Package 'BaySIC'

February 19, 2015

Type Package

Title Bayesian Analysis of Significantly Mutated Genes in Cancer

Version 1.0

Date 2013-03-12

Author Nicholas B. Larson

Maintainer Nicholas B. Larson <larson.nicholas@mayo.edu>

Depends R (>= 2.10), rjags, fields, poibin

Description This R package is the software implementation of the algorithm BaySIC, a Bayesian approach toward analysis of significantly mutated genes in cancer data.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2013-04-04 00:51:05

R topics documented:

BaySIC-package *Bayesian Analysis of Significantly Mutated Genes in Cancer*

Description

Software implementation of the algorithm BaySIC, a Bayesian approach toward analysis of significantly mutated genes in cancer data.

Details

Package:	BaySIC
Type:	Package
Version:	1.0
Date:	2013-03-12
License:	GPL (>=2)

This package provides functions for Bayesian SMG analysis, which includes plotting functions, model definition and fitting, and evaluation of individual genes using posterior predictive methods. BaySIC is a flexible algorithm that can accommodate gene-level covariate data, varying subject-specific sequence coverage, and subtype analysis. It also includes two reference data files (`ccds.18` and `ccds.19`) corresponding to human genome builds hg18 and hg19, which respectively consist of sequence context enumeration of the Consensus Coding Sequence genes in each build.

Author(s)

Nicholas B. Larson

Maintainer: Nicholas B. Larson <larson.nicholas@mayo.edu>

baysic.data *Organizes data for BaySIC functions*

Description

Creates a `list` object from mutation and reference data for use with BaySIC fitting and testing functions

Usage

```
baysic.data(dat, ref.dat, plot = FALSE, N = NULL, silent = TRUE)
```

Arguments

dat	matrix; Mutation input data. Baysic requires a specific format similar to the MUT format file, and should be an $M \times 7$ matrix with column headings "chr", "start", "end", "id","type", "gene","context," where each row details an individual mutation.
ref.dat	a dataframe or list of dataframes; ref.dat is a representation of the sequence content of each gene of interest, for 32 unique trinucleotide sequence contexts, yielding an $G \times 34$ matrix, where G is the total number of genes. If ref.dat is a matrix, it is assumed that all subjects correspond to the same reference data. It is possible that reference data may vary from subject to subject due to different platforms or coverages. In this case, ref.dat can also be a list of N reference data matrices, where N is the number of subjects. The names of each list element should correspond to ids used in the dat file.
plot	logical; if TRUE, a plot summarizing the mutation data at an overall and per subject basis is generated. Defaults to FALSE.
N	an integer (optional); equal to the number of subjects represented in dat. If N=NULL and is.list(ref.dat)==FALSE, N is assumed to the number of unique subject ids in dat. If is.list(ref.dat)=TRUE, then N=length(ref.dat).
silent	logical; if FALSE, mutations defined as 'Synonymous' or 'Silent' will be removed from the dataset and subsequent analyses. Defaults to TRUE.

Details

The mutation data dat is a 7-column matrix similar in style to other popular mutation file formats. The first three columns ("chr","start","end") correspond to the positional information of the somatic mutation. The "id" column represents an identification vector including subject ids for each documented mutation. The "type" column corresponds to the type of mutation for each entry. This is relatively flexible for point mutations, and only requires some form of "silent" or "synonymous" for such mutations if silent=FALSE, but insertion/deletion events should be designated as "INDEL." The "gene" column represents the name of the gene the mutation corresponds to, and must match the gene names used in ref.dat. The "context" entries represent the trinucleotide sequence context of each point mutation (NA for INDELS)

The first two columns of the data matrix (or matrices) in ref.dat should correspond to the gene name and corresponding chromosome, and the column names of the remaining 32 columns should correspond to the trinucleotide motif (e.g. "ACA"). The sequence content entries should be integer values which correspond to the number of nucleotides in the coding content of a given gene which satisify the trinucleotide motif (central base with flanking 5' and 3' bases). Each base should be uniquely represented, such that the sum of all 32 counts is equivalent to the basepair length of the total coding sequence for a given gene.

The baysic.data function has its own trinucleotide naming convention, in that all motifs are in all caps and have either "T" or "C" as the central base. Column names of ref.dat and "context" entries in dat will be adjusted to accommodate this convention if they deviate from it.

