

# Package ‘dStruct’

May 9, 2024

**Type** Package

**Title** Identifying differentially reactive regions from RNA structurome profiling data

**Version** 1.10.0

**Depends** R (>= 4.1)

**Description** dStruct identifies differentially reactive regions from RNA structurome profiling data. dStruct is compatible with a broad range of structurome profiling technologies, e.g., SHAPE-MaP, DMS-MaPseq, Structure-Seq, SHAPE-Seq, etc. See Choudhary et al., Genome Biology, 2019 for the underlying method.

**Imports** zoo, ggplot2, purrr, reshape2, parallel, IRanges, S4Vectors, rlang, grDevices, stats, utils

**License** GPL (>= 2)

**biocViews** StatisticalMethod, StructuralPrediction, Sequencing, Software

**URL** <https://github.com/dataMaster-Kris/dStruct>

**BugReports** <https://github.com/dataMaster-Kris/dStruct/issues>

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

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

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**git\_url** <https://git.bioconductor.org/packages/dStruct>

**git\_branch** RELEASE\_3\_19

**git\_last\_commit** 85159c4

**git\_last\_commit\_date** 2024-04-30

**Repository** Bioconductor 3.19

**Date/Publication** 2024-05-09

**Author** Krishna Choudhary [aut, cre] (<<https://orcid.org/0000-0002-7966-1527>>),  
Sharon Aviran [aut] (<<https://orcid.org/0000-0003-1872-9820>>)

**Maintainer** Krishna Choudhary <kchoudhary@ucdavis.edu>

## Contents

|                             |           |
|-----------------------------|-----------|
| calcDis . . . . .           | 2         |
| dCombs . . . . .            | 3         |
| dStruct . . . . .           | 4         |
| dStructGuided . . . . .     | 6         |
| dStructome . . . . .        | 7         |
| getCombs . . . . .          | 10        |
| getContigRegions . . . . .  | 11        |
| getRegions . . . . .        | 12        |
| lai2019 . . . . .           | 13        |
| normalizer . . . . .        | 14        |
| plotDStructurome . . . . .  | 15        |
| twoEightNormalize . . . . . | 16        |
| wan2014 . . . . .           | 17        |
| <b>Index</b>                | <b>18</b> |

---

|         |                            |
|---------|----------------------------|
| calcDis | <i>Calculates d score.</i> |
|---------|----------------------------|

---

## Description

$d$  score of a nucleotide is a measure of dissimilarity of its normalized reactivity scores. Consider a transcript and its reactivity profiles from a group of samples. Then, the  $d$  score of a nucleotide is  $(2/\pi)$  times the arc-tangent of the ratio of the sample standard deviation of its reactivities to their mean.

## Usage

```
calcDis(x)
```

## Arguments

`x`                    A numeric vector or matrix.

## Value

If input is a numeric vector, a number is returned. For a matrix, a numeric vector is returned.

## Author(s)

Krishna Choudhary

## References

Choudhary, K., Lai, Y. H., Tran, E. J., & Aviran, S. (2019). dStruct: identifying differentially reactive regions from RNA structurome profiling data. *Genome biology*, 20(1), 1-26.

Choudhary K, Shih NP, Deng F, Ledda M, Li B, Aviran S. Metrics for rapid quality control in RNA structure probing experiments. *Bioinformatics*. 2016; 32(23):3575–3583.

## Examples

```
#Lower standard deviation of reactivities results in lower d-score.  
calcDis(rnorm(10, 1, 0.2))  
calcDis(rnorm(10, 1, 0.6))
```

---

dCombs

*Assesses within-group or between-group variation.*

---

## Description

Given the reactivity profiles for a transcript from multiple samples, and a list of sample identifiers, this function computes the dissimilarity of reactivity scores between the specified samples. These are returned as a sequence of nucleotide-wise *d* scores.

## Usage

```
dCombs(rdf, combs)
```

## Arguments

rdf                    Data.frame of reactivities for each sample.  
combs                  Data.frame with each column containing groupings of samples.

## Value

Nucleotide-wise *d* scores.

## Author(s)

Krishna Choudhary

## References

Choudhary, K., Lai, Y. H., Tran, E. J., & Aviran, S. (2019). dStruct: identifying differentially reactive regions from RNA structurome profiling data. *Genome biology*, 20(1), 1-26.

