Package 'comradesOO'

November 10, 2023

Type Package

Title Analysis of COMRADES (Cross-Linking Matched RNA and Deep Sequencing) Data

Version 0.1.1

Maintainer Jonathan Price <jlp76@cam.ac.uk>

Description Analysis of RNA crosslinking data for RNA structure prediction. The package is suitable for the analysis of RNA structure cross-linking data and chemical probing data.

License GPL-3

Encoding UTF-8

BugReports https://github.com/JLP-BioInf/comradesOO/issues

Depends seqinr, GenomicRanges, stats

Imports ggplot2, reshape2, MASS, mixtools, utils, S4Vectors, doParallel, igraph, R4RNA, RColorBrewer, IRanges, foreach,grDevices, heatmap3, TopDom, tidyverse, RRNA, ggrepel, methods, parallel

RoxygenNote 7.2.1

Collate 'comradesOO.R' 'comradesDataSet.R' 'clusterComrades.R' 'clusterComradesMethodsAndHelpers.R' 'commonHelpersAndMethods.R' 'commonStatsAndPlots.R' 'comradesDataSetMethodsAndHelpers.R' 'comradesOO-package.R' 'foldComrades.R' 'foldComradesMethodsAndHelpers.R' 'genericMethods.R'

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Jonathan Price [aut, cre]

Repository CRAN

Date/Publication 2023-11-10 13:00:02 UTC

R topics documented:

`clusterComrades` *clusterComrades*

Description

This method clusters the duplexes.

Usage

```
clusterComrades(cds, cores = 3, stepCount = 2, clusterCutoff = 20)
```

Arguments

`cds`	comradesDataSet object created with comradesDataSet
`cores`	numeric - The number of cores to use
`stepCount`	Stringency for clustering
`clusterCutoff`	The minimum number of reads a cluster requires

Value

A `comradesDataSet` object

Examples

```
cds = makeExampleComradesDataSet()

clusterComrades(cds,
                cores = 1,
                stepCount = 1,
                clusterCutoff = 0)
```

`clusterGrangesList` *clusterGrangesList*

Description

Extract the cluster coordinates in granges format

Usage

```
clusterGrangesList(x)
```

Arguments

x	A comradesDataSet object

Value

A list of Granges objects showing the positions of each cluster, one entry for each sample

Examples

```
cds = makeExampleComradesDataSet()

clusterGrangesList(cds)
```

`clusterGrangesList<-` *clusterGrangesList<-*

Description

Set new clusterGrangesList slot

Usage

```
clusterGrangesList(x) <- value
```

Arguments

x	A comradesDataSet object
value	A replacement

Value

No return - Sets a new clusterGrangesList slot

Examples

```
cds = makeExampleComradesDataSet()

newclusterGrangesList <- clusterGrangesList(cds)
clusterGrangesList(cds) <- newclusterGrangesList
```

clusterNumbers *clusterNumbers*

Description

This method prints a table showing the number of clusters in each step of the analysis

Usage

```
clusterNumbers(knowClusteredCds, rna)
```

Arguments

knowClusteredCds	A comradesDataSet object after clustering has been performed
rna	The RNA ID of interest - use rna(cdsObject).

Value

A data.frame shoing the number of clusters for each sample

Examples

```
cds = makeExampleComradesDataSet()

clusteredCds = clusterComrades(cds,
                cores = 1,
                stepCount = 1,
                clusterCutoff = 1)
clusterNumbers(clusteredCds)
```

clusterTableFolded *clusterTableFolded*

Description

Extract the cluster coordinates with fold prediciton in data frame format

Usage

```
clusterTableFolded(x)
```

Arguments

x	A comradesDataSet object

Value

A table showing the vienna structures of each cluster

Examples

```
cds = makeExampleComradesDataSet()

clusterTableFolded(cds)
```

`clusterTableList` *clusterTableList*

Description

Extract the cluster coordinates in data frame format

Usage

```
clusterTableList(x)
```

Arguments

x	A comradesDataSet object

Value

A list of tables showing the vienna structures of each cluster

Examples

```
cds = makeExampleComradesDataSet()

clusterTableList(cds)
```

`clusterTableList<-` *clusterTableList<-*

Description

Set new clusterTableList slot

Usage

```
clusterTableList(x) <- value
```

Arguments

x	A comradesDataSet object
value	A replacement

Value

No return - Sets a new clusterTableList slot

Examples

```
cds = makeExampleComradesDataSet()

newclusterGrangesList <- clusterTableList(cds)
clusterTableList(cds) <- newclusterGrangesList
```