Value

Returns a list data structure with the following components:

all.dat	Original mutation data object dat
ref.dat	Original reference data object ref.dat
N	Number of subjects with observed data
genes	Vector of length G of gene names included in analysis, where G is the total number of genes. Derived from ref.dat
snv.dat	A $G \times 32$ matrix of total number of SNV mutations per sequence context and gene
indel.dat	Vector of length G of total number of indel mutations per gene

Author(s)

Nicholas B. Larson

See Also

baysic.fit,baysic.test

Examples

```
## Not run:
data(example.dat)
data(ccds.19)
baysic.dat.ex<-baysic.data(example.dat,ccds.19)

## End(Not run)
```

---

baysic.fit *Fits BaySIC BMR model*

Description

Generates an MCMC model fit of the BaySIC BMR model

Usage

```
baysic.fit(dat.out, snv.cat, covar = NULL, excl.list = NULL, burn.in = 10000,n.samp = 25000, fn.jags =
```

Arguments

dat.out	Output from baysic.data
snv.cat	a list of length C, where C is the number of sequence categories desired to be modeled ($C \leq 32$). Each element of snv.cat should be a vector of character strings of trinucleotide motifs (e.g., c("ATA","ACA")) which define a group of motifs which are assumed to have the same background mutation rate.
covar	optional $G \times Q$ matrix of gene-level covariate data, where G is the total number of genes and Q the number of covariates.

excl.list	optional vector of genes to be excluded from model fitting process. The format of excl.list can be either character or numeric, the former indicating the names of genes and the latter their order in ref.dat.
burn.in	an integer; represents the burn-in size to apply in the MCMC model fitting using JAGS. Defaults to 10,000
n.samp	an integer; represents the size of the MCMC posterior sample draw from the fitted model. Defaults to 25,000
fn.jags	a character string; corresponds to the file name and location of the JAGS model file to be written. Defaults to "baysic.jags" in the current working directory.
prior	optional vector of prior distribution specifications (as character strings). If is.null(prior)==FALSE, prior should be of length equal to all of the model parameters and formatted to follow the distributional notation of the JAGS model language. The order of the prior specification follows the format: SNV categories, any covariates (optional), indel λ parameter.

Value

Returns a list object with the following components:

fit.post	an mcmc object of the posterior draws of the BaySIC BMR model parameters
covar	covar object (if included in baysic.fit argument)
snv.cat	the snv.cat object in the original call
excl.list	excl.list object (if included in baysic.fit argument)

Author(s)

Nicholas B. Larson

See Also

baysic.data,baysic.test

Examples

```
## Not run:
data(example.dat)
data(ccds.19)
baysic.dat.ex<-baysic.data(example.dat,ccds.19)
snv.cat.ex<-list()
snv.cat.ex[[1]]<-grep("[^T]C[^G]",colnames(ccds.19)[-c(1:2)])
snv.cat.ex[[2]]<-unique(c(grep("TC.",colnames(ccds.19)[-c(1:2)]),grep(".CG",colnames(ccds.19)[-c(1:2)])))
snv.cat.ex[[3]]<-grep(".T.",colnames(ccds.19)[-c(1:2)])
baysic.fit.ex<-baysic.fit(baysic.dat.ex,snv.cat.ex)

## End(Not run)
```

baysic.test *BaySIC Evaluation of SMGs*

Description

Evaluates genes for SMGs using Bayesian posterior predictive methods

Usage

```
baysic.test(dat.out, fit.out, fdr.level = 0.15, fuzzy.cnt = 10000, r = NULL,subtype = NULL, PB.approx
```

Arguments

dat.out	output from baysic.data
fit.out	output from baysic.fit which utilized dat.out
fdr.level	numeric ($\in (0,1)$) defining FDR level for multiple assessment passed to fuzzy.FDR.approx. Defaults to 0.15
fuzzy.cnt	number of Monte Carlo iterations to use in approximating fuzzy FDR values passed to fuzzy.FDR.approx. Defaults to 10000.
r	Optional number of MCMC draws to thin to for Monte Carlo integration, such that r<R, where R is the total number of MCMC draws.
subtype	Optional $N_s \times 2$ dataframe that defines membership of cancer subtype(s), where $N_s \leq N$. The first column of subtype should consist of subject ids (same as in dat) and the second the corresponding subtype membership. When subtype is provided, baysic.test will also generate analysis results for subtype-specific analyses.
PB.approx	logical; if TRUE, the Refined Normal Approximation (RNA) of the Poisson-Binomial distribution is used when ref.dat is a list. Defaults to FALSE.

