Package: ricalc (via r-universe)

May 9, 2026

Version 0.56-2

Date 2021-08-05

Title Calculations for Recombinant Inbred Lines

Author Karl W Broman [aut, cre]
(<https://orcid.org/0000-0002-4914-6671>)

Maintainer Karl W Broman <broman@wisc.edu>

Description Calculate two- and three-locus probabilities in multiple-strain recombinant inbred lines, and simulate such lines. Broman (2005) <doi:10.1534/genetics.104.035212>.

Depends R (>= 2.10.1)

Imports stats, utils

License GPL-3

URL https://github.com/kbroman/ricalc

BugReports https://github.com/kbroman/ricalc/issues

Encoding UTF-8

ByteCompile true

Repository https://kbroman.r-universe.dev

Date/Publication 2026-05-09 20:19:00 UTC

RemoteUrl https://github.com/kbroman/ricalc

RemoteRef HEAD

RemoteSha e2915d698bbdffd221f3307971abde8baa93c570

Contents

ricalc-package 2
coinc.gam 3
convert.full.tm 4
convert2gen 5
count.absorb 6

fulltm . . . 7
genAILped . . . 7
get.full.tm . . . 8
get.full.tm.symbolic . . . 9
get.ril.coinc . . . 10
get.ril.prob . . . 11
get.start . . . 12
imf.gam . . . 13
lookup . . . 14
mf.gam . . . 14
mouseL . . . 15
mtm . . . 16
ricalcversion . . . 17
sim.ped . . . 17
sim.ri . . . 18
threept . . . 20

Index **21**

`ricalc-package` *Overview of R/ricalc package*

Description

A brief introduction to the contents of the R/ricalc package.

Stuff for the gamma model

- `coinc.gam`: Calculate the three-point coincidence function for the gamma model.
- `mf.gam`: Map function for the gamma model.
- `imf.gam`: Inverse of map function for the gamma model.

Data

- `mouseL`: Estimated genetic lengths of mouse chromosomes.
- `threept.AI`: Three-point haplotype probabilities for 8-way RIL, autosome, under interference.
- `threept.ANI`: Three-point haplotype probabilities for 8-way RIL, autosome, with no interference.
- `threept.XI`: Three-point haplotype probabilities for 8-way RIL, X chromosome, under interference.
- `threept.XNI`: Three-point haplotype probabilities for 8-way RIL, X chromosome, with no interference.

Symbolic data

- fulltm: Generation-to-generation transition matrices for 2- and 4-way RIL by selfing and by sibling mating (in symbolic notation...lists of lists of character strings).
- mtm: Transition matrices for meiosis, again in symbols...lists of lists of character strings).

Counting and enumerating parental types

- lookup: Lookup tables for parental types in the generation of multiple-strain RIL, representing the
- get.start: Symbolic notation for the starting state in the generation of multiple-strain RIL.
- count.absorb: Count the absorbing states in multiple-strain RIL.

Calculations and simulations

- sim.ri: Simulate multiple-strain RIL.
- get.ril.coinc: Numerical calculation of the 3-point coincidence-type quantity for 2- and 4-way RIL.
- get.ril.prob: Numerical calculation of haplotype probabilities on 2- or 4-way RIL chromosomes.
- get.full.tm: Calculate the generation-to-generation transition matrix for the formation of RIL.
- get.full.tm.symbolic: Calculate the generation-to-generation transition matrix for the formation of RIL, in symbolic form.
- convert.full.tm: Convert a sparse version of a transition matrix into a full matrix.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Broman, K. W. (2005) The genomes of recombinant inbred lines. *Genetics* **169**, 1133–1146.

Teuscher, F. and Broman, K. W. (2007) Haplotype probabilities for multiple-strain recombinant inbred lines. *Genetics* **175**, 1267–1274.

---

coinc.gam *Three-point coincidence under the gamma renewal model*

Description

Calculates the three-point coincidence under the gamma renewal model, under the assumption that the two defined intervals have the same recombination fraction.

Usage

```
coinc.gam(r, nu=1, tol=1e-12)
```

Arguments

r	Recombination fraction for each interval (can be a vector).
nu	Interference parameter (nu = m+1 in the chi-square model; nu=0 is NI).
tol	Tolerance for doing numerical integration.

Details

Uses the functions mf.gam and imf.gam.

