Package 'MADSEQ'

April 12, 2018

Type Package

Title Mosaic Aneuploidy Detection and Quantification using Massive Parallel Sequencing Data

Version 1.4.1

Date 2017-04-29

Author Yu Kong, Adam Auton, John Murray Greally

Maintainer Yu Kong <yu.kong@phd.einstein.yu.edu>

Description The MADSEQ package provides a group of hierarchical Bayeisan models for the detection of mosaic aneuploidy, the inference of the type of aneuploidy and also for the quantification of the fraction of aneuploid cells in the sample.

License GPL(>=2)

Depends R(>= 3.4), rjags(>= 4-6),

Suggests knitr

VignetteBuilder knitr

LazyData True

Imports VGAM, coda, BSgenome, BSgenome.Hsapiens.UCSC.hg19, S4Vectors, methods, preprocessCore, GenomicAlignments, Rsamtools, Biostrings, GenomicRanges, IRanges, VariantAnnotation, SummarizedExperiment, GenomeInfoDb, rtracklayer, graphics, stats, grDevices, utils, zlibbioc

biocViews GenomicVariation, SomaticMutation, VariantDetection, Bayesian, CopyNumberVariation, Sequencing, Coverage

URL https://github.com/ykong2/MADSEQ

BugReports https://github.com/ykong2/MADSEQ/issues

RoxygenNote 6.0.1

NeedsCompilation no

R topics documented:

`MADSEQ-package` *Mosaic Aneuploidy Detection using Massive Parallel Sequencing Data (MADSEQ)*

Description

The MADSEQ package provides a group of hierarchical Bayesian models for the detection and quantification of mosaic aneuploidy using massive parallele sequencing data.

Details

MADSEQ is a group of hierarchical Bayesian models used for the detection and quantification of mosaic aneuploidy. The package takes bam file and vcf file as input. There are functions for the calculation of the coverage for the sequencing data; the normalization of the coverage to correct GC bias; the detection and quantification of mosaic aneuploidy and the inference of the type of aneuploidy (monosomy, mitotic trisomy, meiotic trisomy, loss of heterozygosity). The package also includes function to visualize the estimated distribution for detected mosaic aneuploidy. To fully understand how to use the MADSEQ package, please check the documentation. The manual explains what data do you need, and how to process the data to be ready for the model, what steps to follow and how to interpret the output from our model.

Author(s)

Yu Kong

References

Martyn Plummer (2016). rjags: Bayesian Graphical Models using MCMC. R package version 4-6. http://CRAN.R-project.org/package=rjags
C. Alkan, J. Kidd, T. Marques-Bonet et al (2009). Personalized copy number and segmental duplication maps using next-generation sequencing. Nature Genetics, 41(10):1061-7.

aneuploidy_chr18 *An S4 class MadSeq object*

Description

An MadSeq object returned by the function runMadSeq, the object contains the posterior distribution and deltaBIC value of a trisomy chromosome 18

Usage

```
aneuploidy_chr18
```

Format

An MadSeq object

Value

MadSeq object returned from runMadSeq function, mitotic trisomy has been detected for the chromosome18

Examples

```
## to load the data
data(aneuploidy_chr18)
## check statistics of the data
summary(aneuploidy_chr18)
```

deltaBIC *Accessing delta BIC of MadSeq object*

Description

An S4 method to access the delta BIC values of MadSeq object

Usage

```
deltaBIC(object)

## S4 method for signature 'MadSeq'
deltaBIC(object)
```

Arguments

object	A MadSeq object returned by runMadSeq function

Value

A numeric vector containing deltaBIC values between selected model and other models

Author(s)

Yu Kong

See Also

MadSeq, runMadSeq

Examples

```
## load the example MadSeq object come with the package
data("aneuploidy_chr18")

## access deltaBIC
deltaBIC(aneuploidy_chr18)
```

MadSeq-class *The MadSeq class*

Description

An S4 class contains estimated result returned from runMadSeq function

Slots

`posterior` A `matrix` contains the posterior distribution from the selected model

`deltaBIC` A `numeric vector` contains the deltaBIC value between selected model and other models. The deltaBIC between models indicate the confidence level that selected model against other models: deltaBIC ~ [0,2]: Not worth more than a bare mention deltaBIC ~ [2,6]: Positive deltaBIC ~ [6,10]: Strong deltaBIC >10: Very Strong

Accessors

In the code below, `x` is a `MadSeq` object.

