

Package ‘introgress’

October 20, 2009

Type Package

Title methods for analyzing introgression between divergent lineages

Version 1.2.1

Date 2009-09-07

Author Zachariah Gompert, C. Alex Buerkle

Maintainer C. Alex Buerkle <buerkle@uwo.edu>

Depends nnet, genetics, RColorBrewer

Description introgress provides functions for analyzing introgression of genotypes between divergent, hybridizing lineages, including estimating genomic clines from multi-locus genotype data and testing for deviations from neutral expectations. Functions are also provided for maximum likelihood estimation of molecular hybrid index and graphical analysis.

License GPL (>= 2)

URL <http://www.uwo.edu/buerkle/software/introgress/>

LazyLoad yes

Repository CRAN

Date/Publication 2009-10-20 06:56:44

R topics documented:

introgress-package	2
AdmixDataSim1	3
AdmixDataSim2	4
AdmixDom	5
calc.intersp.het	6
clines.plot	7
compare.clines	9
delta	11

est.h	12
fit.c.clines	14
fit.invariant.clines	14
fixup.combos.touse	15
genomic.clines	15
h.func	19
like.h	19
LociD	20
LociDataSim1	20
LociDataSim2	21
mk.image	22
p1D	24
p1DataSim2	24
p2D	25
p2DataSim2	26
per.locus.like	26
prepare.data	27
s.wrapper	30
support.limit	30
test.combinations	30
test.data.objects	31
test.genotypes	31
triangle.plot	31

Index	34
--------------	-----------

introgress-package *introgress*

Description

`introgress` provides functions for analyzing introgression of genotypes between divergent, hybridizing lineages, including estimating genomic clines from multi-locus genotype data and testing for deviations from neutral expectations. Functions are also provided for maximum likelihood estimation of molecular hybrid index and graphical analysis.

Details

`introgress` includes several core functions that are needed to implement the genomic clines method of Gompert and Buerkle (2009a). `prepare.data` estimates the counts of alleles inherited from each of two parental population at each locus for admixed individuals. These estimates are based on genotype data from admixed population(s) and specified parental populations. The function `est.h` uses the list output returned from `prepare.data` and the raw genotype data for the admixed population(s) to compute maximum likelihood estimates of hybrid index. The list output returned by `prepare.data` and the data.frame output returned by `est.h` are used to estimate genomic clines, which is accomplished with the function `genomic.clines`. The `genomic.clines` function includes arguments specifying whether significance testing is performed and the procedure used to generate null expectations for significance testing. The `clines.plot`

function provides graphical output for genomic clines analysis similar to Gompert and Buerkle (2009a, 2009b). In addition, the `introgress` package includes the `compare.clines` function to contrast patterns of introgression between hybrid zones.

`introgress` implements additional functions for graphical analysis of hybrid zones. `triangle.plot` plots the relationship between hybrid index and inter-class heterozygosity (`calc.intersp.het`) for an admixed population. The `mk.image` function produces a graphical representation of marker ancestry allowing visual inspection of variation in patterns of introgression among markers.

For further information regarding the above functions and additional `introgress` functions, please see the appropriate function help pages.

Author(s)

Zachariah Gompert (zgomper@uwyo.edu), C. Alex Buerkle (buerkle@uwyo.edu)

References

Gompert Z. and Buerkle C. A. (2009) A powerful regression-based method for admixture mapping of isolation across the genome of hybrids. *Molecular Ecology*, **18**, 1207-1224.

Gompert Z. and Buerkle C. A. (2009) *introgress*: a software package for mapping components of isolation in hybrids. *Molecular Ecology Resources*, in preparation.

AdmixDataSim1

Admixed Population Genotype Data from Simulation 1

Description

This data set contains genotypes for a simulated admixed population. The simulated admixed population consisted of 500 diploid hermaphroditic individuals with ten pairs of chromosomes, each one Morgan in length. The admixed population resulted from an initial hybridization event between two parental populations (combined in equal proportions) and mating within the admixed population continued for five additional generations. Fitness of hybrids was determined based on a marker at the center of chromosome 1; individuals homozygous at this location had a fitness of 1, while heterozygotes had a fitness of 0.1. Two hundred admixed individuals were sampled at the completion of the simulation and scored for 110 co-dominant markers spaced evenly across all 10 linkage groups (at 10 cM intervals).

The data are provided as a matrix with rows and columns corresponding to markers and individuals, respectively. Alleles inherited from parental population 1 are recorded as *P1* and alleles inherited from parental population 2 are recorded as *P2*.

Usage

```
data(AdmixDataSim1)
```

Format

A matrix with 110 rows and 200 columns.

Source

From simulations conducted by CAB and ZG.

References

Gompert Z. and Buerkle C. A. (2009) A powerful regression-based method for admixture mapping of isolation across the genome of hybrids. *Molecular Ecology*, **18**, 1207-1224.

Gompert Z. and Buerkle C. A. (2009) *introgress*: a software package for mapping components of isolation in hybrids. *Molecular Ecology Resources*, in preparation.

See Also

[LociDataSim1](#)

AdmixDataSim2

Admixed Population Genotype Data from Simulation 2

Description

This data set contains genotypes for a simulated admixed population. The admixed population consists of 100 individuals scored for 90 co-dominant markers and 10 dominant markers. Admixed individuals were simulated by sampling genotypes from parental populations (`p1DataSim2` and `p2DataSim2`). The first and second rows of this matrix data object give population and individual identifications for each admixed individual. The following rows contain the genotype data; rows correspond to markers and columns correspond to individuals.

Usage

```
data(AdmixDataSim2)
```

Format

A matrix with 102 rows and 100 columns.

Source

From simulations conducted by CAB and ZG.

See Also

[LociDataSim2](#), [p1DataSim2](#), [p2DataSim2](#)

Description

This data set contains genotypes for a simulated admixed population. The simulated admixed populations consisted of 500 diploid hermaphroditic individuals with ten pairs of chromosomes, each one Morgan in length. The admixed population resulted from an initial hybridization event between two parental populations (combined in equal proportions) and mating within the admixed population continued for five additional generations. Fitness of hybrids was determined based on a marker at the center of chromosome 1; individuals homozygous at this location had a fitness of 1, while heterozygotes had a fitness of 0.1. Two hundred admixed individuals were sampled at the completion of the simulation and scored for 110 dominant markers spaced evenly across all 10 linkage groups (at 10 cM intervals). Alleles inherited from parental population 1 were treated as the dominant alleles.

