Package 'SOMNiBUS'

October 14, 2021

Title Smooth modeling of bisulfite sequencing

Version 1.0.0

Description This package aims to analyse count-based methylation data on predefined genomic regions, such as those obtained by targeted sequencing, and thus to identify differentially methylated regions (DMRs) that are associated with phenotypes or traits. The method is built a rich flexible model that allows for the effects, on the methylation levels, of multiple covariates to vary smoothly along genomic regions. At the same time, this method also allows for sequencing errors and can adjust for variability in cell type mixture.

License MIT + file LICENSE

URL https://github.com/kaiqiong/SOMNiBUS

BugReports https://github.com/kaiqiong/SOMNiBUS/issues

Depends R (>= 4.1.0)

Imports graphics, Matrix, mgcv, stats, VGAM

Suggests BiocStyle, covr, devtools, dplyr, knitr, magick, rmarkdown, testthat

VignetteBuilder knitr

biocViews DNAMethylation, Regression, Epigenetics, DifferentialMethylation, Sequencing, FunctionalPrediction

Encoding UTF-8

Language en-US

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

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

git_branch RELEASE_3_13

git_last_commit c53c179

git_last_commit_date 2021-05-19

Date/Publication 2021-10-14

Author Kaiqiong Zhao [aut], Kathleen Klein [cre]

Maintainer Kathleen Klein <kathleen.klein@mail.mcgill.ca>

R topics documented:

`binomRegMethModel` *A smoothed-EM algorithm to estimate covariate effects and test regional association in Bisulfite Sequencing-derived methylation data*

Description

This function fits a (dispersion-adjusted) binomial regression model to regional methylation data, and reports the estimated smooth covariate effects and regional p-values for the test of DMRs (differentially methylation regions). Over or under dispersion across loci is accounted for in the model by the combination of a multiplicative dispersion parameter (or scale parameter) and a sample-specific random effect.

This method can deal with outcomes, i.e. the number of methylated reads in a region, that are contaminated by known false methylation calling rate (`p0`) and false non-methylation calling rate (`1-p1`).

The covariate effects are assumed to smoothly vary across genomic regions. In order to estimate them, the algorithm first represents the functional paramters by a linear combination of a set of restricted cubic splines (with dimention `n.k`), and a smoothness penalization term which depends on the smoothing parameters `lambdas` is also added to control smoothness. The estimation is performed by an iterated EM algorithm. Each M step constitutes an outer Newton's iteration to estimate smoothing parameters `lambdas` and an inner P-IRLS iteration to estimate spline coefficients `alpha` for the covariate effects. Currently, the computation in the M step depends the implementation of `gam()` in package `mgcv`.

Usage

```
binomRegMethModel(
  data,
  n.k,
  p0 = 0.003,
  p1 = 0.9,
  Quasi = TRUE,
  epsilon = 10^(-6),
  epsilon.lambda = 10^(-3),
  maxStep = 200,
  detail = FALSE,
  binom.link = "logit",
```

```
  method = "REML",
  covs = NULL,
  RanEff = TRUE,
  reml.scale = FALSE,
  scale = -2
)
```

Arguments

data	a data frame with rows as individual CpGs appeared in all the samples. The first 4 columns should contain the information of Meth_Counts (methylated counts), Total_Counts (read depths), Position (Genomic position for the CpG site) and ID (sample ID). The covariate information, such as disease status or cell type composition, are listed in column 5 and onwards.
n.k	a vector of basis dimensions for the intercept and individual covariates. n.k specifies an upper limit of the degrees of each functional parameters.
p0	the probability of observing a methylated read when the underlying true status is unmethylated. p0 is the rate of false methylation calls, i.e. false positive rate.
p1	the probability of observing a methylated read when the underlying true status is methylated. 1-p1 is the rate of false non-methylation calls, i.e. false negative rate.
Quasi	whether a Quasi-likelihood estimation approach will be used; in other words, whether a multiplicative dispersion is added in the model or not.
epsilon	numeric; stopping criterion for the closeness of estimates of spline coefficients from two consecutive iterations.
epsilon.lambda	numeric; stopping criterion for the closeness of estimates of smoothing parameter lambda from two consecutive iterations.
maxStep	the algorithm will step if the iteration steps exceed maxStep
detail	indicate whether print the number of iterations
binom.link	the link function used in the binomial regression model; the default is the logit link
method	the method used to estimate the smoothing parameters. The default is the 'REML' method which is generally better than prediction based criterion GCV.cp
covs	a vector of covariate names. The covariates with names in covs will be included in the model and their covariate effects will be estimated. The default is to fit all covariates in data
RanEff	whether sample-level random effects are added or not
reml.scale	whether a REML-based scale (dispersion) estimator is used. The default is Fletcher-based estimator
scale	nagative values mean scale paramter should be estimated; if a positive value is provided, a fixed scale will be used.

