Package ‘GenomicDistributionsData’

March 28, 2024

Title Reference data for GenomicDistributions package
Version 1.10.0
Description This package provides ready to use reference data for GenomicDistributions package. Raw data was obtained from ensemblldb and processed with helper functions. Data files are available for the following genome assemblies: hg19, hg38, mm9 and mm10.
License BSD_2_clause + file LICENSE
biocViews ExperimentHub, ExperimentData, Genome
Encoding UTF-8
Roxygen list(markdown = TRUE)
RoxygenNote 7.1.1
Depends R (>= 4.1)
Imports ExperimentHub (>= 1.14.0), AnnotationHub, AnnotationFilter, data.table, utils, BSgenome, GenomeInfoDb, GenomicFeatures, GenomicRanges, ensemblldb
Suggests knitr, BiocStyle, rmarkdown
VignetteBuilder knitr
BugReports http://github.com/databio/GenomicDistributionsData
git_url https://git.bioconductor.org/packages/GenomicDistributionsData
git_branch RELEASE_3_18
git_last_commit 62e25be
git_last_commit_date 2023-10-24
Repository Bioconductor 3.18
Date/Publication 2024-03-28
Author Michal Stolarczyk [aut], Kristyna Kupkova [aut, cre], Jose Verdezoto [aut], John Lawson [ctb], Jason Smith [ctb], Nathan C. Sheffield [ctb]
Maintainer Kristyna Kupkova <kristynakupkova@gmail.com>

### R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>.requireAndReturn</td>
<td>2</td>
</tr>
<tr>
<td>buildChromSizes</td>
<td>3</td>
</tr>
<tr>
<td>buildGeneModels</td>
<td>3</td>
</tr>
<tr>
<td>buildOpenSignalMatrix</td>
<td>4</td>
</tr>
<tr>
<td>buildTSS</td>
<td>4</td>
</tr>
<tr>
<td>cellTypeMetadata</td>
<td>5</td>
</tr>
<tr>
<td>chromSizes_hg19</td>
<td>5</td>
</tr>
<tr>
<td>chromSizes_hg38</td>
<td>6</td>
</tr>
<tr>
<td>chromSizes_mm10</td>
<td>7</td>
</tr>
<tr>
<td>chromSizes_mm9</td>
<td>7</td>
</tr>
<tr>
<td>geneModels_hg19</td>
<td>8</td>
</tr>
<tr>
<td>geneModels_hg38</td>
<td>9</td>
</tr>
<tr>
<td>geneModels_mm10</td>
<td>9</td>
</tr>
<tr>
<td>geneModels_mm9</td>
<td>10</td>
</tr>
<tr>
<td>GenomicDistributionsData</td>
<td>11</td>
</tr>
<tr>
<td>loadBSgenome</td>
<td>11</td>
</tr>
<tr>
<td>loadEnsDb</td>
<td>12</td>
</tr>
<tr>
<td>loadTxDb</td>
<td>13</td>
</tr>
<tr>
<td>openSignalMatrix_hg19</td>
<td>13</td>
</tr>
<tr>
<td>openSignalMatrix_hg38</td>
<td>14</td>
</tr>
<tr>
<td>openSignalMatrix_mm10</td>
<td>15</td>
</tr>
<tr>
<td>TSS_hg19</td>
<td>16</td>
</tr>
<tr>
<td>TSS_hg38</td>
<td>17</td>
</tr>
<tr>
<td>TSS_mm10</td>
<td>18</td>
</tr>
<tr>
<td>TSS_mm9</td>
<td>18</td>
</tr>
</tbody>
</table>

### Index

<table>
<thead>
<tr>
<th>Item</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>.requireAndReturn</td>
<td>2</td>
</tr>
</tbody>
</table>

**Description**

Checks to make sure a package object is installed, and if so, returns it. If the library is not installed, it issues a warning and returns NULL.

**Usage**

```
.requireAndReturn(BSgenomeString)
```

**Arguments**

- `BSgenomeString` A BSgenome compatible genome string.
**buildChromSizes**

**Value**

A BSgenome object if installed.