---

compareKnown *compareKnown*

Description

This method compares the current object to a know structure.run `trimClusters()` on the `comradesDataSet` first

Usage

```
compareKnown(trimmedClusters, knownMat, type)
```

Arguments

trimmedClusters	a `comradesDataSet` object, run `trimClusters()` on the `comradesDataSet` first
knownMat	Matrix - A marix(ncol = lengthRNA,nrow = lengthRNA) where a value in matrix[x,y] would indicate a known interation between nucleotide x and nucleotide y
type	string - the Analysis stage of clusters you would like to compare you can find available types by just running the objects name

Value

Returns a `comradesClusteredDataSet` object

The 3 attributes matrixList, clusterTableList and clusterGrangesList will gain the `types` "known" and "novel" and "knownAndNovel"

Examples

```
cds = makeExampleComradesDataSet()

clusteredCds = clusterComrades(cds,
                cores = 1,
                stepCount = 1,
                clusterCutoff = 0)
knownMat = matrix(0, ncol = rnaSize(cds), nrow = rnaSize(cds))
knownMat[7,27] = 1
# use compare known to gett he known and not know clusters
knowClusteredCds = compareKnown(clusteredCds,
                                knownMat,
                                "original")
clusterNumbers(knowClusteredCds)
```

`compareKnownStructures` *compareKnownStructures*

Description

This method compares the predicted structures to a set of known interactions

Usage

```
compareKnownStructures(foldedCds, file)
```

Arguments

`foldedCds`	`comradesDataSet` after running foldComrades
`file`	a two column file with column header i and j with numeric values showing nucleoide i binds to nucleotide j

Value

Returns a dataframe

a tables showing the number of predicted interactions and their agreement

Examples

```
## Not run:
cds = makeExampleComradesDataSet()
clusteredCds = clusterComrades(cds = cds,
                               cores = 3,
                               stepCount = 2,
                               clusterCutoff = 1)
```

```
trimmedClusters = trimClusters(clusteredCds = clusteredCds,trimFactor = 1, clusterCutoff = 1)



fasta = paste(c(rep('A',25),
                rep('T',25),
                rep('A',10),
                rep('T',23)),collapse = "")

header = '>transcript1'


fastaFile = tempfile()
writeLines(paste(header,fasta,sep = "\n"),con = fastaFile)


rnaRefs = list()
rnaRefs[[rnas(cds)]] = read.fasta(fastaFile)
rnaRefs



foldedCds = foldComrades(trimmedClusters,
                         rnaRefs = rnaRefs,
                         start = 1,
                         end = 83,
                         shape = 0,
                         ensembl = 5,
                         constraintNumber  = 1,
                         evCutoff = 1)


# make an example table of "know" interactions
file = data.frame(V1 = c(6),
                  V2 = c(80))
compareKnownStructures(foldedCds,file)

## End(Not run)
```

---

comradesDataSet-class *comradesDataSet*

Description

comradesDataSet objects are used to store the input meta-data, data and create a framework for the storage of results. Whilst creating the object, the original hyb files are also filtered for the RNA of interest. Check the package vignette for more information.

Usage

```
comradesDataSet(rnas, rnaSize = 0, sampleTable)
```

Arguments

rnas	vector - The names of the RNA interest, these must be displayed the same way as in the input Hyb Files.
rnaSize	named list - The sizes (nt) of the RNAs of interest, the list elements must have same names as the rnas vector and each each contain one numeric value.
sampleTable	string - The address of the sample table, the sample table must have 4 columns, fileName (the full path and file name of the input hyb file for each sample), group ("s" - sample or "c" - control), sample (1,2,3, etc), sampleName (must be unique).

Value

A comradesDataSet object.

