Package 'xoi'

October 14, 2022

Version 0.70

Date 2022-01-21

Depends R (>= 2.10.1), qtl

Title Tools for Analyzing Crossover Interference

Author Karl W Broman [aut, cre] (<https://orcid.org/0000-0002-4914-6671>), Il-Youp Kwak [ctb] (<https://orcid.org/0000-0002-7117-7669>)

Maintainer Karl W Broman <broman@wisc.edu>

Description Analysis of crossover interference in experimental crosses, particularly regarding the gamma model. See, for example, Broman and Weber (2000) <doi:10.1086/302923>.

Imports stats, utils

Suggests devtools, roxygen2, testthat

License GPL-3

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

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

RoxygenNote 7.1.0

Encoding UTF-8

ByteCompile true

LazyData true

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-01-21 17:30:03 UTC

R topics documented:

`bssbsb` *BSS/BSB backcross data*

Description

Data from two densely genotyped backcrosses.

Format

An object of class `cross`. See `qtl::read.cross()` for details.

Details

There are 94 individuals from each of two interspecific backcross: (C57BL/6J × *M. spretus*) × C57BL/6J and (C57BL/6J × SPRET/Ei) × SPRET/Ei. They were typed on 1372 and 4913 genetic markers, respectively, with 904 markers in common.

These data are from September, 2000. Updated data are available.

Source

Lucy Rowe, Jackson Laboratory

References

Rowe, L. B., Nadeau, J. H., Turner, R., Frankel, W. N., Letts, V. A., Eppig, J. T., Ko, M. S., Thurston, S. J. and Birkenmeier, E. H. (1994) Maps from two interspecific backcross DNA panels available as a community genetic mapping resource. *Mamm. Genome* **5**, 253–274.

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

Examples

```
data(bssbsb)
summary(bssbsb)
## Not run: plot(bssbsb)
```

`chiasma` *Estimate chiasma distribution from crossover counts*

Description

Fit several models, with an assumption of no chromatid interference, to crossover count data to obtain fitted distributions of the number of chiasmata.

Usage

```
chiasma(
  xo,
  max.chiasma = max(xo) * 2 + 5,
  n.iter = 10000,
  tol = 0.000001,
  verbose = FALSE
)
```

Arguments

`xo`	Vector of non-negative integers; the number of crossovers in a set of meiotic products.
`max.chiasma`	Maximum number of chiasmata to allow.
`n.iter`	Maximum number of iterations in the EM algorithm.
`tol`	Tolerance for convergence of the EM algorithm.
`verbose`	If TRUE, print number of interations for each of the 4 models at the end.

Details

See Broman and Weber (2000) for details of the method.

We use R's `stats::integrate()` function for numerical integrals, `stats::optimize()` for optimizing the likelihood, and `stats::uniroot()` for identifying the endpoints of the likelihood support interval.

Value

A list with three components.

First, a matrix containing the observed distribution of the numbers of crossovers, followed by the fitted distributions under the Poisson model, the truncated Poisson model (assuming an obligate chiasma), the obligate chiasma model, and the freely varying model. In all cases we assume no chromatid interference.

Second, a matrix containing the estimated distributions of the number of chiasmata on the four-strand bundle for the above four models.

Third, the estimated average number of crossovers under the Poisson and truncated Poisson models.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Broman, K. W. and Weber, J. L. (2000) Characterization of human crossover interference. *Am. J. Hum. Genet.* **66**, 1911–1926.

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

Yu, K. and Feinbold, E. (2001) Estimating the frequency distribution of crossovers during meiosis from recombination data. *Biometrics* **57**, 427–434.

See Also

`fitGamma()`, `qtl::fitstahl()`, `countxo()`

Examples

```
data(bssbsb)

# estimated number of crossovers on chr 1
nxo <- countxo(bssbsb, chr=1)

# estimate chiasma distribution
## Not run: chiasma(nxo)
```

`coincidence` *Estimate coincidence function*

Description

Estimate coincidence function for a chromosome.

Usage

```
coincidence(cross, chr = NULL, window = 5, ncalc = 500)
```

Arguments

`cross`	Cross object; must be a backcross. See `qtl::read.cross()` for format details.
`chr`	Chromosome to consider (only one is allowed). If NULL, the first chromosome is considered.
`window`	Window size
`ncalc`	Total number of points for calculations.

Value

Data frame with columns `distance` and `coincidence`. The input argument `window` is kept as an attribute.

Author(s)

Il youp Kwak

See Also

`intensity()`, `est.coi()`

Examples

```
map1 <- sim.map(103, n.mar=104, anchor=TRUE, include.x=FALSE, eq=TRUE)
x <- sim.cross(map1, n.ind=2000, m=6, type="bc")

out <- coincidence(x, ncalc=101)
plot(out, type="l", lwd=2, ylim=c(0, max(out[,2])))
```

convertxoloc *Convert format of crossover locations data*

Description

Convert the format of data on crossover locations to that needed for the function `fitGamma`.

Usage

```
convertxoloc(breaks)
```

Arguments

breaks	A list of crossover locations, as output by `find.breaks()` or `simStahl()`.

Value

A data frame with two columns: the inter-crossover and crossover-to chromosome end differences (`"distance"`) and indicators of censoring type (`"censor"`), with 0 = distance between crossovers, 1=start of chromosome to first crossover, 2 = crossover to end of chromosome, and 3 = whole chromosome.

Author(s)

Karl W Broman, <broman@wisc.edu>

See Also

`find.breaks()`, `fitGamma()`, `simStahl()`

Examples

```
data(bssbsb)

# crossover locations on chromosome 1
xoloc1 <- convertxoloc(find.breaks(bssbsb, chr=1))

# crossover locations on all chromosomes
xoloc <- convertxoloc(find.breaks(bssbsb))
```

countxo *Estimate number of crossovers*

Description

Estimate the number of crossovers in each meiosis in a backcross.

Usage

```
countxo(cross, chr = NULL)
```

Arguments

cross	An object of class cross. (This must be a backcross.) See qtl::read.cross() for details.
chr	Optional set of chromosomes across which to count crossovers. If NULL, the total number of crossovers, genome-wide, is counted.

Details

This works only a backcross. We use the internal function (within R/qtl) locate.xo.

Value

A vector with the estimated number of crossovers for each individual.