---

**buildChromSizes**  
*Build chromosome sizes object*

**Description**

Build chromosome sizes object

**Usage**

`buildChromSizes(assembly)`

**Arguments**

- `assembly`  
  string, reference assembly identifier to chromosome sizes for

**Value**

named int, sizes of chromosome with respective names

**Examples**

```r
## Not run:
buildChromSizes("hg19")
## End(Not run)
```

---

**buildGeneModels**  
*Build gene models*

**Description**

Build gene models

**Usage**

`buildGeneModels(refAssembly)`

**Arguments**

- `refAssembly`  
  string, reference assembly identifier to build gene model for

**Value**

a list of four GRanges objects: genesGR, exonsGR, threeUTRGR, fiveUTRGR
buildTSS

Description

Build TSS

Usage

buildTSS(assembly)

Arguments

assembly

string, reference assembly identifier to build gene model for

Value

data.frame, rows represent whole selection of open chromatin regions across all cell types defined by ENCODE, columns are individual cell types and values are normalized open chromatin signal values

Examples

## Not run:
buildOpenSignalMatrix("hg19")

## End(Not run)

buildOpenSignalMatrix

Description

Build open signal matrix

Usage

buildOpenSignalMatrix(assembly)

Arguments

assembly

string, reference assembly identifier to build gene model for

Value

data.frame, rows represent whole selection of open chromatin regions across all cell types defined by ENCODE, columns are individual cell types and values are normalized open chromatin signal values

Examples

## Not run:
buildOpenSignalMatrix("hg19")

## End(Not run)

buildTSS

Description

Build TSS

Usage

buildTSS(assembly)
cellTypeMetadata

Arguments

 assembly  string, reference assembly identifier to TSS for

Value

 GRanges object that consists of transcription start sites locations

Examples

## Not run:
tss = buildTSS("hg19")

## End(Not run)

---

Table the maps cell types to tissues and groups

cellTypeMetadata

Description

Table the maps cell types to tissues and groups

Format

data.table with 3 columns (cellType, tissue and group) and 74 rows (one per cellType)

Source

self-curated dataset

---

chromSizes_hg19  hg19 chromosome sizes

Description

A dataset containing chromosome sizes for Homo sapiens hg19 genome assembly

Usage

chromSizes_hg19(metadata=FALSE)

Arguments

 metadata  logical value indicating whether only metadata should be returned or if the resource should be loaded
**Format**
A named vectors of lengths with one item per chromosome

**Value**
A vector of lengths

**Source**
BSgenome.Hsapiens.UCSC.hg19 package

**Examples**
```
hg19c = chromSizes_hg19()
```

---

**chromSizes_hg38**

**hg38 chromosome sizes**

**Description**
A dataset containing chromosome sizes for Homo sapiens hg38 genome assembly

**Usage**
```
chromSizes_hg38(metadata=FALSE)
```

**Arguments**
- `metadata` logical value indicating whether only metadata should be returned or if the resource should be loaded

**Format**
A named vectors of lengths with one item per chromosome

**Value**
A vector of lengths

**Source**
BSgenome.Hsapiens.UCSC.hg38 package

**Examples**
```
hg38c = chromSizes_hg38()
```
### chromSizes_mm10

**mm10 chromosome sizes**

**Description**

A dataset containing chromosome sizes for Mus musculus mm10 genome assembly

**Usage**

```r
chromSizes_mm10(metadata=FALSE)
```

**Arguments**

- `metadata` logical value indicating whether only metadata should be returned or if the resource should be loaded

**Format**

A named vectors of lengths with one item per chromosome

**Value**

A vector of lengths

**Source**

BSgenome.MMusculus.UCSC.mm10 package

**Examples**

```r
mm10c = chromSizes_mm10()
```

---

### chromSizes_mm9

**mm9 chromosome sizes**

**Description**

A dataset containing chromosome sizes for Mus musculus mm9 genome assembly

**Usage**

```r
chromSizes_mm9(metadata=FALSE)
```

**Arguments**

- `metadata` logical value indicating whether only metadata should be returned or if the resource should be loaded
**geneModels_hg19**

**Format**
A named vectors of lengths with one item per chromosome

**Value**
A vector of lengths

**Source**
BSgenome.MMusculus.UCSC.mm9 package

**Examples**
nm9c = chromSizes_mm9()

---

geneModels_hg19  

**Description**
A dataset containing gene models for Homo sapiens hg19 genome assembly

**Usage**
geneModels_hg19(metadata=FALSE)

**Arguments**
metadata  
logical value indicating whether only metadata should be returned or if the resource should be loaded

**Format**
A list of two GRanges objects, with genes and exons locations

**Value**
A list with two GRanges objects.

**Source**
EnsDb.Hsapiens.v75 package

**Examples**
hg19GeneModels = geneModels_hg19()
**geneModels_hg38**

---

### Description

A dataset containing gene models for Homo sapiens hg38 genome assembly

### Usage

```r
geneModels_hg38(metadata=FALSE)
```

### Arguments

- `metadata`: logical value indicating whether only metadata should be returned or if the resource should be loaded.