The data are provided as a matrix with rows and columns corresponding to markers and individuals, respectively. For each marker an individual's genotype is given by a 1 (homozygous for population 1 alleles or heterozygous) or 0 (homozygous for population 2 alleles).

Usage

```
data (AdmixDom)
```

Format

A matrix with 110 rows and 200 columns.

Source

From simulations conducted by CAB and ZG.

References

Gompert Z. and Buerkle C. A. (2009) A powerful regression-based method for admixture mapping of isolation across the genome of hybrids. *Molecular Ecology*, **18**, 1207-1224.

Gompert Z. and Buerkle C. A. (2009) *introgress*: a software package for mapping components of isolation in hybrids. *Molecular Ecology Resources, in preparation*.

See Also

[Locid](#)

calc.intersp.het *Calculate Interspecific Heterozygosity*

Description

This function calculates interspecific heterozygosity of individuals from a matrix of allele counts.

Usage

```
calc.intersp.het(introgress.data=NULL)
```

Arguments

```
introgress.data
```

a list produced by `prepare.data` or a matrix with allele counts.

Details

This function calculates an admixed individual's interspecific heterozygosity (i.e. the proportion of the individual's genome with alleles inherited from both parental populations) based on allele counts. This function should only be used for co-dominant markers.

Value

A numerical vector with an estimate of interspecific heterozygosity for each individual.

Author(s)

Zachariah Gompert (zgomper@uwyo.edu), C. Alex Buerkle (buerkle@uwyo.edu)

See Also

[prepare.data](#)

Examples

```
## Not run:
## load simulated data
## markers have fixed differences, with
## alleles coded as 'P1' and 'P2'
data(AdmixDataSim1)
data(LociDataSim1)

## use prepare.data to produce introgress.data
introgress.data<-prepare.data(admix.gen=AdmixDataSim1,
                             loci.data=LociDataSim1,
                             parental1="P1", parental2="P2",
                             pop.id=FALSE, ind.id=FALSE,
                             fixed=TRUE)
```

```
## Estimate inter-specific heterozygosity
int.het<-calc.intersp.het(introgress.data=introgress.data)
## End(Not run)
```

clines.plot

Clines Plot

Description

This function produces graphical plots of genomic clines using the output from the function `genomic.clines`.

Usage

```
clines.plot(cline.data=NULL,marker.order=NULL,rplots=3,cplots=3,pdf=TRUE,
            out.file="clines.pdf",colors=c("#005A32","#41AB5D"),
            quantiles=FALSE,lb.cd=rep(0.025,3),ub.cd=rep(0.975,3),
            lb.dh=rep(0.025,2),ub.dh=rep(0.975,2),cd=c("AA","Aa","aa"),dh=c("A","a"))
```

Arguments

<code>cline.data</code>	a list that is the product of the <code>genomic.clines</code> function.
<code>marker.order</code>	an optional numeric or character vector specifying the order to plot marker results in; if <code>marker.order = NULL</code> markers are plotted in the order they were originally provided.
<code>rplots</code>	numerical value specifying the number of plots per row in the output.
<code>cplots</code>	numerical value specifying the number of plots per column in the output.
<code>pdf</code>	logical specifying whether to print the plots to a pdf file; if <code>pdf=FALSE</code> plots are printed to the current graphical device.
<code>out.file</code>	a character string for the filename for the output if <code>pdf=TRUE</code> .
<code>colors</code>	a vector of two colors to be used for cline plots.
<code>quantiles</code>	logical specifying whether to include genotype specific quantile information on the plots.
<code>lb.cd</code>	numeric vector of three proportions specifying the lower bounds for co-dominant markers that is used to determine whether each genotype is under-represented.
<code>ub.cd</code>	numeric vector of three proportions specifying the upper bounds for co-dominant markers that is used to determine whether each genotype is over-represented.
<code>lb.dh</code>	numeric vector of two proportions specifying the lower bounds for dominant and haploid markers that is used to determine whether each genotype is under-represented.
<code>ub.dh</code>	numeric vector of two proportions specifying the upper bounds for dominant and haploid markers that is used to determine whether each genotype is over-represented.

cd	vector with three characters used for labeling genotypes of co-dominant markers in cline plots.
dh	vector with two characters used for labeling genotypes of dominant and haploid markers in cline plots.

Details

This function produces graphical plots based on the data object produced by the `genomic.clines` function. A separate plot is produced for each marker. Plots depict the probability of a given genotype as a function of hybrid index. Plots for the observed data are shown with solid (homozygous population 1 genotype for co-dominant markers, or population 1 allele for dominant or haploid markers) and dashed lines (inter-specific heterozygotes for co-dominant markers, not shown for dominant or haploid markers). If significance testing was conducted with the `genomic.clines` function, the corresponding 95% confidence intervals are shown in dark and light green. Circles indicate individuals that are homozygous for alleles from population 1 (co-dominant) or population 1 alleles (dominant or haploid) on the top line, interspecific heterozygotes (co-dominant only) in the middle, and homozygotes for alleles from population 2 (co-dominant) or population 2 alleles (dominant or haploid) on the bottom line. The number of individuals with each genotype is printed on the right vertical axis. The locus name and *P*-value are printed in each plot. If `quantiles = TRUE` and genotype-specific quantiles were computed with the `genomic.clines` function by setting `classification = TRUE`, a title is given for each individual plot that includes whether each genotype (as labeled by `cd` or `dh`) was over- (+) or under-represented (-) compared to neutral expectation using `lb.cd`, `ub.cd`, `lb.dh`, `ub.dh` as thresholds.

See Gompert and Buerkle (2009a, 2009b) for additional details and examples.

Value

A plot is produced, but a data object is not returned.

Author(s)

Zachariah Gompert (zgomper@uwyo.edu), C. Alex Buerkle (buerkle@uwyo.edu)

References

Gompert Z. and Buerkle C. A. (2009) A powerful regression-based method for admixture mapping of isolation across the genome of hybrids. *Molecular Ecology*, **18**, 1207-1224.

Gompert Z. and Buerkle C. A. (2009) *introgress*: a software package for mapping components of isolation in hybrids. *Molecular Ecology Resources*, in preparation.

See Also

[genomic.clines](#)

Examples

```
## this code assumes the data object clines.out2 has been produced using
## the sample code for the "genomic.clines" function
```

```
## produce plots of genomic clines
## Not run:
clines.plot(cline.data=clines.out2)
## End(Not run)
```

compare.clines *Compare Clines*

Description

This function contrasts patterns of introgression between two hybrid zones.

