Package 'BUMHMM'

November 11, 2026

Type Package

Title Computational pipeline for computing probability of modification from structure probing experiment data

Version 1.36.0

Date 2021-11-20

Author Alina Selega (alina.selega@gmail.com), Sander Granneman, Guido Sanguinetti

Maintainer Alina Selega <alina.selega@gmail.com>

Description This is a probabilistic modelling pipeline for computing per- nucleotide posterior probabilities of modification from the data collected in structure probing experiments. The model supports multiple experimental replicates and empirically corrects coverage- and sequence-dependent biases. The model utilises the measure of a ``drop-off rate'' for each nucleotide, which is compared between replicates through a log-ratio (LDR). The LDRs between control replicates define a null distribution of variability in drop-off rate observed by chance and LDRs between treatment and control replicates gets compared to this distribution. Resulting empirical p-values (probability of being ``drawn'' from the null distribution) are used as observations in a Hidden Markov Model with a Beta-Uniform Mixture model used as an emission model. The resulting posterior probabilities indicate the probability of a nucleotide of having being modified in a structure probing experiment.

Depends R (>= 3.4)

License GPL-3

Imports devtools, stringi, gtools, stats, utils, SummarizedExperiment, Biostrings, IRanges

Suggests testthat, knitr, BiocStyle

VignetteBuilder knitr

LazyData true

biocViews ImmunoOncology, GeneticVariability, Transcription, GeneExpression, GeneRegulation, Coverage, Genetics, StructuralPrediction, Transcriptomics, Bayesian, Classification, FeatureExtraction, HiddenMarkovModel, Regression, RNASeq, Sequencing

git_url https://git.bioconductor.org/packages/BUMHMM

git_branch RELEASE_3_23

git_last_commit c2c2d55

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Contents

computeProbs 2
computeStretches 5
covFile 6
docFile 7
dorFile 8
findPatternPos 9
nuclPerm 10
scaleDOR 11
se 12
selectNuclPos 14
seq 15
stabiliseVariance 16

Index **18**

`computeProbs` *Function to compute posterior probabilities of modification with the BUM-HMM model for all nucleotides.*

Description

This function computes posterior probabilities of chemical modification in a structure probing experiment. It uses empirical p-values, which quantify the difference between LDRs (log-ratios of drop-off rates) from treatment-control and control-control experimental comparisons, as observations in a HMM with an emission model defined as a Beta-Uniform mixture.

Usage

```
computeProbs(LDR_C, LDR_CT, Nc, Nt, strand, nuclPosition, analysedC,
                    analysedCT, stretches, optimise)
```

Arguments

LDR_C	A matrix of transformed LDRs for control-control comparisons as returned by stabiliseVariance. The matrix rows correspond to nucleotide positions and columns to control-control comparisons.
LDR_CT	A matrix of transformed LDRs for treatment-control comparisons as returned by stabiliseVariance. The matrix rows correspond to nucleotide positions and columns to treatment-control comparisons.
Nc	Number of control experimental replicates. Must be at least 2.
Nt	Number of treatment experimental replicates. Must be at least 2.
strand	A character, indicating the plus (+) or minus strand (-).
nuclPosition	A list where each component corresponds to a pattern (indicated by the field names) and contains indices of the middle nucleotides of that pattern's occurrences within the sequence, as returned by findPatternPos. If the sequence bias is not addressed and all nucleotide positions are used together (e.g. for short transcripts), the only component of the list should contain all nucleotide positions of the sequence.
analysedC	The first element of the list returned by selectNuclPos, containing the positions of nucleotides selected for all control-control comparisons.
analysedCT	The second element of the list returned by selectNuclPos, containing the positions of nucleotides selected for all treatment-control comparisons.
stretches	An IRanges object storing each uninterrupted stretch of nucleotides for which posterior probabilities are to be computed, as returned by computeStretches.
optimise	An indicator variable to use the EM optimisation of shape parameters of the Beta distribution (emission model of the HMM for the modified state). If optimisation is to be used, the variable should be set to the desired tolerance for parameter estimation (once the estimates stop changing within this tolerance, the algorithm stops). Set to NULL by default.