`posterior(x)`: Get the matrix containing posterior distribution of selected model.

`deltaBIC(x)`: Get the deltaBIC between selected model and other models

Summary

In the code below, `x` is a `MadSeq` object.

`summary(x)`: summarize the posterior distribution

MadSeq Methods

In the code below, `x` is a `MadSeq` object.

`plotMadSeq(x)`: Plot the posterior distribution of all parameters in selected model.

`plotFraction(x)`: Plot the estimated distribution of the fraction of aneuploid sample.

`plotMixture(x)`: Plot the distribution of AAF estimated from the selected model.

Author(s)

Yu Kong

See Also

runMadSeq, plotMadSeq

---

normalizeCoverage *correct coverage bias due to GC content*

Description

function to normalize coverage by GC content and quantile normalization

Usage

```
normalizeCoverage(object, ..., control = NULL, writeToFile = TRUE,
  destination = NULL, plot = TRUE)
```

Arguments

object	A GRanges object returned from prepareCoverageGC function.
...	additional GRanges object to pass. **Note1:** If there is only one Granges object given, then coverage will be corrected by GC content. If there are more than one GRanges object from multiple samples are given, the function will first quantile normalize coverage across samples, then correct coverage by GC content in each sample. **Note2:** If more than one GRanges object provided, make sure they are different samples sequenced by the same protocol, which means the targeted region is the same **Note3:** If your input samples contain female and male, we suggest you separate them to get a more accurate normalization.
control	A GRanges object returned from prepareCoverageGC function. **Default value: NULL**. If you have a control normal sample, then put it here
writeToFile	Boolean Default: TRUE. If TRUE, normalized coverage table for each sample provided will be written to destination specified, the file will be named as "sample_normed_depth.txt". If set to FALSE, a GRangesList object will be returned
destination	A character, specify the path to the location where the normalized coverage table will be written. Default: NULL, the file will be written to current working directory
plot	Boolean Default: TRUE. If TRUE, the coverage vs. GC content plot before and after normalization will be plotted And the average coverage for each chromosome before and after normalization will be plotted

Value

If writeToFile is set to TRUE, normalized coverage will be written to the destination. Otherwise, a GRangesList object containing each of input sample will be returned.

Note

The normalize function works better when you have multiple samples sequenced using the same protocol, namely have the same targeted regions. And if you have female sample and male sample, the best way is to normalize them separately.

Author(s)

Yu Kong

References

C. Alkan, J. Kidd, T. Marques-Bonet et al (2009). Personalized copy number and segmental duplication maps using next-generation sequencing. Nature Genetics, 41(10):1061-7.

See Also

prepareCoverageGC

Examples

```
##------------------------------------
##if you deal with single sample
##------------------------------------
## 1. prepare coverage and gc
## specify the path to the location of bed file
target = system.file("extdata","target.bed",package="MADSEQ")

## specify the path to the bam file
aneuploidy_bam = system.file("extdata","aneuploidy.bam",package="MADSEQ")

## prepare coverage data for the aneuploidy sample
aneuploidy_cov_gc = prepareCoverageGC(target,aneuploidy_bam,"hg19")

## normalize the coverage
##---- if not write to file ----
aneuploidy_norm = normalizeCoverage(aneuploidy_cov_gc,writeToFile=FALSE)
## check the GRangesList and subset your sample
aneuploidy_norm
names(aneuploidy_norm)
aneuploidy_norm["aneuploidy_cov_gc"]

##---- if write to file ----
normalizeCoverage(aneuploidy_cov_gc,writeToFile=TRUE,destination=".")

##------------------------------------------------------------
##if you deal with multiple samples without normal control
##------------------------------------------------------------
## specify the path to the location of bed file
target = system.file("extdata","target.bed",package="MADSEQ")

## specify the path to the bam file
aneuploidy_bam = system.file("extdata","aneuploidy.bam",package="MADSEQ")
normal_bam = system.file("extdata","normal.bam",package="MADSEQ")

## prepare coverage data for the samples
aneuploidy_cov_gc = prepareCoverageGC(target,aneuploidy_bam,"hg19")
```

```
normal_cov_gc = prepareCoverageGC(target,normal_bam,"hg19")

## normalize the coverage
normed=normalizeCoverage(aneuploidy_cov_gc,normal_cov_gc,writeToFile=FALSE)
names(normed)
normed["aneuploidy_cov_gc"]
normed["normal_cov_gc"]
## or
normalizeCoverage(aneuploidy_cov_gc,normal_cov_gc,
                  writeToFile=TRUE,destination=".")

##------------------------------------------------------------
##if you deal with multiple samples with a normal control
##------------------------------------------------------------
## specify the path to the location of bed file
target = system.file("extdata","target.bed",package="MADSEQ")

## specify the path to the bam file
aneuploidy_bam = system.file("extdata","aneuploidy.bam",package="MADSEQ")
normal_bam = system.file("extdata","normal.bam",package="MADSEQ")

## prepare coverage data for the samples
aneuploidy_cov_gc = prepareCoverageGC(target,aneuploidy_bam,"hg19")
normal_cov_gc = prepareCoverageGC(target,normal_bam,"hg19")

## normalize the coverage
normed = normalizeCoverage(aneuploidy_cov_gc,
                           control=normal_cov_gc,writeToFile=FALSE)
## or
normalizeCoverage(aneuploidy_cov_gc,control=normal_cov_gc,
                  writeToFile=TRUE,destination=".")
```

