Package 'contiBAIT'

October 17, 2020

Type Package

Title Improves Early Build Genome Assemblies using Strand-Seq Data

Version 1.16.0

Date 2020-05-15

Author Kieran O'Neill, Mark Hills, Mike Gottlieb

Maintainer Kieran O'Neill <koneill@bcgsc.ca>

Description Using strand inheritance data from multiple single cells
from the organism whose genome is to be assembled, contiBAIT
can cluster unbridged contigs together into putative
chromosomes, and order the contigs within those chromosomes.

License BSD_2_clause + file LICENSE

Depends BH (>= 1.51.0-3), Rsamtools (>= 1.21)

LinkingTo Rcpp, BH

Imports data.table, grDevices, clue, cluster, gplots, BiocGenerics (>=
0.31.6), S4Vectors, IRanges, GenomicRanges, Rcpp, TSP,
GenomicFiles, gtools, rtracklayer, BiocParallel, DNAcopy,
colorspace, reshape2, ggplot2, methods, exomeCopy,
GenomicAlignments, diagram

Suggests BiocStyle

biocViews ImmunoOncology, CellBasedAssays, QualityControl,
WholeGenome, Genetics, GenomeAssembly

Collate 'AllClasses.R' 'AllGenerics.R' 'BAIT.R'
'barplotLinkageGroupCalls.R' 'clusterContigs.R'
'computeConsensus.R' 'computeSim.R' 'contiBAIT.R'
'findSexGroups.R' 'findSimilarLibraries.R' 'fixLinkageGroups.R'
'flipOrderedLinkageGroups.R' 'highlightAssemblyErrors.R'
'ideogramPlot.R' 'locateMisorients.R' 'makeBoxPlot.R'
'makeChrTable.R' 'mapGapFromOverlap.R' 'mergeFlankedLGs.R'
'orderAllLinkageGroups.R' 'orderContigsGreedy.R'
'orderContigsTSP.R' 'plotContigOrder.R' 'plotLGDistances.R'
'plotWCdistribution.R' 'preprocessStrandTable.R'
'reorientAndMergeLGs.R' 'strandSeqFreqTable.R' 'thoroughBed.R'
'writeBed.R'

RoxygenNote 6.0.1.9000

NeedsCompilation yes

R **topics documented:**

BAIT	*BAIT – master function to process strand-seq libraries into BAIT ideograms*

Description

BAIT – master function to process strand-seq libraries into BAIT ideograms

Usage

```
BAIT(path = ".", splitBy = 2e+05, readQual = 10, pairedEnd = TRUE,
  plotBy = "lib", plotName = NULL, chroms = "all", verbose = TRUE)
```

Arguments

path	String denoting location of Strand-seq bam files (default is ".")
splitBy	numeric value of binsize for plotting. default is 200000
readQual	Integer dictating the minimal mapping quality required for a read to be accepted. Default is 10.
pairedEnd	Whether the bam files being read are in paired end format. Default is TRUE. Note,
plotBy	Whether to plot by library ('lib') or chromosome ('chr')
plotName	character which determines file name to be saved. Default is to open an R plot from the terminal
chroms	vector of chromosome number to prevent contig plotting. eg for humans use 1:24. Default is 'all'
verbose	prints messages to the terminal (default is TRUE)

Value

ideogram plots

Examples

```
## Not run:

#Not run because this takes a minute or two:
bamPath=file.path(system.file(package='contiBAIT'), 'extdata')

BAIT(bamPath, pairedEnd = FALSE) #our example data is single-end

## End(Not run)
```

---

barplotLinkageGroupCalls,LinkageGroupList,ChrTable-method

Bar plot all linkage groups, with the true chromosomes of contigs coloured.

Description

Bar plot all linkage groups, with the true chromosomes of contigs coloured.

Usage

```
## S4 method for signature 'LinkageGroupList,ChrTable'
barplotLinkageGroupCalls(object, chrTable,
  by = "lg", bySize = TRUE, returnTable = NULL, whichGroup = NULL,
  percentage = NULL)
```

Arguments

object	LinkageGroupList, as generated by clusterContigs
chrTable	GRanges object containing assembly information about the contigs, including a meta column called 'name' that has names matching the object. Note that the rownames of chrTable should be the contig names, as they are used in object, and the first column (chromosome name) will used to order by chromosome if 'chr' option used in by parameter. To use a bam file header, the product of makeChrTable(bamFile) is suitable for input
by	whether to plot by linkage group (if 'lg') or chromosomes ('chr')
bySize	logical value to return barplot either with LGs sorted by number of contigs or size (in Mb). Default is TRUE.
returnTable	Logical to return chromosome length matrix. Default is NULL
whichGroup	Numeric vector of groups to analyse. If by='lg', then only the subset of LGs selected by whichGroup will be analyzed. If by='chr', then only subset of chromosomes selected by whichGroup will be analyzed. Note to include legend, use legend=rownames(chr.table) for by='lg', and legend=colnames(chr.table) for by='chr'
percentage	Logical that returns the percentage of different chromosomes or LG within the barplot. Default is NULL

Value

a matrix of lengths of each chromosome (rows) in each linkage group (columns)

Examples

```
data("exampleLGList")
data("exampleDividedChr")

barplotLinkageGroupCalls(exampleLGList, exampleDividedChr)
```

ChrTable-class   *A class for storing chromosome/fragment lengths*

Description

This class is a GRanges object with a meta column called name, which represents the fragment name.