Author(s)

Karl W Broman, <broman@wisc.edu>

See Also

find.breaks()

Examples

```
data(bssbsb)

# estimated number of crossovers on chr 1
nxo <- countxo(bssbsb, chr=1)

# estimated number of crossovers genome-wide
nxo <- countxo(bssbsb)
```

`distance.given.two` *Distance between crossovers given there are two*

Description

Calculates the density of the distance between the crossovers on a meiotic product, given that there are precisely two crossovers, for the gamma model.

Usage

```
distance.given.two(
  nu,
  L = 103,
  x = NULL,
  n = 400,
  max.conv = 25,
  integr.tol = 0.00000001,
  max.subd = 1000,
  min.subd = 10
)
```

Arguments

`nu`	The interference parameter in the gamma model.
`L`	The length of the chromsome in cM.
`x`	If specified, points at which to calculate the density.
`n`	Number of points at which to calculate the density. The points will be evenly distributed between 0 and `L`. Ignored if `x` is specified.
`max.conv`	Maximum limit for summation in the convolutions to get inter-crossover distance distribution from the inter-chiasma distance distributions. This should be greater than the maximum number of chiasmata on the 4-strand bundle.
`integr.tol`	Tolerance for convergence of numerical integration.
`max.subd`	Maximum number of subdivisions in numerical integration.
`min.subd`	Minimum number of subdivisions in numerical integration.

Details

Let $f(x;\nu)$ denote the density of a gamma random variable with parameters shape=ν and rate=2ν, and let $f_k(x;\nu)$ denote the density of a gamma random variable with parameters shape=$k\nu$ and rate=2ν.

The distribution of the distance from one crossover to the next is $f^*(x;\nu) = \sum_{k=1}^{\infty} f_k(x;\nu)/2^k$.

The distribution of the distance from the start of the chromosome to the first crossover is $g^*(x;\nu) = 1 - F^*(x;\nu)$ where F^* is the cdf of f^*.

We calculate the distribution of the distance between crossovers on a product with two crossovers by first calculating the joint distribution of the location of the two crossovers, given that they both occur before L and the third occurs after L, and then integrating out the location of the first crossover.

Value

A data frame with two columns: `x` is the distance (between 0 and `L`, in cM) at which the density was calculated and `f` is the density.

Warning

We sometimes have difficulty with the numerical integrals. You may need to use large `min.subd` **(e.g. 25) to get accurate results.**

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Broman, K. W. and Weber, J. L. (2000) Characterization of human crossover interference. *Am. J. Hum. Genet.* **66**, 1911–1926.

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

McPeek, M. S. and Speed, T. P. (1995) Modeling interference in genetic recombination. *Genetics* **139**, 1031–1044.

See Also

`location.given.one()`, `first.given.two()`, `joint.given.two()`, `ioden()`, `firstden()`, `xoprob()`, `gammacoi()`

Examples

```
f1 <- distance.given.two(1, L=200, n=101)
plot(f1, type="l", lwd=2, las=1,
     ylim=c(0,0.0122), yaxs="i", xaxs="i", xlim=c(0,200))

f2 <- distance.given.two(2.6, L=200, n=101)
lines(f2, col="blue", lwd=2)

## Not run:
f3 <- distance.given.two(4.3, L=200, n=101)
lines(f3, col="red", lwd=2)

f4 <- distance.given.two(7.6, L=200, n=101)
lines(f4, col="green", lwd=2)

## End(Not run)
```

`est.coi` *Estimate the coincidence function*

Description

Estimate the coincidence function from backcross data.

Usage

```
est.coi(
  cross,
  chr = NULL,
  pos = NULL,
  window = 0,
  fill.method = c("imp", "argmax"),
  error.prob = 0.0000000001,
  map.function = c("haldane", "kosambi", "c-f", "morgan")
)
```

Arguments

`cross`	Cross object; must be a backcross. See `qtl::read.cross()` for format details.
`chr`	Chromosome to consider (only one is allowed). If NULL, the first chromosome is considered.
`pos`	If provided, these are used as the marker positions. (This could be useful if you want to do things with respect to physical distance.)
`window`	Window size used to smooth the estimates.
`fill.method`	Method used to impute missing data.
`error.prob`	Genotyping error probability used in imputation of missing data.
`map.function`	Map function used in imputation of missing data.

Details

The coincidence function is the probability of a recombination event in both of two intervals, divided by the product of the two recombination fractions. We estimate this as a function of the distance between the two intervals.

Note that we first call `qtl::fill.geno()` to impute any missing genotype data.

Value

A data.frame containing the distance between intervals and the corresponding estimate of the coincidence. There are actually two columns of estimates of the coincidence. In the first estimate, we take a running mean of each of the numerator and denominator and then divide. In the second estimate, we first take a ratio and then take a running mean.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

McPeek, M. S. and Speed, T. P. (1995) Modeling interference in genetic recombination. *Genetics* **139**, 1031–1044.

See Also

gammacoi(), stahlcoi(), kfunc()

Examples

```
map1 <- sim.map(103, n.mar=104, anchor=TRUE, include.x=FALSE, eq=TRUE)
x <- sim.cross(map1, n.ind=2000, m=6, type="bc")

out <- est.coi(x, window=5)
plot(coi1 ~ d, data=out, type="l", lwd=2, col="blue")
lines(coi2 ~ d, data=out, lwd=2, col="green")
lines(gammacoi(7), lwd=2, col="red", lty=2)
```

---

est.coi.um *Estimate the coincidence as a function of micron distance*

Description

Estimate the coincidence as a function of micron distance, with data on XO locations in microns plus SC length in microns.

Usage

```
est.coi.um(
  xoloc,
  sclength,
  centromeres = NULL,
  group = NULL,
  intwindow = 0.05,
  coiwindow = NULL,
  intloc = NULL,
  coiloc = NULL
)
```

Arguments

xoloc	list of crossover locations (in microns) for each of several oocytes or spermatocytes.
sclength	vector of SC lengths (in microns).
centromeres	vector of centromere locations (in microns). If NULL, taken to be sclength/2.
group	nominal vector of groups; the intensity function of the crossover process will be estimated separately for each group, but a joint coincidence function will be estimated.
intwindow	Window size used to smooth the estimated intensity function.
coiwindow	Window size used to smooth the estimated coincidence function.
intloc	Locations at which to estimate the intensity function, in the interval [0,1]
coiloc	Values at which the coincidence function is to be estimated, in microns, less than max(sclength)

Details

The coincidence function is the probability of a recombination event in both of two intervals, divided by the product of the two intensity function for the two intervals.

We estimate this as a function of the distance between the two intervals in microns, taking account of varying SC lengths,.