Slots

clusterTableFolded table - a table similar to the clusterTableList it contains coordinates of the clusters along with vienna format fold and RNA sequences for each cluster

clusterTableList List - Follows the pattern for list slots of comradesDataSet objects, matrixList(cds)[[rna]][[type]] contains a table with coordinates and information about the clusters identified

clusterGrangesList List - Follows the pattern for list slots of comradesDataSet objects, matrixList(cds)[[rna]][[type] contains GRanges objects of the original duplexes with their cluster membership

sampleTable table - Column names; fileName, group (s or c), sample (1,2,3, etc), sampleName (must be unique)

rnas string - a single RNA to analyse - must be present in rnas(cdsObject)

rnaSize if set to 0 this will be calculated

matrixList List - Follows the pattern for list slots of comradesDataSet objects, matrixList(cds)[[rna]][[type]][[samp Contains a set of contact matrices, each cell contains the number of duplexes identified for position x,y.

hybFiles List - Follows the pattern for list slots of comradesDataSet objects, hybFiles(cds)[[rna]][[type]][[sample]] Contains a set of tables, these are the original Hyb files that were read in.

interactionTable Table of interactions discovered in step1 of the folding

viennaStructures List of vienna format structures from final prediction

dgs List of free energies

Examples

```
# make example input
c4 = c(rep("transcript1",100),rep("transcript2",100) )
c10 = c(rep("transcript1",200) )
c1 = 1:200
```

```
c2 = rep(paste(rep("A", 40), collapse = ""),200)
c3 = rep(".",200)
c9 = rep(".",200)
c15 = rep(".",200)
c5 = rep(1,200)
c11 = rep(21,200)
c6 = rep(20,200)
c12= rep(40,200)
# short distance 50
c7 = sample(1:5, 50, replace = TRUE)
c8 = sample(20:25, 50, replace = TRUE)
c13 = sample(20:25, 50, replace = TRUE)
c14 = sample(40:45, 50, replace = TRUE)
# long distance 50
c7 = c(c7,sample(1:5, 50, replace = TRUE))
c8 = c(c8,sample(20:25, 50, replace = TRUE))
c13 = c(c13,sample(60:70, 50, replace = TRUE))
c14 = c(c14,sample(80:83, 50, replace = TRUE))
# inter RNA 100
c7 = c(c7,sample(1:5, 100, replace = TRUE))
c8 = c(c8,sample(20:25, 100, replace = TRUE))
c13 = c(c13,sample(1:5, 100, replace = TRUE))
c14 = c(c14,sample(20:25, 100, replace = TRUE))
exampleInput = data.frame(V1 = c1,
                          V2 = c2,
                          V3 = c3,
                          V4 = c4,
                          V5 = as.numeric(c5),
                          V6 = as.numeric(c6),
                          V7 = as.numeric(c7),
                          V8 = as.numeric(c8),
                          V9 = c9,
                          V10 = c10,
                          V11 = as.numeric(c11),
                          V12 = as.numeric(c12),
                          V13 = as.numeric(c13),
                          V14 = as.numeric(c14),
                          V15 = c15)
file = tempfile()
write.table(exampleInput,file = file,
quote = FALSE,
row.names = FALSE, sep = "\t", col.names = FALSE)


# Set up the sample table
sampleTabler1 = c(file, "s", "1", "s1")
sampleTabler2 = c(file, "c", "1", "c1")
# make the sample table
sampleTable2 = rbind.data.frame(sampleTabler1, sampleTabler2)
# add the column names
colnames(sampleTable2) = c("file", "group", "sample", "sampleName")

# Choose RNA and set up the object ----
```

```
rna = c("transcript1")


# load the object
cds = comradesDataSet(rnas = rna,
                      rnaSize = 0,
                      sampleTable = sampleTable2)

cds
```

`featureInfo` *featureInfo*

Description

Produces a list list of 2 elemnts 'transcript' and 'family' Each element contains a table with the counts for each RNA in each sample.