Value

This function return a `list` including objects:

- `est`: estimates of the spline basis coefficients `alpha`
- `lambda`: estimates of the smoothing parameters for each functional paramters
- `est.pi`: predicted methylation levels for each row in the input `data`
- `ite.points`: estimates of `est`, `lambda` at each EM iteration
- `cov1`: estimated variance-covariance matrix of the basis coefficients `alphas`
- `reg.out`: regional testing output obtained using Fletcher-based dispersion estimate; an additional 'ID' row would appear if RanEff is TRUE
- `reg.out.reml.scale`:regional testing output obtained sing REML-based dispersion estimate;
- `reg.out.gam`:regional testing output obtained using (Fletcher-based) dispersion estimate from mgcv package;
- `phi_fletcher`: Fletcher-based estimate of the (multiplicative) dispersion parameter
- `phi_reml`: REML-based estimate of the (multiplicative) dispersion parameter
- `phi_gam`: Estimated dispersion parameter reported by mgcv
- `SE.out`: a matrix of the estimated pointwise Standard Errors (SE); number of rows are the number of unique CpG sites in the input data and the number of columns equal to the total number of covariates fitted in the model (the first one is the intercept)
- `SE.out.REML.scale`: a matrix of the estimated pointwise Standard Errors (SE); the SE calculated from the REML-based dispersion estimates
- `uni.pos`: the genomic postions for each row of CpG sites in the matrix `SE.out`
- `Beta.out`: a matrix of the estimated covariate effects beta(t), here t denots the genomic positions.
- `ncovs`: number of functional paramters in the model (including the intercept)
- `sigma00`: estimated variance for the random effect if RanEff is TRUE; NA if RanEff is FALSE

Author(s)

Kaiqiong Zhao

See Also

gam

Examples

```
#------------------------------------------------------------#
data(RAdat)
head(RAdat)
RAdat.f <- na.omit(RAdat[RAdat$Total_Counts != 0, ])
out <- binomRegMethModel(
  data=RAdat.f, n.k=rep(5, 3), p0=0.003307034, p1=0.9,
  epsilon=10^(-6), epsilon.lambda=10^(-3), maxStep=200,
  detail=FALSE
)
```

`binomRegMethModelPlot` *Plot the smooth covariate effect*

Description

This function accepts an output object from function `binomRegMethModel` and print out a plot of the estimated covariate effect across the region for each test covariate.

Usage

```
binomRegMethModelPlot(BEM.obj, mfrow = NULL, same.range = FALSE)
```

Arguments

`BEM.obj`	an output object from function `binomRegMethModel`
`mfrow`	the plot parameters to specify the layout of each plot
`same.range`	specify whether the plots should be in the same vertical scale

Value

This function prints out a plot of smooth covariate effects and its pointwise confidence intervals

Author(s)

Kaiqiong Zhao

Examples

```
#-----------------------------------------------------------#
head(RAdat)
RAdat.f <- na.omit(RAdat[RAdat$Total_Counts != 0, ])
out <- binomRegMethModel(
  data=RAdat.f, n.k=rep(5, 3), p0=0.003307034, p1=0.9,
  epsilon=10^(-6), epsilon.lambda=10^(-3), maxStep=200, detail=FALSE,
  Quasi = FALSE, RanEff = FALSE
)
binomRegMethModelPlot(out, same.range=FALSE)
```

binomRegMethModelPred *A smoothed-EM algorithm to estimate covariate effects and test regional association in Bisulfite Sequencing-derived methylation data*

Description

This function returns the predicted methylation levels

Usage

```
binomRegMethModelPred(BEM.obj, newdata = NULL, type = "proportion")
```

Arguments

BEM.obj	an output from the function binomRegMethModel
newdata	the data set whose predictions are calculated; with columns 'Position', and covariate names that can be matched to the BEM.obj
type	return the predicted methylation proportion or the predicted response (in logit or other binom.link scale)

Value

This function returns the predicted methylation levels

Author(s)

Kaiqiong Zhao

Examples

```
#-------------------------------------------------------------#
head(RAdat)
RAdat.f <- na.omit(RAdat[RAdat$Total_Counts != 0, ])
out <- binomRegMethModel(
  data=RAdat.f, n.k=rep(5, 3), p0=0.003307034, p1=0.9,
  epsilon=10^(-6), epsilon.lambda=10^(-3), maxStep=200, detail=FALSE,
  Quasi = FALSE, RanEff = FALSE
)
binomRegMethModelPred(out)
```

`binomRegMethModelSim` *Simulate Bisulfite sequencing data from specified smooth covariate effects*

Description

Simulate Bisulfite sequencing data from a Generalized Additive Model with functional parameters varying with the genomic position. Both the true methylated counts and observed methylated counts are generated, given the error/conversion rate parameters `p0` and `p1`. In addition, the true methylated counts can be simulated from a binomial or a dispersed binomial distribution (Beta-binomial distribution).