## Examples

```
#Example of a data frame with reactivities.
reacs <- data.frame(matrix(runif(30, 0, 10), 10, 3))

#The columns of data frame with must indicate sample grouping and id.
colnames(reacs) <- c("A1", "A2", "B1")

#Get nucleotide-wise dissimilarity scores for a set of samples.
dCombs(rdf = reacs, combs = data.frame(c("A1", "B1")))
```

---

dStruct

*Performs de novo discovery of differentially reactive regions.*

---

## Description

This function takes reactivity profiles for samples of two groups as input and identifies differentially reactive regions in three steps (see Choudhary et al., *Genome Biology*, 2019 for details). First, it regroups the samples into homogeneous and heterogeneous sub-groups, which are used to compute the within-group and between-group nucleotide-wise  $d$  scores. Second, smoothed between- and within-group  $d$  score profiles are compared to construct candidate differential regions. Finally, unsmoothed between- and within-group  $d$  scores are compared using the Wilcoxon signed-rank test. The resulting p-values quantify the significance of difference in reactivity patterns between the two input groups.

## Usage

```
dStruct(
  rdf,
  reps_A,
  reps_B,
  batches = FALSE,
  min_length = 11,
  check_signal_strength = TRUE,
  check_nucs = TRUE,
  check_quality = TRUE,
  quality = "auto",
  evidence = 0,
  signal_strength = 0.1,
  within_combs = NULL,
  between_combs = NULL,
  ind_regions = TRUE,
  gap = 1,
  get_FDR = TRUE,
  proximity_assisted = FALSE,
  proximity = 10,
  proximity_defined_length = 30
)
```

**Arguments**

|                          |  |
|--------------------------|--|
| rdf                      | Dataframe of reactivities for each sample.   |
| reps_A                   | Number of replicates of group A.   |
| reps_B                   | Number of replicates of group B.   |
| batches                  | Logical suggesting if replicates of group A and B were performed in batches and are labelled accordingly. If TRUE, a heterogeneous/homogeneous subset may not have multiple samples from the same batch. |
| min_length               | Minimum length of constructed regions.   |
| check_signal_strength    | Logical, if TRUE, construction of regions must be based on nucleotides that have a minimum absolute value of reactivity.   |
| check_nucs               | Logical, if TRUE, constructed regions must have a minimum number of nucleotides participating in Wilcoxon signed rank test.  |
| check_quality            | Logical, if TRUE, check constructed regions for quality.   |
| quality                  | Worst allowed quality for a region to be tested.   |
| evidence                 | Minimum evidence of increase in variation from within-group comparisons to between-group comparisons for a region to be tested.  |
| signal_strength          | Threshold for minimum signal strength.   |
| within_combs             | Data.frame with each column containing groupings of replicates of groups A or B, which will be used to assess within-group variation.  |
| between_combs            | Dataframe with each column containing groupings of replicates of groups A and B, which will be used to assess between-group variation.   |
| ind_regions              | Logical, if TRUE, test each region found in the transcript separately.   |
| gap                      | Integer. Join regions if they are separated by these many nucleotides.   |
| get_FDR                  | Logical, if FALSE, FDR is not reported.  |
| proximity_assisted       | Logical, if TRUE, proximally located regions are tested together.  |
| proximity                | Maximum distance between constructed regions for them to be considered proximal.   |
| proximity_defined_length | If performing a "proximity-assisted" test, minimum end-to-end length of a region to be tested.   |

**Value**

Constructs regions, reports p-value and median difference of between-group and within-group d-scores for each region, and FDR for them.

**Author(s)**

Krishna Choudhary

## References

Choudhary, K., Lai, Y. H., Tran, E. J., & Aviran, S. (2019). dStruct: identifying differentially reactive regions from RNA structure profiling data. *Genome biology*, 20(1), 1-26.

