Package 'MIRA'

November 12, 2026

Version 1.34.0

Date 2021-11-21

Title Methylation-Based Inference of Regulatory Activity

Description DNA methylation contains information about the regulatory state of the cell. MIRA aggregates genome-scale DNA methylation data into a DNA methylation profile for a given region set with shared biological annotation. Using this profile, MIRA infers and scores the collective regulatory activity for the region set. MIRA facilitates regulatory analysis in situations where classical regulatory assays would be difficult and allows public sources of region sets to be leveraged for novel insight into the regulatory state of DNA methylation datasets.

Author Nathan Sheffield <http://www.databio.org> [aut],
Christoph Bock [ctb],
John Lawson [aut, cre]

Maintainer John Lawson <jtl2hk@virginia.edu>

Depends R (>= 3.5)

Imports BiocGenerics, S4Vectors, IRanges, GenomicRanges, data.table, ggplot2, Biobase, stats, bsseq, methods

Suggests knitr, parallel, testthat, BiocStyle, rmarkdown, AnnotationHub, LOLA

VignetteBuilder knitr

License GPL-3

biocViews ImmunoOncology, DNAMethylation, GeneRegulation, GenomeAnnotation, SystemsBiology, FunctionalGenomics, ChIPSeq, MethylSeq, Sequencing, Epigenetics, Coverage

URL http://databio.org/mira

BugReports https://github.com/databio/MIRA

RoxygenNote 6.1.0

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

git_branch RELEASE_3_23

git_last_commit 34b8921

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Contents

addMethPropCol . . . 2
aggregateMethyl . . . 3
bigBinDT1 . . . 4
bigBinDT2 . . . 5
binRegion . . . 5
BSBinAggregate . . . 6
BSreadBiSeq . . . 7
bsseqToDataTable . . . 8
calcMIRAScore . . . 9
exampleAnnoDT1 . . . 11
exampleAnnoDT2 . . . 11
exampleBins . . . 11
exampleBSDT . . . 12
exampleBSseqObj . . . 12
exampleRegionSet . . . 13
MIRA . . . 13
plotMIRAProfiles . . . 14
plotMIRAScores . . . 15
rbindNamedList . . . 16
SummarizedExperimentToDataTable . . . 16

Index **19**

`addMethPropCol` *Add column for proportion of methylation*

Description

Adding methylProp column that has proportion of reads that were methylated for each site based on number of methylated reads divided by total number of reads. Note: Assigns methylProp column by reference with ":="

Usage

```
addMethPropCol(BSDTList)
```

Arguments

BSDTList	A bisulfite datatable or list of datatables with a column for number of methylated reads (methylCount) and a column for number of total reads (coverage) for each cytosine that was measured.

Value

The BSDTList but with extra ‘methylProp‘ column on each data.table in list.

Examples

```
data("exampleBSDT", package = "MIRA")
exampleBSDT[, methylProp := NULL] # removing methylProp column
addMethPropCol(list(exampleBSDT))
```

---

aggregateMethyl *Aggregate methylation data to get a summary methylation profile for each region set*

Description

The main function for aggregating methylation data in MIRA analysis. Aggregates methylation across all regions in a given region set to give a summary methylation profile for each region set.

Usage

```
aggregateMethyl(BSDT, GRList, binNum = 11, minBaseCovPerBin = 500)
```

Arguments

BSDT	A single data.table that has DNA methylation data on individual sites. Alternatively a BSseq object is allowed which will be converted internally to data.tables. The data.table input should have columns: "chr" for chromosome, "start" for cytosine coordinate, "methylProp" for proportion of methylation (0 to 1), optionally "methylCount" for number of methylated reads, and optionally "coverage" for total number of reads.
GRList	A GRangesList object containing region sets, each set corresponding to a type of regulatory element. Each region set in the list should be named. A named list of data.tables also works.
binNum	How many bins each region should be split into for aggregation of the DNA methylation data.
minBaseCovPerBin	Screen out region sets that have any bins in the final methylation profile with 'sumCoverage' below the 'minBaseCovPerBin' threshold. 'sumCoverage' is an output column: during aggregation, the 'coverage' values for each base in a bin are added, then these sums are added for corresponding bins from all regions, producing a 'sumCoverage' value for each bin. 'minBaseCovPerBin' is

only used if there is a "coverage" column in the input methylation data.table. 'sumCoverage' is greater than or equal to the number of separate reads that contributed to a given bin.

