

Package ‘octopus’

October 12, 2018

Type Package

Version 0.3.7

Encoding UTF-8

Date 2018-05-21

Title Tools for GLM NB analysis

Author Wei Zhang [aut, cre],
Jason J. Corwin [aut],
Daniel J. Kliebenstein [aut],
Guocai Chen [ctb]

Maintainer Wei Zhang<vivianzhang317@gmail.com>

Depends R (>= 3.3.2), Rbowtie, MASS, lsmeans, R.utils, digest, tools, edgeR

Description Tools for GLM NB analysis

License GPL-2

LazyData TRUE

RoxygenNote 6.1.0

R topics documented:

auto.library	2
create.info	3
octopus.glm.nb	4
octopus.normalize	6
octopus.short_reads	6
slice	9
tutorial	10
Index	12

auto.library	<i>Load package, install automatically if missing.</i>
--------------	--

Description

Load package, install highest version
from **bioconductor**
or **rstudio** if missing.

Usage

```
auto.library(package)
```

Arguments

package	the name of a package
---------	-----------------------

Value

invisible TRUE

Examples

```
{  
  auto.library(stringr) #use package name  
  auto.library("stringr") #use package name string  
  auto.library("package_no_exist") # Print warning message when package is not found
```

```
tutorial("auto.library") # print tutorial to console
```

```
}
```

create.info	<i>Print info with timestamp</i>
-------------	----------------------------------

Description

Print debug message, progress reports... with timestamp.

- Default on
- Turn off by `info.enable(FALSE)`
- Turn on by `info.enable(TRUE)`

Usage

```
create.info(prefix = "")
```

Arguments

prefix prefix of message.

Value

object of class `info`

Examples

```
{
  info.self <- create.info() #create object of info class
  print(info.self,"print message without prefix.")

  info.self <- create.info("tutorial.info") #create object of info class with "tutorial.info" as prefix
  print(info.self,"print message with prefix, use identifier for prefix usually.")

  # turn it off
  info.enable(FALSE) #turn information printing off all together
  print(info.self,"will print nothing from now on.")
  print(info.self,"conform print nothing.")

  # turn it on
  info.enable(TRUE) #turn information printing on all together
  print(info.self,"start printing again.")
}
```

```
tutorial("info") # print tutorial to console
```

```
}
```

octopus.glm.nb

Fit a Negative Binomial Generalized Linear Model

Description

Fit glm.nb , add anova to result, add lsm to result.

Before fitting, data and factors will be merged by row.names, mismatching rows will be removed.

If formula start with ~, will fit model with each data column of data as respond y.

Usage

```
octopus.glm.nb(formula, data, factors, specs4lsmeans = NULL)
```

Arguments

formula	formula
data	normalized reads, expecting gene as column, sample as row
factors	factors, expecting factor name as column, sample as row
specs4lsmeans	specs for lsmeans

Value

model if formula has respond y; flattened anova table, lsm table and se table if formula starts with ~.

Examples

```

{
  data(sample_keys) # facotrs
  for(cur_col in colnames(sample_keys)){ sample_keys[,cur_col] <- as.factor(sample_keys[,cur_col]) }

  data(sample_reads) # reads
  # reads normalization
  norm.reads <- octopus.normalize(sample_reads)

  print(row.names(sample_keys))
  print(row.names(norm.reads))
  # transpose it to meet input requirements of founction octopus.glm.nb()
  norm.reads <- data.frame(t(norm.reads))
  print(row.names(norm.reads))

  # fit model with first gene column, column name is Bcin01g00040.1
  result <- octopus.glm.nb(Bcin01g00040.1 ~ Experiment + Experiment/GrowingFlat + Experiment/GrowingFlat/AgarFlat
                           ,norm.reads
                           ,sample_keys)

  print(result$anova)

  # calculate LSMeans
  result$lsmeans <- lsmeans(result,~ Isolate | HostGenotype)
  summary(result$lsmeans)

  # fit model and pull LSMeans in one call
  result <- octopus.glm.nb(Bcin01g00040.1 ~ Experiment + Experiment/GrowingFlat + Experiment/GrowingFlat/AgarFlat
                           ,norm.reads
                           ,sample_keys
                           ,specs4lsmeans = ~ Isolate | HostGenotype)

  print(result$anova)
  summary(result$lsmeans)

  # iterate through all data
  result <- octopus.glm.nb( ~ Experiment + Experiment/GrowingFlat + Experiment/GrowingFlat/AgarFlat + Isolate + HostGenotype
                           ,norm.reads
                           ,sample_keys
                           ,specs4lsmeans = ~ Isolate | HostGenotype)

  print(result$anova)
  print(result$lsmeans)
  print(result$se)
}

```

octopus.normalize	<i>Normalize</i>
-------------------	------------------

Description

NA will be treated as 0

Usage

```
octopus.normalize(x, dist_dir = "results/")
```

Arguments

`x` number matrix, expecting gene as row, sample as column
`dist_dir`

Value

number matrix

See Also

[octopus.glm.nb](#)

Examples

```
tutorial("octopus.glm.nb") # to print tutorial
```

octopus.short_reads	<i>Extract Short Reads</i>
---------------------	----------------------------

Description

Extract short reads from seq_files using **bowtie**.

Output 2 files, dist_dir/orig.reads.csv and dist_dir/orig.reads.info.csv

Remove dist_dir/reusing/ folder if don't want reuse data from that folder

Setup ./tmp/ as an RAM disk folder will avoid lots of disk IOs, speed things up and protect you SSD.

Example for linux :

- cd to curent folder.