Details

When is.list{ref.dat} is TRUE, BaySIC evaluates whether or not a gene is an SMG using the Poisson-Binomial rather than the traditional binomial distribution. This accomodates subject-specific mutation rates given varying sequence content. When N is relatively large (e.g., $N \geq 50$) it is recommended that optional arguments r and PB.approx be considered to alleviate computational burden.

Value

Returns a list object with the following components:

test.res	a matrix with G rows containing the SMG analysis results from BaySIC. This includes the gene, the posterior predictive p-values, and fuzzy rejection probabilities under FDR level fdr.level. It will also contain results for any subtype analyses if subtype is specified.
fdr.level	value of fdr.level used

fuzzy.cnt	value of fuzzy.cnt used
subtype	value of subtype, if supplied

Author(s)

Nicholas B. Larson

Examples

```
## Not run:
data(example.dat)
data(ccds.19)
baysic.dat.ex<-baysic.data(example.dat,ccds.19)
snv.cat.ex<-list()
snv.cat.ex[[1]]<-grep("[^T]C[^G]",colnames(ccds.19)[-c(1:2)])
snv.cat.ex[[2]]<-unique(c(grep("TC.",colnames(ccds.19)[-c(1:2)]),grep(".CG",colnames(ccds.19)[-c(1:2)])))
snv.cat.ex[[3]]<-grep(".T.",colnames(ccds.19)[-c(1:2)])
baysic.fit.ex<-baysic.fit(baysic.dat.ex,snv.cat.ex)
baysic.test.ex<-baysic.test(baysic.dat.ex,baysic.fit.ex)

## End(Not run)
```

BMR.plot *Visualize Sequence Context BMRs*

Description

Generates a heatmap of mutation rates by sequence context to assist in determining somatic point sequence context categories for BMR model

Usage

```
BMR.plot(dat.out)
```

Arguments

dat.out	output from baysic.data

Value

Generates a heatmap of point mutation rates by trinucleotide sequence context motif, which is corrected for values in ref.dat, on the log10 scale

Author(s)

Nicholas B. Larson

See Also

baysic.data

Examples

```
## Not run:
data(example.dat)
data(ccds.19)
baysic.dat.ex<-baysic.data(example.dat,ccds.19)
BMR.plot(baysic.dat.ex)

## End(Not run)
```

---

ccds.18 *CCDS Reference Data (Build hg18)*

Description

A `ref.dat` object for the Consensus Coding Sequence (CCDS) data from UCSC human (*Homo sapiens*) build hg18

Usage

```
data(ccds.18)
```

Format

A data frame `ref.dat` object with 16631 genes on 34 variables (1 Gene column, 1 chromosome column, and 32 sequence context motifs

Details

Each sequence context motif column corresponds to the enumeration of CCDS bases in a given gene that satifies that motif. The gene column corresponds to HUGO gene ids.

Source

http://genome.ucsc.edu/

References

http://www.ncbi.nlm.nih.gov/CCDS/CcdsBrowse.cgi

ccds.19 *CCDS Reference Data (Build hg19)*

Description

A `ref.dat` object for the Consensus Coding Sequence (CCDS) data from UCSC human (*Homo sapiens*) build hg19

Usage

```
data(ccds.19)
```

Format

A data frame `ref.dat` object with 18305 genes on 34 variables (1 Gene column, 1 chromosome column, and 32 sequence context motifs

Details

Each sequence context motif column corresponds to the enumeration of CCDS bases in a given gene that satifies that motif. The gene column corresponds to HUGO gene ids.