Details

Each region is split into bins. For a given set of regions, methylation is first aggregated (averaged) within each bin in each region. Then methylation from corresponding bins from each region are aggregated (averaged) across all regions (all first bins together, all second bins together, etc.), giving an aggregate methylation profile. This process is done for each region set.

Value

a data.table with binNum rows for each region set containing aggregated methylation data. If the input was a BSseq object with multiple samples, a list of data.tables will be returned with one data.table for each sample. Each region was split into bins; methylation was put in these bins; Output contains sum of the all corresponding bins for the regions of each region set, ie for all regions in each region set: first bins summed, second bins summed, etc. Columns of the output should be "bin", "methylProp", "sumCoverage" (only if coverage was an input column, described below), "featureID" (ID for the region set). For information on symmetry of bins and output when a region set has strand info, see ?BSBinAggregate. 'sumCoverage' is calculated as follows: during aggregation, the 'coverage' values for each base in a bin are added, then these sums are added for corresponding bins from all regions, producing a 'sumCoverage' value for each bin.

Examples

```
data("exampleBSDT", package = "MIRA")
data("exampleRegionSet", package = "MIRA")
exBinDT <- aggregateMethyl(exampleBSDT, exampleRegionSet)
```

---

bigBinDT1 *DNA methylation data for the "Applying MIRA to a Biological Question" vignette (1st part).*

Description

A data.table with binned methylation data for the first part of the "Applying MIRA to a Biological Question" vignette. Has data for 3 Ewing samples and 3 muscle-related samples (human skeletal myoblasts and other related samples). It is the product of 'aggregateMethyl'.

Usage

```
data(bigBinDT1)
```

Format

A data.table, data.frame object

bigBinDT2 — *DNA methylation data for the "Applying MIRA to a Biological Question" vignette (2nd part).*

Description

A data.table with binned methylation data for the second part of the "Applying MIRA to a Biological Question" vignette. Has data for 12 Ewing samples and 12 muscle-related samples (human skeletal myoblasts and other related samples). It is the product of ‘aggregateMethyl‘.

Usage

```
data(bigBinDT2)
```

Format

A data.table, data.frame object

binRegion — *Divide region into similarly sized bins.*

Description

Given a start, end, and number of bins, to divide, this function will split the regions into bins. Bins will be only approximately the same size, due to rounding. (they should not be more than 1 different).

Usage

```
binRegion(start, end, bins, idDF = NULL, strand = "*")
```

Arguments

start	Coordinate for beginning of range/range.
end	Coordinate for end of range/region.
bins	How many bins to divide this range/region.
idDF	A string/vector of strings that has chromosome (e.g. "chr1") for given start and end values
strand	"strand" column of the data.table (or single strand value if binRegion is only used on one region). Default is "*".

Details

Use case: take a set of regions, like CG islands, and bin them; now you can aggregate signal scores across the bins, giving you an aggregate signal in bins across many regions of the same type.

In theory, this just runs on 3 values, but you can run it inside a data.table j expression to divide a bunch of regions in the same way.

Value

A data.table, expanded to nrow = number of bins, with these id columns: id: region ID binID: repeating ID (this is the value to aggregate across) ubinID: unique bin IDs

Examples

```
library(data.table)
start <- c(100, 1000, 3000)
end <- c(500, 1400, 3400)
chr <- c("chr1", "chr1", "chr2")
strand <- c("*", "*", "*")
# strand not included in object
# since MIRA assumes "*" already unless given something else
regionsToBinDT <- data.table(chr, start, end)
numberOfBins <- 15
# data.table "j command" using column names and numberOfBins variable
binnedRegionDT <- regionsToBinDT[, binRegion(start, end, numberOfBins, chr)]
```

BSBinAggregate *Bins regions and aggregates methylation data across the regions by bin*

Description

First bins regions and averages the proportion of methylation for all methylation sites within each bin (ie the methylation of all sites within region 1, bin 1 are averaged, then all sites within region 1, bin 2 are averaged, etc.) Then aggregates methylation across all regions by bin by averaging the proportion of methylation in each corresponding bin (ie all bin1's together, all bin2's together, etc.).