Value

Vector of three-point coincidences, of same length as the input r.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Zhao, H. and Speed, T. P. (1996) On genetic map functions. *Genetics* **142**, 1369–1377.

Broman, K. W., Rowe, L. B., Churchill, G. A. and Paigen, K. (2002) Crossover interference in the mouse. *Genetics* **160**, 1123–1131.

See Also

mf.gam, imf.gam

Examples

```
coinc.gam(seq(0.01,0.49,by=0.01), 11.3)
```

---

convert.full.tm *Convert sparse version of transition matrix to full form*

Description

Convert a sparse version of a transition matrix (with only the non-zero elements) into a full matrix.

Usage

```
convert.full.tm(full.tm)
```

Arguments

full.tm	A list whose elements are the rows of the matrix, each being a vector with just the non-zero elements of the matrix.

Value

A full matrix version of the input.

Author(s)

Karl W Broman, <broman@wisc.edu>

See Also

get.full.tm, get.full.tm.symbolic

---

convert2gen *Convert simulated pedigree data to genotypes at markers*

Description

Convert data generated by sim.ped to a matrix of genotypes.

Usage

```
convert2gen(xodat, map)
```

Arguments

xodat	Allele information with continuous crossover locations, as generated by sim.ped.
map	A vector of marker locations, in cM.

Details

The marker locations (map) must be sorted and should span a length that is less then what was used to generated xodat.

If there are just two possible alleles (1 and 2) in xodat, the genotypes are coded as 1, 2, and 3, for genotypes 11, 12 and 22. Otherwise, they are converted to binary codes. For example, with possible alleles 1, 2, and 3, an individual with genotype a,b will be given the value $2^(a-1) + 2^(b-1)$.

Value

The output is a matrix with length(xodat) rows and length(map) columns. Genotype codes are described in Details.

Author(s)

Karl W Broman, <broman@wisc.edu>

See Also

sim.ped, genAILped

Examples

```
ailped <- genAILped(ngen=2, npairs=2, sibship.size=2)
aildat <- sim.ped(ailped, 100)
map <- seq(0, 100, by=10)
names(map) <- paste("M", 1:length(map), sep="")
ailgen <- convert2gen(aildat, map)
```

---

count.absorb *Count absorbing states*

Description

Determine the absorbing states of the Markov chain for the generation of recombinant inbred lines.

Usage

```
count.absorb(n.strains=c("2","4"),type=c("selfing","sibmating"),
         chrtype=c("A","X"),n.loci=c("2","3"))
```

Arguments

n.strains	Number of parental strains.
type	Method for mating.
chrtype	Indicates autosome or X chromosome (when type=sibmating only).
n.loci	Number of loci.

Value

A vector whose names indicate the minimal set of absorbing states, and whose elements are the number of the full states corresponding to each.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Haldane, J. B. S. and Waddington, C. H. (1931) Inbreeding and linkage. *Genetics* **16**, 357–374.

Broman, K. W. (2005) The genomes of recombinant inbred lines. *Genetics* **169**, 1133–1146.

Examples

```
count.absorb()
```

fulltm *Symbolic generation-to-generation transition matrices*

Description

Symbolic generation-to-generation transition matrices for the process of creating multiple strain recombinant inbred lines.

Usage

```
data(fulltm)
```

Format

A list of lists; each list contains the non-zero elements in the rows of the transition matrices.

Source

These were formed using the internal function `get.full.tm.symbolic`, with help from mathematica to simplify the expressions.

See Also

`get.full.tm`, `get.full.tm.symbolic`, `convert.full.tm`

Examples

```
data(fulltm)
fulltm["2sibX2"]]["AA|BB x AA"]
```

genAILped *Generate a pedigree matrix for AIL*

Description

Generate a pedigree matrix, for use with the simulation function `sim.ped`, for advanced intercross lines.

Usage

```
genAILped(ngen=8, npairs=20, sibship.size=10)
```

Arguments

`ngen`	Number of generations of crosses.
`npairs`	Number of mating pairs per generation.
`sibship.size`	Size of sibships in the final generation.

Details

We generate equal numbers of males and females at each generation. Matings are with random male/female pairs, with no attempt to avoid matings between siblings.

Value

The output is a matrix with five columns: `"id"`, `"sex"` (coded as `0`=female, `1`=male), `"mom"`, `"dad"`, and `"generation"`.