Details

The function first computes percentiles of LDR null distributions, for each percent from 1 to 89 and for a tenth of a percent from 90 to 100. If multiple nucleotide patterns are used, then null distributions are computed separately for each pattern. Then, LDRs from treatment-control comparisons are compared to the percentiles of a corresponding null distribution via computing the p-values, defined as 1 - closest percentile. Thus, if the closest percentile to an LDR is 99th, then its p-value will be 0.01, reflecting a small probability for it to be from this null distribution. The p-values are then used as observations in a hidden Markov model. P-values corresponding to multiple comparisons at the same nucleotide position are considered to be independent measurements.

HMM has a binary latent state corresponding to the true state of a nucleotide, modified or unmodified. Transition probabilities are set according to the empirically derived expectations of unmodified nucleotide stretches of average length 20 and modified stretches of average length 5. Emission probabilites come from a Beta-Uniform mixture, where the Uniform component represents the conditional model for the unmodifed state, and the Beta component represents the conditional model for the modified state. The shape parameters of Beta distribution are set to (1, 10) to approximately

match the likelihood under the null hypothesis (unmodified state) for the LDRs in the top quintile of the distribution.

We also provide an option to optimise the shape parameters of Beta distribution with the EM algorithm. To do this, make the function call with the last parameter `optimise` set to the desired tolerance, within which the parameter estimates are allowed not to change at the next iteration. The EM algorithm is run for 10 iterations and each M-step is performed with Newton's optimisation method and is allowed 20 iterations. We remark that during our experiments, this optimisation appeared vulnerable to local minima. Thus, the default version of the pipeline does not use this optimisation.

Value

An n-by-2 matrix of posterior probabilities for all selected nucleotides. First column corresponds to the probability of the nucleotide being unmodified (given the observations) and the second column - to the probability of the nucleotide being modified. Those nucleotides not belonging to stretches for which the HMM was run are assigned an 'NA'.

Error

The following errors are returned if:

"Number of control and treatment replicates must be at least 2." the number of control or treatment experimental replicates is less than 2;

"All lists of positions selected for pair-wise comparisons should be non-empty." a list of nucleotides for control-control or treatment-control comparisons is empty;

"Strand should be either plus or minus, specified with a sign." strand is not specified as "+" or "-";

"The list of considered nucleotide positions should be non-empty. If patterns are not used, a vector with all positions should be provided in the first element of the list." the list of considered nucleotides (which should either correspond to patterns if addressing sequence bias or list all positions) is empty;

"The matrix of control-control (treatment-control) LDRs should have as many columns as there are control-control (treatment-control) comparisons." dimensionalities of matrices with LDRs do not agree with the number of pair-wise comparisons;

"Please provide a tolerance if shape parameters are to be optimised with EM algorithm." the `optimise` parameter was not set to a (real) tolerance value.

Author(s)

Alina Selega, Sander Granneman, Guido Sanguinetti

References

Selega et al. "Robust statistical modeling improves sensitivity of high-throughput RNA structure probing experiments", Nature Methods (2016).

See Also

See Also `stabiliseVariance`, `selectNuclPos`, `findPatternPos`, and `computeStretches`.

Examples

```
library(SummarizedExperiment)

Nc <- 3
Nt <- 3
t <- 1

nuclSelection <- selectNuclPos(se, Nc, Nt, t)

assay(se, "dropoff_rate") <- scaleDOR(se, nuclSelection, Nc, Nt)

stretches <- computeStretches(se, t)

varStab <- stabiliseVariance(se, nuclSelection, Nc, Nt)

LDR_C <- varStab$LDR_C
LDR_CT <- varStab$LDR_CT

sequence <- subject(rowData(se)$nucl)
nuclPosition <- list()
nuclPosition[[1]] <- 1:nchar(sequence)

posteriors <- computeProbs(LDR_C, LDR_CT, Nc, Nt, '+', nuclPosition,
                           nuclSelection$analysedC, nuclSelection$analysedCT,
                           stretches)
```

`computeStretches` *Function to compute continuous stretches of nucleotide positions.*

Description

This function computes continuous stretches of nucleotide positions, for which posterior probabilities are to be computed. Such positions should have the minimum allowed coverage (as defined with the parameter `t`) in all experimental replicates and a non-zero drop-off count in at least one of the treatment replicates. The returned stretches are at least two nucleotides long.

Usage

```
computeStretches(se, t)
```

Arguments

`se`	A `SummarizedExperiment` object storing structure probing data and the associated genomic sequence. The documentation for the example data set provided with the package `se` outlines how the object should be defined. `computeStretches` uses the assays "coverage" and "dropoff_count".
`t`	Threshold for the minimum allowed coverage. Must be non-negative.

