

Package ‘ngsroi’

July 1, 2013

Type Package

Title NGS Regions of Interest Analysis

Version 1.0

Date 2013-04-12

Maintainer Manuel Holtgrewe <manuel.holtgrewe@fu-berlin.de>

Description Routines for I/O of NGS ROI files and manipulation thereof.

License BSD + file LICENSE

URL http://www.segan.de/projects/ngs_roi

Author Bernd Jagla Developer [aut, cre, cph], Manuel Holtgrewe Developer [ctb]

R topics documented:

ngsroi-package	1
readROI	2
writeROI	3

ngsroi-package	<i>NGS Regions of Interest Analysis</i>
----------------	---

Description

Routines for I/O of NGS ROI files and manipulation thereof.

Details

Package: ngsroi
Type: Package
Version: 0.1
Date: 2013-04-12
License: BSD_3_clause

Author(s)

Bernd Jagla <bernd.jagla@pasteur.fr>
Manuel Holtgrewe <manuel.holtgrewe@fu-berlin.de>
Maintainer: Manuel Holtgrewe <manuel.holtgrewe@fu-berlin.de>

References

Jagla, B, Holtgrewe, M, Reinert, K: NGS ROI. To appear.

Examples

```
library(ngsroi)

# Load ROI file into data.frame.
#roi = readROI("dmel.bowtie.sam.roi")

# Compute some metrics.
#roi$min = as.numeric(lapply(roi$counts ,min))
#roi$median = as.numeric(lapply(roi$counts ,median))
#roi$mean = as.numeric(lapply(roi$counts ,mean))
#roi$quantile75 = as.numeric(lapply(roi$counts ,quantile, probs=0.75))
#roi$quantile95 = as.numeric(lapply(roi$counts ,quantile, probs=0.95))

# Write data.frame into ROI file again.
#writeROI(roi, "dmel.bowtie.sam.trans.roi");
```

readROI

Read ROI file.

Description

Read ROI file into data.frame.

Usage

```
readROI(file.name)
```

Arguments

file.name The path to the ROI file to read.

Value

The function returns a data.frame with the data from the ROI file.

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (fpname)
{
  fi <- function(x, i) {
    x[i]
  }
  fni <- function(x, i) {
    as.numeric(x[[i]])
  }
  getVec <- function(x) {
    mylen = as.numeric(x[5])
    vecLen = length(x)
    as.numeric(unlist(strsplit(x[vecLen], ",")))
  }
  getVals <- function(y, x, columnNames) {
    vecLen = length(columnNames)
    t1 = vecLen - 1
    for (colN in c(7:t1)) {
      y[, columnNames[colN]] = unlist(lapply(x, fni, colN))
    }
    return(y)
  }
  con = gzfile(fpname)
  rLines = readLines(con)
  close(con)
  values = rLines[substr(rLines, 1, 1) != "#"]
  values = strsplit(values, "\t")
  if (length(rLines[substr(rLines, 1, 2) == "##"]) == 0) {
    columnNames = unlist(list("##ref", "begin_pos", "end_pos",
      "region_name", "length", "strand", "max_count", "cg_content",
      "counts"))
  }
  else {
    columnNames = unlist(strsplit(rLines[substr(rLines, 1,
      2) == "##"], "\t"))
  }
  df = data.frame(ref = unlist(lapply(values, fi, 1)), begin_pos = as.integer(unlist(la
    fni, 2))), end_pos = as.integer(unlist(lapply(values,
    fni, 3))), region_name = unlist(lapply(values, fi, 4)),
    length = as.integer(unlist(lapply(values, fni, 5))),
    strand = unlist(lapply(values, fi, 6)))
  df = getVals(df, values, columnNames)
  df$counts = lapply(values, getVec)
  df$counts = lapply(df$counts, unlist)
  roiNames = names(df)
  return(df)
}
```

Description

Write data.frame to a ROI file.

Usage

```
writeROI(roi, file.name)
```

Arguments

roi	The data.frame to write to file.
file.name	The path to the file to write to.

Examples

```
##----- Should be DIRECTLY executable !! -----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (roi, fpname)
{
  fpConn <- file(fpname, "w")
  colCount = 1
  write("# ROI written from R", fpConn, append = F)
  colNames = names(roi)
  colNames = colNames[!colNames == "counts"]
  outStr = ""
  outStr = paste(c(outStr, "##", colNames[1]), collapse = "")
  for (colIds in c(2:length(colNames))) {
    outStr = paste(c(outStr, colNames[colIds]), collapse = "\t")
  }
  outStr = paste(c(outStr, "counts"), collapse = "\t")
  write(outStr, file = fpConn, append = TRUE)
  apply(roi, 1, writeRoiLine, fpConn)
  close(fpConn)
}
```