Package 'TFBS.QSAM'

December 8, 2013

Type Package

Title The analysis of TFBS by QSAM

Version 1.0.3

Date 2013-12-04

Author Evgenia Temlyakova

Maintainer Evgenia Temlyakova <evgenia.teml@gmail.com>

Description The package provides various techniques to analyse physical and chemical properties of DNA sequencies. The most attention is given to transription factor binding sites.

License GPL (>= 2)

Depends R (>= 2.14), seqinr

Suggests testthat

R topics documented:

TFBS.QSAM-package *The analysis of TFBS by QSAM*

Description

The package provides various techniques to analyse physical and chemical properties of DNA sequencies. The most attention is given to transription factor binding sites.

Details

Package:	TFBS.QSAM
Type:	Package
Title:	The analysis of TFBS by QSAM
Version:	1.0.2
Date:	2013-12-04
Author:	Evgenia Temlyakova
Maintainer:	Evgenia Temlyakova <evgenia.teml@gmail.com>
License:	GPL (>= 2)

Author(s)

Evgenia Temlyakova

coli *E.coli genome*

Usage

```
data(coli)
```

Format

The format is: chr "AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAA-
GAGTGTCTGATAGCAGCTTCTGAACTGGTTACCTGCCGTGAGTAAATTAAAATTTTATTGACTTAGGTCACTA
__truncated__

Examples

```
data(coli)
## maybe str(coli) ; plot(coli) ...
```

coli.tfbs *E.coli TFBS*

Description

The function returns main information about all known TFBS of E.coli genome.

Usage

```
coli.tfbs()
```

Author(s)

Evgenia Temlyakova

evidence *Short evidence representation*

Description

The function converts long RegulonDB strings about the evidence into short and clear representation with letters |S| and |W|.

Usage

```
evidence(evidence.string)
```

Arguments

`evidence.string`
: RegulonDB string, containing the evidence for a TFBS or a promoter

Author(s)

Evgenia Temlyakova

`pls.analysis` *PLS-DA analysis*

Description

The function performs PLS-DA analysis and returns the detailed result about the created model. There are two possible input formats: set1 and set2 (and the function would prepare and form all nessacery sets by itself), and train and test (user defines these sets in the proper format).

Usage

```
pls.analysis(set1 = NULL, set2 = NULL, train = NULL, test = NULL,
    shuffle = TRUE, train.part = 50)
```

Arguments

`set1`	a matrix or a data.frame for set1 with objects in rows and descriptors in columns
`set2`	a matrix or a data.frame for set2 with objects in rows and descriptors in columns
`train`	a data.frame with two subobjects of class AsIs - descriptors and binary classes
`test`	a data.frame with two subobjects of class AsIs - descriptors and binary classes
`shuffle`	logical: do you want to shuffle objects order in each set?
`train.part`	a proportion of objects to be put in the training process; only if you specify set1 and set2

Author(s)

Evgenia Temlyakova

Examples

```
set1<-matrix(rnorm(500, 1), nrow=50,  byrow=TRUE)
set2<-matrix(rnorm(500, 2), nrow=50,  byrow=TRUE)
res<-pls.analysis(set1=set1, set2=set2)

train.set<-rbind(set1[1:25,], set2[1:25,])
test.set<-rbind(set1[26:50,], set2[26:50,])
train<-data.frame(CH=I(train.set), TY=I(c(rep(1, 25), rep(0, 25))))
test<-data.frame(CH=I(test.set), TY=I(c(rep(1, 25), rep(0, 25))))
res<-pls.analysis(train=train, test=test)
```

process.pred *Prediction data processing*

Description

The function combines and precess primary prediction data of a few classifications by various predefined schemes.

Usage

```
process.pred(pred, scheme = "model1+", bin, window)
```

Arguments

pred	predictions dataframe
scheme	processing scheme to use; possible scemes are 'model1+', 'equal.contribution'
bin	list with 3-4 sublists called 'exp', 'pro', 'reg', 'rtb' containing binary vectors where 1 corresponds to TFBS
window	window for sliding; we use a length of TFBS

Author(s)

Evgenia Temlyakova

QSAM.seq *DNA sequences from QSAM-vector*

Description

The function converts QSAM-vector back into DNA sequence. There are 2 predefined types of QSAM matricies for nuclotides, but it is also possible to set another one for calculations.