Author(s)

Karl W Broman, <broman@wisc.edu>

See Also

`sim.ped`

Examples

```
ailped <- genAILped(ngen=5, npairs=5, sibship.size=2)
```

`get.full.tm` *Calculate the transition matrix for the formation of RILs*

Description

Calculate the full generation-to-generation transition matrix for the formation of RILs.

Usage

```
get.full.tm(r, coinc=1, n.strains=c("2","4"), type=c("selfing","sibmating"),
         chrtype=c("A","X"), n.loci=c("2","3"))
```

Arguments

`r`	Recombination fraction between markers.
`coinc`	The 3-point coincidence (used only when `n.loci=3`).
`n.strains`	Number of parental strains.
`type`	Method for mating.
`chrtype`	Indicates autosome or X chromosome (when `type=sibmating` only).
`n.loci`	Number of loci.

Value

The transition matrix as a list whose components are the rows of the matrix. Only the non-zero elements of each row are returned.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Haldane, J. B. S. and Waddington, C. H. (1931) Inbreeding and linkage. *Genetics* **16**, 357–374.

Broman, K. W. (2005) The genomes of recombinant inbred lines. *Genetics* **169**, 1133–1146.

See Also

`get.full.tm.symbolic`, `fulltm`, `convert.full.tm`

Examples

```
fulltm2self2 <- get.full.tm(0.1)
```

`get.full.tm.symbolic` *Calculate symbolic form of transition matrix for the formation of RILs*

Description

Calculate the full generation-to-generation transition matrix for the formation of RILs, in symbolic form.

Usage

```
get.full.tm.symbolic(n.strains=c("2","4"), type=c("selfing","sibmating"),
         chrtype=c("A","X"), n.loci=c("2","3"))
```

Arguments

`n.strains`	Number of parental strains.
`type`	Method for mating.
`chrtype`	Indicates autosome or X chromosome (when `type=sibmating` only).
`n.loci`	Number of loci.

Value

The transition matrix as a list whose components are the rows of the matrix. Only the non-zero elements of each row are returned, and they are returned as character strings.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Haldane, J. B. S. and Waddington, C. H. (1931) Inbreeding and linkage. *Genetics* **16**, 357–374.

Broman, K. W., Rowe, L. B., Churchill, G. A. and Paigen, K. (2002) Crossover interference in the mouse. *Genetics* **160**, 1123–1131.

See Also

`get.full.tm`, `convert.full.tm`

Examples

```
fulltm2self2 <- get.full.tm.symbolic()
```

`get.ril.coinc` *Calculate 3-point coincidence on an RIL chromosome*

Description

Calculate (numerically) the quantity analogous to a three-point coincidence on a 2- or 4-way RIL chromosome.

Usage

```
get.ril.coinc(r, coinc=1, n.strains=c("2","4"), type=c("selfing","sibmating"),
         chrtype=c("A","X"), verbose=TRUE)
```

Arguments

`r`	Recombination fraction between markers.
`coinc`	The 3-point coincidence (used only when `n.loci=3`).
`n.strains`	Number of parental strains.
`type`	Method for mating.
`chrtype`	Indicates autosome or X chromosome (when `type=sibmating` only).
`verbose`	If TRUE, print stuff as things progress.

Value

A single number.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Broman, K. W. (2005) The genomes of recombinant inbred lines. *Genetics* **169**, 1133–1146.

See Also

get.ril.prob

Examples

```
get.ril.coinc(0.01)
```

`get.ril.prob` *Calculate the haplotype probabilities for an RIL chromosome*

Description

Calculate (numerically) the haplotype probabilities on a 2- or 4-way RIL chromosome.

Usage

```
get.ril.prob(r, coinc=1, n.strains=c("2","4"), type=c("selfing","sibmating"),
         chrtype=c("A","X"), n.loci=c("2","3"), verbose=TRUE)
```

Arguments

`r`	Recombination fraction between markers.
`coinc`	The 3-point coincidence (used only when `n.loci=3`).
`n.strains`	Number of parental strains.
`type`	Method for mating.
`chrtype`	Indicates autosome or X chromosome (when `type=sibmating` only).
`n.loci`	Number of loci.
`verbose`	If TRUE, print stuff as things progress.

Value

A vector of probabilities, with absorbing states in their reduced form.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Haldane, J. B. S. and Waddington, C. H. (1931) Inbreeding and linkage. *Genetics* **16**, 357–374.