Value

A list containing the estimated coincidence (as a matrix with two columns, micron distance and corresponding estimated coincidence) and the estimated intensity functions (as a matrix with length(group)+1 columns (the locations at which the intensity functions were estimated followed by the group-specific estimates).

Author(s)

Karl W Broman, <broman@wisc.edu>

See Also

gammacoi(), stahlcoi(), kfunc(), est.coi()

Examples

```
# simple example using data simulated with no crossover interference
ncells <- 1000
L <- 2                      # chr lengths in Morgans (constant here)
nchi <- rpois(ncells, 2*L)  # number of chiasmata
xoloc <- lapply(nchi, function(a) runif(a, 0, L)) # chi locations
coi <- est.coi.um(xoloc, rep(L, ncells))

# plot estimated coincidence and intensity
#    (intensity is after scaling chromosome to length 1)
```

```
par(mfrow=c(2,1), las=1)
plot(coi$coincidence, type="l", lwd=2, ylim=c(0, max(coi$coincidence[,2])))
plot(coi$intensity, type="l", lwd=2, ylim=c(0, max(coi$intensity[,2])))
```

`est.recrate` *Estimate recombination rate*

Description

Obtain a smoothed estimate of the recombination rate along a chromosome, using the cM and Mbp position of markers.

Usage

```
est.recrate(genmap, phymap, pos = NULL, window = 5)
```

Arguments

`genmap`	Vector of cM positions of markers, or a list of such vectors.
`phymap`	Vector of Mbp positions of markers, or a list of such vectors; same length as `genmap`.
`pos`	Vector of positions at which the recombination rate should be estimated, or a list of such vectors. If NULL, we use the physical marker positions plus a grid with 4 positions per Mbp.
`window`	Length of sliding window (in Mbp).

Details

We assume constant recombination rate within each marker interval.

Value

A data.frame containing the positions and estimate recombination rates.

Author(s)

Karl W Broman, <broman@wisc.edu>

See Also

`est.coi()`, `intensity()`

Examples

```
# create equally-spaced map
pmap <- sim.map(100, n.mar=51, anchor=TRUE, include.x=FALSE, eq.spacing=TRUE)

# simulate cross
x <- sim.cross(pmap, type="bc", n.ind=501)

# estimate map for that cross
emap <- est.map(x)

# empirical estimate of recombination rate
rr <- est.recrate(emap[[1]], pmap[[1]], window=5)
plot(rr, type="l", lwd=2)
```

---

find.breaks *Estimate crossover locations*

Description

Estimate the locations of crossovers in a backcross.

Usage

```
find.breaks(cross, chr = NULL)
```

Arguments

cross	An object of class cross. (This must be a backcross, RIL, or intercross.) See qtl::read.cross() for details.
chr	Optional set of chromosomes on which to look for crossovers. If NULL, all chromosomes are considered.

Details

This works only a backcross, RIL, or intercross. We use the function qtl::locateXO() in R/qtl. Crossovers are estimated to be at the midpoint of the interval between the nearest flanking typed markers.

Value

If only one chromosome is considered, this is a list with one component for each individual. If multiple chromosomes were considered, this is a list with one element for each chromosome, each of which is a list with one element for each individual, as above.

For backcrosses and RIL, the componenets for the individuals are numeric(0) if there were no crossovers or a vector giving the crossover locations. The length of the chromosome (in cM) is saved as an attribute. (Note that the format is the same as the output of simStahl().)

For an intercross, the components for the individuals are themselves lists with all possible allocations of the crossovers to the two meiotic products; each component of this list is itself a list with two components, corresponding to the two meiotic products.

Author(s)

Karl W Broman, <broman@wisc.edu>

See Also

convertxoloc(), fitGamma(), simStahl()

Examples

```
data(bssbsb)

# crossover locations on chromosome 1
xoloc1 <- find.breaks(bssbsb, chr=1)

# crossover locations on all chromosomes
xoloc <- find.breaks(bssbsb)
```

---

first.given.two *Location of first crossover given there are two*

Description

Calculates the density of the location of the first crossover on a random meiotic product, given that there are precisely two crossovers, for the gamma model.

Usage

```
first.given.two(
  nu,
  L = 103,
  x = NULL,
  n = 400,
  max.conv = 25,
  integr.tol = 0.00000001,
  max.subd = 1000,
  min.subd = 10
)
```

Arguments

nu	The interference parameter in the gamma model.
L	The length of the chromsome in cM.
x	If specified, points at which to calculate the density.
n	Number of points at which to calculate the density. The points will be evenly distributed between 0 and L. Ignored if x is specified.
max.conv	Maximum limit for summation in the convolutions to get inter-crossover distance distribution from the inter-chiasma distance distributions. This should be greater than the maximum number of chiasmata on the 4-strand bundle.
integr.tol	Tolerance for convergence of numerical integration.
max.subd	Maximum number of subdivisions in numerical integration.
min.subd	Minimum number of subdivisions in numerical integration.

Details

Let $f(x;\nu)$ denote the density of a gamma random variable with parameters shape=ν and rate=2ν, and let $f_k(x;\nu)$ denote the density of a gamma random variable with parameters shape=$k\nu$ and rate=2ν.

The distribution of the distance from one crossover to the next is $f^*(x;\nu) = \sum_{k=1}^{\infty} f_k(x;\nu)/2^k$.

The distribution of the distance from the start of the chromosome to the first crossover is $g^*(x;\nu) = 1 - F^*(x;\nu)$ where F^* is the cdf of f^*.

We calculate the distribution of the location of the first crossover in a product with two crossovers by calculating the joint distribution of the location of the two crossovers, given that they both occur before L and the third occurs after L, and then integrating out the location of the second crossover.

Value

A data frame with two columns: x is the location (between 0 and L, in cM) at which the density was calculated and f is the density.

Warning

We sometimes have difficulty with the numerical integrals. You may need to use large min.subd **(e.g. 25) to get accurate results.**

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Broman, K. W. and Weber, J. L. (2000) Characterization of human crossover interference. *Am. J. Hum. Genet.* **66**, 1911–1926.

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

McPeek, M. S. and Speed, T. P. (1995) Modeling interference in genetic recombination. *Genetics* **139**, 1031–1044.