Usage

```
QSAM.seq(num, QSAM = "QSAM1")
```

Arguments

num	QSAM-vector
QSAM	QSAM matrix 4*n, where n is a number of properties; possible values are 'QSAM1' and 'QSAM2'

Value

string of latters A, T, C, G

Author(s)

Evgenia Temlyakova

See Also

seq.QSAM

Examples

```
num<-c(-2.23, 0.79, -1.15, -0.78, 2.37, 0.72, -2.07, 1.77, 2.37, 0.72, -2.07, 1.77)
QSAM.seq(num)
```

---

seq.QSAM *QSAM-transformation of DNA sequences*

Description

The function converts a DNA sequence into QSAM-vector. There are 2 predefined types of QSAM values nuclotides, but it is also possible to set another one for calculations.

Usage

```
## S3 method for class 'QSAM'
seq(s, QSAM = "QSAM1")
```

Arguments

s	DNA-sequence
QSAM	QSAM matrix 4*n, where n is a number of properties; possible values are 'QSAM1' and 'QSAM2'

Value

a numeric vector with length equal to number_of_nucleotides*n

Author(s)

Evgenia Temlyakova

See Also

QSAM.seq

Examples

```
seq.QSAM('AAATTGCGC')

myQSAM<-as.data.frame(matrix(c(-2.23, 0.79, 2.37, 0.72,  0.32, -3.76, 1.92, 2.52), nrow=4, byrow=TRUE))
rownames(myQSAM)<-c('A', 'C', 'G', 'T')
seq.QSAM('AAATTGCGC', QSAM=myQSAM)
```

sliding.sum *Calculation of sliding sums*

Description

Converts pred for each DNA position into sliding sum representation

Usage

```
sliding.sum(pred, window, dim = 1, columns = c(1, 3, 5), tail = 500)
```

Arguments

pred	table with pred in columns
window	window for summation
dim	dimensions; use 1 for rows and 2 for columns
columns	colomns to use
tail	number of flanking positions at the end of the prediction table

Author(s)

Evgenia Temlyakova

tfbs.pos *TFBS positions on the E.coli chromosome*

Description

The function locates all binding sites for specified TF on E.coli chromosome for one or both DNA strands. It uses information taken from RegulonDB version 8.2.

Usage

```
tfbs.pos(tfname, strand = "both")
```

Arguments

tfname	a name of TF
strand	

Author(s)

Evgenia Temlyakova

`tf_dataset` *The list contains all information about E.coli TFBS taken from RegulonDB 8.2*

Usage

```
data(tf_dataset)
```

Format

The format is: List of 160 $ AcrR :List of 3 ..$ AcrR1:List of 8 .. ..$ EC : chr "ECK120015994" .. ..$ pos : num [1:2] 484933 484956 .. ..$ strand : chr "reverse" .. ..$ seq : chr "gcgttagattTACATACATTTGTGAATGTATGTAccatagcacg" .. ..$ effect : chr "-" .. ..$ promoter: chr "acrAp" .. ..$ from_tss: num -22.5 .. ..$ evidence: chr "[BCE|W|Binding of cellular extracts],[GEA|W|Gene expression analysis]" [list output truncated]

Source

http://regulondb.ccg.unam.mx/menu/download/datasets/files/BindingSiteSet.txt

See Also

`data(tf_equallength)`

Examples

```
data(tf_dataset)
## maybe str(tf_dataset) ; plot(tf_dataset) ...
```

`tf_equallength` *The list contains an information taken from RegulonDB 8.2 about 8 E.coli TFBS (ArcA, CRP, Fis, FNR, IHF, Lrp, NarL, Fur) with equal length.*

Description

Very often RegulonDB contains entries for one TF with different binding site lengths. It might cause difficulties for TFBS analysis. To avoid the problem we created this list, all entries in it provide sequences with equal (most frequent) lengths.

Usage

```
data(tf_equallength)
```

Format

The format is: List of 8 $ ArcA:List of 104 ..$:List of 11 .. ..$ EC : chr "ECK120011345" .. ..$ pos : num [1:2] 41981 41995 .. ..$ strand : chr "reverse" .. ..$ seq : chr "tatattaaatGTTAACAAAAATAAAacaaacggga" .. ..$ effect : chr "+" .. ..$ promoter: chr "caiTp" .. ..$ from_tss: num 50 .. ..$ evidence: chr "[GEA|W|Gene expression analysis],[AIBSCS|W|Automated inference based on similarity to consensus sequences]" .. ..$ length : int 15 .. ..$ mpot : num [1:121] -0.0613 -0.0612 -0.0611 -0.0611 -0.0611 ... .$ weight : num -6.2 .. [list output truncated]

Source

http://regulondb.ccg.unam.mx/menu/download/datasets/files/BindingSiteSet.txt

See Also

data(tf_dataset)

Examples

```
data(tf_equallength)
## maybe str(tf_equallength) ; plot(tf_equallength) ...
```

Index