Usage

```
ChrTable(chrRanges = GRanges())
```

Arguments

chrRanges   a GRanges object with a meta column called name, which represents the fragment name

Value

a `ChrTable`

---

clusterContigs,StrandStateMatrix-method

                clusterContigs – agglomeratively clusters contigs into linkage groups
                based on strand inheritance

Description

clusterContigs – agglomeratively clusters contigs into linkage groups based on strand inheritance

Usage

```
## S4 method for signature 'StrandStateMatrix'
clusterContigs(object, similarityCutoff = 0.7,
  recluster = NULL, minimumLibraryOverlap = 5, randomise = TRUE,
  randomSeed = NULL, randomWeight = NULL, clusterParam = NULL,
  clusterBy = "hetero", verbose = TRUE)
```

Arguments

object	data.frame containing strand inheritance information for every contig (rows) in every library (columns). This should be the product of strandSeqFreqTable
similarityCutoff	
	place contigs in a cluster when their strand state is at least this similar
recluster	Number of times to recluster and take the consensus of. If NULL, clustering is run only once.
minimumLibraryOverlap	
	for two contigs to be clustered together, the strand inheritance must be present for both contigs in at least this many libraries (in addition to their similarity being at least similarityCutoff)
randomise	whether to reorder contigs before clustering
randomSeed	random seed to initialize clustering
randomWeight	vector of weights for contigs for resampling. If NULL, uniform resampling is used. Typically this should be a measure of contig quality, such as library coverage, so that clustering tends to start from the better quality contigs.
clusterParam	optional BiocParallelParam specifying cluster to use for parallel execution. When NULL, execution will be serial.
clusterBy	Method for performing clustering. Default is 'hetero' (for comparing heterozygous calls to homozygous). Alternative is 'homo' (for compairson between the two homozygous calls)
verbose	prints function progress

Details

Note that a more stringent similarity cutoff will result in more clusters, and a longer run time, since at every iteration a distance is computed to the existing clusters. However, in lower-quality data, a more stringent cutoff may be necessary to reduce the number of contigs that are erroneously grouped.

Note that clusterParam requires BiocParallel to be installed.

Value

LinkageGroupList of vectors containing labels of contigs belonging to each linkage group

Examples

```
data("exampleWCMatrix")

clusteredContigs <- clusterContigs(exampleWCMatrix, verbose=FALSE)
show(clusteredContigs)
show(clusteredContigs[[1]])
```

---

contiBAIT	*contiBAIT – master function to process strand-seq libraries into ordered linkage groups*

Description

contiBAIT – master function to process strand-seq libraries into ordered linkage groups

Usage

```
contiBAIT(path = ".", cluster = 1, clusterParam = NULL,
  saveName = FALSE, filter = FALSE, readQual = 10, readLimit = 10,
  pairedEnd = TRUE, makePlots = FALSE, verbose = TRUE)
```

Arguments

path	String denoting location of Strand-seq bam files (default is ".")
cluster	Integer denoting the number of reclusterings to be performed for creating linkage groups (default is 1)
clusterParam	Number of parallel processors to use when clustering contigs. Default is NULL.
saveName	String denoting the file name for saved data. If FALSE, no intermediate files are saved (default is FALSE)
filter	additional file to split chromosomes based on locations. If this parameter is blank, a filter table will be automatically generated from the header of the first file in bamFileList
readQual	Integer dictating the minimal mapping quality required for a read to be accepted. Default is 10.
readLimit	Minimum number of reads on a contig to make a strand call. Default is 10
pairedEnd	Whether the bam files being read are in paired end format. Default is TRUE. Note, since paired reads will be the same direction, only first mate read of pair is used in output
makePlots	Logical determining whether plots should be created. Default is TRUE
verbose	prints messages to the terminal (default is TRUE)

Value

ordered contigs in bed format. Depending on options, intermediate files and plots will also be generated

Examples

```
#Get a list of BAM files containing libraries for cells from the same organism, aligned to the same genome
#In this case these are the example BAM files provided with the package (hence the call to system.file);
data("exampleDividedChr")
library(BiocParallel)

example.dir <- file.path(system.file(package='contiBAIT'), 'extdata')

orderedContigs <- contiBAIT(path=example.dir,
```

```
cluster=1,
clusterParam=MulticoreParam(),
filter=exampleDividedChr,
pairedEnd=FALSE)
```

ContigOrdering-class   *A class for storing contig ordering of a linkage group*

Description

This class is a matrix of two character vectors that represent the calculated ordering of a linkage group. The first element of this matrix is the Linkage Group sub-setted by contigs with equal strand states across all libraries in the calculated order. The second element is the names of names of each contig in the calculated order.

Usage

```
ContigOrdering(ordering = character())
```

Arguments

ordering   a matrix of two character vectors that represent the calculated ordering of a linkage group. The first element of this matrix is the Linkage Group sub-setted by contigs with equal strand states across all libraries in the calculated order. The second element is the names of names of each contig in the calculated order.

Value

a ContigOrdering

Examples

```
thisOrdering <- ContigOrdering(matrix(ncol=2, c( "LG1.11", "chr2:1000820-2001640",
"LG1.1", "chr2:3002461-4003281")))
```

---

exampleChrTable   *Example of a ChromosomeTable, containing contigs and their lengths*

Description

Example of a ChromosomeTable, containing contigs and their lengths

---

exampleContigOrder   *Example of a ContigOrdering table, containing a list with a matrix of ordered groups element and a StrandStateList element*

Description

Example of a ContigOrdering table, containing a list with a matrix of ordered groups element and a StrandStateList element

exampleCrickFreq	*An example Crick strand frequences extracted from BAMS by strand-SeqFreqTable where BAITtables=TRUE*

Description

An example Crick strand frequences extracted from BAMS by strandSeqFreqTable where BAITtables=TRUE

exampleDividedChr	*Example of a divided chromosome, containing contigs and their lengths*

Description

Example of a divided chromosome, containing contigs and their lengths

exampleLGList	*Example of a LinkageGroupList output from clusterContigs*

Description

Example of a LinkageGroupList output from clusterContigs

exampleLibList	*Example of a LibraryGroupList, containing library names*

Description

Example of a LibraryGroupList, containing library names

exampleReadCounts	*Example of read counts extracted from BAMS by strandSeqFreqTable*

Description

Example of read counts extracted from BAMS by strandSeqFreqTable

exampleStrandFreq	*Example of strand frequencies extracted from BAMS by strandSeqFreqTable*

Description

Example of strand frequencies extracted from BAMS by strandSeqFreqTable

exampleWatsonFreq	*An example Watson strand frequences extracted from BAMS by strandSeqFreqTable where BAITtables=TRUE*

Description

An example Watson strand frequences extracted from BAMS by strandSeqFreqTable where BAITtables=TRUE

exampleWCMatrix	*An example StrandStateMatrix containing WW, CC and WC calls for contigs*

Description

An example StrandStateMatrix containing WW, CC and WC calls for contigs

findSexGroups,LinkageGroupList,StrandStateMatrix-method	
	Bar plot all linkage groups, with the true chromosomes of contigs coloured.

