

# Package ‘negenes’

January 2, 2012

**Version** 0.99-2

**Date** 7 Nov 2011

**Title** Estimating the number of essential genes in a genome

**Author** Karl W Broman <kbroman@biostat.wisc.edu>

**Maintainer** Karl W Broman <kbroman@biostat.wisc.edu>

**Description** Estimating the number of essential genes in a genome on the basis of data from a random transposon mutagenesis experiment, through the use of a Gibbs sampler.

**Depends** R (>= 2.10.1), stats

**License** GPL (>= 2)

**URL** <http://www.biostat.wisc.edu/~kbroman/software/negenes.html>

**Repository** CRAN

**Date/Publication** 2011-11-29 06:57:40

## R topics documented:

Mtb80 . . . . .	2
negenes . . . . .	3
sim.mutants . . . . .	5

<b>Index</b>	<b>7</b>
--------------	----------

Mtb80

*Number of insertion sites in each gene in Mtb CDC1551***Description**

Number of insertion sites in the initial 80% of each gene in the *Mycobacterium tuberculosis* CDC1551 genome.

**Usage**

```
data(Mtb80)
```

**Format**

A matrix with two columns. Each row corresponds to a gene. (The row names are the MT numbers of the genes.) The element in the first column is the number of transposon insertion sites in the initial 80% that appear in the corresponding gene and in no other gene. The element in the second column is the number of transposon insertion sites in the initial 80% of both that gene and the following gene. There are 4204 rows; the 46 genes with no such site are not included.

**Source**

<http://www.tigr.org>

**References**

Blades, N. J. and Broman, K. W. (2002) Estimating the number of essential genes in a genome by random transposon mutagenesis. Technical Report MS02-20, Department of Biostatistics, Johns Hopkins University, Baltimore, MD. <http://www.biostat.biostat.wisc.edu/~kbroman/publications/ms0220.pdf>

**See Also**

[negenes](#), [sim.mutants](#)

**Examples**

```
## Not run: data(Mtb80)

# simulate 44% of genes to be essential
essential <- rep(0,nrow(Mtb80))
essential[sample(1:nrow(Mtb80),ceiling(nrow(Mtb80)*0.44))] <- 1

# simulate 759 mutants
counts <- sim.mutants(Mtb80[,1], essential, Mtb80[,2], 759)

# run the Gibbs sampler
output <- negenes(Mtb80[,1], counts[,1], Mtb80[,2], counts[,2])
## End(Not run)
```

negenes

*Estimate the number of essential genes in a genome***Description**

Estimate, via a Gibbs sampler, the posterior distribution of the number of essential genes in a genome with data from a random transposon mutagenesis experiment.

**Usage**

```
negenes(n.sites, counts, n.sites2, counts2,
        n.mcmc=5000, skip=49, burnin=500,
        startp=1, trace=TRUE,
        calc.prob=FALSE, return.output=FALSE)
```

**Arguments**

n.sites	A vector specifying the number of transposon insertion sites in each gene (alone). All elements must be strictly positive.
counts	A vector specifying the number of mutants observed for each gene (alone). Must be the same length as n.sites, and all elements must be non-negative integers.
n.sites2	A vector specifying the number of transposon insertion sites shared by adjacent genes. The <i>i</i> th element is the number of insertion sites shared by genes <i>i</i> and <i>i</i> +1. The last element is for sites shared by genes <i>N</i> and 1.
counts2	A vector specifying the number of mutants shared by adjacent gene (analogous to n.sites2).
n.mcmc	Number of Gibbs steps to perform.
skip	An integer; only save every skip + 1st step.
burnin	Number of initial Gibbs steps to run (output discarded).
startp	Initial proportion of genes for which no mutant was observed that will be assumed essential for the Gibbs sampler. (Genes for which a mutant was observed are assumed non-essential; other genes are assumed essential independent with this probability.)
trace	If TRUE, print iteration number occasionally.
calc.prob	If TRUE, return the log posterior probability (up to an additive constant) for each saved iteration.
return.output	If TRUE, include detailed Gibbs results in the output.

**Details**

See the technical report cited below.

**Value**

A list with components `n.essential` (containing the total number of essential genes at each iteration of the Gibbs sampler) `summary` (a vector containing the estimated mean, SD, 2.5 percentile and 97.5 percentile of the posterior distribution of the number of essential genes).

