Package ‘metaFunction’

June 20, 2014

Type Package

Title A package for accurately estimating the relative abundance for putative functions in a metagenomic sample

Version 1.0

Date 2014-06-19

Author Lingling An

Maintainer Lingling An <anling@email.arizona.edu>

Description metaFunction is a powerful tool in accurate profiling functions in a metagenomic sample. It allows multiple-function assignment for a gene sequence (usually short read) and also provides visualization plot for estimated relative abundance of functions/subsystems. In addition, metaFunction provides statistical inference (confidence intervals) for the relative abundances.

License None

R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>metaFunction-package</td>
<td>2</td>
</tr>
<tr>
<td>bootstrap</td>
<td>2</td>
</tr>
<tr>
<td>ExampleData</td>
<td>3</td>
</tr>
<tr>
<td>LowHigh.B</td>
<td>4</td>
</tr>
<tr>
<td>LowHigh.NP</td>
<td>4</td>
</tr>
<tr>
<td>mismatch</td>
<td>5</td>
</tr>
<tr>
<td>plot.abundance</td>
<td>5</td>
</tr>
<tr>
<td>preprocess</td>
<td>6</td>
</tr>
<tr>
<td>refseq</td>
<td>7</td>
</tr>
<tr>
<td>subs</td>
<td>7</td>
</tr>
<tr>
<td>Twosteps</td>
<td>8</td>
</tr>
</tbody>
</table>

Index 9
**metaFunction-package**

*metaFunction is a package for accurately estimating the relative abundance for putative functions in a metagenomic sample. What the package does (short line)*

**Description**

metaFunction is a powerful tool in accurate profiling functions in a metagenomic sample. It allows multiple-function assignment for a gene sequence (usually short read) and also provides visualization plot for estimated relative abundance of functions/subsystems. In addition, metaFunction provides statistical inference (confidence intervals) for the relative abundances.

**Details**

- **Package:** metaFunction
- **Type:** Package
- **Version:** 1.0
- **Date:** 2014-06-19
- **License:** What license is it under?

**Author(s)**

Lingling An

Maintainer: Lingling An <anling@email.arizona.edu>

**Usage**

`bootstrap(original, boot.n, epsilon, method)`

**Arguments**

- **original**
  - The preprocessed data from blast output.
- **boot.n**
  - Bootstrap size, i.e., how many resamplings need to do?
- **epsilon**
  - Small probability for controlling the multiple function assignment, see the function "Twosteps".
- **method**
  - Two methods for calculating the confidence intervals: non-parametric way and Bonferroni method.

**Details**

Two methods are provided for calculating the confidence interval based on the bootstrap results: 1) non-parametric way, i.e, no correction for multiple/simultaneous confidence intervals. 2) Bonferroni method for correction of simultaneous confidence intervals.
ExampleData

Examples

```r
library(metaFunction)
require(data.table)
require(mefa)
require(lattice)

data(ExampleData)
data(refseq)
data(subs)

pre=preprocess(ExampleData)
epsilon=0.01
est=Twosteps(pre, epsilon)

bootsiz=100
boot.data=bootstrap(pre, bootsiz, epsilon, method="B")
```

ExampleData  Example Data

Description

An example dataset is provided for checking the functions in the metaFunction package.

Usage

```r
data(ExampleData)
```

Format

A data frame with 651661 observations on the following 19 variables.

Details

Blastx output which is generated from format 7. Other formats can be used too after modify a little on the names of input data in the function of "preprocess".

Examples

```r
data(ExampleData)
## maybe str(ExampleData).
```
LowHigh.B  

*Bonferroni correction for simultaneous confidence intervals.*

**Description**  
The input is the output from the bootstrap function.

**Usage**  

```r
LowHigh.B(x, y)
```

**Arguments**

- **x**: Estimated relative abundances for the functions/subsystems from the Twostep method on the preprocessed dataset.
- **y**: Bootstrap results of relative abundances for the functions/subsystems. Note: each bootstrap result is obtained by Twostep method.

**Examples**

```r
x.value=c(0.01, 0.05, 0.1, 0.2, 0.3, 0.34)
fname=c("feature1","feature2","feature3","feature4","feature5","feature6")
x=data.frame(name=fname, value=x.value)
temp=t(sapply(x.value, function(i)(rnorm(100, i, i/10))))
y.value=sapply(1:dim(temp)[2], function(i)(temp[,i]/sum(temp[,i])))
y=data.frame(name=fname, value=y.value)
LowHigh.B(x,y)
```

LowHigh.NP  

*Non-parametric way for confidence intervals*

**Description**  
The input is the output from the bootstrap function.

**Usage**  

```r
LowHigh.NP(x, y)
```

**Arguments**

- **x**: Estimated relative abundances for the functions/subsystems from the Twostep method on the preprocessed dataset.
- **y**: Bootstrap results of relative abundances for the functions/subsystems. Note: each bootstrap result is obtained by Twostep method.
**mismatch**