### Format

A list of two GRanges objects, with genes and exons locations

### Value

A list with two GRanges objects.

### Source

EnsDb.Hsapiens.v86 package

### Examples

```r
hg38GeneModels = geneModels_hg38()
```

---

**geneModels_mm10**

---

### Description

A dataset containing gene models for Mus musculus mm10 genome assembly

### Usage

```r
geneModels_mm10(metadata=FALSE)
```
geneModels_mm9

Arguments

metadata logical value indicating whether only metadata should be returned or if the resource should be loaded

Format

A list of two GRanges objects, with genes and exons locations

Value

A list with two GRanges objects.

Source

EnsDb.Mmusculus.v79 package

Examples

mm10GeneModels = geneModels_mm10()

---

geneModels_mm9 mm9 gene models

Description

A dataset containing gene models for Mus musculus mm9 genome assembly

Usage

geneModels_mm9(metadata=FALSE)

Arguments

metadata logical value indicating whether only metadata should be returned or if the resource should be loaded

Format

A list of two GRanges objects, with genes and exons locations

Value

A list with two GRanges objects.

Source

TxDb.Mmusculus.UCSC.mm9.knownGene package
Examples

```r
mm9GeneModels = geneModels_mm9()
```

Description

Data (hg19, hg38, mm9, mm10) for the GenomicDistributions package.

Details

The vignette details how to access the data from ExperimentHub: `browseVignettes("GenomicDistributionsData")`

Details on how the data files were created can be found under the R directory in `utils.R` and `build.R`

Source

UCSC hg19, hg38, mm9, mm10 genome packages and ensembldb

Examples

```r
## Not run:
library(ExperimentHub)
hub = ExperimentHub()
q = query(hub, "GenomicDistributionsData")
q[[1]]
q[["EH3472"]]

## End(Not run)
```

loadBSgenome

**Loads BSgenome objects from UCSC-style character vectors.**

Description

This function will let you use a simple character vector (e.g. ’hg19’) to load and then return BSgenome objects. This lets you avoid having to use the more complex annotation for a complete BSgenome object (e.g. BSgenome.Hsapiens.UCSC.hg38.masked)

Usage

```r
loadBSgenome(genomeBuild, masked = TRUE)
```
loadEnsDb

Load selected EnsDb library

Description

Load selected EnsDb library

Usage

loadEnsDb(genomeBuild)

Arguments

genomeBuild string, genome identifier

Value

loaded library

Examples

## Not run:
loadEnsDb("hg19")

## End(Not run)
loadTxDb

Description
Load selected TxDb library

Usage
loadTxDb(genomeBuild)

Arguments

  genomeBuild       string, genome identifier

Value
loaded library

Examples

  ```r
  ## Not run:
  loadTxDb("hg19")
  ## End(Not run)
  ```

openSignalMatrix_hg19

Description
A dataset containing open chromatin regions across all cell types defined by ENCODE for Homo Sapiens hg19

Usage

  openSignalMatrix_hg19(metadata=FALSE)
openSignalMatrix_hg38

Arguments

metadata logical value indicating whether only metadata should be returned or if the resource should be loaded

Format

data.frame, rows represent whole selection of open chromatin regions across all cell types defined by ENCODE, columns are individual cell types and values are normalized open chromatin signal values.

Value

A data.frame with hg19 open chromatin regions.

Source

http://big.databio.org/open_chromatin_matrix/openSignalMatrix_hg19_percentile99_01_quantNormalized_round4d.txt.gz

Examples

## Not run:
hg19OpenSignal = openSignalMatrix_hg19()
## End(Not run)

openSignalMatrix_hg38 A dataset containing open chromatin regions across all cell types defined by ENCODE for Homo sapiens hg38

Description

Preparation steps:

1. made a universe of regions by merging regions across cell types defined as opened in ENCODE
2. took bigwig files from ENCODE for individual cell types, merged replicates, filtered out blacklisted sites
3. evaluated the signal above regions defined by previous step
4. performed quantile normalization

Usage

openSignalMatrix_hg38(metadata=FALSE)
openSignalMatrix_mm10

Arguments

metadata logical value indicating whether only metadata should be returned or if the resource should be loaded

Format
data.frame, rows represent whole selection of open chromatin regions across all cell types defined by ENCODE, columns are individual cell types and values are normalized open chromatin signal values.