- create ram disk in console : `mount -t tmpfs -o size=4g tmpfs ./tmp/`

Example for mac :

- `cd` to curent folder.
- `mkdir -p tmp`
- `sudo mount -t tmpfs -o size=4096M tmpfs ./tmp/`

For windows, there are a number of RAM disk softerwares you can use.

Usage

```
octopus.short_reads(seq_files, references, ..., type = "single",

dist_dir = "results/")
```

Arguments

seq_files	Sequencing files , accepts .fastq or .gz format for files.
	If seq_files is a List or Vector, index of seq_files is assumed to be sampe names of sequencing files.
	If seq_files is a data.frame, row.names is assumed to be sampe names of sequencing files, the first column is assumed to be sequencing files, when type is paired the second column is assumed to be the second mate pair sequences.
references	A comma-separated list of FASTA files containing the reference sequences to be aligned to
...	Additional arguments to be passed on to the binaries. See ... of bowtie
type	Could be one of c("single", "paired", "crossbow").
	If single, the input sequences are interpreted as single reads.
	If paired, they are supposed to be mate pair reads.
	If crossbow, they are considered to be Crossbow-style reads.
dist_dir	folder for result file orig.reads.csv and orig.reads.info.csv

Examples

```
{
  # single
  references <- "seq_data/cdna/Arabidopsis_thaliana.TAIR10.25.cdna.all.fa"
  seq_files <- data.frame(seq_file=c("seq_data/1_AACGTGAT_L003_R1_001.fastq.gz", "seq_data/1_AACGTGAT_L007_R1_001.fastq.gz"),
    ,sample_name=c("sample1", "sample2", "sample3")
    ,stringsAsFactors = FALSE)
  row.names(seq_files) <- seq_files$sample_name

  octopus.short_reads(seq_files,references
    ,p=3 # number of alignment threads to launch
    ,`phred33-quals`=TRUE # input quals are Phred+33
    ,t=TRUE # print wall-clock time taken by search phases
    ,quiet=TRUE # print nothing but the alignments
    ,trim5=10 # trim <int> bases from 5' (left) end of reads
  )

  # paired

  references <- "seq_data/cdna/Arabidopsis_thaliana.TAIR10.25.cdna.all.fa"
  seq_files <- data.frame(seq_file=c("seq_data/A9_S1_L001_R1_001.fastq.gz", "seq_data/xxx_R1_001.fastq.gz", "seq_data/xxx_R2_001.fastq.gz"),
    ,seq_file_pair=c("seq_data/A9_S1_L001_R2_001.fastq.gz", "seq_data/xxx_R2_001.fastq.gz", "seq_data/xxx_R1_001.fastq.gz"),
    ,sample_name=c("sample1", "sample2", "sample3")
    ,stringsAsFactors = FALSE
  )
  row.names(seq_files) <- seq_files$sample_name

  octopus.short_reads(seq_files,references
    ,p=3 # number of alignment threads to launch
    ,`phred33-quals`=TRUE # input quals are Phred+33
    ,t=TRUE # print wall-clock time taken by search phases
    ,quiet=TRUE # print nothing but the alignments
    ,trim5=10 # trim <int> bases from 5' (left) end of reads
    # ,y=TRUE # more sensitive but much slower, see http://bowtie-bio.sourceforge.net/manual.shtml#bowtie2
    ,type="paired"
    ,dist_dir="results_paired/"
  )

  # multiple referencing files
  references <- c("seq_data/ref_sequences/Botrytisfusarivirus1.txt", "seq_data/ref_sequences/BotrytisHypovirus1.txt")
  seq_files <- data.frame(seq_file=c("seq_data/1_AACGTGAT_L003_R1_001.fastq.gz", "seq_data/1_AACGTGAT_L007_R1_001.fastq.gz"),
    ,sample_name=c("sample1", "sample2", "sample3")
    ,stringsAsFactors = FALSE)
  row.names(seq_files) <- seq_files$sample_name

  octopus.short_reads(seq_files,references
    ,p=3 # number of alignment threads to launch
    ,`phred33-quals`=TRUE # input quals are Phred+33
    ,t=TRUE # print wall-clock time taken by search phases
```



```
        ,quiet=TRUE # print nothing but the alignments
        ,trim5=10 # trim <int> bases from 5' (left) end of reads
    )
}
```

slice*Slice list into sublists by n*

Description

Slice list into sublists by n

Usage

```
slice(x, n)
```

Arguments

x	list
n	desired length of sublist

Value

A list of sublists

Examples

```
{
  x <- 5:505
  slice(x,100) # 5 sublists
  slice(x,99)  # 6 sublists
  slice(x,101) # 4 sublists
}
```

```
tutorial("slice") # print tutorial to console
```

```
}
```

tutorial	<i>Print Tutorial</i>
----------	-----------------------

Description

Print tutorial for function/class and return tutorial code as a function. It is a generic function.

Usage

```
tutorial(topic)
```

Arguments

topic	tutorial topic, it is usually a function name.
-------	--

Value

tutorial code as a function

Examples

```
{
  # access tutorial code for class info
  tutorial("info")      # use class name string
  tutorial(create.info()) # use object

  tutorial(create.info)  # use function name
  tutorial("create.info") # use function name string

  tutorial_func <- tutorial("info")
  tutorial_func() # run tutorial code, suggest read it first
```

```
tutorial("tutorial") # print tutorial to console
```

```
}
```

Index

`auto.library`, [2](#)

`bowtie`, [7](#)

`create.info`, [3](#)

`octopus.glm.nb`, [4](#), [6](#)

`octopus.normalize`, [6](#)

`octopus.short_reads`, [6](#)

`slice`, [9](#)

`tutorial`, [10](#)