Usage

```
binomRegMethModelSim(
  n,
  posit,
  theta.0,
  beta,
  phi,
  random.eff = FALSE,
  mu.e = 0,
  sigma.ee = 1,
  p0 = 0.003,
  p1 = 0.9,
  X,
  Z,
  binom.link = "logit"
)
```

Arguments

`n`	sample size
`posit`	genomic position; a numeric vector of size `p` (the number of CpG sites in the considered region).
`theta.0`	a functional parameter for the intercept of the GAMM model; a numeric vector of size `p`.
`beta`	a functional parameter for the slope of cell type composition. a numeric vector of size `p`
`phi`	multiplicative dispersion parameter for each loci in a region. a vector of length `p`. The dispersed-Binomial counts are simulated from beta-binomial distribution, so each element of phi has to be greater than 1.
`random.eff`	indicate whether adding the subject-specific random effect term `e`.
`mu.e`	the mean of the random effect; a single number.
`sigma.ee`	variance of the random effect; a single positive number.

p0	the probability of observing a methylated read when the underlying true status is unmethylated. p0 is the rate of false methylation calls, i.e. false positive rate.
p1	the probability of observing a methylated read when the underlying true status is methylated. 1-p1 is the rate of false non-methylation calls, i.e. false negative rate.
X	the matrix of the read coverage for each CpG in each sample; a matrix of n rows and p columns
Z	numeric matrix with p columns and n rows storing the covariate information
binom.link	the link function used for simulation

Value

The function returns a list of following objects

- S the true methylation counts; a numeric matrix of n rows and p columns
- Y the observed methylation counts; a numeric matrix of n rows and p columns
- theta the methylation parameter (after the logit transformation); a numeric matrix of n rows and p columns
- pi the true methylation proportions used to simulate the data; a numeric matrix of n rows and p columns

Author(s)

Kaiqiong Zhao

Examples

```
#------------------------------------------------------------#
data(RAdat)
RAdat.f <- na.omit(RAdat[RAdat$Total_Counts != 0, ])
out <- binomRegMethModel(
   data=RAdat.f, n.k=rep(5, 3), p0=0, p1=1,
   epsilon=10^(-6), epsilon.lambda=10^(-3), maxStep=200,
   detail=FALSE, RanEff = FALSE
)
Z = as.matrix(RAdat.f[match(unique(RAdat.f$ID), RAdat.f$ID),
c('T_cell', 'RA')])
set.seed(123)
X = matrix(sample(80, nrow(Z)*length(out$uni.pos), replace = TRUE),
nrow = nrow(Z), ncol = length(out$uni.pos))+10
simdat = binomRegMethModelSim(n=nrow(Z), posit= out$uni.pos,
theta.0=out$Beta.out[,1], beta= out$Beta.out[,-1], random.eff=FALSE,
mu.e=0,sigma.ee=1, p0=0.003, p1=0.9,X=X , Z=Z, binom.link='logit',
phi = rep(1, length(out$uni.pos)))
```

RAdat — *Methylation data from a rheumatoid arthritis study*

Description

A dataset containing methylation levels on one targeted region on chromosome 4 near gene BANK1 from cases with rheumatoid arthritis (RA) and controls

Usage

```
RAdat
```

Format

A data frame of 5289 rows and 6 columns. Each row represents a CpG site for a sample. Columns include in order

Meth_Counts Number of methylated reads

Total_Counts Total number of reads; read-depth

Position Genomic position (in bp) for the CpG site

ID indicates which sample the CpG site belongs to

T_cell whether a sample is from T cell or monocyte

RA whether a sample is an RA patient or control

Details

This example data include methylation levels of cell type separated blood samples of 22 rheumatoid arthritis (RA) patients and 21 healthy individuals. In the data set, 123 CpG sites are measured and there are 25 samples from circulating T cells and 18 samples from monocytes.

Source

Dr. Marie Hudson (McGill University)

RAdat2 — *A simulated methylation dataset based on a real data.*

Description

This example data include methylation levels on a region with 208 CpGs for 116 blood samples.

Usage

```
RAdat2
```

Format

A data frame of 6064 rows and 13 columns. Each row represents a CpG site for a sample. Columns include in order

Meth_Counts Number of methylated reads

Total_Counts Total number of reads; read-depth

Position Genomic position (in bp) for the CpG site

ID indicates which sample the CpG site belongs to

ACPA4 binary indicator for a biomarker anti-citrullinated protein antibody

Age Age

Sex 2-female; 1-male

Smoking 1-current or ex-smoker; 0-non-smoker

Smoking_NA 1-Smoking info is NA; 0-Smoking info is available

PC1 PC1 for the cell type proportions

PC2 PC2 for the cell type proportions

PC3 PC3 for the cell type proportions

PC4 PC4 for the cell type proportions

Source

simulation is based a real data set provided by PI Dr. Sasha Bernatsky (McGill University)

Index