Usage

```
compare.clines(cline.data1=NULL, cline.data2=NULL, sig.test=FALSE,
               n.reps=1000)
```

Arguments

`cline.data1` a list that is the product of the `genomic.clines` function.
`cline.data2` a list that is the product of the `genomic.clines` function.
`sig.test` a logical specifying whether to perform significance testing.
`n.reps` numeric value specifying the number of permutations to conduct for significance testing.

Details

This function estimates that likelihood of the count data from `cline.data2` given the regression models from `cline.data1` and `cline.data2` and returns the log ratio of the latter to the former. `cline.data1` and `cline.data2` are lists returned by `genomic.clines` each based on a different hybrid zone between the same parental populations or species; the molecular markers and parental allele frequencies should be the same for both hybrid zones. The range of hybrid index estimates for individuals comprising `cline.data2` should exceed the range of hybrid index estimated for individuals comprising `cline.data1` to avoid predicting allele counts using the regression model from `cline.data1` beyond the range of hybrid indexes that were included in the original model. The log likelihood ratios returned by this function can be used to determine the degree of congruence for marker specific patterns of introgression between the analyzed hybrid zones. This function includes a permutation procedure for testing whether log likelihood ratios are significantly greater than expected by chance, which is invoked by setting `sig.test = TRUE`. This option should only be employed when both hybrid zones contain many individuals with low and high hybrid index estimates to minimized the problem described above.

See Gompert and Buerkle (2009a, 2009b) for additional details and examples.

Value

A matrix with log likelihood ratios for each marker and P-values from significance testing if `sig.test = TRUE`.

Author(s)

Zachariah Gompert (zgomper@uwyo.edu), C. Alex Buerkle (buerkle@uwyo.edu)

References

Gompert Z. and Buerkle C. A. (2009) A powerful regression-based method for admixture mapping of isolation across the genome of hybrids. *Molecular Ecology*, **18**, 1207-1224.

Gompert Z. and Buerkle C. A. (2009) *introgress*: a software package for mapping components of isolation in hybrids. *Molecular Ecology Resources*, in preparation.

See Also

[genomic.clines](#), [prepare.data](#), [est.h](#)

Examples

```
## Not run:
## load simulated data
## markers have fixed differences, with
## alleles coded as 'P1' and 'P2'
data(AdmixDataSim1)
data(LociDataSim1)

## use prepare.data to produce introgress.data
introgress.data<-prepare.data(admix.gen=AdmixDataSim1,
                             loci.data=LociDataSim1,
                             parental1="P1", parental2="P2",
                             pop.id=FALSE, ind.id=FALSE, fixed=TRUE)

## estimate hybrid index
hi.index<-est.h(introgress.data=introgress.data,
               loci.data=LociDataSim1, p1.allele="P1",
               p2.allele="P2")

## random sampling to divide data into two sets of 100 individuals,
## this creates two admixed populations (hybrid zones)
numbs<-sample(1:200,200,replace=FALSE)
sam1<-numbs[1:100]
sam2<-numbs[101:200]

## estimate genomic clines for each data set,
## significance testing is not conducted
clines.out1<-genomic.clines(introgress.data=introgress.data,
                           hi.index=hi.index, loci.data=LociDataSim1,
                           sig.test=FALSE, ind.touse=sam1)

clines.out2<-genomic.clines(introgress.data=introgress.data,
                           hi.index=hi.index, loci.data=LociDataSim1,
                           sig.test=FALSE, ind.touse=sam2)

## compare clines between data sets, with significance testing
```

```

comp.out<-compare.clines(clines.out1,clines.out2,sig.test=TRUE,
                        n.reps=1000)

write.table(comp.out, file="compareClines.txt",
            quote=FALSE, sep=", ")
## End(Not run)

```

delta

Delta

Description

This function calculates delta (δ), the differential between allelic frequencies in two populations. It is called by the function `prepare.data`.

Usage

```
delta(SpA, SpB)
```

Arguments

SpA a vector of population allele frequencies.
SpB a vector of population allele frequencies.

Details

This function calculates the allele frequency differential for a pair of species or populations (δ , Gregorius and Roberds 1986) for a single locus.

$$\delta = \sum_i \frac{|f_{i1} - f_{i2}|}{2}$$

where f_{i1} and f_{i2} denote the frequency of the i th allele in populations one and two, respectively. δ ranges from 0, for identical populations, to 1, for populations that share no alleles in common.

Value

This function returns a vector of length one with the value of δ .

Author(s)

Zachariah Gompert (zgomper@uwyo.edu), C. Alex Buerkle (buerkle@uwyo.edu)

References

Gregorius, H. R. and Roberds, J. H. (1986) Measurement of genetical differentiation among sub-populations. *Theoretical and Applied Genetics*, **71**, 826-834.

See Also

[prepare.data](#)

Examples

```
## allele frequencies for two populations
pop.a<-c(0.3,0.5,0.2)
pop.b<-c(0.1,0.1,0.8)

## delta calculation
delta(pop.a,pop.b)
```

est.h

Estimate Hybrid Index

Description

This function finds maximum likelihood estimates of hybrid index as described by Buerkle (2005).

Usage

```
est.h(introgress.data=NULL, loci.data=NULL, ind.touse=NULL,
      fixed=FALSE, p1.allele=NULL, p2.allele=NULL)
```

Arguments

introgress.data	a list produced by <code>prepare.data</code> or a matrix with allele counts.
loci.data	a matrix or array providing marker information.
ind.touse	vector of individual identifications, numeric indexes, or logicals that specify a subset of individuals for analysis, if <code>NULL</code> all individuals are included.
fixed	a logical specifying whether different alleles are fixed for each parental population for all markers.
p1.allele	if <code>fixed=TRUE</code> provides the character used to specify parental population 1 alleles.
p2.allele	if <code>fixed=TRUE</code> provides the character used to specify parental population 2 alleles.

Details

`introgress.data` may either be the list that is returned by the function `prepare.data`, or, if `fixed=TRUE`, `introgress.data` may simply be a matrix or array providing counts of the number of alleles derived from parental population 1 for each admixed individual. If `introgress.data` is a matrix or array, rows and columns correspond to loci and individuals, respectively.

`loci.data` is a matrix or array where each row provides information on one locus. The first column gives a unique locus name (e.g. "locus3"), and the second column specifies whether the locus is co-dominant ("C" or "c"), haploid ("H" or "h"), or dominant ("D" or "d"). These first two columns in `loci.data` are required. The third column, which is optional, is a numeric value specifying the linkage groups for the marker. The fourth column, which is also optional, is a numeric value specifying both the linkage group and location on the linkage group (e.g. 3.70, for a marker at 70 cM on linkage group 3). These optional columns can be used for ordering markers for the `mk.image`, `genomic.clines`, and `clines.plot` functions.