Description

Bar plot all linkage groups, with the true chromosomes of contigs coloured.

Usage

```
## S4 method for signature 'LinkageGroupList,StrandStateMatrix'
findSexGroups(linkageGroupList,
  strandStateMatrix, callThreshold = 0.2)
```

Arguments

linkageGroupList
> a LinkageGroupList object as generated by clusterContigs

strandStateMatrix
> a StrandStateMatrix object as generated by preprocessStrandTable

callThreshold   a value between 0 and 1 to threshold the proportion of libraries in which scaffolds are heterozygous (ie unlikely to be a monosome sex chromosome). Default value is 0.2

Value

the same LinkageGroupList entered into the function, but with potential sex chromosomes highlighted in the name attribute

---

findSimilarLibraries,StrandStateMatrix,StrandReadMatrix,ChrTable-method

findSimilarLibraries – function to identify libraries that hare similar WC patterns on chromosomes

---

Description

findSimilarLibraries – function to identify libraries that hare similar WC patterns on chromosomes

Usage

```
   ## S4 method for signature 'StrandStateMatrix,StrandReadMatrix,ChrTable'
findSimilarLibraries(strandStateMatrix,
   strandReadMatrix, chrGrange, chrNum, cluster = 1, clusterParam = NULL,
   verbose = TRUE)
```

Arguments

strandStateMatrix
> A strandStateMatrix object for all libraries across split fragments, derived from preprocessStrandTable

strandReadMatrix
> The number of reads present for each strandStateMatrix element. An object of type strandReadMatrix from strandSeqFreqTable

chrGrange   File of type ChrTable (a GRanges object with a meta column titled 'name' determining contig name) to split chromosomes based on locations. name meta should match the rownames of strandStateMatrix and strandReadMatrix

chrNum   The chromosome number to analyse

cluster   Number of times to recluster and take the consensus of. If NULL, clustering is run only once

clusterParam   optional `BiocParallelParam` specifying cluster to use for parallel execution. When `NULL`, execution will be serial.

verbose   prints messages to the terminal (default is TRUE)

Value

a list of type LinkageGroupList with two elements; libraries that are mostly Watson, and those that
are mostly Crick

Examples

```
#Get a list of BAM files containing libraries for cells from the same organism, aligned to the same genome
#In this case these are the example BAM files provided with the package (hence the call to system.file);
data("exampleDividedChr")
data("exampleWCMatrix")
data("exampleReadCounts")

library(BiocParallel)

example.dir <- file.path(system.file(package='contiBAIT'), 'extdata')

chrGrange <- exampleDividedChr[which(exampleDividedChr$name %in% rownames(exampleWCMatrix))]

exampleLibList <- lapply(seq_len(length(unique(seqnames(chrGrange)))), function(x) findSimilarLibraries(exam
exampleLibList <- exampleLibList[!sapply(exampleLibList, is.null)]
exampleLibList <- LibraryGroupList(exampleLibList)

show(exampleLibList)
```

```
fixLinkageGroups,ContigOrdering,StrandStateList,LinkageGroupList-method
```
                    fixLinkageGroups – searches for discrepancies within ordered contigs
                            to highlight erroneously merged fragments.

Description

fixLinkageGroups – searches for discrepancies within ordered contigs to highlight erroneously
merged fragments.

Usage

```
   ## S4 method for signature 'ContigOrdering,StrandStateList,LinkageGroupList'
fixLinkageGroups(contigOrdering,
   orderFrame, linkageGroupList, whichLG = NULL, relatedCutOff = 0.6,
   verbose = TRUE)
```

Arguments

contigOrdering   a data.frame of ordered contigs with linkage group names of class ContigOrder-
                  ing

orderFrame   a list of StrandStateMatrix elements of class StrandStateList

linkageGroupList

                  List of vectors containing names of contigs belonging to each LG of type Link-
                  ageGroupList.

whichLG   vector of integers specifying the element(s) of linkageGroupList to be ordered
                  (i.e. which specific linkage groups to try to order). Default is all LGs.

relatedCutOff	The minimal dissimilarity between adjacent contigs to subset a linkage group into multiple smaller groups. Default is 0.6
verbose	Outputs information to the terminal. Default is TRUE.

Value

a LinkageGroupList with erroneously clustered contigs seperated into their own groups

---

flipOrderedLinkageGroups,ContigOrdering,StrandStateList,LinkageGroupList,StrandStateMatrix-method

flipOrderedLinkageGroups – Simple greedy algorithm to find misorientations within previously ordered LG.

Description

flipOrderedLinkageGroups – Simple greedy algorithm to find misorientations within previously ordered LG.

Usage

```
  ## S4 method for signature
## 'ContigOrdering,StrandStateList,LinkageGroupList,StrandStateMatrix'
flipOrderedLinkageGroups(contigOrdering,
  orderFrame, linkageGroupList, strandStateMatrix, whichLG = NULL,
  maxiter = 100, dissimilarityCutoff = 0.3)
```

Arguments

contigOrdering	a data.frame of ordered contigs with linkage group names of class ContigOrdering
orderFrame	a list of StrandStateMatrix elements of class StrandStateList
linkageGroupList	
	List of vectors containing names of contigs belonging to each LG of type LinkageGroupList. Should be ordered as the output of orderAllLinkageGroups[[3]]
strandStateMatrix	
	a StrandStateMatrix to be reoriented
whichLG	vector of integers specifying the element(s) of linkageGroupList to be ordered (i.e. which specific linkage groups to try to order). Default is all LGs.
maxiter	The number of iterations used to find the optimal path. Default is 100
dissimilarityCutoff	
	The minimum amount of dissimilarity needed to start reorienting contigs. Default is 0.3.