Broman, K. W. (2005) The genomes of recombinant inbred lines. *Genetics* **169**, 1133–1146.

See Also

get.ril.coinc

`get.start` *Determine the initial state in the generation of RILs*

Description

Determine the initial state in the Markov chain to generate 2- or 4-way recombination inbred lines.

Usage

```
get.start(n.strains=c("2","4"),type=c("selfing","sibmating"),
         chrtype=c("A","X"),n.loci=c("2","3"))
```

Arguments

`n.strains`	Number of parental strains.
`type`	Method for mating.
`chrtype`	Indicates autosome or X chromosome (when `type=sibmating` only).
`n.loci`	Number of loci.

Value

A character string.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Haldane, J. B. S. and Waddington, C. H. (1931) Inbreeding and linkage. *Genetics* **16**, 357–374.

Broman, K. W. (2005) The genomes of recombinant inbred lines. *Genetics* **169**, 1133–1146.

See Also

`count.absorb`

Examples

```
get.start()
```

imf.gam *Inverse map function for gamma renewal model*

Description

Calculates the genetic length (in cM) corresponding to a particular recombination fraction for the gamma renewal model for recombination at meiosis.

Usage

```
imf.gam(r, nu=1, tol=1e-12)
```

Arguments

r	Recombination fraction, potentially a vector.
nu	Interference parameter (nu = m+1 in the chi-square model; nu=0 is NI).
tol	Tolerance for doing numerical integration and finding root.

Details

Uses uniroot to solve the r = mf.gam(d, nu).

Value

Vector of interval lengths (in cM), of same length as the input r.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Zhao, H. and Speed, T. P. (1996) On genetic map functions. *Genetics* **142**, 1369–1377.

Broman, K. W., Rowe, L. B., Churchill, G. A. and Paigen, K. (2002) Crossover interference in the mouse. *Genetics* **160**, 1123–1131.

See Also

mf.gam, coinc.gam

Examples

```
imf.gam(seq(0,0.49,by=0.01), 11.3)
```

lookup *Look-up tables for multi-locus parental types*

Description

Look-up tables for multi-locus parental types in the generation of 2-way or 4-way recombinant inbred lines by selfing, sibling mating with the X chromosome, or sibling mating with an autosome, and for cases of two or three loci. The tables connect all possible parental types to the prototypical versions after collapsing according to various symmetries.

Usage

```
data(lookup)
```

Format

A list of vectors, each corresponding to a different set of conditions (for example, 2-way selfing at 3 points). Each vector has names corresponding to all possible parental types and with entries being the corresponding prototypes after collapsing according to various symmetries.

Source

These were formed using the internal function `gtypes`.

References

Broman, K. W. (2005) The genomes of recombinant inbred lines. *Genetics* **169**, 1133–1146.

Examples

```
data(lookup)
length(lookup[["2self3"]])
length(unique(lookup[["2self3"]]))
```

mf.gam *Map function for gamma renewal model*

Description

Calculates the recombination fraction corresponding to a particular cM interval length for the gamma renewal model for recombination at meiosis.

Usage

```
mf.gam(d, nu=1, tol=1e-12)
```

Arguments

d	Interval length (in cM), potentially a vector.
nu	Interference parameter (nu = m+1 in the chi-square model; nu=0 is NI).
tol	Tolerance for doing numerical integration.

Details

Uses the function integrate to do numerical integration.

Value

Vector of recombination fractions, of same length as the input d.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Zhao, H. and Speed, T. P. (1996) On genetic map functions. *Genetics* **142**, 1369–1377.

Broman, K. W., Rowe, L. B., Churchill, G. A. and Paigen, K. (2002) Crossover interference in the mouse. *Genetics* **160**, 1123–1131.

See Also

imf.gam, coinc.gam

Examples

```
mf.gam(0:25, 11.3)
```

---

mouseL *Estimated genetic lengths of mouse chromosomes*

Description

Estimated genetic lengths (in cM) of mouse chromosomes.

Usage

```
data(mouseL)
```

Format

An vector of length 20, each element being the length (in cM) of the corresponding chromosome.

