Package `leeBamViews`

February 1, 2017

Title leeBamViews -- multiple yeast RNAseq samples excerpted from Lee 2009

Version 1.10.0

Author VJ Carey <stvjc@channing.harvard.edu>

Description data from PMID 19096707; prototype for managing multiple NGS samples

Depends R (>= 2.15.0), Biobase, Rsamtools (>= 0.1.50), BSgenome

Imports GenomicRanges, GenomicAlignments, methods

Suggests GenomeGraphs, biomaRt, org.Sc.sgd.db, edgeR

Enhances multicore

Maintainer VJ Carey <stvjc@channing.harvard.edu>

License Artistic 2.0

LazyLoad yes

biocViews ExperimentData, Saccharomyces_cerevisiae_Data, SequencingData, RNASeqData, SNPData

NeedsCompilation no

R topics documented:

bs1 ................................................................. 1
leeRPKM .......................................................... 4
leeUnn ............................................................ 5
tabulateReads .................................................... 6
totalReadCounts .................................................. 7

Index 8

---

bs1 BamViews instance construction related to yeast RNA-seq

Description

BamViews instance construction related to yeast RNA-seq
Format

The format is: Formal class 'BamViews' [package "Rsamtools"] with 5 slots
..@ bamSamples : Formal class 'DataFrame' [package "IRanges"] with 6 slots
  .. ..@ rownames : chr [1:8] "isowt.5" "isowt.6" "rlp.5" "rlp.6" ...
  .. ..@ nrows : int 8
  .. ..@ elementType : chr "ANY"
  .. ..@ metadata : list()
  .. ..@ listData : List of 2
  .. .. $ geno: chr [1:8] "isowt" "isowt" "rlp" "rlp" ...
  .. .. $ lane: chr [1:8] "5" "6" "5" "6" ...
..@ bamRanges : Formal class 'GRanges' [package "GenomicRanges"] with 7 slots
  .. ..@ seqnames : Formal class 'Rle' [package "IRanges"] with 5 slots
  .. .. ..@ values : Factor w/ 1 level "Scchr13": 1
  .. .. ..@ lengths : int 27
  .. .. ..@ elementMetadata: NULL
  .. .. ..@ elementType : chr "ANY"
  .. .. ..@ metadata : list()
  .. .. ..@ ranges : Formal class 'IRanges' [package "IRanges"] with 6 slots
  .. .. .. ..@ start : int [1:27] 798517 801771 804455 808999 810465 811088 818826 820255 822762 832338 ...
  .. .. .. ..@ width : int [1:27] 2862 933 636 234 114 108 1122 2199 1869 915 ...
  .. .. .. ..@ NAMES : NULL
  .. .. .. ..@ elementMetadata: NULL
  .. .. .. ..@ elementType : chr "integer"
  .. .. .. ..@ metadata : list()
  .. .. .. ..@ strand : Formal class 'Rle' [package "IRanges"] with 5 slots
  .. .. .. .. ..@ values : Factor w/ 3 levels "+", ",", "+": 1
  .. .. .. .. ..@ lengths : int 27
  .. .. .. .. ..@ elementMetadata: NULL
  .. .. .. .. ..@ elementType : chr "ANY"
  .. .. .. .. ..@ metadata : list()
  .. .. .. .. ..@ seqlengths : Named int NA
  .. .. .. .. .. @ attr(*, "names")= chr "Scchr13"
  .. .. .. .. ..@ elementMetadata: Formal class 'DataFrame' [package "IRanges"] with 6 slots
  .. .. .. .. ..@ rownames : NULL
  .. .. .. .. ..@ nrows : int 27
  .. .. .. .. ..@ elementType : chr "ANY"
  .. .. .. .. ..@ metadata : list()
  .. .. .. .. ..@ listData : List of 1
  .. .. .. .. .. $ name: chr [1:27] "YMR266W" "YMR267W" "YMR269W" "YMRWdelta20" ...
  .. .. .. .. ..@ elementType : chr "ANY"
  .. .. .. .. ..@ metadata : list()
Details

Illumina short reads from a very small segment of yeast chr XIII have been collected