Usage

```
BSBinAggregate(BSDT, rangeDT, binNum, minBaseCovPerBin = 500,
  byRegionGroup = TRUE, splitFactor = NULL, hasCoverage = TRUE)
```

Arguments

BSDT	A single data table that has DNA methylation data on individual sites including a "chr" column with chromosome, a "start" column with the coordinate number for the cytosine, a "methylProp" column with proportion of methylation (0 to 1), optionally a "methylCount" column with number of methylated reads for each site, and optionally a "coverage" column with total number of reads for each site (hasCoverage param).

rangeDT	A data table with the sets of regions to be binned, with columns named "start", "end". Strand may also be given and will affect the output. See "Value" section.
binNum	Number of bins across the region.
minBaseCovPerBin	Filter out bins where the sum of coverage values is less than X before returning.
byRegionGroup	Default TRUE will aggregate methylation over corresponding bins for each region (all bin1's aggregated, all bin2's, etc). byRegionGroup = FALSE is deprecated.
splitFactor	With default NULL, aggregation will be done separately/individually for each sample.
hasCoverage	Default TRUE. Whether there is a coverage column

Value

With splitFactor = NULL, it will return a data.table with binNum rows, containing aggregated methylation data over regions in region set "rangeDT". Each region was split into bins; methylation was put in these bins; Output contains sum of the all corresponding bins for the regions of each region set ie for all regions in each region set: first bins summed, second bins summed, etc. Columns of the output should be "bin", "methylProp", and, if coverage was included as input col, "coverage"

Info about how strand of rangeDT affects output: The MIRA profile will be symmetrical if no strand information is given for the regions (produced by averaging the profile with the reverse of the profile), because the orientation of the regions is arbitrary with respect to biological features (like a promoter for instance) that could be oriented directionally (e.g. 5' to 3'). If strand information is given, regions on the minus strand will be flipped before being aggregated with plus strand regions so the MIRA profile will be in 5' to 3' orientation.

Examples

```
data("exampleBSDT") # exampleBSDT
data("exampleRegionSet") # exampleRegionSet
exampleBSDT <- addMethPropCol(exampleBSDT)
aggregateBins <- BSBinAggregate(BSDT = exampleBSDT,
                                rangeDT = exampleRegionSet,
                                binNum = 11, splitFactor = NULL)
```

BSreadBiSeq *Read in files from biseq meth caller*

Description

Parses the x/y format of methylation calls, splitting them into individual columns: "methylCount" column for number of methylated reads for site and "coverage" column for total number of reads covering that site. Input files should have the following columns: "chr", "start", "end", "meth", "rate", "strand".

Usage

```
BSreadBiSeq(files, contrastList = NULL, cores = 4,
  returnAsList = FALSE)
```

Arguments

files	a vector of file paths
contrastList	Generally not needed for MIRA. A list of named character vectors, each with length equal to the number of items in files. These will translate into column names in the final table.
cores	number of processors.
returnAsList	Whether to return the output as a list or as one big data.table.

Details

This can run into memory problems if there are too many files... because of the way parallel lacks long vector support. The solution is to just use a single core; or to pass mc.preschedule = FALSE; This makes it so that each file is processed as a separate job. Much better.

Value

Data from each input file joined together into one big data.table. If returnAsList = TRUE, then input from each file will be in its own data.table in a list.

Examples

```
shortBSDTFile <- system.file("extdata", "shortRRBS.bed", package = "MIRA")
shortBSDT <- BSreadBiSeq(shortBSDTFile)
```

---

bsseqToDataTable *Convert a BSseq object to data.table format for MIRA.*

Description

Converts a BSseq object to a list of data.tables with one data.table per sample. A wrapper of the SummarizedExperimentToDataTable function.

Usage

```
bsseqToDataTable(BSseqObj)
```

Arguments

BSseqObj	An object of class BSseq, can have smoothed or raw methylation data.

Value

MIRAFormatBSDTList A list of data.tables containing the methylation data. One data.table per sample with the column names: 'chr', 'start' (methylation coordinate), 'methylCount' (number of methylated reads), 'coverage' (total number of reads), and 'methylProp' (proportion of methylated reads). The order of the list is the order of samples in the columns of the BSseq object. If sample names are in the BSseq object, then a named list will be returned.

Examples

```
data("exampleBSseqObj")
MIRAFormatBSDTList <- bsseqToDataTable(exampleBSseqObj)
```

---

calcMIRAScore *Score methylation profile based on its shape*

Description

This will take a data.table that has the methylation level in each bin in the MIRA profile and return a single score. For the "logRatio" method, this score summarizes how large the 'dip' in methylation is at the center of that methylation profile. A column for sample ID/name and a column for region set ID/name should be included in the data.table because a separate score will be given for each sample/region set combination.