`plotFraction` *histgram for the fraction of aneuploid cells estimated by MadSeq model*

Description

histgram of the posterior distribution of the fraction of aneuploid cells estimated by the selected model.

Usage

```
plotFraction(object, prob = 0.95)

## S4 method for signature 'MadSeq'
plotFraction(object, prob = 0.95)
```

Arguments

`object`	A `MadSeq` object returned by `runMadSeq` function.
`prob`	A `numeric` value between 0~1 specify the highest posterior interval (similar to credible interval) for the distribution. Default: 0.95.

Value

the histgram of posterior distribution of the fraction

Note

If normal model has been selected by runMadSeq function, no fraction plot will be produced by this function.

Author(s)

Yu Kong

Yu Kong

See Also

runMadSeq, plotMadSeq, plotMixture

Examples

```
## load the example MadSeq object come with the package
data("aneuploidy_chr18")

## plot estimated fraction of aneuploid cells
plotFraction(aneuploidy_chr18)
```

---

plotMadSeq *density plot for posterior distribution of selected model*

Description

plot the density plot for each of the parameters in the posterior distribution from selected model

Usage

```
plotMadSeq(object)

## S4 method for signature 'MadSeq'
plotMadSeq(object)
```

Arguments

object	A MadSeq object returned by runMadSeq function.

Value

the density plot for parameters in the posterior distribution of selected model.

Author(s)

Yu Kong

Yu Kong

See Also

runMadSeq, plotFraction, plotMixture

Examples

```
## load the example MadSeq object come with the package
data("aneuploidy_chr18")

## plot the posterior distribution
plotMadSeq(aneuploidy_chr18)
```

---

plotMixture *density plot for the posterior distribution of alternative allele frequency estimated from the selected model*

Description

density plot presents the posterior distribution of alternative allele frequency (AAF) estimated from selected model

Usage

```
plotMixture(object)

## S4 method for signature 'MadSeq'
plotMixture(object)
```

Arguments

object	A MadSeq object returned by runMadSeq function.

Value

density plot for the posterior distribution of AAF

Author(s)

Yu Kong

Yu Kong

See Also

runMadSeq, plotMadSeq, plotFraction

Examples

```
## load the example MadSeq object come with the package
data("aneuploidy_chr18")

## plot the distribution of estimated AAF
plotMixture(aneuploidy_chr18)
```

posterior *Accessing posterior distribution of MadSeq object*

Description

An S4 method to access the posterior distribution of MadSeq object

Usage

```
posterior(object)

## S4 method for signature 'MadSeq'
posterior(object)
```

Arguments

object	A MadSeq object returned by runMadSeq function

Value

A matrix containing posterior distribution of selected model

Author(s)

Yu Kong

Yu Kong

See Also

MadSeq, runMadSeq

Examples

```
## load the example MadSeq object come with the package
data("aneuploidy_chr18")

## access posterior distribution
posterior(aneuploidy_chr18)
```

prepareCoverageGC *get sequencing coverage and GC content for targeted regions*

Description

Given a bam file and a bed file containing targeted regions, return sequencing coverage and GC content for each targeted region

