

# Package ‘rnaCrosslinkOO’

July 23, 2025

**Type** Package

**Title** Analysis of RNA Crosslinking Data

**Version** 0.1.4

**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/rnaCrosslinkOO/issues>

**Depends** seqinr, GenomicRanges, stats

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

**RoxygenNote** 7.3.1

**Collate** 'rnaCrosslinkOO.R' 'rnaCrosslinkDataSet.R'  
'clusterrnaCrosslink.R'  
'clusterrnaCrosslinkMethodsAndHelpers.R'  
'commonHelpersAndMethods.R' 'commonStatsAndPlots.R'  
'foldrnaCrosslink.R' 'foldrnaCrosslinkMethodsAndHelpers.R'  
'genericMethods.R' 'rnaCrosslinkDataSetMethodsAndHelpers.R'  
'rnaCrosslinkOO-package.R' 'rnaCrosslinkQC.R'

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Jonathan Price [aut, cre] (ORCID:  
<<https://orcid.org/0000-0001-6554-5667>>),  
Andrew Lim [ctb]

**Repository** CRAN

**Date/Publication** 2024-07-10 23:40:02 UTC

## Contents

clusterGrangesList	3
clusterGrangesList<-	3
clusterNumbers	4
clusterrnaCrosslink	5
clusterTableFolded	5
clusterTableList	6
clusterTableList<-	6
compareKnown	7
compareKnownStructures	8
featureInfo	9
findBasePairsRNAcoFold2	10
findBasePairsRNAfold	11
findBasePairsRNAfold2	11
foldrnaCrosslink	12
getAdjacencyMat	13
getClusterClusterShortRangeWhole	14
getData	15
getInteractions	15
getMatrices	16
getReverseInteractions	17
group	17
InputFiles	18
InputToGRanges	18
makeExemplernaCrosslinkDataSet	19
matrixList	19
matrixList<-	20
plotClusterAgreement	20
plotClusterAgreementHeat	21
plotCombinedMatrix	22
plotComparisonArc	23
plotEnsemblePCA	24
plotInteractions	26
plotInteractionsAverage	27
plotMatrices	28
plotMatricesAverage	29
plotStructure	30
printClustersFast	31
readNumbers	32
rnaCrosslinkDataSet-class	33
rnaCrosslinkQC	34
rnas	37
rnaSize	38
sampleChimeras	38
sampleNames	39
sampleTable	39
subsetInputList2	40

<i>clusterGrangesList</i>	3
swapInputs	40
swapInputs2	41
swapInputs3	41
topInteractors	42
topInteractions	43
topTranscripts	43
trimClusters	44
<b>Index</b>	<b>45</b>

---

`clusterGrangesList`     *clusterGrangesList*

---

### Description

Extract the cluster coordinates in granges format

### Usage

```
clusterGrangesList(x)
```

### Arguments

`x`                    A `rnaCrosslinkDataSet` object

### Value

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

### Examples

```
cds = makeExemplernaCrosslinkDataSet()
clusterGrangesList(cds)
```

---

`clusterGrangesList<-`     *clusterGrangesList<-*

---

### Description

Set new `clusterGrangesList` slot

### Usage

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

**Arguments**

x                    A rnaCrosslinkDataSet object  
value                A replacement

**Value**

No return - Sets a new clusterGrangesList slot

**Examples**

```
cds = makeExemplernaCrosslinkDataSet()

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

---

clusterNumbers	<i>clusterNumbers</i>
----------------	-----------------------

---

**Description**

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

**Usage**

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

**Arguments**

knowClusteredCds                    A rnaCrosslinkDataSet 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 = makeExemplernaCrosslinkDataSet()

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

`clusterrnaCrosslink`    *clusterrnaCrosslink*

**Description**

This method clusters the duplexes.

**Usage**

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

**Arguments**

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

**Value**

A `rnaCrosslinkDataSet` object

**Examples**

```
cds = makeExemplernaCrosslinkDataSet()