See 'method' parameter for details on scoring calculations.

Usage

```
calcMIRAScore(binnedDT, shoulderShift = "auto", method = "logRatio",
  usedStrand = FALSE, regionSetIDColName = "featureID",
  sampleIDColName = "sampleName")
```

Arguments

binnedDT	A data.table with columns for: bin ("bin"), methylation level ("methylProp"), region set ID/name (default expected column name is "featureID" but this is configurable via a parameter), sample name (default expected column name is "sampleName" but this is configurable via a parameter). The bin column is not used for calculations since it is assumed by the function that the rows will be in the order of the bins (so the function will work without a bin column although the bin column assists in human readability of the input data.table)
shoulderShift	Used to determine the number of bins away from the center to use as the shoulders. Default value "auto" optimizes the shoulderShift variable for each sample/region set combination to try find the outside edges of the dip. shoulderShift may be manually set as an integer that will be used for all sample/region set combinations. "auto" does not currently work with region sets that include strand info.

Brief description/example of the algorithm for "auto": for concave up MIRA profiles with an odd number of bins, the first potential shoulder will be the 2nd bin from the center (if the center was 0). This is the first bin that could not be included in center methylation value for scoring. From there, the shoulder will be changed to the next bin towards the outside of the profile if the next bin over has a higher methylation value or if the average methylation value of the next two bins are higher than the methylation value of the current shoulder bin. The potential shoulder bin keeps moving outward toward the edges of the MIRA profile until neither of these conditions are met and whatever bin it stops on is used for the shoulder in the MIRA score calculation. For symmetrical MIRA profiles (the general use case), the shoulders picked on both sides of the center will be the same number of bins away from the center bin (symmetrical shoulders).

`method` The scoring method. "logRatio" is the log of the ratio of outer edges to the middle. This ratio is the average of outside values of the dip (shoulders) divided by either the center value if it is lower than the two surrounding values (lower for concave up profiles or higher for concave down profiles) or if it is not lower (higher for concave down profiles), an average of the three middle values. For an even binNum, the middle four values would be averaged with the 1st and 4th being weighted by half (as if there were 3 values). A higher score with "logRatio" corresponds to a deeper dip. "logRatio" is the only scoring method currently but more methods may be added in the future.

`usedStrand` If strand information is included as part of an input region set when aggregating methylation, the MIRA profile will probably not be symmetrical. In this case, the automatic shoulderShift sensing (done when shoulderShift="auto") needs to be done for both sides of the dip instead of just one side so set usedStrand=TRUE if strand was included for a region set. usedStrand=TRUE only has an effect on the function when shoulderShift="auto".

`regionSetIDColName`
A character object. The name of the column that has region set names/identifiers.

`sampleIDColName`
A character object. The name of the column that has sample names/identifiers.

Value

A data.table with a column for region set ID (default name is featureID), sample ID (default name is sampleName), and MIRA score (with name "score"). There will be one row and MIRA score for each sample/region set combination. The MIRA score quantifies the "dip" of the MIRA profile which is an aggregation of methylation over all regions in a region set.

Examples

```
data("exampleBins")
calcMIRAScore(exampleBins)
```

exampleAnnoDT1 *A data.table with annotation for a bisulfite sample.*

Description

Has columns for name and sample type.

Usage

```
data(exampleAnnoDT1)
```

Format

A data.table, data.frame object

exampleAnnoDT2 *Annotation data for the "Applying MIRA to a Biological Question" vignette.*

Description

A data.table with annotation data (sampleName and sampleType columns) for 24 Ewing samples and 24 muscle-related samples (human skeletal myoblasts and other related samples).

Usage

```
data(exampleAnnoDT2)
```

Format

A data.table, data.frame object

exampleBins *A data.table with artificial binned methylation data*

Description

Artificial data for two samples/conditions.

Usage

```
data(exampleBins)
```

Format

A data.table, data.frame object

exampleBSDT *A data.table containing DNA methylation data.*

Description

This object contains reduced representation bisulfite sequencing data from the GM06990 B-lymphoblastoid cell line that was generated by the ENCODE project (GEO accession GSE27584, sample GSM980583). Has columns for position (chr, start), number methylated reads, and number of reads total. The sample name is Gm06990_1. Only a subset of the full data is used in this object and the data was switched from hg19 to hg38.