If the parental populations exhibit fixed allelic differences for all markers scored (i.e. `fixed=TRUE`) then `p1.allele` and `p2.allele` should give the character used to specify alleles derived from parental populations one and two, respectively (e.g. `p1.allele="p1"` and `p2.allele="p2"`).

`est.h` uses a maximum likelihood method to estimate the hybrid index for each admixed individual. This estimate is simply the proportion of alleles derived from population 2 if `fixed=TRUE`. In contrast, if populations share alleles (i.e. `fixed=FALSE`), the estimate accounts for uncertainty in the ancestry of alleles, but the estimate of hybrid index is still an estimate of the proportion of the genome that is inherited from population 2. For each individual, `est.h` returns a point estimate and 95% confidence interval of hybrid index.

See Buerkle (2005) for additional details.

Value

A data frame with point estimates of hybrid index and upper and lower limits of 95% confidence intervals (interval of hybrid index that falls within two support units of the ML estimate):

<code>lower</code>	Lower limit of 95% confidence interval.
<code>h</code>	Maximum-likelihood estimate of hybrid index.
<code>upper</code>	Upper limit of 95% confidence interval.

Author(s)

Zachariah Gompert (zgomper@uwyo.edu), C. Alex Buerkle (buerkle@uwyo.edu)

References

- Buerkle C. A. (2005) Maximum-likelihood estimation of a hybrid index based on molecular markers. *Molecular Ecology Notes*, **5**, 684-687.
- Gompert Z. and Buerkle C. A. (2009) A powerful regression-based method for admixture mapping of isolation across the genome of hybrids. *Molecular Ecology*, **18**, 1207-1224.
- Gompert Z. and Buerkle C. A. (2009) *introgress*: a software package for mapping components of isolation in hybrids. *Molecular Ecology Resources*, in preparation.

See Also

[prepare.data](#)

Examples

```
## Not run:
## load simulated data
## markers have fixed differences, with
## alleles coded as 'P1' and 'P2'
data(AdmixDataSim1)
data(LociDataSim1)

## use prepare.data to produce introgress.data
introgress.data<-prepare.data(admix.gen=AdmixDataSim1,
                             loci.data=LociDataSim1,
                             parental1="P1", parental2="P2",
                             pop.id=FALSE, ind.id=FALSE,
                             fixed=TRUE)

## estimate hybrid index
hi.index<-est.h(introgress.data=introgress.data,
               loci.data=LociDataSim1, ind.touse=NULL, fixed=TRUE,
               p1.allele="P1", p2.allele="P2")

write.table(hi.index, file="hindex.txt", quote=FALSE, sep=",")
## End(Not run)
```

fit.c.clines
Fit Co-dominant Clines

Description

This is an internal function called by `genomic.clines`.

Author(s)

Zachariah Gompert (<zgompert@uwyo.edu>), C. Alex Buerkle (<buerkle@uwyo.edu>)

See Also

[genomic.clines](#)

fit.invariant.clines
Fit Invariant Clines

Description

This is an internal function called by `genomic.clines`.

Author(s)

Zachariah Gompert (zgomper@uwyo.edu), C. Alex Buerkle (buerkle@uwyo.edu)

See Also

[genomic.clines](#)

fixup.combos.touse *Fix Combos To Use*

Description

This is an internal function called by `prepare.data`.

Author(s)

Zachariah Gompert (zgomper@uwyo.edu), C. Alex Buerkle (buerkle@uwyo.edu)

See Also

[test.combinations](#), [prepare.data](#)

genomic.clines *Genomic Clines*

Description

This function fits genomic clines to genotypic data using the method described by Gompert and Buerkle (2009a). Significance testing is included, but optional.

Usage

```
genomic.clines(introgress.data=NULL, hi.index=NULL, loci.data=NULL,
               sig.test=FALSE, method="permutation", n.reps=1000,
               classification=FALSE, het.cor=TRUE, loci.touse=NULL, ind.touse=NULL)
```

Arguments

<code>introgress.data</code>	a list produced by <code>prepare.data</code> or a matrix with allele counts.
<code>hi.index</code>	a data frame produced by <code>est.h</code> or a numeric vector of hybrid index estimates.
<code>loci.data</code>	a matrix or array providing marker information.
<code>sig.test</code>	a logical specifying whether to perform significance testing.
<code>method</code>	method to generate null distribution; either "permutation" or "parametric".

<code>n.reps</code>	numeric value specifying number of neutral simulations.
<code>classification</code>	a logical specifying whether to calculate genotype specific quantiles (ignored if <code>sig.test=FALSE</code>).
<code>het.cor</code>	a logical specifying whether to correct for deviations from expected heterozygosity when conducting neutral simulations using the parametric method (ignored with permutation method).
<code>loci.touse</code>	vector of loci names, numeric indexes, or logicals that specify a subset of loci for analysis, if <code>NULL</code> all loci are included.
<code>ind.touse</code>	vector of individual identifications, numeric indexes, or logicals that specify a subset of individuals for analysis, if <code>NULL</code> all individuals are included.

Details

`introgress.data` may either be the list that is returned by the function `prepare.data`, or, if `fixed=TRUE`, `introgress.data` may simply be a matrix or array providing counts of the number of alleles derived from parental population 1 for each admixed individual. If `introgress.data` is a matrix or array, rows and columns correspond to loci and individuals, respectively.

`loci.data` is a matrix or array where each row provides information on one locus. The first column gives a unique locus name (e.g. `"locus3"`), and the second column specifies whether the locus is co-dominant (`"C"` or `"c"`), haploid (`"H"` or `"h"`), or dominant (`"D"` or `"d"`). These first two columns in `loci.data` are required. The third column, which is optional, is a numeric value specifying the linkage groups for the marker. The fourth column, which is also optional, is a numeric value specifying both the linkage group and location on the linkage group (e.g. `3.70`, for a marker at 70 cM on linkage group 3). These optional columns can be used for ordering markers for the `mk.image`, `genomic.clines`, and `clines.plot` functions.