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

`clusterTableFolded`    *clusterTableFolded*

**Description**

Extract the cluster coordinates with fold prediciton in data frame format

**Usage**

```
clusterTableFolded(x)
```

**Arguments**

`x`                        A `rnaCrosslinkDataSet` object

**Value**

A table showing the vienna structures of each cluster

**Examples**

```
cds = makeExemplernaCrosslinkDataSet()
```

```
clusterTableFolded(cds)
```

---

<code>clusterTableList</code>	<code><i>clusterTableList</i></code>
-------------------------------	--------------------------------------

---

**Description**

Extract the cluster coordinates in data frame format

**Usage**

```
clusterTableList(x)
```

**Arguments**

`x` A `rnaCrosslinkDataSet` object

**Value**

A list of tables showing the vienna structures of each cluster

**Examples**

```
cds = makeExemplernaCrosslinkDataSet()
```

```
clusterTableList(cds)
```

---

<code>clusterTableList&lt;-</code>	<code><i>clusterTableList&lt;-</i></code>
------------------------------------	---

---

**Description**

Set new `clusterTableList` slot

**Usage**

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

**Arguments**

x                    A rnaCrosslinkDataSet object  
 value                A replacement

**Value**

No return - Sets a new clusterTableList slot

**Examples**

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

---

compareKnown	<i>compareKnown</i>
--------------	---------------------

---

**Description**

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

**Usage**

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

**Arguments**

trimmedClusters    a rnaCrosslinkDataSet object, run trimClusters() on the rnaCrosslinkDataSet first

knownMat           Matrix - A matrix(ncol = lengthRNA,nrow = lengthRNA) where a value in matrix[x,y] would indicate a known interaction 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 rnaCrosslinkClusteredDataSet object

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

**Examples**

```

cds = makeExemplernaCrosslinkDataSet()

clusteredCds = clusterrnaCrosslink(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	rnaCrosslinkDataSet after running foldrnaCrosslink
file	a two column file with column header i and j with numeric values showing nucleotide i binds to nucleotide j

**Value**

Returns a dataframe  
a tables showing the number of predicted interactions and their agreement

**Examples**

```

## Not run:
cds = makeExemplernaCrosslinkDataSet()
clusteredCds = clusterrnaCrosslink(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 = foldrnaCrosslink(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)

```

---

featureInfo

*featureInfo*


---

### Description

Produces a list list of 2 elements 'transcript' and 'family' Each element contains a table with the counts for each RNA in each sample that interact with the target RNA

**Usage**

```
featureInfo(cds)
```

**Arguments**

cds                    a rnaCrosslinkDataSet object

**Value**

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

**Examples**

```
cds = makeExemplernaCrosslinkDataSet()
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

---

foldrnaCrosslink      *foldrnaCrosslink*

---

### Description

This methods folds an ensembl of structures for the whole RNA or chosen region of the RNA. See rnaCrosslinkDataSet for slot information.

### Usage

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

### Arguments

cdsObject	rnaCrosslinkDataSet object created with rnaCrosslinkDataSet
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 segment to fold
end	End of segment to fold
evCutoff	Minimum number of read support for constraint to be included in folding
ensembl	Number of structures to make
constraintNumber	Number of constraints to add to each final fold
shape	shape reactivities (0 for no constraints)

### Value

a rnaCrosslinkDataSet object

### Examples

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

clusteredCds = clusterrnaCrosslink(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 = foldrnaCrosslink(trimmedClusters,
                             rnaRefs = rnaRefs,
                             start = 1,
                             end = 83,
                             shape = 0,
                             ensembl = 5,
                             constraintNumber = 1,
                             evCutoff = 1)

foldedCds

## End(Not run)
```

---

getAdjacencyMat

*getAdjacencyMat*

---

### Description

Makes and adjacency matrix list (for clustering)

### Usage

```
getAdjacencyMat(InputGranges, nucleotideOrPerc, cutoff)
```

**Arguments**

InputGranges    list created with InputToGRanges (but just the gap section of the list)  
 nucleotideOrPerc    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 Adjacency 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 rnaCrosslinkDataSet object
data	The data type to return <InputFiles   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 = makeExemplernaCrosslinkDataSet()
getData(cds, 'matrixList', 'original')
```

getInteractions              *getInteractions*

**Description**

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

**Usage**

```
getInteractions(cds, interactors)
```

**Arguments**

cds	a rnaCrosslinkDataSet object
interactors	A vector containing the names of RNAs to show interactions with