Usage

```
featureInfo(cds)
```

Arguments

`cds`	a `comradesDataSet` object

Value

A list - Feature level and transcript level counts for each sample

Examples

```
cds = makeExampleComradesDataSet()
featureInfo(cds)
```

`findBasePairsRNAcoFold2`
findBasePairsRNAcoFold2

Description

Folds the clusters using Vienna RNAfold

Usage

```
findBasePairsRNAcoFold2(
  startPos1,
  endPos1,
  seq1,
  startPos2,
  endPos2,
  seq2,
  fasta,
  shape
)
```

Arguments

startPos1	Start of the cluster side x
endPos1	End of the cluster side x
seq1	Sequence of x
startPos2	Start of the cluster side y
endPos2	End of the cluster side y
seq2	Sequence of y
fasta	rnaRefs
shape	shape reactivities

Value

A table of clusters and coordinates with folds

---

findBasePairsRNAfold *findBasePairsRNAfold*

Description

Folds the clusters using Vienna RNA duplex

Usage

```
findBasePairsRNAfold(startPos, endPos, seqs, fasta, shape)
```

Arguments

startPos	Start of the cluster side x
endPos	End of the cluster side x
seqs	Sequence of x
fasta	rnaRefs
shape	shape reactivities

Value

A table of clusters and coordinates with folds

`findBasePairsRNAfold2` *findBasePairsRNAfold2*

Description

Folds the clusters using Vienna RNA duplex

Usage

```
findBasePairsRNAfold2(startPos, endPos, seqs, fasta)
```

Arguments

`startPos`	Start of the cluster side x
`endPos`	End of the cluster side x
`seqs`	Sequence of x
`fasta`	`rnaRefs`

Value

A table of clusters and coordinates with folds

`foldComrades` *foldComrades*

Description

This methods folds an ensebl of structures for the whole RNA or chosen region of the RNA. See `comradesDataSet` for slot information.

Usage

```
foldComrades(
  cdsObject,
  rnaRefs,
  start,
  end,
  evCutoff = 1,
  ensembl = 50,
  constraintNumber = 20,
  shape = 0
)
```

Arguments

cdsObject	comradesDataSet object created with comradesDataSet
rnaRefs	named List - a list with named elements that correspond to the .RNA of interest. The element of the list must be a fasta file that has been read with seqinr::read.fasta()
start	Start of segmnent to fold
end	End of segmnent to fold
evCutoff	Mininum number of read support for contraint to be included in folding
ensembl	Number of structures to Nake
constraintNumber	Number of constraints to add to each final fold
shape	shape reactivities (0 for no constraints)

Value

a comradesDataSet object

Examples

```
## Not run:
cds = makeExampleComradesDataSet()

clusteredCds = clusterComrades(cds,
                cores = 1,
                stepCount = 1,
                clusterCutoff = 0)

trimmedClusters = trimClusters(clusteredCds = clusteredCds,
             trimFactor = 1,
             clusterCutoff = 0)



fasta = paste(c(rep('A',25),
                rep('T',25),
                rep('A',10),
                rep('T',23)),collapse = "")

header = '>transcript1'


fastaFile = tempfile()
writeLines(paste(header,fasta,sep = "\n"),con = fastaFile)


rnaRefs = list()
rnaRefs[[rnas(cds)]] = read.fasta(fastaFile)
rnaRefs
```

```
foldedCds = foldComrades(trimmedClusters,
                         rnaRefs = rnaRefs,
                         start = 1,
                         end = 83,
                         shape = 0,
                         ensembl = 5,
                         constraintNumber  = 1,
                         evCutoff = 1)
foldedCds

## End(Not run)
```

`getAdjacancyMat` *getAdjacancyMat*

Description

Makes and adjacency matrix list (for clustering)

Usage

```
getAdjacancyMat(hybGranges, nucletideOrPerc, cutoff)
```

Arguments

`hybGranges`	list created with hybToGRanges (but just the gap section of the list)
`nucletideOrPerc`	measure difference by percentage or nucleotides
`cutoff`	The maximum difference before giving these two gaps 0

Details

Makes and adjacency matrix list (for clustering)

Value

A list of Adjacancy matrices

getClusterClusterShortRangeWhole *getClusterClusterShortRangeWhole*

Description

Decides if a cluster is long or short range then either grabs the whole sequence or the sequence of the two sides of the interaction separately.

Usage

```
getClusterClusterShortRangeWhole(cluster, seq)
```

Arguments

cluster	cluster positions
seq	sequence of transcript

Value

The same table with an extra column

getData *getData*

Description

Get data is more generic method for retrieving data from the object and returns a list, the number of entries in the list is number of samples in the dataset and the list contain entries of the data type and analysis stage you select.