This function (`genomic.clines`) estimates genomic clines in genotype frequency for admixed populations using the genomic clines method described by Gompert and Buerkle (2009a). If `sig.test = FALSE`, genomic clines are estimated for the admixed population, but no significance testing is done. If `sig.test = TRUE`, the genomic cline for each locus is evaluated for deviations from neutral expectations. Either the permutation method (`method = "permutation"`) or the parametric method (`method = "parametric"`) described by Gompert and Buerkle (2009a) can be used to generate the neutral distribution for significance testing. The permutation method cannot be used if both co-dominant and dominant (or haploid) data are included in the analysis. If `classification = TRUE` this function also returns the proportion of neutral simulations/permutations yielding a model with a higher total probability of a given genotype for a given marker than the model from the observed data

The function will issue a warning if an invariant locus is included (all individuals have the same genotype). In this case the probability of one of the genotypes does not vary with hybrid index.

See Gompert and Buerkle (2009a, 2009b) for additional details.

Value

A list with the following components:

`Summary.data` a matrix with the locus data including log likelihood ratios and P-values from significance testing if `sig.test = TRUE`.

Fitted.AA	a matrix with fitted values for the population 1 homozygote for each locus (row) and individual (column).
Fitted.Aa	a matrix with fitted values for inter-population heterozygotes for each locus (row) and individual (column).
Fitted.aa	a matrix with fitted values for the population 2 homozygote for each locus (row) and individual (column).
Neutral.AA	a matrix with upper and lower bounds for the empirical 95% confidence interval for the expected population 1 homozygote genomic clines under neutrality for each locus (row) and individual (column); these confidence intervals are based on the neutral simulations/permutations.
Neutral.Aa	a matrix with upper and lower bounds for the empirical 95% confidence interval for the expected inter-population heterozygote genomic clines under neutrality for each locus (row) and individual (column); these confidence intervals are based on the neutral simulations/permutations.
Neutral.aa	a matrix with upper and lower bounds for the empirical 95% confidence interval for the expected population 2 homozygote genomic clines under neutrality for each locus (row) and individual (column); these confidence intervals are based on the neutral simulations/permutations.
Count.matrix	the user supplied count.matrix.
hybrid.index	the user supplied hi.index.
Loci.data	the user supplied loci.data.
Quantiles	a matrix with the proportion of neutral simulations/permutations yielding a model with a higher total probability of a given genotype (column) than the model from the observed data; proportions are provided for each locus (row).

Author(s)

Zachariah Gompert (zgomper@uwyo.edu), C. Alex Buerkle (buerkle@uwyo.edu)

References

- Gompert Z. and Buerkle C. A. (2009) A powerful regression-based method for admixture mapping of isolation across the genome of hybrids. *Molecular Ecology*, **18**, 1207-1224.
- Gompert Z. and Buerkle C. A. (2009) *introgress*: a software package for mapping components of isolation in hybrids. *Molecular Ecology Resources, in preparation*.

See Also

[prepare.data.est.h](#)

Examples

```
## Not run:
## Example 1, genomic clines analysis without significance testing, or
## with significance testing on a subset of the data

## load simulated data
```



```
write.table(clines.out2$Summary.data, file="clines.txt",
            quote=FALSE, sep=",")
## End(Not run)
```

h.func

Hybrid Index Function

Description

This is an internal function called by `est.h`.

Author(s)

Zachariah Gompert (zgomper@uwyo.edu), C. Alex Buerkle (buerkle@uwyo.edu)

See Also

[est.h](#)

like.h

Likelihood Hybrid Index

Description

This is an internal function called by `est.h`.

Author(s)

Zachariah Gompert (zgomper@uwyo.edu), C. Alex Buerkle (buerkle@uwyo.edu)

See Also

[est.h](#)

`LociD`*Dominant Marker Information for Simulation 1*

Description

This data object provides the marker information associated with `AdmixDom`. Rows correspond to markers (110 markers spaced evenly across 10 linkage groups at 10 cM intervals). The first column gives the locus name, (`c1.f` is the marker at the location of the genomic region that determined each individual's fitness, see `AdmixDom`). The second column (`type`) indicates that all of the markers are dominant (D). The third and fourth columns specify each markers linkage group (between 1 and 10) and relative location on each linkage group, respectively.

Usage

```
data(LociD)
```

Format

A matrix with 110 rows and 4 columns.

Source

From simulations conducted by CAB and ZG.

References

Gompert Z. and Buerkle C. A. (2009) A powerful regression-based method for admixture mapping of isolation across the genome of hybrids. *Molecular Ecology*, **18**, 1207-1224.

Gompert Z. and Buerkle C. A. (2009) *introgress*: a software package for mapping components of isolation in hybrids. *Molecular Ecology Resources*, in preparation.

See Also

[AdmixDom](#)

`LociDataSim1`*Marker Information for Simulation 1*

Description

This data object provides the marker information associated with `AdmixSimData1`. Rows correspond to markers (110 markers spaced evenly across 10 linkage groups at 10 cM intervals). The first column gives the locus name, (`c1.f` is the marker at the location of the genomic region that determined each individual's fitness, see `AdmixSimData1`). The second column (`type`) indicates that all of the markers are co-dominant (C). The third and fourth columns specify each markers linkage group (between 1 and 10) and relative location on each linkage group, respectively.

Usage

```
data(LociDataSim1)
```

Format

A matrix with 110 rows and 4 columns.

Source

From simulations conducted by CAB and ZG.

References

Gompert Z. and Buerkle C. A. (2009) A powerful regression-based method for admixture mapping of isolation across the genome of hybrids. *Molecular Ecology*, **18**, 1207-1224.

Gompert Z. and Buerkle C. A. (2009) *introgress*: a software package for mapping components of isolation in hybrids. *Molecular Ecology Resources, in preparation*.

See Also

[AdmixDataSim1](#)

LociDataSim2

Marker Information for Simulation 2

Description

This data object provides the marker information associated with `AdmixDataSim2`. Rows correspond to markers (100 markers total). The first column gives the locus name and the second column gives the locus type; co-dominant (C) for the first 90 markers, dominant (D) for the last 10 markers. Map information is not provided for these markers.

Usage

```
data(LociDataSim2)
```

Format

A matrix with 100 rows and 2 columns.

Source

From simulations conducted by CAB and ZG.

See Also

[AdmixDataSim2](#), [p1DataSim2](#), [p2DataSim2](#)

mk.image

*Make Image***Description**

This function produces a graphical representation of marker ancestry across individuals and facilitates visual inspection of variation in patterns of introgression among markers.