**Value**

A table showing the read coverage of the specified interacting RNAs

**Examples**

```
cds = makeExemplernaCrosslinkDataSet()
getInteractions(cds, c("transcript1", "transcript2"))
```

---

getMatrices

*getMatrices*

---

**Description**

Make a matrix of contact interactions

**Usage**

```
getMatrices(InputList, rna, size)
```

**Arguments**

InputList	the original InputList created with readInputFiles or subsetInputList
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 `rnaCrosslinkDataSet` object  
`interactor`            The rna to show interactions with

**Value**

A long format table shoing the read coverage of chosen RNA

**Examples**

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

---

`group`                    *group*

---

**Description**

Extract the indeces for each group for the instance

**Usage**

```
group(x)
```

**Arguments**

`x`                        A `rnaCrosslinkDataSet` object

**Value**

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

**Examples**

```

cds = makeExemplernaCrosslinkDataSet()

group(cds)

```

---

InputFiles

*InputFiles*


---

**Description**

Extract the data in original format

**Usage**

```
InputFiles(x)
```

**Arguments**

x                    A rnaCrosslinkDataSet object

**Value**

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

**Examples**

```

cds = makeExemplernaCrosslinkDataSet()

InputFiles(cds)

```

---

InputToGRanges

*InputToGRanges*


---

**Description**

This function is useful to turn a list of Input 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**

```
InputToGRanges(InputList, rna)
```

**Arguments**

InputList            the original InputList created with readInputFiles or subsetInputList  
rna                    The rna of interest

**Value**

A list of GRanges data in Input format

---

`makeExemplernaCrosslinkDataSet`  
*makeExemplernaCrosslinkDataSet*

---

**Description**

Creat a minimal example `rnaCrosslinkDataSetObject`

**Usage**

`makeExemplernaCrosslinkDataSet()`

**Value**

An example `rnaCrosslinkDataSet` object

**Examples**

`cds = makeExemplernaCrosslinkDataSet()`

---

`matrixList`                    *matrixList*

---

**Description**

Extract the contact matrices

**Usage**

`matrixList(x)`

**Arguments**

`x`                    A `rnaCrosslinkDataSet` object

**Value**

A list of contract matrices, one entry for each sample

**Examples**

`cds = makeExemplernaCrosslinkDataSet()`

`matrixList(cds)`

---

```
matrixList<-          matrixList
```

---

**Description**

Set new matrixList slot

**Usage**

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

**Arguments**

x                    A rnaCrosslinkDataSet object  
value                A replacement

**Value**

No return - Sets a new matrixList slot

**Examples**

```
cds = makeExemplernaCrosslinkDataSet()

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

---

```
plotClusterAgreement Plot a heatmap that plots the agreements between replicates after clus-  
terrnaCrosslink has been performed
```

---

**Description**

Plot a heatmap that plots the agreements between replicates after clusterrnaCrosslink has been performed

**Usage**

```
plotClusterAgreement(cds, analysisStage = "originalClusters")
```

**Arguments**

cds                    A rnaCrosslinkDataSet object  
analysisStage        The stage of the analysis to plot

**Value**

A heatmap of the agreement between replicates in the analysis stage chosen

**Examples**

```
cds = makeExemplernaCrosslinkDataSet()
```

```
clusteredCds = clusterrnaCrosslink(cds,  
  cores = 1,  
  stepCount = 1,  
  clusterCutoff = 0)
```

```
plotClusterAgreement(cds)
```

---

plotClusterAgreementHeat

*Plot a heatmap that plots the agreements between replicates after clusterrnaCrosslink has been performed*

---

**Description**

Plot a heatmap that plots the agreements between replicates after clusterrnaCrosslink has been performed

**Usage**

```
plotClusterAgreementHeat(cds, analysisStage = "originalClusters")
```

**Arguments**

cds                    A rnaCrosslinkDataSet object  
analysisStage        The stage of the analysis to plot

**Value**

A heatmap of the agreement between replicates in the analysis stage chosen

**Examples**