Usage

```
getData(x, data, type)
```

Arguments

x	A comradesDataSet object
data	The data type to return <hybFiles | matrixList | clusterGrangesList | clusterTableList>
type	The analysis stage <original | noHost | originalClusters | trimmedClusters>

Value

A list of the chosen data type - one entry for each sample

Examples

```
cds = makeExampleComradesDataSet()

getData(cds, 'matrixList','original')
```

`getInteractions` *getInteractions*

Description

This method returns a table interactions of an RNA (interactor) on the RNA of interest use topInteracters.

Usage

```
getInteractions(cds, interactor)
```

Arguments

`cds`	a `comradesDataSet` object
`interactor`	The rna to show interactions with

Value

A table showign the read coverage of the interacting RNA

Examples

```
cds = makeExampleComradesDataSet()
getInteractions(cds, 'transcript2')
```

`getMatrices` *getMatrices*

Description

Make a matrix of contact interactions

Usage

```
getMatrices(hybList, rna, size)
```

Arguments

hybList	the original hybList created with readHybFiles or subsetHybList
rna	the RNA of interest that you want to subset
size	The size of the RNA

Details

Function used to create a list of matrices for plotting with plotMatrixList or plotMatrixListFull, the output list will be same as the input except for an extra list layer for the specific RNA

Value

A list of matrices

---

getReverseInteractions *getReverseInteractions*

Description

This method prints interactions of the RNA of interest on another RNA transcript.

Usage

```
getReverseInteractions(cds, interactor)
```

Arguments

cds	a comradesDataSet object
interactor	The rna to show interactions with

Value

A long format table shoing the read coverage of chosen RNA

Examples

```
cds = makeExampleComradesDataSet()
getReverseInteractions(cds, 'transcript2')
```

group *group*

Description

Extract the indeces for each group for the instance

Usage

```
group(x)
```

Arguments

x	A comradesDataSet object

Value

A list - The indices of the sample in the control and sample groups

Examples

```
cds = makeExampleComradesDataSet()

group(cds)
```

hybFiles *hybFiles*

Description

Extract the data in original format

Usage

```
hybFiles(x)
```

Arguments

x	A comradesDataSet object

Value

A list of tables in the original input format, one entry for each sample

Examples

```
cds = makeExampleComradesDataSet()

hybFiles(cds)
```

hybToGRanges *hybToGRanges*

Description

This function is useful to turn a list of hyb data into lists of GRanges It creates a list for each sample one for the left side one for the right side and one for the gap in the middle.

Usage

```
hybToGRanges(hybList, rna)
```

Arguments

hybList	the original hybList created with readHybFiles or subsetHybList
rna	The rna of interest

Value

A list of GRanges data in hyb format

---

makeExampleComradesDataSet *makeExampleComradesDataSet*

Description

Creat a minimal example comradesdataSetObject

Usage

```
makeExampleComradesDataSet()
```

Value

An example comradesDataSet objext

Examples

```
cds = makeExampleComradesDataSet()
```

matrixList *matrixList*

Description

Extract the contact matrices

Usage

```
matrixList(x)
```

Arguments

x	A comradesDataSet object

Value

A list of contract matrices, one entry for each sample

Examples

```
cds = makeExampleComradesDataSet()

matrixList(cds)
```

matrixList<- *matrixList*

Description

Set new matrixList slot

Usage

```
matrixList(x) <- value
```

Arguments

x	A comradesDataSet object
value	A replacement

Value

No return - Sets a new matrixList slot

Examples

```
cds = makeExampleComradesDataSet()

newMatrixList <- matrixList(cds)
matrixList(cds) <- newMatrixList
```

---

plotComparisonArc *plotComparisonArc*

Description

This method plots two structures chosen from the plotEnsemblePCA method

Usage

```
plotComparisonArc(foldedCds, s1 = "s1", s2 = "s2", n1 = 1, n2 = 2)
```

Arguments

foldedCds	comradesDataSet after running foldComrades
s1	sample of structure 1
s2	sample of structure 2
n1	number of structure from first sample
n2	number of structure from first sample