Source

http://genome.ucsc.edu/

References

http://www.ncbi.nlm.nih.gov/CCDS/CcdsBrowse.cgi

example.dat *Example Mutation Data (Build hg19)*

Description

An example simulated dataset in the input format used by BaySIC (MUT-style), consisting of observed somatic mutations for 200 subjects

Usage

```
data(example.dat)
```

Format

A dataframe object with 9882 observations on 7 variables

Details

BaySIC utilizes a MUT-style format for input into its functions, which is a $M \times 7$ dataframe where M is the number of observed mutations, and has the following structure:

- `chr`character string; chromosome (e.g.,"chr#","chrX","chrY")
- `start`integer; start basepair position
- `end`integer; end basepair position
- `id`character string; subject identification
- `type`character string; type of somatic mutation (e.g., "SNV", "INDEL")
- `gene`character string; corresponding gene of mutation
- `context`character string; sequence context (trinucleotide motif) of point mutation (`NA` for INDEL)

`fn.cat` *Collapses SNV and reference data into sequence mutation categories*

Description

Subroutine for `baysic.fit` and `baysic.test` which generates reduced data representations of mutation and reference data by collapsing sequence categories into single columns

Usage

```
fn.cat(dat, snv.cat)
```

Arguments

`dat`	$G \times 32$ `ref.dat` or `snv.dat` data matrix
`snv.cat`	a `list` of length C, where C is the number of sequence categories desired to be modeled ($C \leq 32$). Each element of `snv.cat` should be a vector of character strings of trinucleotide motifs (e.g., c("ATA","ACA")) which define a group of motifs which are assumed to have the same background mutation rate.

Value

a $G \times C$ matrix where each column corresponds to an SNV sequence category in `snv.cat`

Author(s)

Nicholas B. Larson

See Also

`baysic.fit`,`baysic.test`

`fuzzy.FDR.approx` *Generate Approximate Fuzzy Rejection Probabilites*

Description

For hypothesis tests with discrete reference distributions, obtains fuzzy rejection probabilites for a given level of false discovery rate control

Usage

```
fuzzy.FDR.approx(pprev, p, alpha, N)
```

Arguments

`pprev`	numeric vector of p-values of length l, corresponding to strict inequality of test statistic values in a one-sided test (i.e., $P(T > t)$)
`p`	length l numeric vector of p-values corresponding to traditional one-sided test (i.e., $P(T \geq t)$).
`alpha`	FDR level of interest (under Benjamini-Hochberg FDR procedure)
`N`	Number of Monte Carlo samples used to generate fuzzy rejection probability approximations.

Details

This is a Monte Carlo implementation of the fuzzy FDR work developed by Kulinskaya et al. (2007)

Value

Returns a vector of length l corresponding to the fuzzy rejection probabilities of the hypotheses represented in `pprev` and `p`, under FDR level `alpha`

Author(s)

Nicholas B. Larson

References

http://www.bgx.org.uk/alex/Kulinskaya-Lewin-resubmit.pdf

revcomp *DNA Reverse Complementation*

Description

Returns the reverse complement of a given DNA character string

Usage

```
revcomp(dna.seq)
```

Arguments

dna.seq	character string; genetic sequence composed of "A","C","T", and "G" characters, of which the reverse complement sequence is desired

Value

A character string that is the reverse complement of dna.seq

Author(s)

Nicholas B. Larson

Examples

```
test.sequence<-"ACTGATGAT"
revcomp(test.sequence)
```

write.baysic *Write BaySIC JAGS model files*

Description

Procedurally writes JAGS model files based upon the arguments for the BaySIC model fitting function baysic.fit.

Usage

```
write.baysic(mut.dat, covar = NULL, prior = NULL, fn.jags = "baysic.jags")
```

Arguments

mut.dat	matrix or dataframe containing the observed SNV and indel. The indel counts should be contained in the final column. Column names from this object will be used to create the model file.
covar	optional matrix or dataframe of gene-level covariates. Column names from this object will be used to create the model file.
prior	optional vector of character strings which define prior distributions on the model parameters in the JAGS language format. If prior is non-NULL, it should be of length equal to all possible model parameters (sum of number of columns of mut.dat and covar)
fn.jags	file name of JAGS model file to be used. Defaults to "baysic.jags"

Details

This function is a subroutine used in baysic.fit

Value

Writes JAGS file to the location specified by fn.jags

Author(s)

Nicholas B. Larson

See Also

baysic.test

Index