Value

a StrandStateMatrix with misorientations resolved

---

highlightAssemblyErrors

> *highlightAssemblyErrors – Master function to identify misorientations*
> *and chimeras in the assembly*

Description

highlightAssemblyErrors – Master function to identify misorientations and chimeras in the assembly

Usage

```
highlightAssemblyErrors(path, splitBy = 1e+06, cluster = 1,
  clusterParam = NULL, pairedEnd = TRUE, qual = 10, gapFile = NULL,
  verbose = TRUE)
```

Arguments

path	String denoting location of Strand-seq bam files
splitBy	integer determining the average size contigs should be split by
cluster	Number of times to recluster and take the consensus of. If NULL, clustering is run only once.
clusterParam	optional `BiocParallelParam` specifying cluster to use for parallel execution. When `NULL`, execution will be serial.
pairedEnd	Whether the bam files being read are in paired end format. Default is TRUE. Note,
qual	Mapping quality threshold. Default is 1
gapFile	A GRanges object consisting of start and end locations of assembly gaps (defaul it NULL)
verbose	prints messages to the terminal (default is TRUE)

Value

a directional ChrTable object that can be used in downstream functions (strandSeqFreqTable)

---

ideogramPlot,StrandReadMatrix,StrandReadMatrix,ChrTable-method
> *ideogramPlot – plots BAIT-like ideograms*

Description

ideogramPlot – plots BAIT-like ideograms

Usage

```
    ## S4 method for signature 'StrandReadMatrix,StrandReadMatrix,ChrTable'
ideogramPlot(WatsonFreqList,
  CrickFreqList, chrTable, plotBy = "lib", sizeProportional = NULL,
  showPage = NULL, orderFrame = NULL, orientationData = NULL,
  verbose = TRUE)
```

Arguments

WatsonFreqList   data.frame of Watson calls. Product of strandSeqFreqTable[[3]] when BAITta-
bles=TRUE

CrickFreqList   data.frame of Crick calls. Product of strandSeqFreqTable[[4]] when BAITta-
bles=TRUE

chrTable   A data.frame consisting of chromosomes and lengths. Generated by makeChrTable().
Note rownames equal to chromosome names are required

plotBy   Whether to generate a plot for each library ('lib') or a plot for each chromosome
('chr')

sizeProportional
                 The plotter will divide each fragment into a discrete number of bins of size
                 sizeProportional and distribute reads across them. For example, when set to
                 200000, the all contigs in each linkage group will be split into 200kb fragments
                 and each fragment will have an equal proportion of the reads derived from that
                 fragment. This ultimately rescales the ideograms such that LGs with the greatest
                 DNA content are the biggest, as opposed to teh LGs with the most contigs.
                 Default value is NULL

showPage   Integer specifying which LG (if plotBy='chr') or libraries (if plotBy='lib') to
                 plot. Useful when not plotting to a file, or when wishing to subset data. Default
                 is NULL

orderFrame   ordered data.frame of contigs (produced by orderAllLinkageGroups). Default is
                 FALSE, where plots will be made from elements in chrTable.

orientationData
                 ChrTable of contig orientations telling which reads to flip Watson and Crick
                 counts

verbose   prints messages to the terminal (default is TRUE)

Value

ordered contigs in bed format. Depending on options, intermediate files and plots will also be
generated

Examples

```
data("exampleWatsonFreq")
data("exampleCrickFreq")
data('exampleDividedChr')

singleWatsonLibrary <- StrandReadMatrix(exampleWatsonFreq[,2, drop=FALSE])
singleCrickLibrary <- StrandReadMatrix(exampleCrickFreq[,2, drop=FALSE])

ideogramPlot(singleWatsonLibrary, singleCrickLibrary, exampleDividedChr)
```

LibraryGroupList-class

A class for storing library group calls for contigs

Description

This class is a list of lists, each primary list element is a chromosome/contig, and contains 2 sub-list elements: a list of 'Mostly Watson" and "Mostly Crick" library names.

Usage

```
LibraryGroupList(libraryGroups = list(), names = character())
```

Arguments

libraryGroups   a list of lists, with each primary list element representing a chromosome with two internal list elements; a character vector of mostly watson library names, and a character vector of mostly Crick library names

names   a vector of names of linkage groups

Value

a LibraryGroupList

Examples

```
lg1 <- LinkageGroupList(list(a=c('library1', 'library2'), b=c('library3')), names=c('chr1_Mostly_Crick', 'chr
lg2 <- LinkageGroupList(list(a=c('library1'), b=c('library6', 'library4')), names=c('chr2_Mostly_Crick', 'chr
libList <- LibraryGroupList(list(lg1, lg2))
```

LinkageGroupList-class

A class for storing linkage group calls for contigs

Description

This class is simply a list of character strings containing the names of linkage groups.

Usage

```
LinkageGroupList(linkageGroups = list(), names = character())
```

Arguments

linkageGroups   a list of character vectors of names of contigs in each LG

names   a vector of names of linkage groups

Value

a LinkageGroupList

Examples

```
lgList <- LinkageGroupList(list(lg1=c('contig1', 'contig2'), lg2=c('contig3')),
names=c('lg1', 'lg20'))
```

---

locateMisorients,GRanges-method

> *locateMisorients – function to identify libraries that hare similar WC patterns on chromosomes*

Description

locateMisorients – function to identify libraries that hare similar WC patterns on chromosomes

Usage

```
## S4 method for signature 'GRanges'
locateMisorients(compiledGrange, gapFile = NULL,
  stateNum = 3, readCutOff = 40, verbose = TRUE)
```