Value

an ark diagram of the two strcutures

Examples

```
## Not run:
cds = makeExampleComradesDataSet()
clusteredCds = clusterComrades(cds = cds,
                               cores = 3,
                               stepCount = 2,
                               clusterCutoff = 1)


trimmedClusters = trimClusters(clusteredCds = clusteredCds,trimFactor = 1, clusterCutoff = 1)



fasta = paste(c(rep('A',25),
                rep('T',25),
                rep('A',10),
                rep('T',23)),collapse = "")
```

```
header = '>transcript1'


fastaFile = tempfile()
writeLines(paste(header,fasta,sep = "\n"),con = fastaFile)


rnaRefs = list()
rnaRefs[[rnas(cds)]] = read.fasta(fastaFile)
rnaRefs



foldedCds = foldComrades(trimmedClusters,
                         rnaRefs = rnaRefs,
                         start = 1,
                         end = 83,
                         shape = 0,
                         ensembl = 5,
                         constraintNumber  = 1,
                         evCutoff = 1)


plotComparisonArc(foldedCds,"s1","s1",1,3)

## End(Not run)
```

`plotEnsemblePCA` *plotEnsemblePCA*

Description

This method plots a PCA of the ensembl

Usage

```
plotEnsemblePCA(foldedCds, labels = TRUE, split = TRUE)
```

Arguments

`foldedCds`	`comradesDataSet` after running foldComrades
`labels`	plot with labels or not (TRUE/FALSE)
`split`	split the plot using facets based on the samples (TRUE/FALSE)

Value

a PCA plot of the ensembl

Examples

```
## Not run:
cds = makeExampleComradesDataSet()
clusteredCds = clusterComrades(cds = cds,
                               cores = 3,
                               stepCount = 2,
                               clusterCutoff = 1)


trimmedClusters = trimClusters(clusteredCds = clusteredCds,trimFactor = 1, clusterCutoff = 1)



fasta = paste(c(rep('A',25),
                rep('T',25),
                rep('A',10),
                rep('T',23)),collapse = "")

header = '>transcript1'


fastaFile = tempfile()
writeLines(paste(header,fasta,sep = "\n"),con = fastaFile)


rnaRefs = list()
rnaRefs[[rnas(cds)]] = read.fasta(fastaFile)
rnaRefs



foldedCds = foldComrades(trimmedClusters,
                         rnaRefs = rnaRefs,
                         start = 1,
                         end = 83,
                         shape = 0,
                         ensembl = 5,
                         constraintNumber  = 1,
                         evCutoff = 1)


plotEnsemblePCA(foldedCds)

## End(Not run)
```

`plotMatrices` *Plots a number of contact maps to file of each sample for a stage in the analysis*

Description

Plots a number of contact maps to file of each sample for a stage in the analysis

Usage

```
plotMatrices(
  cds,
  type = "original",
  directory = 0,
  a = 1,
  b = 50,
  c = 1,
  d = 50,
  h = 3
)
```

Arguments

cds	A comradesDataSet object
type	The analysis stage to plot
directory	An output directory for the heatmap (use 0 for no output)
a	To make a subsetted plot (left value on x)
b	To make a subsetted plot (right value on x)
c	To make a subsetted plot (left value on y)
d	To make a subsetted plot (right value on y)
h	Height of image (inches) (only useful if plotting)

Value

A heatmap of the reads in the analysis stage chosen

Examples

```
cds = makeExampleComradesDataSet()

plotMatrices(cds,
            b = rnaSize(cds),
            d = rnaSize(cds))
```

plotMatricesAverage *plotMatricesAverage*

Description

Plots a contact map to file of all samples for a stage in the analysis

Usage

```
plotMatricesAverage(
  cds,
  type = "original",
  directory = 0,
  a = 1,
  b = 50,
  c = 1,
  d = 50,
  h = 3
)
```

Arguments

cds	A comradesDataSet object
type	The analysis stage to plot
directory	An output directory for the heatmap (use 0 for no output)
a	To make a subsetted plot (left value on x)
b	To make a subsetted plot (right value on x)
c	To make a subsetted plot (left value on y)
d	To make a subsetted plot (right value on y)
h	Height of image (inches) (only useful if plotting)

