Package 'DMRcate'

October 11, 2016

Title Methylation array and sequencing spatial analysis methods

Version 1.8.6

Date 2016-06-07

Author Tim Peters

Maintainer Tim Peters <t.peters@garvan.org.au>

Description

De novo identification and extraction of differentially methylated regions (DMRs) from the human genome using Whole Genome Bisulphite Sequencing (WGBS) and Illumina Infinium Array (450K and EPIC) data. Provides functionality for filtering probes possibly confounded by SNPs and cross-hybridisation. Includes GRanges generation and plotting functions.

Depends R (>= 3.3.0), minfi, DSS, DMRcatedata

Imports limma, GenomicRanges, parallel, methods, graphics, plyr, Gviz, IRanges, stats, utils, S4Vectors

biocViews DifferentialMethylation, GeneExpression, Microarray, MethylationArray, Genetics, DifferentialExpression, GenomeAnnotation, DNAMethylation, OneChannel, TwoChannel, MultipleComparison, QualityControl, TimeCourse

Suggests knitr, RUnit, BiocGenerics, IlluminaHumanMethylation450kanno.ilmn12.hg19, IlluminaHumanMethylationEPICanno.ilm10b2.hg19

License file LICENSE

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

DMRcate-package *DMR calling from bisulphite sequencing and Illumina array data*

Description

De novo identification and extraction of differentially methylated regions (DMRs) in the human genome using array and sequencing data. DMRcate extracts and annotates differentially methylated regions (DMRs) using an array-bias corrected smoothed estimate. Functions are provided for filtering probes possibly confounded by SNPs and cross-hybridisation. Includes GRanges generation and plotting functions.

Author(s)

Tim J. Peters <t.peters@garvan.org.au>

References

Peters T.J., Buckley M.J., Statham, A., Pidsley R., Samaras K., Lord R.V., Clark S.J. and Molloy P.L. *De novo* identification of differentially methylated regions in the human genome. *Epigenetics & Chromatin* 2015, **8**:6, doi:10.1186/1756-8935-8-6

Examples

```
data(dmrcatedata)
myMs <- logit2(myBetas)
myMs.noSNPs <- rmSNPandCH(myMs, dist=2, mafcut=0.05)
patient <- factor(sub("-.*", "", colnames(myMs)))
type <- factor(sub(".*-", "", colnames(myMs)))
design <- model.matrix(~patient + type)
myannotation <- cpg.annotate("array", myMs.noSNPs, analysis.type="differential",
    design=design, coef=39)
dmrcoutput <- dmrcate(myannotation, lambda=1000, C=2)
results.ranges <- extractRanges(dmrcoutput, genome = "hg19")
groups <- c(Tumour="magenta", Normal="forestgreen")
cols <- groups[as.character(type)]
samps <- c(1:6, 38+(1:6))
DMR.plot(ranges=results.ranges, dmr=1, CpGs=myBetas, phen.col=cols, genome="hg19", samps=samps)
```

cpg.annotate *Annotate CpGs with their chromosome position and test statistic*

Description

Either: - Annotate a matrix of *M*-values (logit transform of beta) representing 450K or EPIC data with probe weights (depending on `analysis.type`) and chromosomal position, or - Standardise this information from `DSS:::DMLtest()` to the same data format.

Usage

```
cpg.annotate(datatype = c("array", "sequencing"),
             object,
           annotation=c(array = "IlluminaHumanMethylation450k", annotation = "ilmn12.hg19"),
             analysis.type = c("differential", "variability"),
             design,
             contrasts = FALSE,
             cont.matrix = NULL,
             fdr = 0.05,
             coef,
             ...)
```