```
cds = makeExemplernaCrosslinkDataSet()
```

```
clusteredCds = clusterrnaCrosslink(cds,  
  cores = 1,
```

```
stepCount = 1,
clusterCutoff = 0)
```

```
plotClusterAgreementHeat(cds)
```

---

`plotCombinedMatrix`     *Plots a contact map of two chosen samples for chosen stages in the analysis, with each chosen sample on separate halves of the contact map*

---

### Description

Plots a contact map of two chosen samples for chosen stages in the analysis, with each chosen sample on separate halves of the contact map

### Usage

```
plotCombinedMatrix(
  cds,
  type1 = "original",
  type2 = "original",
  sample1 = 1,
  sample2 = 1,
  directory = 0,
  a = 1,
  b = 50,
  c = 1,
  d = 50,
  h = 3,
  returnData = FALSE
)
```

### Arguments

<code>cds</code>	A <code>rnaCrosslinkDataSet</code> object
<code>type1</code>	The analysis stage to plot on the upper half of the heatmap
<code>type2</code>	The analysis stage to plot on the lower half of the heatmap
<code>sample1</code>	The sample number to plot on the upper half of the heatmap
<code>sample2</code>	The sample number to plot on the upper half of the heatmap
<code>directory</code>	An output directory for the heatmap (use 0 for no output)
<code>a</code>	To make a subsetting plot (left value on x)
<code>b</code>	To make a subsetting plot (right value on x)

c	To make a subsetting plot (left value on y)
d	To make a subsetting plot (right value on y)
h	Height of image (inches) (only useful if plotting)
returnData	if TRUE matrix is returned instead of plotting

**Value**

A heatmap of the reads of the chosen sample numbers, in the analysis stages chosen, with each chosen sample on a separate half of the heatmap

**Examples**

```

cds = makeExemplernaCrosslinkDataSet()

plotCombinedMatrix(cds,
  type1 = "original",
  type2 = "noHost",
  b = rnaSize(cds),
  d = rnaSize(cds))

```

---

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	rnaCrosslinkDataSet after running foldrnaCrosslink
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 structures

**Examples**

```

## Not run:
cds = makeExemplernaCrosslinkDataSet()
clusteredCds = clusterrnaCrosslink(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 = foldrnaCrosslink(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	rnaCrosslinkDataSet after running foldrnaCrosslink
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 = makeExemplernaCrosslinkDataSet()
clusteredCds = clusterrnaCrosslink(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 = foldrnaCrosslink(trimmedClusters,
                              rnaRefs = rnaRefs,
                              start = 1,
                              end = 83,
                              shape = 0,
                              ensembl = 5,
```

```

constraintNumber = 1,
evCutoff = 1)

plotEnsemblePCA(foldedCds)

## End(Not run)

```

---

plotInteractions	<i>Plots a contact map of interactions of each sample of an RNA (interactor) on the RNA of interest</i>
------------------	---

---

### Description

Plots a contact map of interactions of each sample of an RNA (interactor) on the RNA of interest

### Usage

```

plotInteractions(
  cds,
  rna,
  interactor,
  directory = 0,
  a = 1,
  b = 50,
  c = 1,
  d = 50,
  h = 3
)

```

### Arguments

cds	A rnaCrosslinkDataSet object
rna	The RNA of interest
interactor	The RNA to show interactions with
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) (use 'max' to plot the whole RNA strand length)
c	To make a subsetted plot (left value on y)
d	To make a subsetted plot (right value on y) (use 'max' to plot the whole RNA strand length)
h	Height of image (inches) (only useful if plotting)

**Value**

A heatmap of interactions of the RNA (interactor) on the RNA of interest

**Examples**

```
cds = makeExemplernaCrosslinkDataSet()

plotInteractions(cds,
                 rna = "transcript1",
                 interactor = "transcript2",
                 b = "max",
                 d = "max")
```

---

plotInteractionsAverage

*Plots a contact map of interactions of all samples of an RNA (interactor) on the RNA of interest*

---

**Description**

Plots a contact map of interactions of all samples of an RNA (interactor) on the RNA of interest

**Usage**