Value

A data.frame with hg38 open chromatin regions.

Source

http://big.databio.org/open_chromatin_matrix/openSignalMatrix_hg38_percentile99_01_quantNormalized_round4d.txt.gz

Examples

## Not run:
hg38OpenSignal = openSignalMatrix_hg38()

## End(Not run)

openSignalMatrix_mm10  A dataset containing open chromatin regions across all cell types defined by ENCODE for Mus musculus mm10

Description

Preparation steps:

1. made a universe of regions by merging regions across cell types defined as opened in ENCODE
2. took bigwig files from ENCODE for individual cell types, merged replicates, filtered out blacklisted sites
3. evaluated the signal above regions defined by previous step
4. performed quantile normalization

Usage

openSignalMatrix_mm10(metadata=FALSE)
**TSS_hg19**

**Description**

A dataset containing Transcription Start Sites for Homo sapiens hg19 genome assembly

**Usage**

`TSS_hg19(metadata=FALSE)`

**Arguments**

- `metadata` logical value indicating whether only metadata should be returned or if the resource should be loaded

**Format**

A GRanges object with Transcription Start Site locations
**Value**

A GRanges object with hg19 TSS locations.

**Source**

EnsDb.Hsapiens.v75 package

**Examples**

```r
hg19TSS = TSS_hg19()
```

---

**TSS_hg38**

**hg38 TSS locations**

**Description**

A dataset containing Transcription Start Sites for the Homo sapiens hg38 genome assembly

**Usage**

```r
TSS_hg38(metadata=FALSE)
```

**Arguments**

- `metadata` logical value indicating whether only metadata should be returned or if the resource should be loaded

**Format**

A GRanges object with Transcription Start Site locations

**Value**

A GRanges object with hg38 TSS locations.

**Source**

EnsDb.Hsapiens.v86 package

**Examples**

```r
hg38TSS = TSS_hg38()
```
TSS_mm10

---

**Description**

A dataset containing Transcription Start Sites for the Mus musculus mm10 genome assembly

**Usage**

TSS_mm10(metadata=FALSE)

**Arguments**

- **metadata** logical value indicating whether only metadata should be returned or if the resource should be loaded

**Format**

A named vectors of lengths with one item per chromosome

**Value**

A GRanges object with mm10 TSS locations.

**Source**

EnsDb.Mmusculus.v79 package

**Examples**

```r
mm10TSS = TSS_mm10()
```

---

TSS_mm9

---

**Description**

A dataset containing Transcription Start Sites for the Mus musculus mm9 genome assembly

**Usage**

TSS_mm9(metadata=FALSE)

**Arguments**

- **metadata** logical value indicating whether only metadata should be returned or if the resource should be loaded
**Format**

A named vectors of lengths with one item per chromosome

**Value**

A GRanges object with mm9 TSS locations.

**Source**

TxDb.Musculus.UCSC.mm9.knownGene package

**Examples**

```r
mm9TSS = TSS_mm9()
```
Index

* datasets
  cellTypeMetadata, 5
  chromSizes_hg19, 5
  chromSizes_hg38, 6
  chromSizes_mm10, 7
  chromSizes_mm9, 7
  geneModels_hg19, 8
  geneModels_hg38, 9
  geneModels_mm10, 9
  geneModels_mm9, 10
  openSignalMatrix_hg19, 13
  openSignalMatrix_hg38, 14
  openSignalMatrix_mm10, 15
  TSS_hg19, 16
  TSS_hg38, 17
  TSS_mm10, 18
  TSS_mm9, 18
.requireAndReturn, 2
buildChromSizes, 3
buildGeneModels, 3
buildOpenSignalMatrix, 4
buildTSS, 4

cellTypeMetadata, 5
chromSizes_hg19, 5
chromSizes_hg38, 6
chromSizes_mm10, 7
chromSizes_mm9, 7

geneModels_hg19, 8
geneModels_hg38, 9
geneModels_mm10, 9
geneModels_mm9, 10
GenomicDistributionsData, 11
loadBSgenome, 11
loadEnsDb, 12
loadTxDb, 13
openSignalMatrix_hg19, 13