Value

An IRanges object storing each stretch.

Error

The following errors are returned if:

"The minumum coverage threshold must be non-negative." the threshold for the minimum considered coverage is negative.

Author(s)

Alina Selega, Sander Granneman, Guido Sanguinetti

References

Selega et al. "Robust statistical modeling improves sensitivity of high-throughput RNA structure probing experiments", Nature Methods (2016).

Examples

```
t <- 1
stretches <- computeStretches(se, t)
```

---

covFile *Example coverage data set.*

Description

A matrix containing 1800x6 entries of per nucleotide coverage information from structure probing experiments.

This matrix is provided as a reference for creating a SummarizedExperiment object for storing all structure probing data used by the BUMHMM package. See se for the example code.

Usage

```
covFile
```

Format

A matrix containing the coverage information for each nucleotide position of a ribosomal RNA 18S obtained in a structure probing experiment using DMS as a probe, where:

1st column control experimental replicate

2nd column control experimental replicate

3rd column control experimental replicate

4th column treatment experimental replicate

5th column treatment experimental replicate

6th column treatment experimental replicate

Value

Coverage information per nucleotide.

References

Hector, R. D. et al. "Snapshots of pre-rRNA structural flexibility reveal eukaryotic 40S assembly dynamics at nucleotide resolution." Nucleic acids research (2014).

`docFile` *Example drop-off count data set.*

Description

A matrix containing 1800x6 entries of per nucleotide drop-off count information from structure probing experiments.

This matrix is provided as a reference for creating a `SummarizedExperiment` object for storing all structure probing data used by the `BUMHMM` package. See `se` for the example code.

Usage

```
docFile
```

Format

A matrix containing the drop-off count information for each nucleotide position of a ribosomal RNA 18S obtained in a structure probing experiment using DMS as a probe, where:

1st column control experimental replicate

2nd column control experimental replicate

3rd column control experimental replicate

4th column treatment experimental replicate

5th column treatment experimental replicate

6th column treatment experimental replicate

Value

Drop-off count information per nucleotide.

References

Hector, R. D. et al. "Snapshots of pre-rRNA structural flexibility reveal eukaryotic 40S assembly dynamics at nucleotide resolution." Nucleic acids research (2014).

dorFile *Example drop-off rate data set.*

Description

A matrix containing 1800x6 entries of per nucleotide drop-off rate information from structure probing experiments.

This matrix is provided as a reference for creating a `SummarizedExperiment` object for storing all structure probing data used by the `BUMHMM` package. See `se` for the example code.

Usage

```
dorFile
```

Format

A matrix containing the drop-off rate information for each nucleotide position of a ribosomal RNA 18S obtained in a structure probing experiment using DMS as a probe, where:

1st column control experimental replicate

2nd column control experimental replicate

3rd column control experimental replicate

4th column treatment experimental replicate

5th column treatment experimental replicate

6th column treatment experimental replicate

Value

Drop-off rate information per nucleotide.

References

Hector, R. D. et al. "Snapshots of pre-rRNA structural flexibility reveal eukaryotic 40S assembly dynamics at nucleotide resolution." Nucleic acids research (2014).

`findPatternPos` *Function to find positions of the nucleotide patterns in the sequence.*

Description

This function finds all occurrences of a nucleotide pattern in the sequence. For each occurrence, the function returns the index of the middle nucleotide, computed as: `ceiling(length(pattern) / 2)`. The function supports data for the plus and minus DNA strands; for the minus strand, all patterns are turned to complementary sequence.

Usage

```
findPatternPos(patterns, sequence, strand)
```

Arguments

`patterns`	A list of nucleotide permutations of length `n`, as returned by `nuclPerm`.
`sequence`	A `DNAString` object storing the reference genomic sequence to search for the patterns in. The sequence corresponding to plus strand is expected.
`strand`	A character, indicating the plus (`+`) or minus strand (`-`). For the minus strand, the occurrences found for a particular pattern will be attributed to the pattern with complementary sequence.

Details

This function uses `stringi::stri_locate_all_fixed()`.

This function aims to assist with addressing sequence bias in structure probing data. The sequence in the neighbourhood of a nucleotide is assumed to have an effect on its structural state. By considering sequence patterns of a certain length (specified by the user), this function finds indices of the middle nucleotide of each pattern's occurrences within the sequence. We then separately analyse the nucleotides occurring in the middle of each pattern, taking into account sequence dependency.