See Also

location.given.one(), distance.given.two(), joint.given.two(), ioden(), firstden(), xoprob(), gammacoi()

Examples

```
f1 <- first.given.two(1, L=200, n=101)
plot(f1, type="l", lwd=2, las=1,
     ylim=c(0,0.011), yaxs="i", xaxs="i", xlim=c(0,200))

f2 <- first.given.two(2.6, L=200, n=101)
lines(f2, col="blue", lwd=2)

## Not run:
f3 <- first.given.two(4.3, L=200, n=101)
lines(f3, col="red", lwd=2)

f4 <- first.given.two(7.6, L=200, n=101)
lines(f4, col="green", lwd=2)

## End(Not run)
```

---

firstden *Distance to a crossover*

Description

Calculates the density of the distance from an arbitrary point to the next crossover, for the gamma model.

Usage

```
firstden(nu, L = 103, x = NULL, n = 400, max.conv = 25)
```

Arguments

nu	The interference parameter in the gamma model.
L	Maximal distance (in cM) at which to calculate the density. Ignored if x is specified.
x	If specified, points at which to calculate the density.
n	Number of points at which to calculate the density. The points will be evenly distributed between 0 and L. Ignored if x is specified.
max.conv	Maximum limit for summation in the convolutions to get inter-crossover distance distribution from the inter-chiasma distance distributions. This should be greater than the maximum number of chiasmata on the 4-strand bundle.

Details

Let $f(x;\nu)$ denote the density of a gamma random variable with parameters shape=ν and rate=2ν, and let $f_k(x;\nu)$ denote the density of a gamma random variable with parameters shape=$k\nu$ and rate=2ν.

The distribution of the distance from one crossover to the next is $f^*(x;\nu) = \sum_{k=1}^{\infty} f_k(x;\nu)/2^k$.

The distribution of the distance from an arbitrary point to the first crossover is $g^*(x;\nu) = 1 - F^*(x;\nu)$ where F^* is the cdf of f^*.

Value

A data frame with two columns: `x` is the distance (between 0 and `L`, in cM) at which the density was calculated and `f` is the density.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Broman, K. W. and Weber, J. L. (2000) Characterization of human crossover interference. *Am. J. Hum. Genet.* **66**, 1911–1926.

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

McPeek, M. S. and Speed, T. P. (1995) Modeling interference in genetic recombination. *Genetics* **139**, 1031–1044.

See Also

`location.given.one()`, `first.given.two()`, `distance.given.two()`, `joint.given.two()`, `ioden()`, `xoprob()`, `gammacoi()`

Examples

```
f1 <- firstden(1, L=200, n=201)
plot(f1, type="l", lwd=2, las=1,
     ylim=c(0,0.012), yaxs="i", xaxs="i", xlim=c(0,200))

f2 <- firstden(2.6, L=200, n=201)
lines(f2, col="blue", lwd=2)

f3 <- firstden(4.3, L=200, n=201)
lines(f3, col="red", lwd=2)

f4 <- firstden(7.6, L=200, n=201)
lines(f4, col="green", lwd=2)
```

`fitGamma` *Fit Gamma model*

Description

Fit the gamma model for crossover interference to data on crossover locations.

Usage

```
fitGamma(
  d,
  censor = NULL,
  nu = NULL,
  lo = NULL,
  hi = NULL,
  se = FALSE,
  supint = FALSE,
  rescale = FALSE,
  drop = 1.5,
  tol = 0.00001,
  maxit = 1000,
  max.conv = 25,
  integr.tol = 0.00000001,
  max.subd = 1000,
  min.subd = 10,
  h = 0.1,
  hstep = 1.5
)
```

Arguments

`d`	A vector of inter-crossover distances in cM. This should include distances from start of chromosome to first crossover, last crossover to end of chromosome, and chromosome length, if there are no crossovers. Alternatively, this may be a matrix with the first column being the distances and second column being the censoring types (`censor`).
`censor`	A vector of the same length as `d`, indicating the censoring type for each distance. `0` = uncensored, `1` = right-censored, `2` = initial crossover on chromosome, `3` = whole chromosome.
`nu`	A vector of interference parameters (ν) at which to calculate the log likelihood. If NULL, `lo` and `hi` must be specified.
`lo`	If `nu` is unspecified, `lo` indicates the lower value of the interval in which to search for the MLE. If `supint=TRUE`, this should be below the lower limit of the support interval.

hi	If nu is unspecified, hi indicates the upper value of the interval in which to search for the MLE. If supint=TRUE, this should be above the upper limit of the support interval.
se	If TRUE and nu was not specified, an estimated SE (based on the second derivative of the log likelihood) is estimated.
supint	If TRUE and nu was not specified, a likelihood support interval is calculated, with drop being the amount to drop in log (base 10).
rescale	If TRUE and nu was specified, re-scale the log likelihoods so that the maximum is at 0.
drop	If supint was specified, this indicates the amount to drop in log (base 10) for the likelihood support interval.
tol	Tolerance for converence to calculate the likelihood, SE, and likelihood support interval.
maxit	Maximum number of iterations in estimating the SE and likelihood support interval.
max.conv	Maximum limit for summation in the convolutions to get inter-crossover distance distribution from the inter-chiasma distance distributions. This should be greater than the maximum number of chiasmata on the 4-strand bundle.
integr.tol	Tolerance for convergence of numerical integration.
max.subd	Maximum number of subdivisions in numerical integration.
min.subd	Minimum number of subdivisions in numerical integration.
h	Step used in estimating the second derivative of the log likelihood.
hstep	factor by which h is decreased in each iteration of the estimation of the second derivative of the log likelihood.

Details

See Broman and Weber (2000) for details of the method.

We use R's `stats::integrate()` function for numerical integrals, `stats::optimize()` for optimizing the likelihood, and `stats::uniroot()` for identifying the endpoints of the likelihood support interval.

Value

If nu is specified, we return a data frame with two columns: nu and the corresponding log (base e) likelihood. If rescale=TRUE, the maximum log likelihood is subtracted off, so that its maximum is at 0.

If lo and hi is specified, the output contains a single row with the MLE of ν and the corresponding log likelihood. If se=TRUE, we also include the estimated SE. If supint=TRUE, we include two additional rows with the lower and upper limits of the likelihood support interval.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Broman, K. W. and Weber, J. L. (2000) Characterization of human crossover interference. *Am. J. Hum. Genet.* **66**, 1911–1926.

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

McPeek, M. S. and Speed, T. P. (1995) Modeling interference in genetic recombination. *Genetics* **139**, 1031–1044.