Arguments

`datatype`	Character string representing the type of data being analysed.
`object`	Either: - A matrix of *M*-values, with unique Illumina probe IDs as rownames and unique sample IDs as column names or, - Output from `DSS:::DMLtest()`.
`annotation`	A vector describing the type of annotation to affix to `object`. Identical context to `minfi`, i.e. `annotation <- annotation(minfiobject)` where `minfiobject` is a `[Genomic](Methyl|Ratio)Set)`. Argument for 450K arrays: `c(array = "IlluminaHumanMethylation450k", annotation = "ilmn12.hg19")`. Argument for EPIC arrays: `c(array = "IlluminaHumanMethylationEPIC", annotation = "ilm10b2.hg19")`. An error will be thrown if you attempt one on an `object` with rownames on the other, due to non-overlapping probes on both platforms. Only applicable when `datatype="array"`.
`analysis.type`	`"differential"` for `dmrcate()` to return DMRs and `"variability"` to return VMRs. Only applicable when `datatype="array"`.
`design`	Study design matrix. Identical context to differential analysis pipeline in `limma`. Must have an intercept if `contrasts=FALSE`. Applies only when `analysis.type="differential"`. Only applicable when `datatype="array"`.

contrasts	Logical denoting whether a limma-style contrast matrix is specified. Only applicable when datatype="array".
cont.matrix	Limma-style contrast matrix for explicit contrasting. For each call to cpg.annotate, only one contrast will be fit. Only applicable when datatype="array".
fdr	FDR cutoff (Benjamini-Hochberg) for which CpG sites are individually called as significant. Used only to determine effect size, and not for downstream thresholding.
coef	The column index in design corresponding to the phenotype comparison. Corresponds to the comparison of interest in design when contrasts=FALSE, otherwise must be a column name in cont.matrix. Applies only when analysis.type="differential" and when datatype="array".
...	Extra arguments passed to the limma function lmFit(). Applies only when analysis.type="differential" and when datatype="array".

Value

An object of class "annot", for passing to dmrcate, containing the vectors:

- ID: Illumina probe ID or row number
- stat: *t*-statistic or Wald statistics between phenotypes for each CpG
- CHR: Chromosome which the CpG maps to
- pos: Genomic coordinate (on CHR) that the CpG maps to
- betafc: The beta fold change according to the given design
- indfdr: Individually-derived FDRs for each CpG
- is.sig: Logical denoting either significance from fdr (analysis.type="differential") or top ventile of variable probes (analysis.type="variability")

Author(s)

Tim J. Peters <t.peters@garvan.org.au>

References

Smyth, G. K. (2005). Limma: linear models for microarray data. In: *Bioinformatics and Computational Biology Solutions using R and Bioconductor*, R. Gentleman, V. Carey, S. Dudoit, R. Irizarry, W. Huber (eds.), Springer, New York, pages 397-420.

Feng, H., Conneely, K. N., & Wu, H. (2014). A Bayesian hierarchical model to detect differentially methylated loci from single nucleotide resolution sequencing data. *Nucleic Acids Research*, **42**(8), e69.

Peters T.J., Buckley M.J., Statham, A., Pidsley R., Samaras K., Lord R.V., Clark S.J. and Molloy P.L. *De novo* identification of differentially methylated regions in the human genome. *Epigenetics & Chromatin* 2015, **8**:6, doi:10.1186/1756-8935-8-6.

Examples

```
## Not run:
data(dmrcatedata)
myMs <- logit2(myBetas)
myMs.noSNPs <- rmSNPandCH(myMs, dist=2, mafcut=0.05)
patient <- factor(sub("-.*", "", colnames(myMs)))
type <- factor(sub(".*-", "", colnames(myMs)))
design <- model.matrix(~patient + type)
myannotation <- cpg.annotate("array", myMs.noSNPs, analysis.type="differential",
    design=design, coef=39)

## End(Not run)
```

`DMR.plot` *Plotting DMRs*

Description

Plots an individual DMR (in context of possibly other DMRs) as found by `dmrcate`. Heatmap and mean methylation plots are shown as well as genomic coordinates and proximal coding regions.