Value

This function returns a list where each component corresponds to a pattern (indicated by the field `names`) and contains indices of the middle nucleotides of that pattern's occurrences within the sequence.

Error

The following errors are returned if:

"Strand should be either plus or minus, specified with a sign." strand is not specified as "+" or "-";

"The sequence should be non-empty." provided sequence is empty;

"The list of patterns should be non-empty." the list of patterns to search for in the sequence is empty.

Author(s)

Alina Selega, Sander Granneman, Guido Sanguinetti

References

Selega et al. "Robust statistical modeling improves sensitivity of high-throughput RNA structure probing experiments", Nature Methods (2016).

See Also

See also `nuclPerm`.

Examples

```
library(SummarizedExperiment)

## Extract the DNA sequence from se
sequence <- subject(rowData(se)$nucl)

## Generate patterns of length 3
n <- 3
patterns <- nuclPerm(n)

## Find positions of pattern occurrences
nuclPosition <- findPatternPos(patterns, sequence, '+')
```

`nuclPerm` *Function to create nucleobase patterns.*

Description

This function creates all permutations of 4 nucleobases (A, T, G, C) of length `n`.

Usage

```
nuclPerm(n)
```

Arguments

`n`	Length of the pattern.

Details

This function uses `gtools::permutations()`.

Value

A vector of characters with each element being a nucleobase pattern of length `n`.

Error

The following error is returned if:

"The length of patterns provided is not a positive number." the provided length of patterns to be generated is not positive.

Author(s)

Alina Selega, Sander Granneman, Guido Sanguinetti

References

Selega et al. "Robust statistical modeling improves sensitivity of high-throughput RNA structure probing experiments", Nature Methods (2016).

See Also

`gtools::permutations()`

Examples

```
nuclPerm(3)
```

`scaleDOR` *Function to scale drop-off rates across replicates to a common median.*

Description

The function extracts all nucleotides selected for control-control and treatment-control comparisons and scales them to have a common median across all replicates. This value is computed as the median of all drop-off rates of the selected nucleotides in control replicate samples. The function requires the output of `selectNuclPos`, which holds lists of nucleotide positions selected for pairwise comparisons.

Usage

```
scaleDOR(se, nuclSelection, Nc, Nt)
```

Arguments

`se`	A `SummarizedExperiment` object storing structure probing data and the associated genomic sequence. The documentation for the example data set provided with the package `se` outlines how the object should be defined. `scaleDOR` uses the assay "dropoff_rate".
`nuclSelection`	A list returned by `selectNuclPos`, containing the positions of nucleotides selected for all control-control and treatment-control comparisons.
`Nc`	Number of control replicate samples. Must be at least 2.
`Nt`	Number of treatment replicate samples. Must be at least 2.

Value

Returns a modified n-by-m matrix of drop-off rates, scaled per replicate to have the same median value, where n is the number of nucleotides and m is the total number of replicate samples.

Error

The following errors are returned if:

"Number of control and treatment replicates must be at least 2." the number of control or treatment experimental replicates is less than 2;

"Nucleotide selection should have two elements." "All lists of positions selected for pair-wise comparisons or for which posteriors will be computed should be non-empty." the list containing positions of nucleotides selected for control-control and treatment-control comparisons does not have 2 elements or any of the elements is empty;

"Drop-off rate matrix should not have any NA entries." the drop-off rate matrix has NA entries.

Author(s)

Alina Selega, Sander Granneman, Guido Sanguinetti

References

Selega et al. "Robust statistical modeling improves sensitivity of high-throughput RNA structure probing experiments", Nature Methods (2016).

See Also

See Also selectNuclPos.

Examples

```
Nc <- 3
Nt <- 3
t <- 1
nuclSelection <- selectNuclPos(se, Nc, Nt, t)
dorFile <- scaleDOR(se, nuclSelection, Nc, Nt)
```

---

se *Example RNA structure probing data set.*

Description

A SummarizedExperiment container storing the genomic sequence of the yeast rRNA 18S and three matrices with coverage, drop-off count, and drop-off rate information for each nucleotide position in three replicates of control and treatment DMS structure probing experiments.