See Also

qtl::fitstahl()

Examples

```
data(bssbsb)


xodist <- convertxoloc(find.breaks(bssbsb, chr=1))

# plot a rough log likelihood curve
## Not run: out <- fitGamma(xodist, nu=seq(1, 19, by=2))

plot(out, type="l", lwd=2)

# get MLE
## Not run: mle <- fitGamma(xodist, lo=8, hi=12)

mle

abline(v=mle[1], h=mle[2], col="blue", lty=2)

# get MLE and SE
## Not run: mle <- fitGamma(xodist, lo=9.5, hi=10.5, se=TRUE)

mle

# get MLE and 10^1.5 support interval
## Not run: int <- fitGamma(xodist, lo=1, hi=20, supint=TRUE)

int
abline(v=mle[2:3,1], h=mle[2:3,2], col="red", lty=2)
```

`fitStahl` *Fit Stahl model*

Description

Fit the Stahl model for crossover interference to data on crossover locations.

Usage

```
fitStahl(
  xoloc,
  chrlen = NULL,
  nu = c(1, 20),
  p = 0.02,
  max.conv = 25,
  integr.tol = 0.00000001,
  max.subd = 1000,
  min.subd = 10,
  verbose = TRUE,
  ...
)
```

Arguments

`xoloc`	A list of crossover locations (in cM), each component being a vector of locations for a different meiotic product.
`chrlen`	Chromosome length (in cM), either of length 1 or the same length as `xoloc`.
`nu`	Interference parameter (ν). This should be a pair of values to be used as end-points to first do a 1-dimensional optimization with $p = 0$.
`p`	Starting value for the proportion of crossovers from the no interference pathway, for the 2-dimensional optimization.
`max.conv`	Maximum limit for summation in the convolutions to get inter-crossover distance distribution from the inter-chiasma distance distributions. This should be greater than the maximum number of chiasmata on the 4-strand bundle.
`integr.tol`	Tolerance for convergence of numerical integration.
`max.subd`	Maximum number of subdivisions in numerical integration.
`min.subd`	Minimum number of subdivisions in numerical integration.
`verbose`	If TRUE, print tracing information. If "..." includes `control`, this is ignored.
`...`	Further arguments sent to `stats::optim()`.

Details

See Housworth and Stahl (2003) and Broman and Weber (2000) for details of the method.

We first use `stats::optimize()` to find the MLE with the contraint `p=0`, followed by use of `stats::optim()` to do a 2-dimensional optimization for the MLEs of the pair.

Value

A vector with the estimates of ν (interference parameter) and p (proportion of crossovers coming from the no interference pathway), the maximized log likelihood, the estimate of nu with p constrained to be 0, the maximized log likelihood in this case, and the log likelihood ratio for comparing the model with p allowed to vary freely versus contrained to be 0. (Note that it's the natural log of the likelihood ratio, and not twice that.)

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Housworth, E. A. and Stahl, F. W. (2003) Crossover interference in humans. *Am. J. Hum. Genet.* **73**, 188–197.

Broman, K. W. and Weber, J. L. (2000) Characterization of human crossover interference. *Am. J. Hum. Genet.* **66**, 1911–1926.

See Also

`fitGamma()`, `stahlLoglik()`, `simStahl()`

Examples

```
data(bssbsb)


xoloc <- find.breaks(bssbsb, chr=1)
L <- attr(xoloc, "L")

# get MLE (limiting maximum iterations to 10, just for speed in this example)
## Not run: mle <- fitStahl(xoloc, L, nu=c(9, 12), control=list(maxit=10))
```

---

gammacoi *Coincidence function for the gamma model*

Description

Calculates the coincidence function for the gamma model.

Usage

```
gammacoi(nu, L = 103, x = NULL, n = 400, max.conv = 25)
```

Arguments

nu	The interference parameter in the gamma model.
L	Maximal distance (in cM) at which to calculate the density. Ignored if x is specified.
x	If specified, points at which to calculate the density.
n	Number of points at which to calculate the density. The points will be evenly distributed between 0 and L. Ignored if x is specified.
max.conv	Maximum limit for summation in the convolution. This should be greater than the maximum number of chiasmata on the 4-strand bundle.

Details

Let $f(x;\nu)$ denote the density of a gamma random variable with parameters shape=ν and rate=2ν, and let $f_k(x;\nu)$ denote the density of a gamma random variable with parameters shape=$k\nu$ and rate=2ν.

The coincidence function for the gamma model is $C(x;\nu) = \sum_{k=1}^{\infty} f_k(x;\nu)/2$.

Value

A data frame with two columns: x is the distance (between 0 and L, in cM) at which the coicidence was calculated and coincidence.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Broman, K. W. and Weber, J. L. (2000) Characterization of human crossover interference. *Am. J. Hum. Genet.* **66**, 1911–1926.

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

McPeek, M. S. and Speed, T. P. (1995) Modeling interference in genetic recombination. *Genetics* **139**, 1031–1044.

See Also

stahlcoi(), location.given.one(), first.given.two(), distance.given.two(), joint.given.two(), ioden(), firstden(), xoprob()

Examples

```
f1 <- gammacoi(1, L=200)
plot(f1, type="l", lwd=2, las=1,
     ylim=c(0,1.25), yaxs="i", xaxs="i", xlim=c(0,200))
```

```
f2 <- gammacoi(2.6, L=200)
lines(f2, col="blue", lwd=2)

f3 <- gammacoi(4.3, L=200)
lines(f3, col="red", lwd=2)

f4 <- gammacoi(7.6, L=200)
lines(f4, col="green", lwd=2)
```

`intensity` *Estimate intensity function*

Description

Estimate intensity function for a chromosome.

Usage

```
intensity(cross, chr = NULL, window = 2.5, ncalc = 500)
```

Arguments

`cross`	Cross object; must be a backcross. See `qtl::read.cross()` for format details.
`chr`	Chromosome to consider (only one is allowed). If NULL, the first chromosome is considered.
`window`	Window size
`ncalc`	Total number of points for calculations.

Value

Data frame with columns `position` and `intensity`. The input argument `window` is kept as an attribute.

Author(s)

Il youp Kwak

See Also

`coincidence()`

Examples

```
map1 <- sim.map(103, n.mar=104, anchor=TRUE, include.x=FALSE, eq=TRUE)
x <- sim.cross(map1, n.ind=2000, m=6, type="bc")

out <- intensity(x)
plot(out, type="l", lwd=2, ylim=c(0, max(out[,2])))
```

---

ioden *Distance between crossovers*

Description

Calculates the density of the distance from a given crossover to the next crossover, for the gamma model.