Source

Mouse Genome Informatics, http://www.informatics.jax.org/mgihome/other/mouse_facts1.shtml

See Also

sim.ri

Examples

```
data(mouseL)
ri <- sim.ri(mouseL, type="selfing", n.str="8")
```

---

mtm *Symbolic transition matrices for meiosis*

Description

Transition matrices for meiosis: the probabilities of the possible haplotypes for a meiotic product from a given parental type. These are symbolic versions.

Usage

```
data(mtm)
```

Format

A list of lists. Each list corresponds to a different condition (for example, 2 alleles for the X chromosome at 3 points). Each list contains a set of vectors, one for each possible parental type, containing a vector of character strings specifying the probabilities of each possible meiotic haplotype.

Source

These were formed using the internal functions `write.meiosis.tm.symbolic` and `read.meiosis.tm.symbolic`, using mathematica to simplify the expressions.

Examples

```
data(mtm)
sapply(mtm[["2A3"]],length)
```

ricalcversion *Installed version of R/ricalc*

Description

Print the version number of the currently installed version of R/ricalc.

Usage

```
ricalcversion()
```

Value

A character string with the version number of the currently installed version of R/ricalc.

Author(s)

Karl W Broman, <broman@wisc.edu>

Examples

```
ricalcversion()
```

sim.ped *Simulate a general pedigree*

Description

Simulates alleles for a general pedigree, using the chi-square model for interference at meiosis.

Usage

```
sim.ped(pedigree, L, sexsp=1, xchr=FALSE, m=10, obligate.chiasma=FALSE)
```

Arguments

pedigree	A matrix with rows corresponding to individuals and columns being individual identifiers ("id"), sex (0/1 or F/M), mom, and dad. Parents must precede their offspring.
L	Length of chromosome in cM.
sexsp	Female:male recombination rate (must be > 0)
xchr	If TRUE, simulate the X chromosome.
m	The interference parameter (a non-negative integer). m=0 corresponds to no interference; m>0 corresponds to positive crossover interference.
obligate.chiasma	Indicates whether there is an obligate chiasma on the four-strand bundle.

Details

Meiosis is simulated by the chi-square model (see Zhao et al. 1995), with the possibility of requiring an obligate chiasma on the four-strand bundle at meiosis.

The `pedigree` matrix must have columns named `"id"`, `"sex"`, `"mom"`/`"dam"` and `"dad"`/`"sire"`. Other columns are ignored. Founders should have 0 in the mom and dad columns; others should have the identifiers for the mom and dad. Each individual must have either both parents in the matrix or neither.

Value

The output is a list with length equal to the number of rows in `pedigree`. Each component is itself a list with two components, the maternal and paternal chromosomes. These are matrices with two rows: The first row consists of the locations of exchanges (0 and `L` are included); the second row contains the allele (coded 1, 2, . . . , no. founders) in the interval to the left.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Zhao, H., Speed, T. P. and McPeek, M. S. (1995) Statistical analysis of crossover interference using the chi-square model. *Genetics* **139**, 1045–1056.

See Also

`sim.ri`, `genAILped`, `convert2gen`

Examples

```
ailped <- genAILped(ngen=5, npairs=5, sibship.size=2)
aildat <- sim.ped(ailped, 100)
```

---

sim.ri *Simulate a recombinant inbred line*

Description

Simulates alleles for a multiple-strain RI line, using the chi-square model for interference at meiosis, performing inbreeding by selfing or sibling mating, until complete fixation has occurred.

Usage

```
sim.ri(L, sexsp=1, type=c("selfing","sibmating"),
       n.strains=c("2","4","8"),
       xchr=FALSE, m=10, obligate.chiasma=FALSE)
```

Arguments

L	Length of chromosome(s) in cM; either a vector or a single number.
sexsp	Female:male recombination rate (must be > 0)
type	Indicates whether the inbreeding is by selfing or by sibling mating.
n.strains	Number of initial parental strains.
xchr	If length(L)==1 and type="sibmating", this indicates to simulate the X chromosome. If length(L) > 1, the chromosomes named "X" or "x" are taken to be the X chromosome and all others are assumed to be autosomes.
m	The interference parameter (a non-negative integer). m=0 corresponds to no interference; m>0 corresponds to positive crossover interference.
obligate.chiasma	Indicates whether there is an obligate chiasma on the four-strand bundle.

Details

Meiosis is simulated by the chi-square model (see Zhao et al. 1995), with the possibility of requiring an obligate chiasma on the four-strand bundle at meiosis.