Usage

```
se
```

Format

An instance of the SummarizedExperiment class storing data obtained in a structure probing experiment for the yeast rRNA 18S using the DMS chemical probing agent. The data can be accessed in the Gene Expression Omnibus with the code GSE52878.

The available data types are summarised in three assays: coverage, drop-off count of the reverse transcriptase, and drop-off rate at each nucleotide position. The coverage and drop-off count data are obtained in a random-primed paired-end sequencing structure probing experiment. The drop-off rate is computed as the ratio between the drop-off count and coverage at each nucleotide position. Control (no reagent added) and treatment experiments are available in triplicates.

The data from each assay is summarised in a rectangular matrix. The rows represent nucleotide position and the columns represent experimental replicates. The replicates are labelled as "control" or "treatment", which is stored as column data and can be accessed through the field `replicate`. The row data stores the associated DNA sequence, represented with a `DNAString` object.

Details

The SummarizedExperiment object can be created from individual matrices and the sequence string as follows. Below:

`seq` a string storing the genomic sequence

`covFile` a matrix storing coverage information, with rows as nucleotide positions and columns as experimental replicates (control replicates come first, followed by treatment replicates)

`docFile` a matrix storing drop-off count information, arranged as `covFile`

`dorFile` a matrix storing drop-off rate information, arranged as `covFile`

The `seq`, `covFile`, `docFile`, and `dorFile` are also provided as accompanying data sets for reference.

Example code:

library(BUMHMM) library(Biostrings) library(SummarizedExperiment)

dna = DNAString(seq) se <- SummarizedExperiment(list(coverage=as.matrix(covFile), dropoff_count=as.matrix(docFile), dropoff_rate=as.matrix(dorFile)), colData=DataFrame(replicate=rep(c("control", "treatment"), each=3)), rowData=DataFrame(nucl=Views(dna, successiveIRanges(rep(1, nchar(dna)))))) colnames(se) <- c('C1', 'C2', 'C3', 'T1', 'T2', 'T3')

Value

RNA structure probing data.

References

Hector, R. D. et al. "Snapshots of pre-rRNA structural flexibility reveal eukaryotic 40S assembly dynamics at nucleotide resolution." Nucleic acids research (2014).

`selectNuclPos` *Function to compute selections of nucleotide positions.*

Description

This function selects pairs of nucleotide positions for computing log-ratios of control-control and treatment-control replicate comparisons.

Usage

```
selectNuclPos(se, Nc, Nt, t)
```

Arguments

`se`	A `SummarizedExperiment` object storing structure probing data and the associated genomic sequence. The documentation for the example data set provided with the package `se` outlines how the object should be defined. `selectNuclPos` uses the assays "coverage" and "dropoff_count".
`Nc`	Number of control experimental replicates. Must be at least 2.
`Nt`	Number of treatment experimental replicates. Must be at least 2.
`t`	Threshold for the minimum allowed coverage. Must be non-negative.

Details

This function uses `combn`.

Value

`analysedC`	List where each element corresponds to a control-control replicate comparison. Each element holds indices of nucleotides that have coverage >= `t` and a drop-off count > 0 in both replicates of that comparison.
`analysedCT`	List where each element corresponds to a treatment-control replicate comparison. Each element holds indices of nucleotides that have coverage >= `t` and a drop-off count > 0 in both replicates of that comparison.

Error

The following errors are returned if:

"The number of experimental replicates must be at least 2." the number of control or treatment experimental replicates is less than 2;

"The minumum coverage threshold must be non-negative." the threshold for the minimum considered coverage is negative;

"The coverage and drop-off count matrices should not have NA entries." the coverage and drop-off count matrices have NA entries.

Author(s)

Alina Selega, Sander Granneman, Guido Sanguinetti

References

Selega et al. "Robust statistical modeling improves sensitivity of high-throughput RNA structure probing experiments", Nature Methods (2016).

Examples

```
selectNuclPos(se, 3, 3, 1)
```

---

seq *Example genomic sequence string.*

Description

A string containing genomic sequence of a ribosomal RNA 18S.

This string is provided as a reference for creating a `SummarizedExperiment` object for storing all structure probing data used by the `BUMHMM` package. See `se` for the example code.

Usage

```
seq
```

Format

A string containing genomic sequence of a ribosomal RNA 18S.

Value

Genomic sequence.

References

RDN18-1, S000006482, Saccharomyces Genome Database.