Value

A heatmap of the reads in the analysis stage chosen

Examples

```
cds = makeExampleComradesDataSet()

plotMatricesAverage(cds,
            b = rnaSize(cds),
            d = rnaSize(cds))
```

plotStructure *plotStructure*

Description

This method plots a structures chosen from the plotEnsemblePCA method

Usage

```
plotStructure(foldedCds, rnaRefs, s = "s1", n = 1)
```

Arguments

foldedCds	comradesDataSet after running foldComrades
rnaRefs	A fasta of the transcript (made with seqinr::read.fasta)
s	sample of structure
n	number of structure

Value

a diagram of the predicted structure

Examples

```
## Not run:
cds = makeExampleComradesDataSet()
clusteredCds = clusterComrades(cds = cds,
                               cores = 3,
                               stepCount = 2,
                               clusterCutoff = 1)


trimmedClusters = trimClusters(clusteredCds = clusteredCds,trimFactor = 1, clusterCutoff = 1)



fasta = paste(c(rep('A',25),
                rep('T',25),
                rep('A',10),
                rep('T',23)),collapse = "")

header = '>transcript1'


fastaFile = tempfile()
writeLines(paste(header,fasta,sep = "\n"),con = fastaFile)
```

```
rnaRefs = list()
rnaRefs[[rnas(cds)]] = read.fasta(fastaFile)
rnaRefs



foldedCds = foldComrades(trimmedClusters,
                         rnaRefs = rnaRefs,
                         start = 1,
                         end = 83,
                         shape = 0,
                         ensembl = 5,
                         constraintNumber  = 1,
                         evCutoff = 1)


plotStructure(foldedCds,rnaRefs,"s1",3)

## End(Not run)
```

`printClustersFast` *printClustersFast*

Description

Makes a table with the coordinates of the clusters

Usage

```
printClustersFast(dir, clustering, highest_clusters, left, right)
```

Arguments

`dir`	the directory that contains the *hybrids.hyb files
`clustering`	The output from the iGraph function cluster_walktrap for the (made with adjacency matrix input)
`highest_clusters`	The cluster you are interested in keeping
`left`	list created with hybToGRanges (but just the left section of the list)
`right`	list created with hybToGRanges (but just the right section of the list)

Details

Does the same as printClusters but is a lot faster and does not create plots of each cluster

Value

A table of clusters and coordinates

readNumbers *readNumbers*

Description

This method prints a table showing the number of duplexes in the clusters in each step of the analysis

Usage

```
readNumbers(knowClusteredCds, rna)
```

Arguments

knowClusteredCds	A comradesDataSet object after clustering has been performed
rna	The RNA ID of interest - use rna(cdsObject).

Value

A data.frame shoing the number of reads in clusters for each sample

Examples

```
cds = makeExampleComradesDataSet()

clusteredCds = clusterComrades(cds,
                cores = 1,
                stepCount = 1,
                clusterCutoff = 1)
readNumbers(clusteredCds)
```

rnas *rnas*

Description

Extract the rna ID for the instance

Usage

```
rnas(x)
```

Arguments

x	A comradesDataSet object

Value

A character - the ID of the RNA

Examples

```
cds = makeExampleComradesDataSet()

rnas(cds)
```

`rnaSize` *rnaSize*

Description

Extract the size of the RNA for the instance

Usage

```
rnaSize(x)
```

Arguments

x	A comradesDataSet object

Value

A numeric - the size of the RNA (nucleotides)

Examples

```
cds = makeExampleComradesDataSet()

rnaSize(cds)
```

`sampleChimeras` *sampleChimeras*

Description

This function samples chimeras into smaller chunks so that clustering is quicker

Usage

```
sampleChimeras(chimeraList)
```

Arguments

chimeraList	list of chimeras

sampleNames *sampleNames*

Description

Extract the sample names for the instance

Usage

```
sampleNames(x)
```

Arguments

x	A comradesDataSet object

Value

A character vector - the sample names

Examples

```
cds = makeExampleComradesDataSet()

sampleNames(cds)
```

sampleTable *sampleTable*

Description

Extract the sample table for the instance

Usage

```
sampleTable(x)
```

Arguments

x	A comradesDataSet object

Value

A data frame - The orginal meta-data table

Examples

```
cds = makeExampleComradesDataSet()

sampleTable(cds)
```

subsetHybList2 *subsetHybList2*

Description

Subset a list of hyb files

Usage

```
subsetHybList2(hybList, min, max, length)
```

Arguments

hybList	the original hybList created with readHybFiles
min	the rna of interest that you want to subset
max	The number of randomly subsetted chimeric reads you need
length	The number of randomly subsetted chimeric reads you need

