Package ‘davidTiling’

February 1, 2017

Version 1.14.0

Title Data and analysis scripts for David, Huber et al. yeast tiling array paper

Author Wolfgang Huber <huber@ebi.ac.uk>, Joern Toedling <toedling@ebi.ac.uk>

Maintainer Wolfgang Huber <huber@ebi.ac.uk>

Depends R (>= 2.10), Biobase (>= 2.5.5), tilingArray, GO.db

Description This package contains the data for the paper by L. David et al. in PNAS 2006 (PMID 16569694): 8 CEL files of Affymetrix genechips, an ExpressionSet object with the raw feature data, a probe annotation data structure for the chip and the yeast genome annotation (GFF file) that was used. In addition, some custom-written analysis functions are provided, as well as R scripts in the scripts directory.

biocViews ExperimentData, Genome, Saccharomyces_cerevisiae_Data, MicroarrayData, ReproducibleResearch

Reference A high-resolution map of transcription in the yeast genome.

License LGPL

URL http://www.ebi.ac.uk/huber

NeedsCompilation no

R topics documented:

data:davidTiling data:gff data:probeAnno getAliGO getAttributeField GOHyperG scatterWithHist scoreSegments showDens yeastFeatures

Index 10
**data:davidTiling**  
*Dataset of class eSet with the raw 'CEL file' intensities*

**Description**

The data are from an experiment that used Affymetrix Scerevisiaetiling chips from 2004, which where custom-made for the Stanford Genome Center. The chips tile the complete genome of *S. cerevisiae* in steps of 8 bases, separately for each strand of each chromosome. The two tiles for one chromosome (Watson and Crick strands) are offset by 4 bases.

Note that the class `eSet` was used instead of `AffyBatch` since the additional overhead of 'CDF environments' in the latter is not needed here.

**Usage**

```r
data("davidTiling")
```

**Format**

Intensity data for 8 arrays. The `phenoData` slot contains the file names and the nucleic acid type.

**Author(s)**

W. Huber `<huber@ebi.ac.uk>`

**Source**

Lior David and Lars Steinmetz, both from the Stanford Genome Center. Lars Steinmetz is also at EMBL Heidelberg.

**Examples**

```r
data("davidTiling")
dim(exprs(davidTiling))
```

---

**data:gff**  
*Genomic features of Saccharomyces cerevisiae*

**Description**

A data frame with genomic features of *Saccharomyces cerevisiae*.

**Usage**

```r
data("gff")
```

**Format**

Object of class `data.frame`. GFF is a file format for annotating genomes, see <insert the URL to the documentation page for GFF at Sanger here>. The format is essentially a rectangular table, and here it is represented as a data frame.
Two GFF files were downloaded: saccharomyces_cerevisiae.gff from ftp://genome-ftp.stanford.edu/pub/yeast on 7 Aug 2005, 18:16 BST, and IGR_v24.2.p001.allowoverlap.GFF from http://jura.wi.mit.edu/fraenkel/download/released_v24/GFF upon suggestion from the SGD curators on 30 Aug 2005. (Future versions of SGD’s GFF files are likely to include the latter as well). They were parsed, combined and written into the gff data.frame with the script makeProbeAnno.R in the inst/scripts directory of this package.

Examples

data("gff")
str(gff)
getAllGO

Usage

data("probeAnno")

Author(s)

W. Huber <huber@ebi.ac.uk>

Source

Probe sequences were obtained from Affymetrix in a file called S.cerevisiae_tiling.1lq. The genomic sequences of the S. cerevisiae chromosomes were downloaded from ftp://genome-ftp.stanford.edu/pub/yeast on 7 Aug 2005, 18:16 BST in 17 files chr01.fsa–chr16.fsa, and chrmt.fsa. The probe sequences were matched against the chromosomal sequences with the program MUMmer, see the script mapProbesToGenome.sh (in the inst/scripts directory of this package). MUMmer results were parsed and processed into the probeAnno environment with the script makeProbeAnno.R (in the inst/scripts directory of this package).

Examples

data("probeAnno")
ls(probeAnno)
str(probeAnno$"1.+.start")

getAllGO

Get all GO categories for a list of genes.

Description

The function uses the GO categories in the data.frame gff to obtain annotated GO categories, then the GO**ANCESTOR data in the GO package to add all parent terms as well.

Usage

getAllGO(x, gff)

Arguments

x  character vector.

Args  data.frame with columns feature, Name, and (Ontology_term or attributes), see details

Details

The elements of x are matched against the column gene in gff. All are required to match. A list of GO terms is then extracted from the corresponding rows in the Ontology_term column. A gene may be annotated by several terms, separated by ",". Then the GO package is used to augment this by all ancestor terms.

Value

List of character vectors.
getAttributeField

Description

Extract the value of a certain field out of a character vector such as in the "attributes" column of a GFF table.

Usage

getAttributeField(x, field, attrsep=";")

Arguments

- x: character vector.
- field: character vector of length 1, containing the field name.
- attrsep: character vector of length 1, containing the separator name.

Details

See example.