## Examples

```
#Load data from Lai et al., 2019
data(lai2019)

#Run dStruct in de novo discovery mode for a transcript with id YAL042W.
dStruct(rdf = lai2019[["YAL042W"]], reps_A = 3, reps_B = 2,
       batches = TRUE, min_length = 21,
       between_combs = data.frame(c("A3", "B1", "B2")),
       within_combs = data.frame(c("A1", "A2", "A3")),
       ind_regions = TRUE)
```

---

|               |  |
|---------------|--|
| dStructGuided | <i>Performs guided discovery of differentially reactive regions.</i> |
|---------------|--|

---

## Description

This function takes as input reactivity profiles for a transcript region from samples of two groups. First, it regroups the samples into homogeneous and heterogeneous sub-groups, which are used to compute the within-group and between-group nucleotide-wise  $d$  scores. If the region meets the quality criteria, the between- and within-group  $d$  scores are compared using the Wilcoxon signed-rank test. The resulting p-values quantify the significance of difference in reactivity patterns between the two input groups.

## Usage

```
dStructGuided(
  rdf,
  reps_A,
  reps_B,
  batches = FALSE,
  within_combs = NULL,
  between_combs = NULL,
  check_quality = TRUE,
  quality = "auto",
  evidence = 0
)
```

## Arguments

|        |   |
|--------|---|
| rdf    | Dataframe of reactivities for each sample. Each column must be labelled as A1, A2, ..., B1, B2, ... |
| reps_A | Number of replicates of group A.  |

|               |  |
|---------------|--|
| reps_B        | Number of replicates of group B.   |
| batches       | Logical suggesting if replicates of group A and B were performed in batches and are labelled accordingly. If TRUE, a heterogeneous/homogeneous subset may not have multiple samples from the same batch. |
| within_combs  | Data.frame with each column containing groupings of replicates of groups A or B, which will be used to assess within-group variation.  |
| between_combs | Dataframe with each column containing groupings of replicates of groups A and B, which will be used to assess between-group variation.   |
| check_quality | Logical, if TRUE, check regions for quality.   |
| quality       | Worst allowed quality for a region to be tested.   |
| evidence      | Minimum evidence of increase in variation from within-group comparisons to between-group comparisons for a region to be tested.  |

**Value**

p-value for the tested region (estimated using one-sided Wilcoxon signed rank test) and the median of nucleotide-wise difference of between-group and within-group d-scores.

**Author(s)**

Krishna Choudhary

**References**

Choudhary, K., Lai, Y. H., Tran, E. J., & Aviran, S. (2019). dStruct: identifying differentially reactive regions from RNA structurome profiling data. *Genome biology*, 20(1), 1-26.

**Examples**

```
#Load Wan et al., 2014 data
data(wan2014)

#Run dStruct in the guided mode on first region in wan2014.
dStructGuided(wan2014[[1]], reps_A = 2, reps_B = 1)
```

---

|            |   |
|------------|---|
| dStructome | <i>Performs de novo or guided discovery of differentially reactive regions for transcriptome-wide data.</i> |
|------------|---|

---

**Description**

This function provides a convenient way to call the dStruct or dStructGuided functions for multiple transcripts simultaneously. By default, the transcripts are processed in using multiple parallel processes if available.

**Usage**

```
dStructome(
  r1,
  reps_A,
  reps_B,
  batches = FALSE,
  min_length = 11,
  check_signal_strength = TRUE,
  check_nucs = TRUE,
  check_quality = TRUE,
  quality = "auto",
  evidence = 0,
  signal_strength = 0.1,
  within_combs = NULL,
  between_combs = NULL,
  ind_regions = TRUE,
  gap = 1,
  processes = "auto",
  method = "denovo",
  proximity_assisted = FALSE,
  proximity = 10,
  proximity_defined_length = 30
)
```

**Arguments**

|                       |  |
|-----------------------|--|
| r1                    | List of dataframes of reactivities for each sample.  |
| reps_A                | Number of replicates of group A.   |
| reps_B                | Number of replicates of group B.   |
| batches               | Logical suggesting if replicates of group A and B were performed in batches and are labelled accordingly. If TRUE, a heterogeneous/homogeneous subset may not have multiple samples from the same batch. |
| min_length            | Minimum length of constructed regions.   |
| check_signal_strength | Logical, if TRUE, construction of regions must be based on nucleotides that have a minimum absolute value of reactivity.   |
| check_nucs            | Logical, if TRUE, constructed regions must have a minimum number of nucleotides participating in Wilcoxon signed rank test.  |
| check_quality         | Logical, if TRUE, check constructed regions for quality.   |
| quality               | Worst allowed quality for a region to be tested.   |
| evidence              | Minimum evidence of increase in variation from within-group comparisons to between-group comparisons for a region to be tested.  |
| signal_strength       | Threshold for minimum signal strength.   |
| within_combs          | Data.frame with each column containing groupings of replicates of groups A or B, which will be used to assess within-group variation.  |