Usage

```
DMR.plot(ranges, dmr, CpGs, phen.col, genome = c("hg19", "hg38", "mm10"),
       array.annotation = c(array = "IlluminaHumanMethylation450k", annotation = "ilmn12.hg19"),
         samps = NULL, ...)
```

Arguments

`ranges`	A GRanges object (ostensibly created by `extractRanges()` describing DMR coordinates.
`dmr`	Index of `ranges` (one integer only) indicating which DMR to be plotted.
`CpGs`	Either: - A matrix of beta values for plotting, with unique Illumina probe IDs as rownames. - A GRanges object describing individual CpGs to be plotted, containing methylated reads and total coverage for each sample. Please see the worked example in the vignette for the correct structure of this object.
`phen.col`	Vector of colors denoting phenotypes of *all* samples described in `CpGs`. See vignette for worked example.
`genome`	Reference genome for annotating DMRs. Can be one of `"hg19"`, `"hg38"` or `"mm10"`
`array.annotation`	A vector describing the type of annotation from which plots are derived. Identical context to `minfi`, i.e. `annotation <- annotation(minfiobject)` where `minfiobject` is a `[Genomic](Methyl|Ratio)Set`). Argument for 450K arrays: `c(array = "IlluminaHumanMethylation450k", annotation = "ilmn12.hg19")`.

	Argument for EPIC arrays: `c(array = "IlluminaHumanMethylationEPIC", annotation = "ilm10b2.hg19")`. An error will be thrown if you attempt one on `CpGs` with rownames on the other, due to non-overlapping probes on both platforms. Only applicable when `datatype="array"`.
`samps`	Vector of samples to be plotted, corresponding to indices of `phen.col`. Default is all samples plotted.
`...`	Extra arguments passed to `Gviz:::plotTracks()`.

Value

A plot to the current device.

Author(s)

Aaron Statham <a.statham@garvan.org.au>, Tim J. Peters <t.peters@garvan.org.au>

Examples

```
## Not run:
data(dmrcatedata)
myMs <- logit2(myBetas)
myMs.noSNPs <- rmSNPandCH(myMs, dist=2, mafcut=0.05)
patient <- factor(sub("-.*", "", colnames(myMs)))
type <- factor(sub(".*-", "", colnames(myMs)))
design <- model.matrix(~patient + type)
myannotation <- cpg.annotate("array", myMs.noSNPs, analysis.type="differential",
    design=design, coef=39)
dmrcoutput <- dmrcate(myannotation, lambda=1000, C=2)
results.ranges <- extractRanges(dmrcoutput, genome = "hg19")
groups <- c(Tumour="magenta", Normal="forestgreen")
cols <- groups[as.character(type)]
samps <- c(1:6, 38+(1:6))
DMR.plot(ranges=results.ranges, dmr=1, CpGs=myBetas, phen.col=cols, genome="hg19", samps=samps)

## End(Not run)
```

`dmrcate` *DMR identification*

Description

The main function of this package. Computes a kernel estimate against a null comparison to identify significantly differentially (or variable) methylated regions.

Usage

```
dmrcate(object,
           lambda = 1000,
           C=NULL,
           p.adjust.method = "BH",
           pcutoff = "fdr",
           consec = FALSE,
           conseclambda = 10,
           betacutoff = NULL,
           min.cpgs = 2,
           mc.cores = 1
           )
```