Usage

```
data(exampleBSDT)
```

Format

A data.table, data.frame object

exampleBSseqObj *A small BSseq object.*

Description

A BSseq object which has the top 20 rows of the BSseq object "BS.chr22" which was included as built-in data in the bsseq package

Usage

```
data(exampleBSseqObj)
```

Format

A BSseq object

exampleRegionSet *A Granges object with Nrf1-binding regions.*

Description

A GRanges object with a ranges corresponding to Nrf1 Chip-seq peaks in the GM12878 B-lymphoblastoid cell line (GEO accession GSE31477, sample GSM935309). Only a subset of the ranges are used in this object and the regions were switched from hg19 to hg38.

Usage

```
data(exampleRegionSet)
```

Format

A GRanges object

MIRA *Methylation-based Inference of Regulatory Activity (MIRA)*

Description

MIRA is a score that infers regulatory activity of genomic elements based on DNA methylation data. It assesses the degree of dip in methylation level surrounding a regulatory site of interest, such as transcription factor binding sites. This package provides functions for aggregating methylation data across region sets, in bins.

Author(s)

Nathan Sheffield

John Lawson

References

http://github.com/databio

plotMIRAProfiles *Plot summary methylation profile*

Description

Plot one or multiple methylation profiles. Displays each region set in a different subplot. If you only want to plot certain region sets, subset with the 'featID' parameter.

Usage

```
plotMIRAProfiles(binnedRegDT, featID = unique(binnedRegDT[, featureID]),
  plotType = "line", colBlindOption = FALSE,
  sampleTypeColName = "sampleType")
```

Arguments

binnedRegDT	A datatable with specific column names containing: bin numbers(binnedRegionDT column), aggregated methylation values (methylProp column), name of the region set (featureID column), case/control column (sampleType column), sample name (sampleName column).
featID	Region set names in a single string or vector of strings.
plotType	Line or jitter (ggplot2).
colBlindOption	If TRUE, function will plot with a color blind friendly palette which could be helpful when plotting multiple colors.
sampleTypeColName	character object. The name of the column that contains sample type or condition information (eg case vs control). Line color will be assigned based on this if it is present and there are more than two unique sample types.

Value

A plot of class "gg"/ "ggplot" that shows MIRA profiles

Examples

```
data("exampleBins", package = "MIRA")
MIRAplot <- plotMIRAProfiles(binnedRegDT = exampleBins)
```

plotMIRAScores *Plot MIRA scores and compare different conditions*

Description

Splits up samples by sample type. Displays each region set in a different subplot. If you only want to plot certain region sets, subset with the ‘featID‘ parameter.

Usage

```
plotMIRAScores(scoreDT, featID = unique(scoreDT[, featureID]),
  colBlindOption = FALSE)
```

Arguments

scoreDT	A datatable with the following columns: score, featureID (names of region sets), ideally include ’sampleType’.
featID	Region set name/names in a single string or vector of strings.
colBlindOption	If TRUE, function will plot with a color blind friendly palette which could be helpful when plotting multiple colors.

Details

Due to the limited number of colors in the palette, a warning will be issued if there are too many (more than 9) region sets (’featureID’s).

Value

a plot of class "gg"/"ggplot" that shows MIRA scores with geom_boxplot and geom_jitter (or geom_violin instead of boxplot if no sampleType column is given).

Examples

```
data(bigBinDT2)
exScores <- calcMIRAScore(bigBinDT2)
# adding annotation
sampleType <- rep(c("Ewing", "Muscle-related"), each = 24)
exScores <- cbind(exScores, sampleType)
exScorePlot <- plotMIRAScores(exScores)
```

`rbindNamedList` *Combine data.tables from a named list into one big data.table with extra column for name.*

Description

Combines data.tables from a list as data.table::rbindlist except an extra column is added which can mark which rows came from which original data.table. The name of this new column can be specified (newColName param) and the values are the names of the input list. This function was originally intended for a list where each data.table corresponds to a separate sample and the list names correspond to the names of the samples. Now all samples can be combined into one data.table but which sample the information came from can be kept.