Usage

```
ioden(nu, L = 103, x = NULL, n = 400, max.conv = 25)
```

Arguments

nu	The interference parameter in the gamma model.
L	Maximal distance (in cM) at which to calculate the density. Ignored if x is specified.
x	If specified, points at which to calculate the density.
n	Number of points at which to calculate the density. The points will be evenly distributed between 0 and L. Ignored if x is specified.
max.conv	Maximum limit for summation in the convolutions to get inter-crossover distance distribution from the inter-chiasma distance distributions. This should be greater than the maximum number of chiasmata on the 4-strand bundle.

Details

Let $f(x;\nu)$ denote the density of a gamma random variable with parameters shape=ν and rate=2ν, and let $f_k(x;\nu)$ denote the density of a gamma random variable with parameters shape=$k\nu$ and rate=2ν.

The distribution of the distance from one crossover to the next is $f^*(x;\nu) = \sum_{k=1}^{\infty} f_k(x;\nu)/2^k$.

Value

A data frame with two columns: x is the distance (between 0 and L, in cM) at which the density was calculated and f is the density.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Broman, K. W. and Weber, J. L. (2000) Characterization of human crossover interference. *Am. J. Hum. Genet.* **66**, 1911–1926.

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

McPeek, M. S. and Speed, T. P. (1995) Modeling interference in genetic recombination. *Genetics* **139**, 1031–1044.

See Also

location.given.one(), first.given.two(), distance.given.two(), joint.given.two(), firstden(), xoprob(), gammacoi()

Examples

```
f1 <- ioden(1, L=200, n=201)
plot(f1, type="l", lwd=2, las=1,
     ylim=c(0,0.014), yaxs="i", xaxs="i", xlim=c(0,200))

f2 <- ioden(2.6, L=200, n=201)
lines(f2, col="blue", lwd=2)

f3 <- ioden(4.3, L=200, n=201)
lines(f3, col="red", lwd=2)

f4 <- ioden(7.6, L=200, n=201)
lines(f4, col="green", lwd=2)
```

---

joint.given.two *Crossover locations given there are two*

Description

Calculates the joint density of the crossover locations on a random meiotic product, given that there are precisely two crossovers, for the gamma model.

Usage

```
joint.given.two(
  nu,
  L = 103,
  x = NULL,
  y = NULL,
  n = 20,
  max.conv = 25,
  integr.tol = 0.00000001,
  max.subd = 1000,
  min.subd = 10
)
```

Arguments

nu	The interference parameter in the gamma model.
L	The length of the chromsome in cM.
x	If specified, locations of the first crossover.
y	If specified, locations of the second crossover.
n	Number of points at which to calculate the density. The points will be evenly distributed between 0 and L. Ignored if x and y are specified.
max.conv	Maximum limit for summation in the convolutions to get inter-crossover distance distribution from the inter-chiasma distance distributions. This should be greater than the maximum number of chiasmata on the 4-strand bundle.
integr.tol	Tolerance for convergence of numerical integration.
max.subd	Maximum number of subdivisions in numerical integration.
min.subd	Minimum number of subdivisions in numerical integration.

Details

Let $f(x;\nu)$ denote the density of a gamma random variable with parameters shape=ν and rate=2ν, and let $f_k(x;\nu)$ denote the density of a gamma random variable with parameters shape=$k\nu$ and rate=2ν.

The distribution of the distance from one crossover to the next is $f^*(x;\nu) = \sum_{k=1}^{\infty} f_k(x;\nu)/2^k$.

The distribution of the distance from the start of the chromosome to the first crossover is $g^*(x;\nu) = 1 - F^*(x;\nu)$ where F^* is the cdf of f^*.

Value

A data frame with three columns: x and y are the locations (between 0 and L, in cM) at which the density was calculated and f is the density.

Warning

We sometimes have difficulty with the numerical integrals. You may need to use large min.subd **(e.g. 25) to get accurate results.**

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Broman, K. W. and Weber, J. L. (2000) Characterization of human crossover interference. *Am. J. Hum. Genet.* **66**, 1911–1926.

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

McPeek, M. S. and Speed, T. P. (1995) Modeling interference in genetic recombination. *Genetics* **139**, 1031–1044.

See Also

location.given.one(), distance.given.two(), first.given.two(), ioden(), firstden(), xoprob(), gammacoi()

Examples

```
# Calculate the distribution of the average of the crossover locations,
# given that there are two and that they are separated by 20 cM
# (for a chromosome of length 200 cM)
L <- 200
d <- 20
x <- seq(0, L-d, by=0.5)
y <- x+d

f <- joint.given.two(4.3, L=L, x, y)
f$f <- f$f / distance.given.two(4.3, L, d)$f
plot((f$x+f$y)/2, f$f, type="l", xlim=c(0, L), ylim=c(0,max(f$f)),
     lwd=2, xlab="Average location", ylab="Density")
abline(v=c(d/2,L-d/2), h=1/(L-d), lty=2, lwd=2)
```

---

kfunc *estimate Ripley's K function*

Description

estimate the 1-d version of Ripley's K function

Usage

```
kfunc(
  x,
  d = seq(0, 100, by = 0.1),
  lengths = NULL,
  exclude = 0,
  tol = 0.000001
)
```

Arguments

x	list with sorted locations of the data
d	values at which to calculate the function
lengths	lengths of segments studied
exclude	distance to exclude
tol	tolerance value

Value

data frame with d, k, and se

See Also

gammacoi(), stahlcoi(), coincidence()

Examples

```
L <- 103
n <- 2000
map1 <- sim.map(L, n.mar=104, anchor=TRUE, include.x=FALSE, eq=TRUE)
x <- sim.cross(map1, n.ind=n, m=6, type="bc")

xoloc <- find.breaks(x)

d <- seq(0, 100, by=0.1)[-1]
kf <- kfunc(xoloc, d=d, lengths=rep(L, n))

plot(k ~ d, data=kf, type="n", yaxs="i", xaxs="i", las=1,
     ylim=c(0, max(kf$k + kf$se)))
polygon(c(kf$d, rev(kf$d)), c(kf$k + kf$se, rev(kf$k-kf$se)),
        border=NA, col="#add8e650")
lines(k ~ d, data=kf)
```

`location.given.one` *Location of crossover given there is one*

Description

Calculates the density of the location of the crossover on a random meiotic product, given that there is precisely one crossover, for the gamma model.