Source


References


Examples

```
library(leeBamViews) # bam files stored in package
bpaths = dir(system.file("bam", package="leeBamViews"), full=TRUE, patt="bam$")
#
# extract genotype and lane information from filenames
#
gt = gsub("/.*", "", bpaths)
gt = gsub("_.*", "", gt)
lane = gsub(".*(.)$", "\\1", gt)
geno = gsub(".$", "", gt)
#
# format the sample-level information appropriately
#
pd = DataFrame(geno=geno, lane=lane, row.names=paste(geno,lane,sep="."))
prd = new("DataFrame") # protocol data could go here
#
# create the views object, adding some arbitrary experiment-level information
#
bs1 = BamViews(bamPaths=bpaths, bamSamples=pd,
   bamExperiment=list(abbreviation="org.Sc.sgd.db"))
bs1
# add ranges and tabulate reads
START=c(861250, 863000)
END=c(862750, 864000)
exc = GRanges(IRanges(start=START, end=END), seqnames="Scchr13", strand="+")
values(exc)$name = c("intv1", "intv2") # necessary
bamRanges(bs1) = exc
bs1
tabulateReads(bs1, "+")
```
Description

supplemental data extract on RNA seq results in yeast

Usage

data(leeRPKM)

Format

A data frame with 6291 observations on the following 16 variables.

chr  a numeric vector
strand a numeric vector
start a numeric vector
end  a numeric vector
name a factor with levels LSR1 NME1 YAL001C YAL002W YAL003W ...
feature a factor with levels CDS CDS_unchar snRNA snoRNA
orf_classification a factor with levels Uncharacterized Verified silenced_gene3AVerified
gene a factor with levels AAC1 AAC3 AAD10 AAD14 AAD15 AAD16 AAD3 AAD4 ...
wt.reads  a numeric vector
rrp.reads a numeric vector
ski.reads  a numeric vector
xrn.reads a numeric vector
wt.rpkm a numeric vector
rrp.rpkm a numeric vector
ski.rpkm a numeric vector
xrn.rpkm a numeric vector

Source

imported from supplemental data

References

Lee et al PLOS genetics December 2008 ; Volume 4 ; Issue 12 ; e1000299

Examples

data(leeRPKM)
leeRPKM[1:5,]
Description

supplemental data extracts on existing evidence of transcription in yeast

Usage

data(leeUnn)

Format

A data frame with 54822 observations on the following 11 variables.

- chr  a numeric vector
- start  a numeric vector
- end  a numeric vector
- strand  a numeric vector
- lengthWithoutMask  a numeric vector
- length  a numeric vector
- lambda  a numeric vector
- background5  a logical vector
- background20  a logical vector
- reads  a numeric vector
- study  a factor with levels David Davis Miura Nagalakshmi

Source

from Lee et al PLoS genetics December 2008 Volume 4 Issue 12 e1000299 supplemental data information on unannotated transcripts for which some evidence of transcription was obtained in this experiment

Examples

data(leeUnn)
leeUnn[1:5,]
**tabulateReads**

*tabulate counts of alignments occurring in specified genomic regions*

**Description**

Tabulate counts of alignments occurring in specified genomic regions.

**Usage**

```
tabulateReads(bv, strandmarker=NULL, as.GRanges=FALSE, applier=lapply)
```

**Arguments**

- `bv` (BamViews-class instance)
- `strandmarker` (character atom: '+' or '-': if missing, ignore strand)
- `as.GRanges` (logical directive to return a GRanges instance instead of a matrix)
- `applier` (lapply-like function; if unspecified and multicore is attached will use mclapply)

**Details**

`readGAlignments` is the basic engine for this task.

**Value**

Annotated matrix with start, end, and samples as rows, regions as columns, and read counts as cell entries.

**Author(s)**

VJ Carey <stvjc@channing.harvard.edu>

**Examples**

```
# counts in a partition
myrn = GRanges(IRanges(start=seq(861250, 862750, 100), width=100),
                seqnames="Scchr13", strand="+")
values(myrn)$name = paste("til", 1:length(myrn), sep=".")
bamRanges(bs1) = myrn
tabulateReads(bs1, "+")
```

# a related computation based on countBam
lapply(bamPaths(bs1)[1:2], function(x)
        countBam(x, param=ScanBamParam(which=bamRanges(bs1))))
Description
scan BAM files for total read counts

Usage
totalReadCounts(x)

Arguments
x 

BamViews-class instance

Details
slow procedure – does lightweight scan of entire file

Value
named integer vector of read counts per sample

Author(s)
VJ Carey <stvjc@channing.harvard.edu>

Examples
example(bs1)
totalReadCounts(bs1)
Index

*Topic datasets
    bs1, 1
    leeRPKM, 4
    leeUnn, 5

*Topic models
    tabulateReads, 6
    totalReadCounts, 7

bs1, 1

leeRPKM, 4
leeUnn, 5

readGAlignments, 6

tabulateReads, 6
tabulateReads,BamViews,characterORNULL,logical,function-method (tabulateReads), 6
tabulateReads,BamViews,characterORNULL,missing,missing-method (tabulateReads), 6
tabulateReads,BamViews,missing,missing,missing-method (tabulateReads), 6
totalReadCounts, 7
totalReadCounts,BamViews-method (totalReadCounts), 7