Arguments

compiledGrange	A GRanges object consisting of read locations. Can be an individual file or the product of thoroughBed
gapFile	A GRanges object consisting of start and end locations of assembly gaps (default NULL)
stateNum	The number of expected strand states. Default is 3 (WW, WC and CC). Function may exhibit unusual behaviour is changed
readCutOff	The minimal number of reads required to make an accurate strand state call. Default is 40.
verbose	prints messages to the terminal (default is TRUE)

Value

a list of ChrTable objects. The first is a ChrTable of misorientations detected. The second is a ChrTable of chimera detected. Output can be used in downstream functions (strandSeqFreqTable)

---

makeBoxPlot,ChrTable,LinkageGroupList-method

> *makeBoxPlot – creates boxplot of contigs included in the analysis vs those excluded*

Description

makeBoxPlot – creates boxplot of contigs included in the analysis vs those excluded

Usage

```
## S4 method for signature 'ChrTable,LinkageGroupList'
makeBoxPlot(chrTable, linkage.contigs)
```

Arguments

chrTable   A GRanges object consisting of contigs and positions. A meta column called
                  'names' must be present with names in the same format as those from link-
                  age.contigs. Generated by makeChrTable().

linkage.contigs
                  A list of clustered contigs, generated by clusterContigs()

Value

a box plot of included and excluded contigs

Examples

```
#make an example barplot of data

data("exampleLGList")
data("exampleChrTable")
makeBoxPlot(exampleChrTable, exampleLGList)
```

---

makeChrTable   *makeChrTable – Pulls out chromosome and length data from the*
                                header of a bam file

Description

makeChrTable – Pulls out chromosome and length data from the header of a bam file

Usage

```
makeChrTable(bamFile, splitFile = NULL, splitBy = NULL, verbose = TRUE)
```

Arguments

bamFile   string of location of a bam file to extract header data from

splitFile   GRanges object (of type chr, start and end: no strand or meta columns) of loca-
                  tions in which to split the assembly, such as previously determined locations of
                  contig chimerism

splitBy   integer determining the average size contigs should be split by

verbose   if FALSE, no messages appear on terminal

Details

makeChrTable creates a table with chromosome name and chromosome length by extracting header
data from the supplied bam file.

Value

a GRanges object of class ChrTable, containing information on the organism's chromosomes as
extracted from the BAM file header.

Examples

```
#Get an example BAM file and generate a chromosome table featuring fragment names and lengths

example.bam <- list.files(file.path(system.file(package='contiBAIT'), 'extdata'), full.names=TRUE)[1]

chrTable <- makeChrTable(example.bam)

show(chrTable)

dividedChr <- makeChrTable(example.bam, splitBy=1000000)

show(dividedChr)
```

mapGapFromOverlap	*mapGapFromOverlap – function to co-localize strand state changes with assembly gaps*

Description

mapGapFromOverlap – function to co-localize strand state changes with assembly gaps

Usage

```
mapGapFromOverlap(sceFile, gapFile, chrTable, verbose = TRUE,
  overlapNum = 4)
```

Arguments

sceFile   GRanges object of strand state change locations in BED format

gapFile   GRanges object of assembly gaps in BED format (can be downloaded from UCSC table browser)

chrTable   GRanges object of chromosome table (product of makeChrTable)

verbose   prints messages to the terminal (default is TRUE)

overlapNum   Minimal number of strand state changes that overlap with a gap before assembly is cut at that location

Value

a GRanges object of all contigs split by regions where the sceFile and gapFile GRanges objects overlap.

---

mergeFlankedLGs,LinkageGroupList,StrandStateMatrix-method

mergeFlankedLGs – searches for similarities at the ends of ordered linkage groups to chain groups together

---

Description

mergeFlankedLGs – searches for similarities at the ends of ordered linkage groups to chain groups together

Usage

```
   ## S4 method for signature 'LinkageGroupList,StrandStateMatrix'
mergeFlankedLGs(linkageGroupList,
   strandStateMatrix, buildConsensus = 1, cluster = NULL,
   clusterParam = NULL, similarityCutoff = 0.7, verbose = TRUE)
```

Arguments

linkageGroupList

        List of ordered vectors containing names of contigs belonging to each LG, of type LinkageGroupList

strandStateMatrix

        Table of type strandStateMatrix encompassing strand state for all contigs. Product of StrandSeqFreqTable.

buildConsensus number of contigs to take at the end of the linkage group to build a consensus strand state. Default is 5

cluster   Number of times to recluster and take the consensus of. If NULL, clustering is run only once.

clusterParam   optional `BiocParallelParam` specifying cluster to use for parallel execution. When NULL, execution will be serial.

similarityCutoff

        merge contigs that are more similar this this

verbose   Outputs information to the terminal. Default is TRUE.

Value

a list containing a revised LinkageGroupList with merged groups, if appropriate, and a StrandStateMatrix with contigs reoriented, if newly merged groups were in opposite orientations.

orderAllLinkageGroups,LinkageGroupList,StrandStateMatrix,StrandFreqMatrix,StrandReadMatrix-method

Function to call contig ordering algorithms iteratively across each linkage group element

Description

Function to call contig ordering algorithms iteratively across each linkage group element

Usage

```
   ## S4 method for signature
## 'LinkageGroupList,StrandStateMatrix,StrandFreqMatrix,StrandReadMatrix'
orderAllLinkageGroups(linkageGroupList,
  strandStateMatrix, strandFreqMatrix, strandReadCount, whichLG = NULL,
  saveOrdered = NULL, orderCall = "greedy", randomAttempts = 75,
  nProcesses = 1, verbose = TRUE)
```

Arguments

linkageGroupList
:   list of vectors, each specifying which contigs belong in which linkage group (product of clusterContigs)

strandStateMatrix
:   table of strand calls for all contigs (product of preprocessStrandTable)

strandFreqMatrix
:   table of W:C read proportions (used for QC) (product of strandSeqFreqTable[[1]])

strandReadCount
:   table of read counts (product of strandSeqFreqTable[[2]])

whichLG
:   vector of integers specifying the element(s) of linkageGroupList to be ordered (i.e. which specific linkage groups to try to order). Default is all LGs.

saveOrdered
:   Will return a pdf of heatmaps for each linkage group; String entered becomes the fileName (default is saveOrderedPDF=FALSE)

orderCall
:   currently either 'greedy' for greedy algorithm or 'TSP' for travelling salesperson alogrithm (default is 'greedy')

randomAttempts
:   itenger specifying number of iterations of either the Monte Carlo or the 2-opt TSP solver to identify the best ordering. Default is 75, but higher values (eg 1000) make more sense for the TSP.

nProcesses
:   number of processes to attempt ordering in parallel

verbose
:   Pringts messages to the terminal. Default is TRUE

Value

a list containing a ContigOrdering object consisting of ordered contigs with linkage group names, a StrandStateList consisting of StrandStateMatrices for each LG, and a LinkageGroupList consisting of reordered contigs within each list element.