```
plotInteractionsAverage(
  cds,
  rna,
  interactor,
  directory = 0,
  a = 1,
  b = 50,
  c = 1,
  d = 50,
  h = 3
)
```

**Arguments**

cds	A rnaCrosslinkDataSet object
rna	The RNA of interest
interactor	The RNA to show interactions with
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) (use 'max' to plot the whole RNA strand length)

c	To make a subsetted plot (left value on y)
d	To make a subsetted plot (right value on y) (use 'max' to plot the whole RNA strand length)
h	Height of image (inches) (only useful if plotting)

**Value**

A heatmap of interactions of all samples of the RNA (interactor) on the RNA of interest

**Examples**

```

cds = makeExemplernaCrosslinkDataSet()

plotInteractionsAverage(cds,
  rna = "transcript1",
  interactor = "transcript2",
  b = "max",
  d = "max")

```

---

plotMatrices	<i>Plots a number of contact maps to file of each sample for a stage in the analysis</i>
--------------	--

---

**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 rnaCrosslinkDataSet 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 = makeExemplernaCrosslinkDataSet()

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

---

plotMatricesAverage    *plotMatricesAverage*

---

**Description**

Plots a contact map of all samples for two chosen stages in the analysis, with each chosen stage on separate halves of the contact map

**Usage**

```
plotMatricesAverage(
  cds,
  type1 = "original",
  type2 = "blank",
  directory = 0,
  a = 1,
  b = 50,
  c = 1,
  d = 50,
  h = 3
)
```

**Arguments**

cds	A rnaCrosslinkDataSet object
type1	The analysis stage to plot on the upper half of the heatmap (use 'blank' to leave this half blank)
type2	The analysis stage to plot on the lower half of the heatmap (use 'blank' to leave this half blank)

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 two analysis stages chosen, with each chosen stage on a separate half of the heatmap

**Examples**

```

cds = makeExemplernaCrosslinkDataSet()

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

```

---

plotStructure	<i>plotStructure</i>
---------------	----------------------

---

**Description**

This method plots a structures chosen from the plotEnsemblePCA method

**Usage**

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

**Arguments**

foldedCds	rnaCrosslinkDataSet after running foldrnaCrosslink
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 = makeExemplernaCrosslinkDataSet()
clusteredCds = clusterrnaCrosslink(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 = foldrnaCrosslink(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 \*Inputrids.Input 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 `InputToGRanges` (but just the left section of the list)

`right` list created with `InputToGRanges` (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	<i>readNumbers</i>
-------------	--------------------

---

**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 `rnaCrosslinkDataSet` object after clustering has been performed

`rna` The RNA ID of interest - use `rna(cdsObject)`.

**Value**

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



**Examples**

```

cds = makeExamplernaCrosslinkDataSet()

clusteredCds = clusterrnaCrosslink(cds,
                                   cores = 1,
                                   stepCount = 1,
                                   clusterCutoff = 1)
readNumbers(clusteredCds)

```

---

```

rnaCrosslinkDataSet-class
      rnaCrosslinkDataSet

```

---

**Description**

rnaCrosslinkDataSet 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 Input files are also filtered for the RNA of interest. Check the package vignette for more information.

**Usage**

```

rnaCrosslinkDataSet(
  rnas,
  rnaSize = 0,
  sampleTable,
  subset = "all",
  sample = "all"
)

```

**Arguments**

rnas	vector - The names of the RNA interest, these must be displayed the same way as in the input Input 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 Input file for each sample ), group ("s" - sample or "c" - control), sample (1,2,3, etc), sampleName (must be unique).
subset	a vector of 4 values to subset based on structural read size. c(l-min,l-max,r-min,r-max)
sample	The number of reads to sample for each sample.

**Value**

A rnaCrosslinkDataSet 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 `rnaCrosslinkDataSet` 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 `rnaCrosslinkDataSet` 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 `rnaCrosslinkDataSet` objects, `matrixList(cds)[[rna]][[type]][[sample]]` Contains a set of contact matrices, each cell contains the number of duplexes identified for position x,y.

`InputFiles` List - Follows the pattern for list slots of `rnaCrosslinkDataSet` objects, `InputFiles(cds)[[rna]][[type]][[sample]]` Contains a set of tables, these are the original Input 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
cds = makeExamplernaCrosslinkDataSet()

cds
```