`stabiliseVariance` *Function to reduce variance of LDRs as a function of coverage.*

Description

This function computes the log drop-off rate ratios (LDRs) for control-control and treatment-control comparisons and applies a transformation to them that reduces their variance as a function of coverage.

Usage

```
stabiliseVariance(se, nuclSelection, Nc, Nt)
```

Arguments

`se`	A `SummarizedExperiment` object storing structure probing data and the associated genomic sequence. The documentation for the example data set provided with the package `se` outlines how the object should be defined. `stabiliseVariance` uses the assays "coverage" and "dropoff_rate".
`nuclSelection`	A list returned by `selectNuclPos`, containing the positions of nucleotides selected for all control-control and treatment-control comparisons.
`Nc`	Number of control experimental replicates. Must be at least 2.
`Nt`	Number of treatment experimental replicates. Must be at least 2.

Details

The variance is reduced by sorting all LDRs in the null distribution by the average coverage and splitting the data in bins. Each bin spans a coverage of 100; or, if maximum coverage is not larger by 100 than the minimum coverage, the range is set to their difference divided by 10.

For each bin, the 95th quantile of LDRs with subtracted mean and the average coverage are computed. Then non-linear least squares are used to fit the following model (with parameters k, b): f = 1/sqrt(n) * k + b, for f - quantiles, n - mean coverage in the bin.

All LDRs are then rescaled by this model according to their corresponding average coverage in the pair of replicates.

Value

`LDR_C`	A matrix of transformed LDRs for control-control comparisons. The matrix rows correspond to nucleotide positions and columns to a control-control comparison. Only those positions selected for a pair-wise comparison will be assigned a value; the rest will be left as an `NA`.
`LDR_CT`	A matrix of transformed LDRs for treatment-control comparisons. The matrix rows correspond to nucleotide positions and columns to a treatment-control comparison. Only those positions selected for a pair-wise comparison will be assigned a value; the rest will be left as an `NA`.

Error

The following errors are returned if:

"Number of control and treatment replicates must be at least 2." the number of control or treatment experimental replicates is less than 2;

"All lists of positions selected for pair-wise comparisons should be non-empty." a list of nucleotides for control-control or treatment-control comparisons is empty;

"The coverage and drop-off count matrices should not have NA entries." the coverage and drop-off count matrices have NA entries;

"Unable to fit the model for correcting the coverage bias." The function `nls` could not execute successfully. The 95th quantiles of the LDR distribution in each bin could be equal to 0 or not have enough elements. This would happen if not enough nucleotides end up in a bin; e.g. one nucleotide per bin.

Author(s)

Alina Selega, Sander Granneman, Guido Sanguinetti

References

Selega et al. "Robust statistical modeling improves sensitivity of high-throughput RNA structure probing experiments", Nature Methods (2016).

See Also

See Also `selectNuclPos`.

Examples

```
library(SummarizedExperiment)
Nc <- 3
Nt <- 3
t <- 1
nuclSelection <- selectNuclPos(se, Nc, Nt, t)
assay(se, "dropoff_rate") <- scaleDOR(se, nuclSelection, Nc, Nt)
varStab <- stabiliseVariance(se, nuclSelection, Nc, Nt)
```

Index

∗ **arith**
computeStretches, 5
∗ **character**
nuclPerm, 10
∗ **classes&character**
findPatternPos, 9
∗ **classif**
computeProbs, 2
∗ **datasets**
covFile, 6
docFile, 7
dorFile, 8
se, 12
seq, 15
∗ **htest**
computeProbs, 2
∗ **manip**
computeStretches, 5
findPatternPos, 9
scaleDOR, 11
selectNuclPos, 14
∗ **models&nonlinear**
stabiliseVariance, 16
∗ **models**
computeProbs, 2
∗ **univar**
scaleDOR, 11

combn, *14*
computeProbs, 2
computeStretches, *3, 4*, 5
covFile, 6

DNAString, *9*
docFile, 7
dorFile, 8

findPatternPos, *3, 4*, 9

gtools::permutations(), *10, 11*

IRanges, *3*, *6*

nls, *17*
nuclPerm, *9, 10*, 10

scaleDOR, 11
se, *5–8*, *11*, 12, *14–16*
selectNuclPos, *3, 4*, *11, 12*, 14, *16, 17*
seq, 15
stabiliseVariance, *3, 4*, 16
stringi::stri_locate_all_fixed(), *9*
SummarizedExperiment, *5*, *11–14*, *16*