Examples

```
#Get a data.frame of ordered contigs from cells from the same organism, aligned to the same genome

data("exampleLGList")
data("exampleWCMatrix")
data("exampleStrandFreq")
data("exampleReadCounts")
contigOrder <- orderAllLinkageGroups(exampleLGList, exampleWCMatrix, exampleStrandFreq, exampleReadCounts)

show(contigOrder)

contigOrder <- orderAllLinkageGroups(exampleLGList, exampleWCMatrix, exampleStrandFreq, exampleReadCounts, o
show(contigOrder)
```

orderContigsGreedy	*Function to order contigs within a single linkage group using a greedy algorithms Attempt to order contigs within*

Description

Function to order contigs within a single linkage group using a greedy algorithms Attempt to order contigs within

Usage

```
orderContigsGreedy(linkageGroupReadTable, randomAttempts = 75,
  nProcesses = 1, verbose = TRUE)
```

Arguments

linkageGroupReadTable

                dataframe of strand calls (product of combineZeroDists or preprocessStrandTable)

randomAttempts   number of times to repeat the greedy algortihm with a random restart

nProcesses   number of processes to attempt ordering in parallel

verbose   whether to print verbose messages

Value

list of two members: 1) contig names in order, 2) the original data.frame entered into function correctly ordered

---

orderContigsTSP   *Attempt to order contigs within linkage groups using travelling sales-person algorithm*

Description

Attempt to order contigs within linkage groups using travelling salesperson algorithm

Usage

```
orderContigsTSP(linkageGroupReadTable, reps)
```

Arguments

linkageGroupReadTable

                dataframe of strand calls (product of combineZeroDists or preprocessStrandTable)

reps   Number of repetitions for he 2-opt TSP solver algorithm

Value

list of two members: 1) contig names in order, 2) the original data.frame entered into function correctly ordered

---

plotContigOrder,ContigOrdering-method
                  Plot ordering of contigs within a single linkage group.

Description

Plot ordering of contigs within a single linkage group.

Usage

```
## S4 method for signature 'ContigOrdering'
plotContigOrder(contigOrder, lg = "all",
  verbose = TRUE)
```

Arguments

contigOrder   matrix from orderAllContigs with the subdivided linkage groups and the names
                of the contigs to plot

lg   Integer specifying the linkage group by which to plot. Default is all

verbose   prints messages to the terminal (default is TRUE)

Value

A ggplot object (which will be plotted automatically if not assigned).

Examples

```
#Get a data.frame of ordered contigs from cells from the same organism, aligned to the same genome

data("exampleContigOrder")

plotContigOrder(exampleContigOrder[[1]], lg=1)
```

---

plotLGDistances,LinkageGroupList,StrandStateMatrix-method

plotLGDistances – plots a heatmap of the distances between linkage groups

---

Description

plotLGDistances – plots a heatmap of the distances between linkage groups

Usage

```
## S4 method for signature 'LinkageGroupList,StrandStateMatrix'
plotLGDistances(object,
  allStrands, lg = "all", labels = TRUE, state = "all",
  alreadyOrdered = FALSE)
```

Arguments

`object`	LinkageGroupList
`allStrands`	StrandStateMatrix for all linkageGroups (usually reoriented by reorientStrandTable)
`lg`	='all' vector of integers to determine which linkage group(s) to plot. 'all' will calculate consensus strand calls for all linkage groups and plot them side by side (default it 'all')
`labels`	=TRUE if TRUE, contig names will be plotted on the axes
`state`	string denoting whether only homozygous states should be used ('homo'; just WW vs CC), if a comparison should be made between homozygous and heterozygous states only ('hetero'; just homo vs hetero), or if all three states ('all'; WW vs WC vs CC). Default is 'all'
`alreadyOrdered`	if TRUE, the function will assume that the linkageGroupList is already ordered and not create a dendrogram. Default is FALSE
`...`	additional parameters to pass to heatmap.2

Value

a heatplot of linkage group calls

Examples

```
data("exampleLGList")
data("exampleWCMatrix")

plotLGDistances(exampleLGList, exampleWCMatrix)
plotLGDistances(exampleLGList, exampleWCMatrix, lg=1)
```

plotWCdistribution,StrandFreqMatrix-method
Creates median distribution boxplots across all libraries and contigs

Description

Creates median distribution boxplots across all libraries and contigs

Usage

```
## S4 method for signature 'StrandFreqMatrix'
plotWCdistribution(object, filterThreshold = 0.8)
```

Arguments

object   object of class StrandFreqMatrix (product of strandSeqFreqTable)

filterThreshold
                numeric value used in assessing the threshold for homozygous strand calls. Default is 0.8.

Value

nothing, just plots.