The next component, `geneprob`, is a vector with one element for each gene, containing the estimated posterior probability that each gene is essential. These are Rao-Blackwellized estimates.

If the argument `calc.prob` was true, there will also be a component `logprob` containing the log (base e) of the posterior probability (up to an additive constant) at each Gibbs step.

If the argument `return.output` was true, there will also be a matrix with `n.mcmc / (skip + 1)` rows (corresponding to the Gibbs steps) and a column for each gene. The entries in the matrix are either 0 (essential gene) or 1 (non-essential gene) according to the state of that gene at that step in the Gibbs sampler.

**Author(s)**

Karl W Broman, <[kbroman@biostat.wisc.edu](mailto:kbroman@biostat.wisc.edu)>

**References**

Blades, N. J. and Broman, K. W. (2002) Estimating the number of essential genes in a genome by random transposon mutagenesis. Technical Report MS02-20, Department of Biostatistics, Johns Hopkins University, Baltimore, MD. <http://www.biostat.biostat.wisc.edu/~kbroman/publications/ms0220.pdf>

**See Also**

[sim.mutants, Mtb80](#)

**Examples**

```
data(Mtb80)

# simulate 44% of genes to be essential
essential <- rep(0,nrow(Mtb80))
essential[sample(1:nrow(Mtb80),ceiling(nrow(Mtb80)*0.44))] <- 1

# simulate 759 mutants
counts <- sim.mutants(Mtb80[,1], essential, Mtb80[,2], 759)

# run the Gibbs sampler without returning detailed output
output <- negenes(Mtb80[,1], counts[,1], Mtb80[,2], counts[,2])

# run the Gibbs sampler, returning the detailed output
output2 <- negenes(Mtb80[,1], counts[,1], Mtb80[,2], counts[,2], return=TRUE)
```

---

sim.mutants	<i>Simulate data for a random transposon mutagenesis experiment</i>
-------------	---

---

**Description**

Simulate data for a random transposon mutagenesis experiment.

**Usage**

```
sim.mutants(n.sites, essential, n.sites2, n.mutants)
```

**Arguments**

n.sites	A vector specifying the number of transposon insertion sites in each gene. All elements must be strictly positive.
essential	A vector containing 1's (indicating that the corresponding gene is essential) and 0's (indicating that the corresponding gene is not essential). Must be the same length as n.sites.
n.sites2	A vector specifying the number of transposon insertion sites shared by adjacent genes. The <i>i</i> th element is the number of insertion sites shared by genes <i>i</i> and <i>i</i> +1. The last element is for sites shared by genes <i>N</i> and 1. If missing, these are assumed to be all 0.
n.mutants	Number of mutants to simulate.

**Value**

If n.sites2 is missing or contains all 0's, a vector is returned containing the number of mutants observed for each gene.

If n.sites2 is not missing and has some positive entries, a matrix with two columns is returned. The first column contains the number of mutants observed for each gene alone; the second column contains the number of mutants observed shared by adjacent genes.

**Author(s)**

Karl W Broman, <[kbroman@biostat.wisc.edu](mailto:kbroman@biostat.wisc.edu)>

**References**

Blades, N. J. and Broman, K. W. (2002) Estimating the number of essential genes in a genome by random transposon mutagenesis. Technical Report MS02-20, Department of Biostatistics, Johns Hopkins University, Baltimore, MD. <http://www.biostat.biostat.wisc.edu/~kbroman/publications/ms0220.pdf>

**See Also**

[negenes](#), [Mtb80](#)

**Examples**

```
## Not run: data(Mtb80)

# simulate 44% of genes to be essential
essential <- rep(0,nrow(Mtb80))
essential[sample(1:nrow(Mtb80),ceiling(nrow(Mtb80)*0.44))] <- 1

# simulate 759 mutants
counts <- sim.mutants(Mtb80[,1], essential, Mtb80[,2], 759)

# run the Gibbs sampler
output <- negenes(Mtb80[,1], counts[,1], Mtb80[,2], counts[,2])
## End(Not run)
```

# Index

\*Topic **datagen**  
sim.mutants, 5

\*Topic **datasets**  
Mtb80, 2

\*Topic **models**  
negenes, 3

Mtb80, 2, 4, 5

negenes, 2, 3, 5

sim.mutants, 2, 4, 5