---

rnaCrosslinkQC

*rnaCrosslinkQC*

---

**Description**

get a plot fo the read lengths and transcripts in the dataset The fuction will make 1 pdf and 2 text file in the directory provided

**Usage**

```
rnaCrosslinkQC(sampleTable, directory, topTranscripts = TRUE)
```

**Arguments**

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 Input file for each sample ), group ("s" - sample or "c" - control), sample (1,2,3, etc), sampleName (must be unique).

directory        A directory address to write the files

topTranscripts   If FALSE a table of top transcripts will not be written to file

**Value**

ggplot and txt file

**Examples**

```

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)

c4 = c(rep("transcript1",55),rep("transcript2",90) )
c10 = c(rep("transcript1",145) )
c1 = 1:145
c2 = rep(paste(rep("A", 40), collapse = ""),145)
c3 = rep(".",145)
c9 = rep(".",145)
c15 = rep(".",145)
c5 = rep(1,145)
c11 = rep(21,145)
c6 = rep(20,145)
c12= rep(40,145)
# short distance 55
c7 = sample(1:5, 55, replace = TRUE)
c8 = sample(20:25, 55, replace = TRUE)
c13 = sample(20:25, 55, replace = TRUE)
c14 = sample(40:45, 55, replace = TRUE)

# inter RNA 100
c7 = c(c7,sample(1:40, 90, replace = TRUE))
c8 = c(c8,sample(20:75, 90, replace = TRUE))
c13 = c(c13,sample(1:40, 90, replace = TRUE))
c14 = c(c14,sample(20:75, 90, 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)

file2 = tempfile()
write.table(exampleInput,
            file = file2,
            quote = FALSE,
            row.names = FALSE,
            sep = "\t",
            col.names = FALSE)

# Set up the sample table. ----
sampleTable1 = c(file, "s", "1", "s1")
sampleTable2 = c(file2, "c", "1", "c1")
# make the sample table
sampleTable2 = rbind.data.frame(sampleTable1, sampleTable2)
# add the column names
colnames(sampleTable2) = c("file", "group", "sample", "sampleName")

rnaCrosslinkQC(sampleTable2, tempdir())
```

---

*rnas*

*rnas*

---

## Description

Extract the rna ID for the instance

## Usage

```
rnas(x)
```

## Arguments

x                    A `rnaCrosslinkDataSet` object

## Value

A character - the ID of the RNA

## Examples

```
cds = makeExemplernaCrosslinkDataSet()

rnas(cds)
```

*rnaSize**rnaSize*

---

**Description**

Extract the size of the RNA for the instance

**Usage**

```
rnaSize(x)
```

**Arguments**

x                    A *rnaCrosslinkDataSet* object

**Value**

A numeric - the size of the RNA (nucleotides)

**Examples**

```
cds = makeExemplernaCrosslinkDataSet()
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	<i>sampleNames</i>
-------------	--------------------

---

**Description**

Extract the sample names for the instance

**Usage**

```
sampleNames(x)
```

**Arguments**

x                    A rnaCrosslinkDataSet object

**Value**

A character vector - the sample names

**Examples**

```
cds = makeExemplernaCrosslinkDataSet()
sampleNames(cds)
```

---

sampleTable	<i>sampleTable</i>
-------------	--------------------

---

**Description**

Extract the sample table for the instance

**Usage**

```
sampleTable(x)
```

**Arguments**

x                    A rnaCrosslinkDataSet object

**Value**

A data frame - The original meta-data table

**Examples**

```
cds = makeExemplernaCrosslinkDataSet()
sampleTable(cds)
```

---

subsetInputList2	<i>subsetInputList2</i>
------------------	-------------------------

---

**Description**

Subset a list of Input files

**Usage**