Arguments

object	A class of type "annot", created from cpg.annotate.
lambda	Gaussian kernel bandwidth for smoothed-function estimation. Also informs DMR bookend definition; gaps >= lambda between significant CpG sites will be in separate DMRs. Support is truncated at 5*lambda. Default is 1000 nucleotides. See details for further info.
C	Scaling factor for bandwidth. Gaussian kernel is calculated where lambda/C = sigma. Empirical testing shows that, for 450k data when lambda=1000, near-optimal prediction of sequencing-derived DMRs is obtained when C is approximately 2, i.e. 1 standard deviation of Gaussian kernel = 500 base pairs. Should be a lot larger for sequencing data - suggest C=50. Cannot be < 0.2.
p.adjust.method	Method for *p*-value adjustment from the significance test. Default is "BH" (Benjamini-Hochberg).
pcutoff	*p*-value cutoff to determine DMRs. Default is automatically determined by the number of significant CpGs returned by either limma or DSS for that contrast, but can be set manually with a numeric value.
consec	Use DMRcate in consecutive mode. Treats CpG sites as equally spaced.
conseclambda	Bandwidth in *CpGs* (rather than nucleotides) to use when consec=TRUE. When specified the variable lambda simply becomes the minumum distance separating DMRs.
betacutoff	Optional filter; removes any region from the results that does not have at least one CpG site with a beta fold change exceeding this value.
min.cpgs	Minimum number of consecutive CpGs constituting a DMR.
mc.cores	When > 1, the processor will attempt to run the kernel smoothing in parallel, 1 chromosome per core. Use with discretion. Default recommended for laptop use. Please use detectCores() and htop in your terminal to check your resource ceiling before increasing the default.

Details

The values of lambda and C should be chosen with care. For array data, we currently recommend that half a kilobase represent 1 standard deviation of support (lambda=1000 and C=2), and 20bp

(C=50) for WGBS data. If lambda is too small or C too large then the kernel estimator will not have enough support to significantly differentiate the weighted estimate from the null distribution. If lambda is too large then dmrcate will report very long DMRs spanning multiple gene loci, and the large amount of support will likely give Type I errors. If you are concerned about Type I errors we recommend using the default value of pcutoff, although this will return no DMRs if no DM CpGs are returned by limma/DSS either.

Value

A list containing 2 data frames (input and results) and a numeric value (cutoff). input contains the contents of the annot object, plus calculated *p*-values:

- ID: As per annotation object input
- stat: As per annotation object input
- CHR: As per annotation object input
- pos: As per annotation object input
- betafc: As per annotation object input
- raw: Raw *p*-values from the significance test
- fdr: Adjusted *p*-values from the significance test
- step.dmr: Vector denoting the start of a new DMR (TRUE), constitutive of a DMR, but not the start (FALSE), or non-DMR (NA).

results contains an annotated data.frame of significant regions, ranked by Stouffer:

- coord: Coordinates of the significant region in hg19. IGV- and UCSC-friendly.
- no.cpgs: Number of CpG sites constituting the significant region. Tie-breaker when sorting by Stouffer.
- minfdr: Minimum adjusted *p*-value from the CpGs constituting the significant region.
- Stouffer: Stouffer transformation of the group of limma- or DSS-derived fdrs for individual CpG sites as DMR constituents.
- maxbetafc: Maximum absolute beta fold change within the region
- meanbetafc: Mean beta fold change within the region.

cutoff is the signficance *p*-value cutoff provided in the call to dmrcate.

Author(s)

Tim J. Peters <t.peters@garvan.org.au>, Mike J. Buckley <Mike.Buckley@csiro.au>, Tim Triche Jr. <tim.triche@usc.edu>

References

Peters T.J., Buckley M.J., Statham, A., Pidsley R., Samaras K., Lord R.V., Clark S.J. and Molloy P.L. *De novo* identification of differentially methylated regions in the human genome. *Epigenetics & Chromatin* 2015, **8**:6, doi:10.1186/1756-8935-8-6

Wand, M.P. & Jones, M.C. (1995) *Kernel Smoothing*. Chapman & Hall.

Duong T. (2013) Local significant differences from nonparametric two-sample tests. *Journal of Nonparametric Statistics*. 2013 **25**(3), 635-645.

Examples

```
## Not run:
data(dmrcatedata)
myMs <- logit2(myBetas)
myMs.noSNPs <- rmSNPandCH(myMs, dist=2, mafcut=0.05)
patient <- factor(sub("-.*", "", colnames(myMs)))
type <- factor(sub(".*-", "", colnames(myMs)))
design <- model.matrix(~patient + type)
myannotation <- cpg.annotate(myMs.noSNPs, analysis.type="differential",
    design=design, coef=39)
dmrcoutput <- dmrcate(myannotation, lambda=1000)

## End(Not run)
```

`extractRanges` *Create GRanges object from* `dmrcate` *output.*

Description

Takes a `dmrcate.output` object and produces the corresponding GRanges object.