**Examples**

```r
x.value <- c(0.01, 0.05, 0.1, 0.2, 0.3, 0.34)
fname <- c("feature1", "feature2", "feature3", "feature4", "feature5", "feature6")
x <- data.frame(name = fname, value = x.value)
temp <- sapply(x.value, function(i) rnorm(1, i, 1))
y.value <- sapply(1:dim(temp)[2], function(i) temp[, i] / sum(temp[, i]))
y <- data.frame(name = fname, value = y.value)
LowHigh.NP(x, y)
```

---

**Calculation of Maximum Mismatches**

**Description**

Assume a binomial distribution for the number of mismatched codons, calculate the maximum allowed matches to meet the pre-specified small probability.

**Usage**

```r
mismatch(Length, CM, p, epsilon)
```

**Arguments**

- **Length**
  - Length of a short read.

- **CM**
  - Maximum perfect matches for a short read across multiple returns in Blastx output.

- **p**
  - Error estimated from the mixture model.

- **epsilon**
  - A pre-specified small probability.

**Examples**

```r
mismatch(32, 32, 0.15, 0.05)
```

---

**plot.abundance**

*Generate plots for estimated relative abundances of functions/subsystems.*

**Description**

Visualizing the relative abundances of the functions or subsystems in a metagenomic sample. 95% confidence intervals for the abundances obtained from bootstrap resampling method are added as error bars.

**Usage**

```r
plot.abundance(boot.dat)
```
**Arguments**

boot.dat  
The output from bootstrap resampling method.

**Examples**

```r
library(metaFunction)
require(data.table)
require(mefa)
require(lattice)

data(ExampleData)
data(refseq)
data(subs)

pre=preprocess(ExampleData)

epsilon=0.01
bootsize=100

boot.data=bootstrap(pre, bootsize, epsilon, method="B")
plot.abundance(boot.data)
```

---

**Description**

Preprocess the output data from Blastx.

**Usage**

```r
preprocess(x)
```

**Arguments**

x  
blastx output

**Details**

Depends on the blastx format for alignment, the input data for the function “preprocess” could be modified a little bit to accompany with the blastx output.

**Examples**

```r
library(metaFunction)
require(data.table)
require(mefa)
require(lattice)

data(ExampleData)
data(refseq)
data(subs)

pre=preprocess(ExampleData)
```
refseq

### Description
Mapping information between accession number and functional roles.

### Usage
```r
data(refseq)
```

### Format
A data frame with 1693417 observations on the following 3 variables.

- **AccessionNumber** a character vector
- **FunctionRole** a character vector
- **ID** a character vector

### Examples
```r
data(refseq)
## maybe str(refseq) ; plot(refseq) ...
```

---

subs

### Description
Mapping information between the functional roles and different levels of subsystems.

### Usage
```r
data(subs)
```

### Format
A data frame with 12808 observations on the following 4 variables.

- **sub3** a character vector
- **sub1** a character vector
- **sub2** a character vector
- **FunctionRole** a character vector

### Examples
```r
data(subs)
## maybe str(subs) ; plot(subs) ...
```
Twosteps

Main function for estimating the relative abundances of functions/subsystems in a metagenomic sample

Description

This function contains two steps: mixture model for estimating (sequencing) error and binomial distribution for determining the multiple function assignment.

Usage

Twosteps(mydata0, epsilon)

Arguments

mydata0 Input data from the output of the preprocess function.
epsilon A small probability for adjusting the function assignment.

Examples

library(metaFunction)
require(data.table)
require(mefa)
require(lattice)

data(ExampleData)
data(refseq)
data(subs)

pre=preprocess(ExampleData)

epsilon=0.01
est=Twosteps(pre, epsilon)
Index

*Topic \textasciitilde kwd1
  bootstrap, 2
  LowHigh.B, 4
  LowHigh.NP, 4
  mismatch, 5
  plot.abundance, 5
  preprocess, 6
  Twosteps, 8
*Topic \textasciitilde kwd2
  bootstrap, 2
  LowHigh.B, 4
  LowHigh.NP, 4
  mismatch, 5
  plot.abundance, 5
  preprocess, 6
  Twosteps, 8
*Topic datasets
  ExampleData, 3
  refseq, 7
  subs, 7
*Topic package
  metaFunction-package, 2
  bootstrap, 2
  ExampleData, 3
  LowHigh.B, 4
  LowHigh.NP, 4
  metaFunction (metaFunction-package), 2
  metaFunction-package, 2
  mismatch, 5
  plot.abundance, 5
  preprocess, 6
  refseq, 7
  subs, 7
  Twosteps, 8