```
subsetInputList2(InputList, min, max, length)
```

**Arguments**

InputList	the original InputList created with readInputFiles
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 Input data created by readInputFiles 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 Input files

---

swapInputs	<i>swapInputs</i>
------------	-------------------

---

**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 swapInputs as it ensure that BOTH duplex sides originate from the RNA of interest.

**Usage**

```
swapInputs(InputList, rna)
```



**Arguments**

InputList	the original InputList created with readInputFiles or subsetInputList
rna	The rna of interest

**Value**

A list of "swapped" Input datas

---

swapInputs2	<i>swapInputs2</i>
-------------	--------------------

---

**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 swapInputs as it ensure that BOTH duplex sides originate from the RNA of interest.

**Usage**

```
swapInputs2(InputList, rna)
```

**Arguments**

InputList	the original InputList created with readInputFiles or subsetInputList
rna	The rna of interest

**Value**

A list of "swapped" Input data

---

swapInputs3	<i>swapInputs3</i>
-------------	--------------------

---

**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 swapInputs as it ensure that BOTH duplex sides originate from the RNA of interest.

**Usage**

```
swapInputs3(InputList, rna)
```

**Arguments**

InputList      the original InputList created with readInputFiles or subsetInputList  
 rna             The rna of interest

**Value**

A list of "swapped" Input datas

---

topInteractors	<i>topInteractors</i>
----------------	-----------------------

---

**Description**

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

**Usage**

```
topInteractors(cds, ntop = 10, sds = TRUE)
```

**Arguments**

cds             a rnaCrosslinkDataSet object  
 ntop            the number of entries to display  
 sds             known bug, doesn't work for small data sets fix incoming

**Value**

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

**Examples**

```
cds = makeExemplernaCrosslinkDataSet()
topInteractors(cds, sds = TRUE)
```

---

topInteractions	<i>topInteractions</i>
-----------------	------------------------

---

**Description**

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

**Usage**

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

**Arguments**

cds	a rnaCrosslinkDataSet object
ntop	the number of entries to display

**Value**

A table, the number of counts per sample per interaction

**Examples**

```
cds = makeExemplernaCrosslinkDataSet()  
topInteractions(cds)
```

---

topTranscripts	<i>topTranscripts</i>
----------------	-----------------------

---

**Description**

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

**Usage**

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

**Arguments**

cds	a rnaCrosslinkDataSet object
ntop	the number of entries to display

**Value**

A table, the number of counts per sample per transcript

**Examples**

```

cds = makeExemplernaCrosslinkDataSet()
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 `rnaCrosslinkDataSet` object where clustering has been performed with the `clusterrnaCrosslink` method and trims the clusters according to the `trimFactor` argument.

**Usage**

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

**Arguments**

`clusteredCds` a `rnaCrosslinkDataSet` 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 `rnaCrosslinkDataSet` object

**Examples**

```

cds = makeExemplernaCrosslinkDataSet()

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

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

```

# Index

clusterGrangesList, 3  
clusterGrangesList, rnaCrosslinkDataSet-method  
    (clusterGrangesList), 3  
clusterGrangesList<-, 3  
clusterGrangesList<-, rnaCrosslinkDataSet-method  
    (clusterGrangesList<-), 3  
clusterNumbers, 4  
clusterNumbers, rnaCrosslinkDataSet-method  
    (clusterNumbers), 4  
clusterrnaCrosslink, 5  
clusterrnaCrosslink, rnaCrosslinkDataSet-method  
    (clusterrnaCrosslink), 5  
clusterTableFolded, 5  
clusterTableFolded, rnaCrosslinkDataSet-method  
    (clusterTableFolded), 5  
clusterTableList, 6  
clusterTableList, rnaCrosslinkDataSet-method  
    (clusterTableList), 6  
clusterTableList<-, 6  
clusterTableList<-, rnaCrosslinkDataSet-method  
    (clusterTableList<-), 6  
compareKnown, 7  
compareKnown, rnaCrosslinkDataSet-method  
    (compareKnown), 7  
compareKnownStructures, 8  
compareKnownStructures, rnaCrosslinkDataSet-method  
    (compareKnownStructures), 8  
  