Value

Character vector.

Author(s)

W. Huber <huber@ebi.ac.uk>

Examples

```r
acol = c("ID=46891;Name=TEL01L-TR;Note=Bla",
         "ID=46892;Name=TEL01L;Note=Di",
         "ID=46893;Name=TEL01L-XR;Note=Bla")

getAttributeField(acol, "Name")
getAttributeField(acol, "ID")
```
GOHyperG

Hypergeometric test for GO category enrichment.

Description
This function is adapted from the function of the same name in GOstats package. Main difference is that it draws the GO annotations of the genes in candidates from the data.frame gff. It also draws a plot.

Usage
GOHyperG(candidates, gff, plotmain)

Arguments
- candidates: character vector
- gff: see getAllGO
- plotmain: character, plot title

Details
The elements of x are matched against the column gene in gff. All are required to match. A list of GO terms is then extracted from the corresponding rows in the Ontology_term column. A gene may be annotated by several terms, separated by ",". Then the GO package is used to augment this by all ancestor terms.

Value
List of character vectors.

Author(s)
W. Huber <huber@ebi.ac.uk>

See Also
getAllGO

scatterWithHist
Scatterplot with histograms of marginal distributions

Description
Scatterplot with histograms of marginal distributions.

Usage
scatterWithHist(x, breaks, barcols, xlab, ylab, ...)

Arguments

- **x**: numeric matrix with 2 columns.
- **breaks**: numeric vector with histogram breaks, see `hist`.
- **barcols**: character vector of length 2, colors for the histogram filling.
- **xlab**: character of length 1, label for x-axis.
- **ylab**: character of length 1, label for y-axis.
- **...**: further arguments that get passed on to `plot`.

Value

The function is called for its side effect.

Author(s)

W. Huber <huber@ebi.ac.uk>

Examples

```r
dx = rexp(100)
xx = cbind(x, x+0.6+rnorm(length(x)))
scatterWithHist(xx,
           breaks=seq(min(x),max(x),length=20),
           barcols=c("mistyrose", "lightblue"),
           xlab="Daffodil", ylab="Petunia", pch=16)
```

scoreSegments

### Description

Score the segments found by a previous call to `findSegments` by comparing to genome annotation.

### Usage

```r
scoreSegments(s, gff,
           nrBasePerSeg = 1500,
           probeLength = 25,
           params = c(overlapFraction = 0.5, oppositeWindow = 100, flankProbes=10),
           verbose = TRUE)
```

### Arguments

- **s**: environment. See details.
- **gff**: GFF dataframe.
- **nrBasePerSeg**: Numeric of length 1. This parameter determines the number of segments.
- **probeLength**: Numeric of length 1.
- **params**: vector of additional parameters, see details.
- **verbose**: Logical.
showDens

Details

This function scores segments. It is typically called after a segmentation. For an example segmentation script, see the script segment.R in the scripts directory of this package. For an example scoring script, which loads the data and then calls this function, see the script scoreSegments.R.

Value

A dataframe with columns as described in the details section.

Author(s)

W. Huber <huber@ebi.ac.uk>

showDens

Description

Plot function for more than one density

Usage

showDens(z, breaks, xat, xtickLabels=paste(xat), col, ylab = "", ...)

Arguments

z List: numeric vectors for computing histograms for
breaks Numeric vector: breaks of the histogram
xat Numeric vector: where to put the x-axis ticks
xtickLabels Character vector: what to write underneath them
col Character vector: colours of the histograms
ylab Character scalar: y-axis label
... further arguments passed on to plot

Details

...

Value

returns scale factor

Author(s)

Wolfgang Huber <huber@ebi.ac.uk>

See Also

hist
yeastFeatures

Examples

```r
data(yeastFeatures)
```

Description

The rows of the data frame correspond to feature categories, such as "gene", "CDS", "telomere". The column `isTranscribed` is a logical vector that denotes whether this feature category is considered to be transcribable.

Usage

```r
data(yeastFeatures)
```

Format

A data.frame

Author(s)

W. Huber <huber@ebi.ac.uk>

Examples

```r
data(yeastFeatures)
```
Index

*Topic **datasets**
  - data:davidTiling, 2
  - data:gff, 2
  - data:probeAnno, 3
  - yeastFeatures, 9

*Topic **hplot**
  - scatterWithHist, 6
  - showDens, 8

*Topic **manip**
  - getAllGO, 4
  - getAttributeField, 5
  - GOHyperG, 6
  - scoreSegments, 7

AffyBatch, 2

  - data:davidTiling, 2
  - data:gff, 2
  - data:probeAnno, 3
  - davidTiling, 3
  - davidTiling (data:davidTiling), 2

  - eSet, 2

  - getAllGO, 4, 6
  - getAttributeField, 5
  - gff (data:gff), 2
  - GOHyperG, 6

  - hist, 7, 8

  - phenoData, 2
  - plot, 7
  - probeAnno (data:probeAnno), 3

  - scatterWithHist, 6
  - scoreSegments, 7
  - showDens, 8

  - yeastFeatures, 9