|                          |  |
|--------------------------|--|
| between_combs            | Dataframe with each column containing groupings of replicates of groups A and B, which will be used to assess between-group variation. |
| ind_regions              | Logical, if TRUE, test each region found in the transcript separately.   |
| gap                      | Integer. Join regions if they are separated by these many nucleotides.   |
| processes                | Number of parallel processes to use.   |
| method                   | Character specifying either guided or de novo discovery approach.  |
| proximity_assisted       | Logical, if TRUE, proximally located regions are tested together.  |
| proximity                | Maximum distance between constructed regions for them to be considered proximal.   |
| proximity_defined_length | If performing a "proximity-assisted" test, minimum end-to-end length of a region to be tested.   |

**Value**

Constructs regions, reports p-value and median difference of between-group and within-group d-scores for each region, and FDR for them.

**Author(s)**

Krishna Choudhary

**References**

Choudhary, K., Lai, Y. H., Tran, E. J., & Aviran, S. (2019). dStruct: identifying differentially reactive regions from RNA structurome profiling data. *Genome biology*, 20(1), 1-26.

**Examples**

```
#Load data from Lai et al., 2019
data(lai2019)

#Run dStruct in de novo discovery mode for all the transcripts in this data in one step.
dStructome(lai2019, 3, 2, batches= TRUE, min_length = 21,
  between_combs = data.frame(c("A3", "B1", "B2")),
  within_combs = data.frame(c("A1", "A2", "A3")),
  ind_regions = TRUE, processes = 1)

#Load data from Wan et al., 2014
data(wan2014)

#Run dStruct in guide discovery mode for all the transcript regions in this data in one step.
dStructome(wan2014, reps_A = 2, reps_B = 1, method = "guided", processes = 1)
```

---

|          |  |
|----------|--|
| getCombs | <i>Identifies subgroupings of replicates for assessing within-group and between-group variation.</i> |
|----------|--|

---

### Description

Regroup all the samples of A and B groups into homogeneous and heterogeneous sub-groups. Each homogenous sub-group contains replicates of either group A only or group B only. Each heterogeneous sub-group has a mix of samples from both the groups A and B.

### Usage

```
getCombs(
  reps_A,
  reps_B,
  batches = FALSE,
  between_combs = NULL,
  within_combs = NULL
)
```

### Arguments

|               |  |
|---------------|--|
| reps_A        | Number of replicates of group A.   |
| reps_B        | Number of replicates of group B.   |
| batches       | Logical suggesting if replicates of group A and B were performed in batches and are labelled accordingly. If TRUE, a heterogeneous/homogeneous subset may not have multiple samples from the same batch. |
| between_combs | Dataframe with each column containing groupings of replicates of groups A and B, which will be used to assess between-group variation.   |
| within_combs  | Data.frame with each column containing groupings of replicates of groups A or B, which will be used to assess within-group variation.  |

### Value

List of two dataframes, containing groupings for within-group and between-group variation.

### Author(s)

Krishna Choudhary

### References

Choudhary, K., Lai, Y. H., Tran, E. J., & Aviran, S. (2019). dStruct: identifying differentially reactive regions from RNA structurome profiling data. *Genome biology*, 20(1), 1-26.

**Examples**

```
#Get heterogeneous and homogeneous set combinations of samples when there are 2 samples of group A and 1 of group B.  
getCombs(2, 1)
```

---

|                  |   |
|------------------|---|
| getContigRegions | <i>Identifies contiguous regions from a list of nucleotide indices.</i> |
|------------------|---|

---

**Description**

Given a sequence of nucleotide indices, this function returns integer ranges covered by the indices. There is an option to merge ranges if they are separated by less than a user-specified distance.

**Usage**

```
getContigRegions(x, gap = 0)
```

**Arguments**

|     |  |
|-----|--|
| x   | A vector of integers.  |
| gap | Include gaps in the ranges if they are shorter than or equal to this length. |

**Value**

IRanges object storing start and end sites of contiguous regions.