Examples

```
data("exampleStrandFreq")

plotWCdistribution(exampleStrandFreq, filterThreshold=0.8)
```

preprocessStrandTable,StrandFreqMatrix-method
preprocessStrandTable – remove low quality libraries and contigs before attempting to build a genome

Description

preprocessStrandTable – remove low quality libraries and contigs before attempting to build a genome

Usage

```
## S4 method for signature 'StrandFreqMatrix'
preprocessStrandTable(strandTable,
  strandTableThreshold = 0.8, filterThreshold = 0.8,
  orderMethod = "libsAndConc", lowQualThreshold = 0.9, verbose = TRUE,
  minLib = 10)
```

Arguments

strandTable	data.frame containing the strand table to use as input
strandTableThreshold	
	threshold at which to call a contig WW or CC rather than WC
filterThreshold	
	maximum number of libraries a contig can be NA or WC in
orderMethod	the method to oder contigs. currently libsAndConc only option. Set to FALSE to not order contigs based on library quality
lowQualThreshold	
	background threshold at which to toss an entire library. If NULL, function will not make an overall assessment of library quality. Very chimeric assemblies can appear low quality across all libraries.
verbose	messages written to terminal
minLib	minimum number of libraries a contig must be present in to be included in the output

Value

A list of one matrix and three quality data.frames – 1: a matrix of WW/WC/WW calls for all contigs; 2: the quality of libraries used (based on frequencies outside expected ranges); 3: A data.frame of libraries that are of low quality and therefore excluded from analysis; 5: A ChrTable object containing contigs that are present as WC in more libraries than expected. These are excluded from the strandStateMatrix, but are potentially worth investigating for chimerism.

Examples

```
data("exampleStrandFreq")

strandStates <- preprocessStrandTable(exampleStrandFreq, lowQualThreshold=0.8)

show(strandStates[[1]]) # WW-WC-CC matrix
```

---

reorientAndMergeLGs,LinkageGroupList,StrandStateMatrix-method

reorientAndMergeLGs uses a simple dissimilarity to find misoriented fragments within linkage groups.

Description

reorientAndMergeLGs uses a simple dissimilarity to find misoriented fragments within linkage groups.

Usage

```
## S4 method for signature 'LinkageGroupList,StrandStateMatrix'
reorientAndMergeLGs(object,
  allStrands, cluster = NULL, clusterParam = NULL, similarityCutoff = 0.9,
  verbose = TRUE)
```

Arguments

`object`	List of vectors containing names of contigs belonging to each LG.
`allStrands`	Table of type `strandStateMatrix` encompassing strand state for all contigs. Product of StrandSeqFreqTable.
`cluster`	Number of times to recluster and take the consensus of. If NULL, clustering is run only once.
`clusterParam`	optional `BiocParallelParam` specifying cluster to use for parallel execution. When NULL, execution will be serial.
`similarityCutoff`	
	merge contigs that are more similar this this
`verbose`	Outputs information to the terminal. Default is TRUE.

Value

a list consisting of a strandStateMatrix (a reoriented version of allStrands), a ChrTable containing contig names and orientations, as '+' or '-' and a merged LinkageGroupList.

Examples

```
data(exampleLGList)
data(exampleWCMatrix)

reorientedMatrix <- reorientAndMergeLGs(exampleLGList,
  exampleWCMatrix)

# Note that in this example data, everything is correctly oreiented to
# to begin with, so all contigs come out as + orientation
```

---

show,ContigOrdering-method
                    show-methods

Description

Shows a ContigOrdering

Usage

```
## S4 method for signature 'ContigOrdering'
show(object)
```

Arguments

`object`	a ContigOrdering

Value

nothing

---

show,LibraryGroupList-method
                              show-methods

Description

Shows a LibraryGroupList

Usage

```
## S4 method for signature 'LibraryGroupList'
show(object)
```

Arguments

object   a LibraryGroupList

Value

nothing

---

show,LinkageGroupList-method
                              show-methods

Description

Shows a LinkageGroupList

Usage

```
## S4 method for signature 'LinkageGroupList'
show(object)
```

Arguments

object   a LinkageGrouplist

Value

nothing

show,OrientationFrame-method
show-methods

Description

Shows a OrientationFrame

Usage

```
## S4 method for signature 'OrientationFrame'
show(object)
```

Arguments

object   a OrientationFrame

Value

nothing

show,StrandFreqMatrix-method
show-methods

Description

Shows a StrandFreqMatrix

Usage

```
## S4 method for signature 'StrandFreqMatrix'
show(object)
```

Arguments

object   a StrandFreqMatrix

Value

nothing

---

show,StrandReadMatrix-method

show-methods

Description

Shows a StrandReadMatrix

Usage

```
## S4 method for signature 'StrandReadMatrix'
show(object)
```

Arguments

object   a StrandReadMatrix

Value

nothing

---

show,StrandStateList-method

show-methods

Description

Shows a StrandStateList

Usage

```
## S4 method for signature 'StrandStateList'
show(object)
```

Arguments

object   a StrandStateList

Value

nothing

show,StrandStateMatrix-method
show-methods

Description

Shows a StrandStateMatrix

Usage

```
## S4 method for signature 'StrandStateMatrix'
show(object)
```

Arguments

object   a StrandStateMatrix

Value

nothing

StrandFreqMatrix-class
A class for storing a matrix of frequencies of Watson to Crick reads for a set of contigs over several libraries

---

Description

The strand information stored in this object is the ratio of Watson to Crick reads mapping to each contig in each library (cell). This should fall within the range (-1,1). This class simply extends matrix, but with additional validity checking.

Usage

```
StrandFreqMatrix(counts = matrix(double()))
```

Arguments

counts   a double matrix of read count ratios

Value

a StrandFreqMatrix

Examples

```
data("exampleWatsonFreq")
data("exampleCrickFreq")
frequencyMatrix <- sapply(1:ncol(exampleCrickFreq),
function(colNum){exampleCrickFreq[,colNum] / exampleWatsonFreq[,colNum]})

StrandFreqMatrix(frequencyMatrix)
```

`StrandReadMatrix-class`

A class for storing read counts for a set of contigs over several libraries

---

Description

The information stored in this class is simple read counts, so should be integers >=0.