Inbreeding is performed until complete fixation. Of course, we assume no mutation and no selection.

Value

If length(L)==1, the output is a two-row matrix whose first row consists of the locations of exchanges along the recombinant inbred chromosome (0 and L are included). The second row contains the allele (coded 1, 2, ..., 8) in the interval to the left. Attributes "prop.het" and "num.het" are included which contain the proportion of the genome that is not yet fixed and the number of segments of such heterozygosity, respectively, after each generation of inbreeding. The attribute "nubreak" gives the number of unique breakpoints at each generation.

If length(L)>1, the output is a list of length length(L), with each component being a two-row matrix, as above. The overall list also has attributes "prop.het" and "num.het", specifying the proportion of the entire genome that has not yet been fixed and the number of segments of such heterozygosity, respectively, after each generation of inbreeding.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Haldane, J. B. S. and Waddington, C. H. (1931) Inbreeding and linkage. *Genetics* **16**, 357–374.

Broman, K. W. (2005) The genomes of recombinant inbred lines. *Genetics* **169**, 1133–1146.

Broman, K. W., Rowe, L. B., Churchill, G. A. and Paigen, K. (2002) Crossover interference in the mouse. *Genetics* **160**, 1123–1131.

Zhao, H., Speed, T. P. and McPeek, M. S. (1995) Statistical analysis of crossover interference using the chi-square model. *Genetics* **139**, 1045–1056.

See Also

sim.ri

Examples

```
data(mouseL)
ri <- sim.ri(mouseL, type="selfing", n.str="8")
```

---

threept *Three-point probabilities for four-way RILs by sibling mating*

Description

Three-point haplotype probabilities for four-way recombinant inbred lines formed by sibling mating. A and X indicate autosome and the X chromosome. I and NI indicate strong crossover interference (gamma model with $\nu = 11.3$) and no interference.

Usage

```
data(threept.AI)
data(threept.ANI)
data(threept.XI)
data(threept.XNI)
```

Format

Each is matrix; the first two columns indicate the recombination fraction between markers (at meiosis) and the three-point coincidence (at meiosis). The remaining columns are the haplotype probabilities for the fixed RIL chromosome. These are for the reduced set of states, taking account of many symmetries.

Use count.absorb to get the numbers of full states in each reduced state.

Source

These were calculated using perl programs included with this package.

References

Haldane, J. B. S. and Waddington, C. H. (1931) Inbreeding and linkage. *Genetics* **16**, 357–374.

Broman, K. W. (2005) The genomes of recombinant inbred lines. *Genetics* **169**, 1133–1146.

See Also

get.ril.prob

Examples

```
data(threept.AI)
plot(threept.AI[,c(1,3)])
```

Index

∗ **datagen**
convert2gen, 5
count.absorb, 6
genAILped, 7
get.full.tm, 8
get.full.tm.symbolic, 9
get.ril.coinc, 10
get.ril.prob, 11
get.start, 12
sim.ped, 17
sim.ri, 18
∗ **datasets**
fulltm, 7
lookup, 14
mouseL, 15
mtm, 16
threept, 20
∗ **manip**
convert.full.tm, 4
∗ **models**
ricalc-package, 2
∗ **print**
ricalcversion, 17
∗ **utilities**
coinc.gam, 3
imf.gam, 13
mf.gam, 14

coinc.gam, *2*, 3, *13*, *15*
convert.full.tm, *3*, 4, *7*, *9*, *10*
convert2gen, 5, *18*
count.absorb, *3*, 6, *12*, *20*

fulltm, *3*, 7, *9*

genAILped, *6*, 7, *18*
get.full.tm, *3*, *5*, *7*, 8, *10*
get.full.tm.symbolic, *3*, *5*, *7*, 9, *9*
get.ril.coinc, *3*, 10, *11*
get.ril.prob, *3*, 11, *11*, *20*
get.start, *3*, 12

imf.gam, *2*, *4*, 13, *15*
integrate, *15*

lookup, *3*, 14

mf.gam, *2*, *4*, *13*, 14
mouseL, *2*, 15
mtm, *3*, 16

ricalc-package, 2
ricalcversion, 17

sim.ped, *5–8*, 17
sim.ri, *3*, *16*, 18, *18*, *20*

threept, 20
threept.AI, *2*
threept.ANI, *2*
threept.XI, *2*
threept.XNI, *2*

uniroot, *13*