featureInfo, 9  
featureInfo, rnaCrosslinkDataSet-method  
    (featureInfo), 9  
findBasePairsRNAcoFold2, 10  
findBasePairsRNAfold, 11  
findBasePairsRNAfold2, 11  
foldrnaCrosslink, 12  
foldrnaCrosslink, rnaCrosslinkDataSet-method  
    (foldrnaCrosslink), 12  
  
getAdjacencyMat, 13  
getClusterClusterShortRangeWhole, 14  
    getData, 15  
getClusterClusterShortRangeWhole, rnaCrosslinkDataSet-method  
    (getData), 15  
getInteractions, 15  
getInteractions, rnaCrosslinkDataSet-method  
    (getInteractions), 15  
getMatrices, 16  
getReverseInteractions, 17  
getReverseInteractions, rnaCrosslinkDataSet-method  
    (getReverseInteractions), 17  
group, 17  
group, rnaCrosslinkDataSet-method  
    (group), 17  
InputFiles, 18  
InputFiles, rnaCrosslinkDataSet-method  
    (InputFiles), 18  
InputToGRanges, 18  
makeExamplernaCrosslinkDataSet, 19  
matrixList, 19  
matrixList, rnaCrosslinkDataSet-method  
    (matrixList), 19  
matrixList<-, 20  
matrixList<-, rnaCrosslinkDataSet-method  
    (matrixList<-), 20  
  
plotClusterAgreement, 20  
plotClusterAgreement, rnaCrosslinkDataSet-method  
    (plotClusterAgreement), 20  
plotClusterAgreementHeat, 21  
plotClusterAgreementHeat, rnaCrosslinkDataSet-method  
    (plotClusterAgreementHeat), 21  
plotCombinedMatrix, 22  
plotCombinedMatrix, rnaCrosslinkDataSet-method  
    (plotCombinedMatrix), 22  
plotComparisonArc, 23  
plotComparisonArc, rnaCrosslinkDataSet-method  
    (plotComparisonArc), 23  
plotEnsemblePCA, 24

plotEnsemblePCA, rnaCrosslinkDataSet-method  
     (plotEnsemblePCA), 24  
 plotInteractions, 26  
 plotInteractions, rnaCrosslinkDataSet-method  
     (plotInteractions), 26  
 plotInteractionsAverage, 27  
 plotInteractionsAverage, rnaCrosslinkDataSet-method  
     (plotInteractionsAverage), 27  
 plotMatrices, 28  
 plotMatrices, rnaCrosslinkDataSet-method  
     (plotMatrices), 28  
 plotMatricesAverage, 29  
 plotMatricesAverage, rnaCrosslinkDataSet-method  
     (plotMatricesAverage), 29  
 plotStructure, 30  
 plotStructure, rnaCrosslinkDataSet-method  
     (plotStructure), 30  
 printClustersFast, 31  
  
 readNumbers, 32  
 readNumbers, rnaCrosslinkDataSet-method  
     (readNumbers), 32  
 rnaCrosslinkDataSet  
     (rnaCrosslinkDataSet-class), 33  
 rnaCrosslinkDataSet-class, 33  
 rnaCrosslinkQC, 34  
 rnas, 37  
 rnas, rnaCrosslinkDataSet-method (rnas),  
     37  
 rnaSize, 38  
 rnaSize, rnaCrosslinkDataSet-method  
     (rnaSize), 38  
  
 sampleChimeras, 38  
 sampleNames, 39  
 sampleNames, rnaCrosslinkDataSet-method  
     (sampleNames), 39  
 sampleTable, 39  
 sampleTable, rnaCrosslinkDataSet-method  
     (sampleTable), 39  
 subsetInputList2, 40  
 swapInputs, 40  
 swapInputs2, 41  
 swapInputs3, 41  
  
 topInteractors, 42  
 topInteractors, rnaCrosslinkDataSet-method  
     (topInteractors), 42  
 topInteractions, 43  
 topInteractions, rnaCrosslinkDataSet-method  
     (topInteractions), 43  
 topTranscripts, 43  
 topTranscripts, rnaCrosslinkDataSet-method  
     (topTranscripts), 43  
 trimClusters, 44  
 trimClusters, rnaCrosslinkDataSet-method  
     (trimClusters), 44