Usage

```
mk.image(introgress.data=NULL, loci.data=NULL,
        marker.order=NULL, hi.index=NULL, ind.touse=NULL,
        loci.touse=NULL, ylab.image="Individuals", main.image="",
        xlab.h="population 2 ancestry", col.image=NULL,
        pdf=TRUE, out.file="image.pdf")
```

Arguments

introgress.data	a list produced by <code>prepare.data</code> or a matrix with allele counts.
loci.data	a matrix or array providing marker information.
marker.order	a numeric or character vector specifying the order in which to plot markers, if <code>marker.order=NULL</code> the markers will be plotted in the order in which they occur in the <code>introgress.data</code> .
hi.index	a data frame produced by <code>est.h</code> or a numeric vector of hybrid index estimates.
ind.touse	vector specifying a subset of individuals to plot, if <code>ind.touse=NULL</code> all individuals will be included.
loci.touse	vector specifying a subset of markers to plot, if <code>loci.touse=NULL</code> all markers will be included.
ylab.image	character string giving the label for the y-axis.
main.image	character string giving the main label for the plot.
xlab.h	character string giving the x-axis label for the hybrid index portion of the plot.
col.image	vector of three colors to be used for the image plot, if <code>col.image=NULL</code> default colors will be used.
pdf	a logical specifying whether to print the plot to a pdf, if <code>pdf=FALSE</code> the current graphical output device will be used.
out.file	character string specifying the name of the output file for the image, if <code>pdf=TRUE</code> .

Details

`introgress.data` may either be the list that is returned by the function `prepare.data`, or, if `fixed=TRUE`, `introgress.data` may simply be a matrix or array providing counts of the number of alleles derived from parental population 1 for each admixed individual. If `introgress.data` is a matrix or array, rows and columns correspond to loci and individuals, respectively.

`loci.data` is a matrix or array where each row provides information on one locus. The first column gives a unique locus name (e.g. "locus3"), and the second column specifies whether the locus is co-dominant ("C" or "c"), haploid ("H" or "h"), or dominant ("D" or "d"). These first two columns in `loci.data` are required. The third column, which is optional, is a numeric value specifying the linkage groups for the marker. The fourth column, which is also optional, is a numeric value specifying both the linkage group and location on the linkage group (e.g. 3.70, for a marker at 70 cM on linkage group 3). If present, the third column will be used to draw breaks between and label linkage groups. The fourth column can be used to generate an order in which to plot markers (specified in `marker.order`).

This function (`mk.image`) produces a plot of marker specific ancestry for each individual. Each row corresponds to an individual and each column to a marker. Each marker/individual combination is colored dark green for homozygotes for alleles from parental population 1 (allele count of 2), medium green for interspecific heterozygotes (allele count of 1), light green for homozygotes for alleles from parental population 2 (allele count of 0), or white (missing data). Individuals are ordered according to hybrid index estimates, which are plotted to the right of the main plot.

Value

A plot is produced, but no value is returned.

Author(s)

Zachariah Gompert (zgomper@uwyo.edu), C. Alex Buerkle (buerkle@uwyo.edu)

References

Gompert Z. and Buerkle C. A. (2009) *introgress*: a software package for mapping components of isolation in hybrids. *Molecular Ecology Resources, in preparation*.

See Also

[prepare.data](#), [est.h](#)

Examples

```
## Not run:
## load simulated data
## markers have fixed differences, with
## alleles coded as 'P1' and 'P2'
data(AdmixDataSim1)
data(LociDataSim1)

## use prepare.data to produce introgress.data
introgress.data<-prepare.data(admix.gen=AdmixDataSim1,
                             loci.data=LociDataSim1,
                             parental1="P1", parental2="P2", pop.id=FALSE,
                             ind.id=FALSE, fixed=TRUE)

## estimate hybrid index
hi.index<-est.h(introgress.data=introgress.data,
               loci.data=LociDataSim1, fixed=TRUE, p1.allele="P1",
```

```

        p2.allele="P2")

## produce image plot of marker ancestry
mk.image(introgress.data=introgress.data, loci.data=LociDataSim1,
        marker.order=NULL, hi.index=hi.index, ind.touse=NULL,
        loci.touse=NULL, ylab.image="Individuals", main.image="",
        xlab.h="population 2 ancestry", col.image=NULL,
        pdf=TRUE, out.file="image.pdf")
## End(Not run)

```

p1D

Parental Population 1 Dominant Genotype Data from Simulation 1

Description

This data set contains genotype data for parental population 1 associated with `AdmixDom`. This data set consists of 110 dominant markers scored for 10 individuals. All individuals were given a genotype of 1, meaning they possessed the dominant allele at each marker.

Usage

```
data(p1D)
```

Format

A matrix with 110 rows and 10 columns.

Source

From simulations conducted by CAB and ZG.

See Also

[AdmixDom](#)

p1DataSim2

Parental Population 1 Dominant Genotype Data from Simulation 2

Description

This data set contains genotype data for parental population 1 associated with `AdmixDataSim2`. This data set consists of 90 co-dominant and 10 dominant markers scored for 50 individuals. The 220 allele occurs at a high frequency for the 90 co-dominant markers in this population, as does the 1 allele (phenotype) for the 10 dominant markers.

Usage

```
data(p1DataSim2)
```

Format

A matrix with 100 rows and 50 columns.

Source

From simulations conducted by CAB and ZG.

See Also

[AdmixDataSim2](#)

p2D

Parental Population 2 Dominant Genotype Data from Simulation 1

Description

This data set contains genotype data for parental population 2 associated with `AdmixDom`. This data set consists of 110 dominant markers scored for 10 individuals. All individuals were given a genotype of 0, meaning that they are homozygous for the recessive allele at each marker.

Usage

```
data(p2D)
```

Format

A matrix with 110 rows and 10 columns.

Source

From simulations conducted by CAB and ZG.

See Also

[AdmixDom](#)

`p2DataSim2`*Parental Population 2 Genotype Data from Simulation 2*

Description

This data set contains genotype data for parental population 2 associated with `AdmixDataSim2`. This data set consists of 90 co-dominant and 10 dominant markers scored for 50 individuals. The 224 and 230 alleles occurs at high frequency for the 90 co-dominant markers in this population, as does the 0 (recessive) allele for the 10 dominant markers.

Usage

```
data(p2DataSim2)
```

Format

A matrix with 100 rows and 50 columns.

Source

From simulations conducted by CAB and ZG.

See Also

[AdmixDataSim2](#)

`per.locus.like`*Per Locus Likelihood*

Description

This is an internal function called by `est.h`.

