othe meta tabulation tables for the unplaceable reads.
...	additional parameters passed along to manta()

Value

A MANTA object

See Also

manta, in2manta

Examples

```
KOG.SQLite.repo <- system.file('extdata','pplacer',package='manta')
pplacer2manta(dir=KOG.SQLite.repo,
        groups=c('coastal','costal','DCM','surface','upwelling'),
        norm=FALSE, disp=FALSE
       )
```

---

readSeastar *Read SEAStAR output format*

Description

This function reads Vaughn Iverson's SEAStAR tabular format output and appends headers to it.

Usage

```
readSeastar(path,
clmn.names=c('seq_id','bit_score','read_count','raw_abundance','fractional_abundance','mean_cove
clmn.class = c("character", rep("numeric", 6), "integer", "numeric", rep("character", 2)),
name.clmn='seq_id', ret.df=FALSE, ret.clmn='read_count', ct.calc=expression(raw_abundance*seq_len
```

Arguments

path	Path to the seastar file
clmn.names	The feild names for the absent header column
clmn.class	The class types for each of the columns
name.clmn	the seastar column to be used to name the rows of the dataframe
ret.df	Return the entire dataframe rather than just the calculated count column

ret.clmn	If ret.df is FALSE, return just this column as a vector. Can be 'count' as calculated by 'ct.calc'
ct.calc	the equation to use to calculate the counts
header	If the seastar tables have headers
...	Additional parameters passed on to read.delim

Value

a SEAStAR formated matrix of per-reference/contig/gene stats (including counts)

See Also

seastar2counts

Examples

```
conditions <- c('ambient','plusFe')
ss.names <- caroline::nv(paste('Pgranii-',conditions,'.seastar', sep=''), conditions)

ss.paths <- system.file("extdata",ss.names, package="manta")

df <- readSeastar(ss.paths[1])
```

---

seastar2counts *Convert seastar output to count data*

Description

This function performs a simple merge between two different SEAStAR tables.

Usage

```
seastar2counts(treat.paths, id.prefix=NA, all=TRUE, uniques=0, ...)
```

Arguments

treat.paths	a named vector of strings of paths to 2 seastar tabular files.
id.prefix	prefix to use when naming the rownames of the merged table
all	used in the same way as the underlying merge function's all parameter.
uniques	determines the replacement value for genes which are unique to the opposite library.
...	additional paramters passed on to readSeastar

Value

a named count matrix or vector

See Also

nerge, merge

Examples

```
conditions <- c('ambient','plusFe')
ss.names <- caroline::nv(paste('Pgranii-',conditions,'.seastar', sep=''), conditions)

ss.paths <- caroline::nv(system.file("extdata",ss.names, package="manta"), conditions)

dfm <- seastar2counts(ss.paths)
```

`summary.manta` *Summarize a MANTA object*

Description

Currently this funciton merely dumps the contents of the meta.sums tables to screen if available.

Usage

```
summary.manta(object, ...)
```

Arguments

`object`	a MANTA object.
`...`	additinal arguments

Value

A MANTA summary printout.

See Also

manta

Examples

```
manta.path <- system.file("extdata","PapaGO-BLAST.results-diatoms.Rdata", package="manta")
load(manta.path)
summary(x)
```

Index