Details

Function used to subset a list of hyb data created by readHybFiles This function produces the same size list as before but it returns ONLY the rna of interest and also Choose duplexes where the nt difference in position between the one side and other side of an interaction is between min and max

Value

A list of subsetted hyb files

swapHybs *swapHybs*

Description

Swap the table to ensure that 3 prime most duplex side is on he left of the table used to make one sides heatmaps and other reasons where having the left of the table coming after the right side is a problem. Different from swapHybs as it ensure that BOTH duplex sides originate from the RNA of interest.

Usage

```
swapHybs(hybList, rna)
```

Arguments

hybList	the original hybList created with readHybFiles or subsetHybList
rna	The rna of interest

Value

A list of "swapped" hyb datas

`swapHybs2` *swapHybs2*

Description

Swap the table to ensure that 3 prime most duplex side is on the left of the table used to make one sides heatmaps and other reasons where having the left of the table coming after the right side is a problem. Different from swapHybs as it ensure that BOTH duplex sides originate from the RNA of interest.

Usage

```
swapHybs2(hybList, rna)
```

Arguments

`hybList`	the original hybList created with readHybFiles or subsetHybList
`rna`	The rna of interest

Value

A list of "swapped" hyb data

`swapHybs3` *swapHybs3*

Description

Swap the table to ensure that 3 prime most duplex side is ont he left of the table used to make one sides heatmaps and other reasons where having the left of the table coming after the right side is a problem. Different from swapHybs as it ensure that BOTH duplex sides originate from the RNA of interest.

Usage

```
swapHybs3(hybList, rna)
```

Arguments

`hybList`	the original hybList created with readHybFiles or subsetHybList
`rna`	The rna of interest

Value

A list of "swapped" hyb datas

topInteracters *topInteracters*

Description

This method prints the top transcripts that have the most duplexes assigned that interact with the transcript of interest

Usage

```
topInteracters(cds, ntop = 10)
```

Arguments

cds	a comradesDataSet object
ntop	the number of entries to display

Value

A table, the number of counts per sample per interacting transcript

Examples

```
cds = makeExampleComradesDataSet()
topInteracters(cds)
```

topInteractions *topInteractions*

Description

This method prints the top transcript interactions that have the most duplexes assigned

Usage

```
topInteractions(cds, ntop = 10)
```

Arguments

cds	a comradesDataSet object
ntop	the number of entries to display

Value

A table, the number of counts per sample per interaction

Examples

```
cds = makeExampleComradesDataSet()
topInteractions(cds)
```

`topTranscripts` *topTranscripts*

Description

This method prints the top transcripts that have the most duplexes assigned

Usage

```
topTranscripts(cds, ntop = 10)
```

Arguments

`cds`	a `comradesDataSet` object
`ntop`	the number of entries to display

Value

A table, the number of counts per sample per transcript

Examples

```
cds = makeExampleComradesDataSet()
topTranscripts(cds)
```

`trimClusters` *trimClusters*

Description

Trimming of the clusters removes redundant information derived from random fragmentation of the reads during library preparation. This method takes a `comradesDataSet` object where clustering has been performed with the clusterCOMRADES method and trims the clusters according to the trimFactor argument.

Usage

```
trimClusters(clusteredCds, trimFactor = 2.5, clusterCutoff = 1)
```

Arguments

clusteredCds	a comradesDataSet object
trimFactor	a positive value that defines how much the clusters will
clusterCutoff	Minimum number of reads before discarding cluster be trimmed = mean + (sd * trimFactor)

Details

The 3 attributes; matrixList, clusterTableList and clusterGrangesList will gain the types "superClusters" and "trimmedClusters"

Value

Returns a comradesDataSet object

Examples

```
cds = makeExampleComradesDataSet()

clusteredCds = clusterComrades(cds,
                cores = 1,
                stepCount = 1,
                clusterCutoff = 0)

trimClusters(clusteredCds = clusteredCds,
             trimFactor = 1,
             clusterCutoff = 0)
```

Index