Author(s)

Zachariah Gompert (zgompert@uwyo.edu), C. Alex Buerkle (buerkle@uwyo.edu)

See Also

[est.h](#)

```
prepare.data          Prepare Data
```

Description

This function is meant to be used with individuals from an admixed population. The function determines the number of alleles inherited from each of two parental populations at each locus. The counts are based on genotype data from specified parental populations, which must be supplied. This function works with both co-dominant and dominant (or haploid) data.

Usage

```
prepare.data(admix.gen=NULL, loci.data=NULL,
             parental1=NULL, parental2=NULL,
             pop.id=TRUE, ind.id=TRUE, fixed=FALSE,
             sep.rows=FALSE, sep.columns=FALSE)
```

Arguments

<code>admix.gen</code>	a matrix, array or data frame with genotype data
<code>loci.data</code>	a matrix or array providing marker information.
<code>parental1</code>	a matrix or two-dimensional array if <code>fixed=FALSE</code> , a single character if <code>fixed = TRUE</code> .
<code>parental2</code>	a matrix or two-dimensional array if <code>fixed=FALSE</code> , a single character if <code>fixed=TRUE</code> .
<code>pop.id</code>	a logical specifying whether <code>admix.gen</code> includes a row specifying sampling localities.
<code>ind.id</code>	a logical specifying whether <code>admix.gen</code> includes a row specifying individual identifications.
<code>fixed</code>	a logical specifying whether all loci scored exhibit fixed differences between the parental populations.
<code>sep.rows</code>	a logical specifying whether genotypes at a locus are recorded using two rows.
<code>sep.columns</code>	a logical specifying whether genotypes at a locus are recorded using two columns.

Details

Genotypic data for individuals are provided in `admix.gen`, a data object with genotypes for each individual at each locus in the format 'A/D' or '110/114' for co-dominant data, 'A' or 'hap1b' for haploid data, and '0' or '1' for dominant data. In other words, for co-dominant and haploid data alleles can be encoded by any simple character string. Each row should contain data for a locus and columns should correspond to individuals. Missing data should be entered as 'NA/NA' or 'NA' for co-dominant and haploid / dominant data, respectively.

Alternatively, in `admix.gen` genotypic data for an individual can be split between two rows (`sep.rows = TRUE`) or two columns (`sep.columns = TRUE`). These options are similar to those of the data format for the program *structure* (Pritchard et al. 2000, Falush et al. 2003), with

the difference that `admix.gen` is transposed relative to the input for *structure*. Thus, after reading in a *structure* file, the data matrix can be transposed with `rawdata <- t(rawdata)` before passing the matrix to `prepare.data`. If genotype data are split across columns or rows, and they include haploid or dominant markers, the second allele for these markers should be recorded as NA.

If `pop.id = TRUE` and `ind.id = TRUE` the first row of `admix.gen` should give the population identification (i.e. sampling locality) of each individual and the second row should provide a unique individual identification; genotype information would then begin on row three.

`loci.data` is a matrix or array data object where each row provides information on one locus. The first column gives a unique locus name (e.g. "locus3"), and the second column specifies whether the locus is co-dominant ("C" or "c"), haploid ("H" or "h"), or dominant ("D" or "d"). These first two columns in `loci.data` are required. The third column, which is optional, is a numeric value specifying the linkage groups for the marker. If present, this column is used in the `mk.image` function for plotting. The fourth column, which is also optional, is a numeric value specifying both the linkage group and location on the linkage group (e.g. 3.70, for a marker at 70 cM on linkage group 3). This last column could be used to generate a different order in which to utilize marker data from `admix.gen` in other functions in the package (specified in the `marker.order` argument to `mk.image` and `clines.plot`). Each column in `loci.data` should have a heading (the second column should be named "type").

If the parental populations exhibit fixed differences for all markers scored (i.e. `fixed = TRUE`) then `parental1` and `parental2` should give the character used to specify alleles derived from parental populations one and two, respectively (e.g. `parental1 = "p1"` and `parental2 = "p2"`). If parental populations exhibit fixed differences at all loci, the count matrix produced by `prepare.data` is simply a count of the number of alleles inherited from parental population 1 for each individual at each locus (0, 1, or 2 for co-dominant marker data; 0 or 1 for dominant or haploid marker data).

If the parental populations do not exhibit fixed differences at all loci scored (i.e. `fixed = FALSE`) then `parental1` and `parental2` should be matrix data objects providing genotype data for individuals sampled from each of the parental populations. These data objects should be in the same format as the `genotype.data` data object, with the difference that they should not contain rows for individual and population identifications at the top. `prepare.data` uses the parental data objects to calculate allele frequencies at each locus for both of the parental populations. Alleles are then binned into allelic classes with maximum (equal to the observed) frequency differentials between parental populations (δ , Gregorius and Roberds 1986). These allelic classes serve as the basis for estimating the count matrix, which is in the same format as described above. In the absence of fixed differences the counts are of alleles from the allelic class associated with population 1 and the frequency of allelic classes in the parental species can be used to account for uncertainty in the ancestry of particular alleles.

See Gompert and Buerkle (2009a, 2009b) for additional details.

Value

A list with the following components:

`Individual.data`

a matrix with `pop.id` and `ind.id` data if they were supplied.

`Count.matrix` the count matrix; each row corresponds to a locus and each column represents an individual.

Combos.to.use	NULL if fixed = TRUE, otherwise this provides the allelic class data needed for genomic.clines.
Parental1.allele.freq	matrix of allele frequencies calculated for parental population 1 where each row is a locus and each column is an allele.
Parental2.allele.freq	matrix of allele frequencies calculated for parental population 2 where each row is a locus and each column is an allele.
Alleles	a matrix specifying the names of the alleles in the same order as they are given in Parental1.allele.freq and Parental2.allele.freq for each locus.
Admix.gen	the matrix of genotype data for the admixed population; each row corresponds to a locus and each column represents an individual.

Author(s)

Zachariah Gompert (zgomper@uwyo.edu), C. Alex Buerkle (buerkle@uwyo.edu)

References

- Falush D., Stephens M., and Pritchard J. K. (2003) Inference of population structure using multilocus genotype data: linked loci and correlated allele frequencies. *Genetics*, **164**, 1567-1587.
- Gompert Z. and Buerkle C. A. (2009) A powerful regression-based method for admixture mapping of isolation across the genome of hybrids. *Molecular Ecology*, **18**, 1207-1224.
- Gompert Z. and Buerkle C. A. (2009) *introgress*: a software package for mapping components of isolation in hybrids. *Molecular Ecology Resources*, in preparation.
- Gregorius H. R. and Roberds J. H. (1986) Measurement of genetical differentiation among subpopulations. *Theoretical and Applied Genetics*, **71**, 826-834.
- Pritchard J. K., Stephens M., and Donnelly P. (2000) Inference of population structure using multi-locus genotype data. *Genetics*, **155**, 945-959.