Usage

```
prepareCoverageGC(target_bed, bam, genome_assembly = "hg19")
```

Arguments

target_bed	A character, specify the path to the location of bed file containing targeted regions.
bam	character, path to the bam file. Please make sure that bam file is sorted, and the index bam is present
genome_assembly	A character, indicating the assembly number of your genome. Default:"hg19". To see available genome_assembly, use available.genomes from BSgenome package

Value

a GRanges object with at least two mcols: depth and GC, each range indicating a targeted region

Note

The bam file should be sorted and indexed.

Author(s)

Yu Kong

See Also

normalizeCoverage

Examples

```
## specify the path to the location of bed file
target = system.file("extdata","target.bed",package="MADSEQ")

## specify the path to the bam file
aneuploidy_bam = system.file("extdata","aneuploidy.bam",package="MADSEQ")
normal_bam = system.file("extdata","normal.bam",package="MADSEQ")

## prepare coverage data for the samples
aneuploidy_cov_gc = prepareCoverageGC(target,aneuploidy_bam,"hg19")
normal_cov_gc = prepareCoverageGC(target,normal_bam,"hg19")
```

---

prepareHetero *prepare heterozygous sites for aneuploidy detection*

Description

given the vcf file and bed file containing targeted region, generate processed heterozygous sites for furthur analysis

Usage

```
prepareHetero(vcffile, target_bed, genome = "hg19", writeToFile = TRUE,
  destination = NULL, plot = FALSE)
```

Arguments

vcffile	A character, specify the path to the location of the vcf.gz file of your sample. **Note:** the vcf file need to be compressed by bgzip. The tool is part of tabix package, can be download from http://www.htslib.org/
target_bed	A character, specify the path to the location of bed file containing targeted regions.
genome	A character, specify the assembly of your genome. Default: hg19. To see available genome assembly, use available.genomes from BSgenome package
writeToFile	Boolean Default: TRUE. If TRUE, processed table containing heterozygous sites will be written to destination specified, the file will be named as "sample_filtered_heterozygous.txt' If set to FALSE, a GRanges object containing processed heterozygous sites will be returned
destination	A character, specify the path to the location where the processed heterozygous sites table will be written. Default: NULL, the file will be written to current working directory
plot	A Boolean Default: FALSE. If TRUE, A plot showing AAF before and after filtering for problematic regions will be generated

Value

If writeToFile is set to TRUE, processed table will be written to the destination. Otherwise, a GRanges object containing each of input sample will be returned.

Note

1. The vcf file you provided need to be compressed by bgzip
2. The vcf file should contain depth and allelic depth for variants in the FORMAT field

Author(s)

Yu Kong

See Also

runMadSeq

Examples

```
## specify the path to the vcf.gz file for the aneuploidy sample
aneuploidy_vcf=system.file("extdata","aneuploidy.vcf.gz",package="MADSEQ")
target = system.file("extdata","target.bed",package="MADSEQ")
##------ if not write to file ------
aneuploidy_hetero=prepareHetero(aneuploidy_vcf,target,writeToFile=FALSE)

##------ if write to file ------
prepareHetero(aneuploidy_vcf, target,writeToFile=TRUE, destination=".")
```

runMadSeq *Model to detect and quantify mosaic aneuploidy*

Description

Take in the heterozygous sites and coverage information, use different models (normal, monosomy, mitotic trisomy, meiotic trisomy, loss of heterozygosity) to fit the data, and select the model fit the data best according to BIC value and return estimation of the fraction of aneuploid cells.