**Author(s)**

Krishna Choudhary

**Examples**

```
#Convert an integer vector of nucleotide positions to an IRanges object containing the coordinates of contiguous regions.  
nucleotide_positions <- c(1, 3, 2, 8, 4:7, 11:20)  
getContigRegions(nucleotide_positions)  
  
#Merge regions if their end points are within 3 nt of each other.  
getContigRegions(nucleotide_positions, gap = 3)
```

---

|            |  |
|------------|--|
| getRegions | <i>Constructs potential differentially reactive regions.</i> |
|------------|--|

---

### Description

This function takes between- and within-group  $d$  scores for a transcript as input and identifies regions where the former is generally larger. Regions that pass minimum quality and minimum signal criteria are returned.

### Usage

```
getRegions(
  d_within,
  d_spec,
  rdf,
  min_length = 11,
  check_signal_strength = TRUE,
  check_nucs = TRUE,
  check_quality = TRUE,
  quality = 0.5,
  evidence = 0,
  signal_strength = 0.1
)
```

### Arguments

|                       |   |
|-----------------------|---|
| d_within              | Nucleotide-wise $d$ score for within-group variation.   |
| d_spec                | Nucleotide-wise $d$ score for between-group variation.  |
| rdf                   | Dataframe of reactivities for each sample.  |
| min_length            | Minimum length of constructed regions.  |
| check_signal_strength | Logical, if TRUE, construction of regions must be based on nucleotides that have a minimum absolute value of reactivity.        |
| check_nucs            | Logical, if TRUE, constructed regions must have a minimum number of nucleotides participating in Wilcoxon signed rank test.     |
| check_quality         | Logical, if TRUE, check constructed regions for quality.  |
| quality               | Worst allowed quality for a region to be tested.  |
| evidence              | Minimum evidence of increase in variation from within-group comparisons to between-group comparisons for a region to be tested. |
| signal_strength       | Threshold for minimum signal strength.  |

### Value

Integer vector of nucleotides that constitute potential differentially reactive regions.

**Author(s)**

Krishna Choudhary

**References**

Choudhary, K., Lai, Y. H., Tran, E. J., & Aviran, S. (2019). dStruct: identifying differentially reactive regions from RNA structure profiling data. *Genome biology*, 20(1), 1-26.

---

lai2019

Saccharomyces cerevisiae *Structure-seq* data

---

**Description**

Data from a Structure-seq assay of five samples of *S. cerevisiae*, three of which were wild-type samples and two mutant samples. The data was pre-processed to obtain DMS reactivities as described by Lai et al. (2019).

**Usage**

```
data("lai2019")
```

**Format**

An object of class "list".

**Source**

Raw data from [Lai et al., 2019](#) in processed form.

**References**

Lai et al. (2019) *Genetics*, Vol. 212, 153–174 ([Genetics](#))

**Examples**

```
data("lai2019")
```

---

|            |   |
|------------|---|
| normalizer | Returns normalizer for reactivity vector. |
|------------|---|

---

### Description

Assesses normalization factor for raw reactivities using the 2-8 % method. Given a reactivity profile, first, remove 2% of the nucleotides with the highest reactivities. Then, the normalization factor is the mean of reactivities of the 8% of the nucleotides with the next highest reactivities.

### Usage

```
normalizer(raw.estimates)
```

### Arguments

`raw.estimates` A vector of raw reactivities.

### Value

The normalization factor.

### Author(s)

Krishna Choudhary

### References

Low JT, Weeks KM. SHAPE-directed RNA secondary structure prediction. *Methods*. 2010; 52(2):150–8.

Sloma MF, Mathews DH, Chen SJ, Burke-Aguero DH. Chapter four – improving RNA secondary structure prediction with structure mapping data. In: *Methods in Enzymology*, vol. 553. Cambridge: Academic Press: 2015. p. 91–114.

Choudhary K, Deng F, Aviran S. Comparative and integrative analysis of RNA structural profiling data: current practices and emerging questions. *Quant Biol*. 2017; 5(1):3–24.

### Examples

```
normalizer(c(NA, rnorm(20, 0.5, 0.3), NA, -999))
```

---

plotDStructurome      *Plots differentially reactive regions.*

---

### Description

Given the table of results from dStruct or dStructGuided and the corresponding lists with reactivity scores for all transcripts, this function saves a PDF file with detailed visualizations of reactivities for all differential regions.