Usage

```
StrandReadMatrix(counts = matrix(integer()))
```

Arguments

counts   an integer matrix of read counts

Value

a `StrandReadMatrix`

Examples

```
data("exampleWatsonFreq")
StrandReadMatrix(exampleWatsonFreq[,2, drop=FALSE])
```

---

strandSeqFreqTable   *strandSeqFreqTable – function to process bam files for contiBAIT*

Description

strandSeqFreqTable – function to process bam files for contiBAIT

Usage

```
strandSeqFreqTable(bamFileList, fieldSep = ".", field = 1, qual = 0,
  rmdup = TRUE, verbose = TRUE, filter = NULL, misorientations = NULL,
  chimeras = NULL, tileChunk = 1e+05, pairedEnd = TRUE,
  BAITtables = FALSE)
```

Arguments

bamFileList   vector containing the location of the bams file to be read

fieldSep   The field seperator of the bam file to use to define the field. Default is '.'

field   The field of the bam file name to use as an index (default is 1)

qual   Mapping quality threshold. Default is 0

rmdup   remove duplicates in output file. Default is TRUE

verbose   prints messages to the terminal (default is TRUE)

filter	additional file of type ChrTable (GRanges with a meta column titled 'name' determining contig name) to split chromosomes based on locations. If this parameter is NULL (default), a filter table will be automatically generated from the header of the first file in bamFileList.
misorientations	
	additional file of type ChrTable (GRanges with a meta column titled 'name' determining contig name) with putative misorientation locations. These location's reads are reversed in the regions of filter that they lie within. Product of locateMisorients[[1]]. Default is NULL
chimeras	additional file of type ChrTable (GRanges with a meta column titled 'name' determining contig name) with putative chimeric locations. These location's reads are masked in the regions of filter that they lie within, and are appended to the end of the output file. Product of locateMisorients[[2]]. Default is NULL
tileChunk	Number of reads to split bam files into (smaller number requires less RAM). Default is 100000.
pairedEnd	Whether the bam files being read are in paired end format. Default is TRUE. Note, since paired reads will be the same direction, only first mate read of pair is used in output
BAITtables	creates additional matrices in the returned list with just Watson and Crick read counts to be used in downstreat BAIT plotting. Default is FALSE

Value

a list containing two matrices: a StrandFreqMatrix of W:C read frequencies, and a StrandReadMatrix of read counts

Examples

```
#Get a list of BAM files containing libraries for cells from the same organism, aligned to the same genome
#In this case these are the example BAM files provided with the package (hence the call to system.file);

example.dir <- file.path(system.file(package='contiBAIT'), 'extdata')
bam.files <- dir(example.dir, full.names=TRUE)

strand.freq <- strandSeqFreqTable(bam.files, pairedEnd = FALSE)

show(strand.freq[[1]])
show(strand.freq[[2]])
```

StrandStateList-class   *A class for storing StrandStateList lists for contigs*

Description

This class is a list of StrandStateMatrices, each a subset of the StrandStateMatrix split by linkagegroups

Usage

```
StrandStateList(strandGroupList = list(), names = character())
```

Arguments

strandGroupList

a list of StrandStateMatrix elements, with each primary element representing a
StrandStateMatrix containing ordered contigs from a LinkageGroupList element

names   a vector of names of StrandStateMatrix elements

Value

a `StrandStateList`

StrandStateMatrix-class

*A class for storing a data frame of discrete strand states of a set of
contigs over several libraries*

Description

The strand information stored in this object is a call of the strand state of each contig in each
library. mapping to each contig in each library (cell). This should fall within the range (-1,1).
This class simply extends matrix, but with additional validity checking.

Usage

```
StrandStateMatrix(states = matrix(integer()))
```

Arguments

states   an integer matrix of strand states by library

Value

a `StrandStateMatrix`

Examples

```
StrandStateMatrix(matrix(ncol=2, c(1,3,1,2)))
```

---

thoroughBed,ANY,LibraryGroupList-method

*thoroughBed – function to merge chromosomes from libraries that
have the same strand states*

Description

thoroughBed – function to merge chromosomes from libraries that have the same strand states

Usage

```
## S4 method for signature 'ANY,LibraryGroupList'
thoroughBed(bamFileList, relatedLibList,
  qual = 10, pairedEnd = TRUE, rmdup = TRUE, verbose = TRUE)
```

Arguments

`bamFileList`	vector containing the location of the bams file to be read
`relatedLibList`	list where each element contains all library names that show similar strand pattern. The product of findSimilarLibraries
`qual`	Mapping quality threshold. Default is 10
`pairedEnd`	Whether the bam files being read are in paired end format. Default is TRUE. Note, since paired reads will be the same direction, only first mate read of pair is used in output to reduce file size
`rmdup`	Logical as to whether to remove duplicate reads within each library. Default is TRUE.
`verbose`	prints messages to the terminal (default is TRUE)

Value

a GRanges object comprising merged directional reads from all libraries in relatedLibList.

Examples

```
#Get a list of BAM files containing libraries for cells from the same organism, aligned to the same genome
#In this case these are the example BAM files provided with the package (hence the call to system.file);
data("exampleLibList")


library(BiocParallel)

example.dir <- file.path(system.file(package='contiBAIT'), 'extdata')

exampleRange <- thoroughBed(example.dir, exampleLibList)

show(exampleRange)
```

---

writeBed,ANY,ChrTable,ContigOrdering-method
                    function to write contig order to BED file

---

Description

function to write contig order to BED file

Usage

```
## S4 method for signature 'ANY,ChrTable,ContigOrdering'
writeBed(chrTable, orientationData,
  contigOrder, libWeight = NULL, file = "contiBAIT_assembly.bed")
```

Arguments

chrTable   a GRanges object with a 'name' meta column matching contig names. Product
                    of makeChrTable

orientationData
                    ChrTable of contig and strand (with rownames matching contig names). Product
                    of reorientAndMergeLGs[[2]]

contigOrder   an object of type ContigOrdering with ordered Linkage Groups and contigs.
                    Product of orderAllLinkageGroups

libWeight   average quality across all libraries for a contig

file   character string for bed file name to write

Value

NULL; BED file written to file

Examples

```
## Not run:

data("exampleDividedChr")

writeBed(exampleDividedChr,
 reorientedMatrix[[2]],
 contigOrder)

## End(Not run)
```

Index