Usage

```
location.given.one(
  nu,
  L = 103,
  x = NULL,
  n = 400,
  max.conv = 25,
  integr.tol = 0.00000001,
  max.subd = 1000,
  min.subd = 10
)
```

Arguments

`nu`	The interference parameter in the gamma model.
`L`	The length of the chromsome in cM.
`x`	If specified, points at which to calculate the density.
`n`	Number of points at which to calculate the density. The points will be evenly distributed between 0 and `L`. Ignored if `x` is specified.
`max.conv`	Maximum limit for summation in the convolutions to get inter-crossover distance distribution from the inter-chiasma distance distributions. This should be greater than the maximum number of chiasmata on the 4-strand bundle.
`integr.tol`	Tolerance for convergence of numerical integration.
`max.subd`	Maximum number of subdivisions in numerical integration.
`min.subd`	Minimum number of subdivisions in numerical integration.

Details

Let $f(x;\nu)$ denote the density of a gamma random variable with parameters shape=ν and rate=2ν, and let $f_k(x;\nu)$ denote the density of a gamma random variable with parameters shape=$k\nu$ and rate=2ν.

The distribution of the distance from one crossover to the next is $f^*(x;\nu) = \sum_{k=1}^{\infty} f_k(x;\nu)/2^k$.

The distribution of the distance from the start of the chromosome to the first crossover is $g^*(x;\nu) = 1 - F^*(x;\nu)$ where F^* is the cdf of f^*.

We calculate the distribution of the location of the crossover on a product with a single crossover as the convolution of g^* with itself, and then rescaled to integrate to 1 on the interval (0,L).

Value

A data frame with two columns: `x` is the location (between 0 and `L`, in cM) at which the density was calculated and `f` is the density.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Broman, K. W. and Weber, J. L. (2000) Characterization of human crossover interference. *Am. J. Hum. Genet.* **66**, 1911–1926.

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

McPeek, M. S. and Speed, T. P. (1995) Modeling interference in genetic recombination. *Genetics* **139**, 1031–1044.

See Also

`first.given.two()`, `distance.given.two()`, `joint.given.two()`, `ioden()`, `firstden()`, `xoprob()`, `gammacoi()`

Examples

```
f1 <- location.given.one(1, L=200, n=201)
plot(f1, type="l", lwd=2, las=1,
     ylim=c(0,0.006), yaxs="i", xaxs="i", xlim=c(0,200))

f2 <- location.given.one(2.6, L=200, n=201)
lines(f2, col="blue", lwd=2)

f3 <- location.given.one(4.3, L=200, n=201)
lines(f3, col="red", lwd=2)

f4 <- location.given.one(7.6, L=200, n=201)
lines(f4, col="green", lwd=2)
```

---

recrate2scanone *Convert recrate to scanone format*

Description

Convert the result of `est.recrate()` to the format output by R/qtl's `qtl::scanone()` function.

Usage

```
recrate2scanone(recrate, phymap = NULL)
```

Arguments

recrate	A list of results from est.recrate()
phymap	A list of vectors of Mbp positions of markers

Value

A data frame with class "scanone", in the format output by qtl::scanone().

Author(s)

Karl W Broman, <broman@wisc.edu>

See Also

est.recrate()

Examples

```
pmap <- sim.map(100, n.mar=51, anchor=TRUE, include.x=FALSE, eq.spacing=TRUE)

# simulate cross
x <- sim.cross(pmap, type="bc", n.ind=501)

# estimate map for that cross
emap <- est.map(x)

# empirical estimate of recombination rate
rr <- est.recrate(emap[[1]], pmap[[1]], window=5)

# make it a list (one component per chromosome, but here just the one chromosome)
rr <- list("1"=rr)

# convert to scanone output and plot
rr_scanone <- recrate2scanone(rr)
plot(rr_scanone)
```

---

simStahl *Simulate crossover locations under the Stahl model*

Description

Simulate crossover locations under the Stahl model.

Usage

```
simStahl(
  n.sim,
  nu = 1,
  p = 0,
  L = 100,
  obligate_chiasma = FALSE,
  n.bins4start = 10000
)
```

Arguments

n.sim	Number of meiotic products to simulate.
nu	The interference parameter in the gamma model.
p	The proportion of chiasmata coming from the no-interference mechanism.
L	Chromosome length (in cM).
obligate_chiasma	Require an obligate chiasma (requires nu to be an integer; if nu is not an integer, it is rounded.
n.bins4start	We approximate the distribution of the location of the first crossover from the mechanism exhibiting interference using a even grid with this many bins. (Only if nu is not an integer.)

Details

The Stahl model is an extension to the gamma model, in which chiasmata occur according to two independent mechanisms. A proportion p come from a mechanism exhibiting no interference, and a proportion 1-p come from a mechanism in which chiasma locations follow a gamma model with interference parameter ν.

Value

A vector of length n.sim, each element being empty (for products with no crossovers) or a vector of crossover locations, in cM. An attribute, L, contains the chromosome length in cM.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Copenhaver, G. P., Housworth, E. A. and Stahl, F. W. (2002) Crossover interference in Arabidopsis. *Genetics* **160**, 1631–1639.

Housworth, E. A. and Stahl, F. W. (2003) Crossover interference in humans. *Am J Hum Genet* **73**, 188–197.

See Also

fitGamma(), qtl::sim.cross()

Examples

```
# simulations with no interference, chromosome of length 80 cM
xoNI <- simStahl(100, nu=1, p=0, L=80)

# simulations under gamma model with nu=7.6
xogamma <- simStahl(100, nu=7.6, p=0, L=80)

# simulations under Stahl model with nu=7.6, p=0.1
xostahl <- simStahl(100, nu=7.6, p=0.1, L=80)

# simulations under chi-square model with nu=11 (m=10) and obligate chiasma
xo_oblchi <- simStahl(100, nu=11, p=0, L=80, obligate_chiasma=TRUE)

# simulations under Stahl model with nu=11, p=0.1, and obligate chiasma
xo_oblchi_stahl <- simStahl(100, nu=11, p=0.1, L=80, obligate_chiasma=TRUE)
```

---

stahlcoi *Coincidence function for the Stahl model*

Description

Calculates the coincidence function for the Stahl model.

Usage

```
stahlcoi(nu, p = 0, L = 103, x = NULL, n = 400, max.conv = 25)
```

Arguments

nu	The interference parameter in the gamma model.
p	The proportion of chiasmata coming from the no-interference mechanism.
L	Maximal distance (in cM) at which to calculate the density. Ignored if x is specified.
x	If specified, points at which to calculate the density.
n	Number of points at which to calculate the density. The points will be evenly distributed between 0 and L. Ignored if x is specified.
max.conv	Maximum limit for summation in the convolution. This should be greater than the maximum number of chiasmata on the 4-strand bundle.