### Usage

```
plotDStructurome(  
  r1,  
  diff_regions,  
  outfile,  
  fdr = 0.05,  
  ylim = c(-0.05, 3),  
  del_d_cutoff = 0.01  
)
```

### Arguments

|              |   |
|--------------|---|
| r1           | List of dataframes of reactivities for each sample.   |
| diff_regions | Output from dStruct or dStructGuided containing coordinates of regions with significance of differentially reactivity.          |
| outfile      | The name for pdf file which will be saved.  |
| fdr          | FDR threshold for plotted regions.  |
| ylim         | Y-axis limits for plots.  |
| del_d_cutoff | Minimum effect size for plotted regions specified in terms of median difference of the between-group and within-group d-scores. |

### Value

Saves a PDF for all differentially reactive regions. Returns NULL.

### Author(s)

Krishna Choudhary

### References

Choudhary, K., Lai, Y. H., Tran, E. J., & Aviran, S. (2019). dStruct: identifying differentially reactive regions from RNA structurome profiling data. *Genome biology*, 20(1), 1-26.

## Examples

```
#Load data from Lai et al., 2019
data(lai2019)

#Run dStruct in de novo discovery mode for all the transcripts in this data in one step.
res <- dStructome(lai2019, 3, 2, batches= TRUE, min_length = 21,
  between_combs = data.frame(c("A3", "B1", "B2")),
  within_combs = data.frame(c("A1", "A2", "A3")),
  ind_regions = TRUE, processes = 1)

#Plot the significant results and save to a PDF file.
plotDStructurome(rl = lai2019,
  diff_regions = res,
  outfile = "significantly_differential_regions",
  fdr = 0.05,
  ylim = c(-0.05, 3))
```

---

|                   |                                      |
|-------------------|--------------------------------------|
| twoEightNormalize | <i>Normalizes reactivity vector.</i> |
|-------------------|--------------------------------------|

---

## Description

Given a reactivity profile, first, remove 2% of the nucleotides with the highest reactivities. Then, the normalization factor is the mean of reactivities of the 8% of the nucleotides with the next highest reactivities. The raw reactivities are divided by the normalization factor to get normalized reactivities. This is called as 2-8 % normalization and has been a common way to normalize data from RNA structurome profiling technologies such as SHAPE-Seq, Structure-Seq, etc. (see Low and Weeks, 2010, Sloma et al., 2015, and Choudhary et al., 2017).

## Usage

```
twoEightNormalize(raw.estimated)
```

## Arguments

`raw.estimated` A vector of raw reactivities.

## Value

A vector of normalized reactivities.

## Author(s)

Krishna Choudhary



## References

- Low JT, Weeks KM. SHAPE-directed RNA secondary structure prediction. *Methods*. 2010; 52(2):150–8.
- Sloma MF, Mathews DH, Chen SJ, Burke-Aguero DH. Chapter four – improving RNA secondary structure prediction with structure mapping data. In: *Methods in Enzymology*, vol. 553. Cambridge: Academic Press; 2015. p. 91–114.
- Choudhary K, Deng F, Aviran S. Comparative and integrative analysis of RNA structural profiling data: current practices and emerging questions. *Quant Biol*. 2017; 5(1):3–24.

## Examples

```
twoEightNormalize(c(NA, rnorm(20, 0.5, 0.3), NA, -999))
```

---

wan2014

Homo sapiens *PARS data*

---

## Description

Data from a PARS assay of a family trio of mother, father, and child. The data was pre-processed to obtain PARS scores as described in Choudhary et al. (2019).

## Usage

```
data(wan2014)
```

## Format

An object of class "list".

## Source

Counts data from [Wan et al., 2014](#) in processed form.

## References

- Wan et al., *Nature*, 505, 706–709 (2014) ([Nature](#))

## Examples

```
data(wan2014)
```

# Index

## \* datasets

lai2019, [13](#)

wan2014, [17](#)

calcDis, [2](#)

dCombs, [3](#)

dStruct, [4](#)

dStructGuided, [6](#)

dStructome, [7](#)

getCombs, [10](#)

getContigRegions, [11](#)

getRegions, [12](#)

lai2019, [13](#)

normalizer, [14](#)

plotDStructurome, [15](#)

twoEightNormalize, [16](#)

wan2014, [17](#)