Usage

```
runMadSeq(hetero, coverage, target_chr, adapt = 10000, burnin = 10000,
  nChain = 2, nStep = 10000, thinSteps = 2, checkConvergence = FALSE,
  plot = TRUE)
```

Arguments

hetero	A character specify the location of processed heterozygous table returned by prepareHetero function, or A GRanges object returned by prepareHetero function
coverage	A character specify the location of normalized coverage table returned by normalizeCoverage function, or A GRanges object from the GRangesList returned by normalizeCoverage function. Look up your sample by names(GRangesList), and subset your the normalized coverage for your sample by GRangesList["sample_name"]. For more details, please check the example.
target_chr	A character specify the chromosome number you want to detect. **Note:** Please check your assembly, use contig name "chr1" or "1" accordingly.
adapt	A integer indicate the adaption steps for the MCMC sampling. Default: 10000
burnin	A integer indicate burnin steps for the MCMC sampling. Default: 10000. If the posterior distribution is not converged, increasing burnin steps can be helpful.
nChain	A integer indicate the number of chains for the MCMC sampling. Default: 2. **Note:** More than 1 chain is required if checkConvergence is set to TRUE.
nStep	A integer indicate the number of steps to be recorded for the MCMC sampling. Default: 10000. Generally, the more steps you record, the more accurate the estimation is.
thinSteps	A integer indicate the number of steps to "thin" (thinSteps=1) means save everystep. Default: 2.
checkConvergence	A Boolean indicate whether to check the convergence of independent MCMC chains. If your data is not converged, you may increase adaption step and burnin step. Default: FALSE
plot	A Boolean. If TRUE, the alternative allele frequency (AAF) for each heterozygous site along the target chromosome will be plotted.

Value

An S4 object of class MadSeq containing the posterior distribution for the selected model, and deltaBIC between five models.

Note

1.If you didn't write normalized coverage into file, please subset the normalized coverage GRanges object from the GRangesList object returned from the normalizeCoverage funciton.
2. When specify target_chr, please make sure it consist with the contig names in your sequencing data, example: "chr1" and "1".
3. If checkConvergence set to TRUE, the nChain has to be >2
4. If it shows that your chains are not converged, helpful options are increasing the adapt and burnin steps.
5. Because the model is an MCMC sampling process, it can take a very long time to finish. Running in the background or HPC is recommended.

Author(s)

Yu Kong

References

Martyn Plummer (2016). rjags: Bayesian Graphical Models using MCMC. R package version 4-6. https://CRAN.R-project.org/package=rjags

See Also

MadSeq, plotMadSeq, plotFraction, plotMixture

Examples

```
## ------------------------------------------------------------------------
## The following example is for the case that normalized coverage and
## processed heterozygous sites have not been written to files. For more
## examples, please check the documentation.
## ------------------------------------------------------------------------
##------Prepare Heterozygous Sites
## specify the path to the vcf.gz file for the aneuploidy sample
aneuploidy_vcf = system.file("extdata","aneuploidy.vcf.gz",package="MADSEQ")
## specify the path to the bed file containing targeted region
target = system.file("extdata","target.bed",package="MADSEQ")
## prepare heterozygous sites
aneuploidy_hetero = prepareHetero(aneuploidy_vcf,target, writeToFile=FALSE)

##------Prepare Normalized Coverage
## specify the path to the bam file
aneuploidy_bam = system.file("extdata","aneuploidy.bam",package="MADSEQ")
normal_bam = system.file("extdata","normal.bam",package="MADSEQ")

## prepare coverage data for the samples
aneuploidy_cov_gc = prepareCoverageGC(target,aneuploidy_bam,"hg19")
normal_cov_gc = prepareCoverageGC(target,normal_bam,"hg19")

## normalize the coverage
normed = normalizeCoverage(aneuploidy_cov_gc,
                           control=normal_cov_gc,writeToFile=FALSE)

##------subset normalized coverage GRanges object
aneuploidy_normed_cov = normed[["aneuploidy_cov_gc"]]
## check chromosome18
```

```
## (to speed up the example, we only run one chain and less steps here,
##  but default settings are recommended in real case)
aneuploidy_chr18 = runMadSeq(aneuploidy_hetero, aneuploidy_normed_cov,
                             target_chr="chr18", adapt=100, burnin=200,
                             nChain =1, nStep = 1000, thinSteps=1)
```

`summary,MadSeq-method` *Summarize statistics of the MadSeq object*

Description

An S4 method to summarize statistics for `MadSeq` object

Usage

```
## S4 method for signature 'MadSeq'
summary(object)
```

Arguments

`object`	A `MadSeq` object returned by `runMadSeq` function

Value

a `table` containing statistics for each parameters in the selected model

Author(s)

Yu Kong

Examples

```
## load the example MadSeq object come with the package
data("aneuploidy_chr18")

## show statistics
summary(aneuploidy_chr18)
```

Index