See Also

[delta.mk.image](#), [genomic.clines](#), [clines.plot](#)

Examples

```
## Not run:
## load simulated data
## markers have fixed differences, with
## alleles coded as 'P1' and 'P2'
data(AdmixDataSim1)
data(LociDataSim1)

## use prepare.data to produce introgress.data
introgress.data<-prepare.data(admix.gen=AdmixDataSim1,
                             loci.data=LociDataSim1,
```

```
parental1="P1", parental2="P2",
pop.id=FALSE, ind.id=FALSE, fixed=TRUE)
```

```
## End(Not run)
```

```
s.wrapper
```

```
Sign Wrapper
```

Description

This is an internal function called by `est.h`.

Author(s)

Zachariah Gompert (zgompert@uwyo.edu), C. Alex Buerkle (buerkle@uwyo.edu)

See Also

[est.h](#)

```
support.limit
```

```
Support Limit
```

Description

This is an internal function called by `est.h`.

Author(s)

Zachariah Gompert (zgompert@uwyo.edu), C. Alex Buerkle (buerkle@uwyo.edu)

See Also

[est.h](#)

```
test.combinations Test Combinations
```

Description

This is an internal function called by `prepare.data`.

Author(s)

Zachariah Gompert (zgompert@uwyo.edu), C. Alex Buerkle (buerkle@uwyo.edu)

See Also

[delta,prepare.data](#)

test.data.objects *Test Data Objects*

Description

This is an internal function called by `prepare.data`.

Author(s)

Zachariah Gompert (zgomper@uwyo.edu), C. Alex Buerkle (buerkle@uwyo.edu)

See Also

[prepare.data](#)

test.genotypes *Test Genotypes*

Description

This is an internal function called by `prepare.data`.

Author(s)

Zachariah Gompert (zgomper@uwyo.edu), C. Alex Buerkle (buerkle@uwyo.edu)

See Also

[prepare.data](#)

triangle.plot *Triangle Plot*

Description

This function makes a plot of interspecific heterozygosity as a function of hybrid index.

Usage

```
triangle.plot(hi.index=NULL, int.het=NULL, pdf=TRUE,  
              out.file="tri_plot.pdf")
```

Arguments

<code>hi.index</code>	a data frame produced by <code>est.h</code> or a numeric vector of hybrid index estimates.
<code>int.het</code>	a vector of numeric interspecific heterozygosity estimates.
<code>pdf</code>	a logical specifying whether results should be output to a pdf file.
<code>out.file</code>	a character string specifying the name of the output file if <code>pdf=TRUE</code> .

Details

This function plots interspecific heterozygosity as a function of hybrid index for individuals from an admixed population. Individuals that are the progeny of at least one parent from one of the pure parental populations should have maximal heterozygosity for the observed hybrid index. The plot has lines that correspond to these theoretical maximum values. Individuals that fall on the maximal line are likely F1s or backcross progeny. Evidence for individuals of this type will be more likely if the data set consists of loci with no alleles in common between parental species (`delta=1`), whereas shared alleles will lead to ambiguity in inferring ancestry. Hybrid index estimates can be obtained from the `est.h` function and interspecific heterozygosity estimates can be obtained from the `calc.intersp.het` function.

Value

A plot is produced, but there is no return value.

Author(s)

Zachariah Gompert (zgomper@uwyo.edu), C. Alex Buerkle (buerkle@uwyo.edu)

See Also

[est.h](#), [calc.intersp.het](#), [delta](#)

Examples

```
## produce triangle plots
## load simulated data
## markers have fixed differences, with
## alleles coded as 'P1' and 'P2'
data(AdmixDataSim1)
data(LociDataSim1)

## use prepare.data to produce introgress.data
introgress.data<-prepare.data(admix.gen=AdmixDataSim1,
                             loci.data=LociDataSim1,
                             parental1="P1", parental2="P2",
                             pop.id=FALSE, ind.id=FALSE,
                             fixed=TRUE)

## estimate hybrid index
hi.index<-est.h(introgress.data=introgress.data,
               loci.data=LociDataSim1, fixed=TRUE, p1.allele="P1",
               p2.allele="P2")
```

```
## Estimate interspecific heterozygosity
int.het<-calc.intersp.het(introgress.data=introgress.data)

## make plot
triangle.plot(hi.index=hi.index, int.het=int.het, pdf=FALSE)
```

Index

*Topic **datasets**

- AdmixDataSim1, [3](#)
- AdmixDataSim2, [4](#)
- AdmixDom, [4](#)
- LociD, [19](#)
- LociDataSim1, [20](#)
- LociDataSim2, [21](#)
- p1D, [24](#)
- p1DataSim2, [24](#)
- p2D, [25](#)
- p2DataSim2, [25](#)

AdmixDataSim1, [3](#), [20](#)

AdmixDataSim2, [4](#), [21](#), [25](#), [26](#)

AdmixDom, [4](#), [20](#), [24](#), [25](#)

calc.intersp.het, [2](#), [5](#), [32](#)

clines.plot, [2](#), [6](#), [29](#)

compare.clines, [2](#), [8](#)

delta, [10](#), [29–32](#)

est.h, [2](#), [9](#), [11](#), [17–19](#), [23](#), [26](#), [29](#), [30](#), [32](#)

fit.c.clines, [13](#)

fit.invariant.clines, [14](#)

fixup.combos.touse, [14](#)

genomic.clines, [2](#), [8](#), [9](#), [14](#), [15](#), [29](#)

h.func, [18](#)

introgress-package, [2](#)

like.h, [19](#)

LociD, [5](#), [19](#)

LociDataSim1, [3](#), [20](#)

LociDataSim2, [4](#), [21](#)

mk.image, [2](#), [21](#), [29](#)

p1D, [24](#)

p1DataSim2, [4](#), [21](#), [24](#)

p2D, [25](#)

p2DataSim2, [4](#), [21](#), [25](#)

per.locus.like, [26](#)

prepare.data, [2](#), [6](#), [9](#), [11](#), [13](#), [14](#), [17](#), [23](#), [26](#), [30](#), [31](#)

s.wrapper, [29](#)

support.limit, [30](#)

test.combinations, [14](#), [30](#)

test.data.objects, [30](#)

test.genotypes, [31](#)

triangle.plot, [2](#), [31](#)