Usage

```
rbindNamedList(namedList, newColName = "sampleName")
```

Arguments

`namedList`	A list of data.tables. One sample per data.table/list item. Names of the list are the sample names.
`newColName`	The name of the column that will be added to the data.tables

Value

mergedDT A single data.table that combines those in the input list (data.table::rbindlist) but also added a column (default, newColName="sampleName") to each one to keep track of which rows were from which sample

Examples

```
# NOTE: generally sample names should be added after running aggregateMethyl()
# rather than before in order to save memory (since data.table has less rows
# after aggregation)
data(exampleBSseqObj)
multiSampleList <- bsseqToDataTable(exampleBSseqObj)
combinedDT <- rbindNamedList(multiSampleList)
```

`SummarizedExperimentToDataTable` *Make MIRA-compatible data.tables using information from SummarizedExperiment-based classes*

Description

Packages that use SummarizedExperiment-based classes for DNA methylation data which could be converted with this function include "bsseq", "methylPipe", and "BiSeq" packages. Of the methylCountDF, coverageDF, and methylPropDF arguments, either methylPropDF must be given or both methylCountDF and coverageDF must be given. After that, whichever is not given will be calculated from the others. If coverageDF and methylCountDF are both not given, coverage will be assumed to be 1 at each site with a methylProp value. Acceptable formats for the three "DF" parameters include: data.frame, data.table, matrix, and DelayedMatrix classes. If converting a BSseq object, see ?bsseqToDataTable for a convenient wrapper of this function.

Usage

```
SummarizedExperimentToDataTable(coordinates, methylCountDF = NULL,
  coverageDF = NULL, methylPropDF = NULL, sample_names = NULL)
```

Arguments

coordinates	Coordinates for the methylation loci as a GRanges (or GPos) object (in same order as methylCountDF/coverageDF/methylPropDF, whichever is given). The start coordinate should be the coordinate of the cytosine.
methylCountDF	An object of matrix/data.frame/similar format that contains the number of reads with methylated C's for the loci in 'coordinates' argument. It should have one column per sample with rows that correspond to 'coordinates'.
coverageDF	An object of matrix/data.frame/similar format that contains the total number of reads for the loci in 'coordinates' argument. It should have one column per sample with rows that correspond to 'coordinates'.
methylPropDF	An object of matrix/data.frame/similar format that contains the total number of reads for the loci in 'coordinates' argument. It should have one column per sample with rows that correspond to 'coordinates'.
sample_names	A character vector with sample names in the same order as the columns of methylCountDF/coverageDF/methylPropDF samples in the columns

Value

MIRAFormatBSDTList A list of data.tables containing the methylation data. One data.table per sample with the column names: 'chr', 'start' (methylation coordinate), 'methylCount' (number of methylated reads), 'coverage' (total number of reads), and 'methylProp' (proportion of methylated reads). The order of the list is the order of samples in the columns of methylCountDF/coverageDF/methylPropDF. If sample names are explicitly given as input ('sample_names' argument) or can be derived from other arguments to the function then a named list will be returned.

Examples

```
data("exampleBSseqObj")
MIRAFormatBSDTList <- SummarizedExperimentToDataTable(coordinates =
    bsseq::granges(exampleBSseqObj),
    methylCountDF = bsseq::getCoverage(BSseq = exampleBSseqObj, type = "M"),
    coverageDF = bsseq::getCoverage(BSseq = exampleBSseqObj, type = "Cov"),
```

```
methylPropDF = bsseq::getMeth(BSseq = exampleBSseqObj, type = "raw"),
sample_names = bsseq::sampleNames(exampleBSseqObj))
```

Index

∗ **datasets**
bigBinDT1, 4
bigBinDT2, 5
exampleAnnoDT1, 11
exampleAnnoDT2, 11
exampleBins, 11
exampleBSDT, 12
exampleBSseqObj, 12
exampleRegionSet, 13

addMethPropCol, 2
aggregateMethyl, 3

bigBinDT1, 4
bigBinDT2, 5
binRegion, 5
BSBinAggregate, 6
BSreadBiSeq, 7
bsseqToDataTable, 8

calcMIRAScore, 9

exampleAnnoDT1, 11
exampleAnnoDT2, 11
exampleBins, 11
exampleBSDT, 12
exampleBSseqObj, 12
exampleRegionSet, 13

MIRA, 13
MIRA-package *(*MIRA*)*, 13

plotMIRAProfiles, 14
plotMIRAScores, 15

rbindNamedList, 16

SummarizedExperimentToDataTable, 16