Details

The Stahl model is an extension to the gamma model, in which chiasmata occur according to two independent mechanisms. A proportion p come from a mechanism exhibiting no interference, and a proportion 1-p come from a mechanism in which chiasma locations follow a gamma model with interference parameter ν.

Let $f(x; \nu, \lambda)$ denote the density of a gamma random variable with parameters shape=ν and rate=λ.

The coincidence function for the Stahl model is $C(x; \nu, p) = [p + \sum_{k=1}^{\infty} f(x; k\nu, 2(1-p)\nu)]/2$.

Value

A data frame with two columns: `x` is the distance (between 0 and `L`, in cM) at which the coicidence was calculated and `coincidence`.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Copenhaver, G. P., Housworth, E. A. and Stahl, F. W. (2002) Crossover interference in Arabidopsis. *Genetics* **160**, 1631–1639.

Housworth, E. A. and Stahl, F. W. (2003) Crossover interference in humans. *Am J Hum Genet* **73**, 188–197.

See Also

`gammacoi()`, `location.given.one()`, `first.given.two()`, `distance.given.two()`, `ioden()`, `firstden()`, `xoprob()`

Examples

```
f1 <- stahlcoi(1, p=0, L=200)
plot(f1, type="l", lwd=2, las=1,
     ylim=c(0,1.25), yaxs="i", xaxs="i", xlim=c(0,200))

f2 <- stahlcoi(2.6, p=0, L=200)
lines(f2, col="blue", lwd=2)

f2s <- stahlcoi(2.6, p=0.1, L=200)
lines(f2s, col="blue", lwd=2, lty=2)

f3 <- stahlcoi(4.3, p=0, L=200)
lines(f3, col="red", lwd=2)

f3s <- stahlcoi(4.3, p=0.1, L=200)
lines(f3s, col="red", lwd=2, lty=2)

f4 <- stahlcoi(7.6, p=0, L=200)
lines(f4, col="green", lwd=2)
```

```
f4s <- stahlcoi(7.6, p=0.1, L=200)
lines(f4s, col="green", lwd=2, lty=2)
```

`stahlLoglik` *Calculate log likelihood for Stahl model*

Description

Calculate the log likelihood for the Stahl model for varying parameters, with data on crossover locations.

Usage

```
stahlLoglik(
  xoloc,
  chrlen = NULL,
  nu,
  p,
  max.conv = 25,
  integr.tol = 0.00000001,
  max.subd = 1000,
  min.subd = 10
)
```

Arguments

`xoloc`	A list of crossover locations (in cM), each component being a vector of locations for a different meiotic product.
`chrlen`	Chromosome length (in cM), either of length 1 or the same length as `xoloc`.
`nu`	A vector of interference parameters (ν) at which to calculate the log likelihood.
`p`	A vector of parameter values for the proportion of crossovers from the no interference pathway.
`max.conv`	Maximum limit for summation in the convolutions to get inter-crossover distance distribution from the inter-chiasma distance distributions. This should be greater than the maximum number of chiasmata on the 4-strand bundle.
`integr.tol`	Tolerance for convergence of numerical integration.
`max.subd`	Maximum number of subdivisions in numerical integration.
`min.subd`	Minimum number of subdivisions in numerical integration.

Details

See Housworth and Stahl (2003) and Broman and Weber (2000) for details of the method.

If neither `nu` nor `p` has length 1, they both must have the same length. If one has length 1 and the other does not, the one with length 1 is repeated so that they both have the same length.

Value

A vector of log likelihoods.

The corresponding values of nu and p are saved as attributes.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Housworth, E. A. and Stahl, F. W. (2003) Crossover interference in humans. *Am. J. Hum. Genet.* **73**, 188–197.

Broman, K. W. and Weber, J. L. (2000) Characterization of human crossover interference. *Am. J. Hum. Genet.* **66**, 1911–1926.

See Also

qtl::fitstahl()

Examples

```
data(bssbsb)
xoloc <- find.breaks(bssbsb, chr=1)

loglik <- stahlLoglik(xoloc, nu=4, p=c(0.05, 0.1, 0.15))
```

---

xoiversion *Installed version of R/xoi*

Description

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

Usage

```
xoiversion()
```

Value

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

Author(s)

Karl W Broman, <broman@wisc.edu>

Examples

```
xoiversion()
```

`xoprob` *Distribution of number of crossovers*

Description

Calculates the probability of 0, 1, 2, or >2 crossovers for a chromosome of a given length, for the gamma model.

Usage

```
xoprob(
  nu,
  L = 103,
  max.conv = 25,
  integr.tol = 0.00000001,
  max.subd = 1000,
  min.subd = 10
)
```

Arguments

`nu`	The interference parameter in the gamma model.
`L`	Length of the chromosome (in cM).
`max.conv`	Maximum limit for summation in the convolutions to get inter-crossover distance distribution from the inter-chiasma distance distributions. This should be greater than the maximum number of chiasmata on the 4-strand bundle.
`integr.tol`	Tolerance for convergence of numerical integration.
`max.subd`	Maximum number of subdivisions in numerical integration.
`min.subd`	Minimum number of subdivisions in numerical integration.

Details

Let $f(x;\nu)$ denote the density of a gamma random variable with parameters shape=ν and rate=2ν, and let $f_k(x;\nu)$ denote the density of a gamma random variable with parameters shape=$k\nu$ and rate=2ν.

The distribution of the distance from one crossover to the next is $f^*(x;\nu) = \sum_{k=1}^{\infty} f_k(x;\nu)/2^k$.

The distribution of the distance from the start of the chromosome to the first crossover is $g^*(x;\nu) = 1 - F^*(x;\nu)$ where F^* is the cdf of f^*.

We calculate the desired probabilities by numerical integration.

Value

A vector of length 4, giving the probabilities of 0, 1, 2, or >2 crossovers, respectively, on a chromosome of length L cM.

Author(s)

Karl W Broman, <broman@wisc.edu>

References

Broman, K. W. and Weber, J. L. (2000) Characterization of human crossover interference. *Am. J. Hum. Genet.* **66**, 1911–1926.

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

McPeek, M. S. and Speed, T. P. (1995) Modeling interference in genetic recombination. *Genetics* **139**, 1031–1044.

See Also

location.given.one(), first.given.two(), distance.given.two(), joint.given.two(), ioden(), firstden(), gammacoi()

Examples

```
xoprob(1, L=103)
xoprob(4.3, L=103)
```

Index