Usage

```
extractRanges(dmrcoutput, genome = c("hg19", "hg38", "mm10"))
```

Arguments

`dmrcoutput`	An object of class `dmrcate.output`.
`genome`	Reference genome for annotating DMRs with promoter overlaps. Can be one of `"hg19"`, `"hg38"` or `"mm10"`

Value

A GRanges object.

Author(s)

Tim Triche Jr. <tim.triche@usc.edu>, Tim Peters <t.peters@garvan.org.au>

Examples

```
## Not run:
data(dmrcatedata)
myMs <- logit2(myBetas)
myMs.noSNPs <- rmSNPandCH(myMs, dist=2, mafcut=0.05)
patient <- factor(sub("-.*", "", colnames(myMs)))
type <- factor(sub(".*-", "", colnames(myMs)))
design <- model.matrix(~patient + type)
myannotation <- cpg.annotate("array", myMs.noSNPs, analysis.type="differential",
```

```
    design=design, coef=39)
dmrcoutput <- dmrcate(myannotation, lambda=1000, C=2)
results.ranges <- extractRanges(dmrcoutput, genome = "hg19")

## End(Not run)
```

---

rmSNPandCH *Filter probes*

Description

Filters a matrix of M-values (or beta values) by distance to SNP. Also (optionally) removes cross-hybridising probes and sex-chromosome probes.

Usage

```
rmSNPandCH(object, dist = 2, mafcut = 0.05, and = TRUE, rmcrosshyb = TRUE, rmXY=FALSE)
```

Arguments

object	A matrix of M-values or beta values, with unique Illumina probe IDs as rownames.
dist	Maximum distance (from CpG to SNP) of probes to be filtered out. See details for when Illumina occasionally lists a CpG-to-SNP distance as being < 0.
mafcut	Minimum minor allele frequency of probes to be filtered out.
and	If TRUE, the probe must have at least 1 SNP binding to it that satisfies **both** requirements in dist and mafcut for it to be filtered out. If FALSE, it will be filtered out if either requirement is satisfied. Default is TRUE.
rmcrosshyb	If TRUE, filters out probes found by Chen et al. (2013) to be cross-reactive with areas of the genome not at the site of interest. Many of these sites are on the X-chromosome, leading to potential confounding if the sample group is a mix of males and females. There are 30,969 probes in total in this list. Default is TRUE.
rmXY	If TRUE, filters out probe hybridising to sex chromosomes. Or-operator applies when combined with other 2 filters.

Details

Probes in -1:dist will be filtered out for any integer specification of dist. When a probe is listed as being "-1" nucleotides from a SNP (7 in total of the 153,113), that SNP is immediately adjacent to the end of the probe, and is likely to confound the measurement, in addition to those listed as 0, 1 or 2 nucleotides away. See vignette for further details.

Value

A matrix, attenuated from object, with rows corresponding to probes matching user input filtered out.

Author(s)

Tim J. Peters <t.peters@garvan.org.au>

References

Chen YA, Lemire M, Choufani S, Butcher DT, Grafodatskaya D, Zanke BW, Gallinger S, Hudson TJ, Weksberg R. Discovery of cross-reactive probes and polymorphic CpGs in the Illumina Infinium HumanMethylation450 microarray. *Epigenetics*. 2013 Jan 11;8(2).

http://supportres.illumina.com/documents/myillumina/88bab663-307c-444a-848e-0ed6c338ee4d/humanmethylation450_15017482_v.1.2.snpupdate.table.v3.txt

Examples

```
## Not run:
data(dmrcatedata)
myMs <- logit2(myBetas)
myMs.noSNPs <- rmSNPandCH(myMs, dist=2, mafcut=0.05)

## End(Not run)
```

Index