

Package ‘Geneland’

June 22, 2010

Title Detection of structure from multilocus genetic data.

Author Arnaud Estoup, Gilles Guillot, Filipe Santos

Description Stochastic simulation and MCMC inference of structure from genetic data.

Maintainer Gilles Guillot <gigu@imm.dtu.dk> , Filipe Santos
<filipe.santos@mail.icav.up.pt>

URL <http://www2.imm.dtu.dk/~gigu/Geneland/>

License GPL

Version 3.2.2

Date Tue Jun 22 16:15:37 2010

Depends RandomFields,fields,mapproj,maps,snow,tcltk

Repository CRAN

Date/Publication 2010-06-22 14:59:23

R topics documented:

Geneland-package	2
coordinates	3
EstimateFreqNA	4
FormatGenotypes	4
Fstat	5
Fstat.output	6
Geneland.GUI	7
genotypes	7
gl2gp	8
MCMC	8
nullify	13
PlotDrift	14
PlotFreq	15
PlotFreqA	15

Plotnpop	16
Plotntile	17
PlotTessDyn	17
PlotTessellation	18
PosteriorMode	18
PostProcessChain	19
PostProcessMultChain	20
show.simdata	22
simdata	23
simFmodel	26

Index	28
--------------	-----------

Geneland-package *Simulation and inference for subdivided populations*

Description

The main function [MCMC](#) takes multilocus genetic data and tries to detect population structure (i.e sub-populations), making use of genetic (and optionally geographic) information.

Details

Package: Geneland
 Type: Package
 Version: 3.0.0
 Date: 2008-11-15
 License: GPL

The main purpose of the program is to perform Bayesian inference of all the parameters involved through Markov Chain Monte-Carlo simulation. This is achieved by the function [MCMC](#). Function [PostProcessChain](#) read some output files of [MCMC](#) and computes some statistics suitable to print maps of inferred populations.

See [Storage format](#) section in [MCMC](#) help page.

The following functions are provided by the package:

[simFmodel](#): simulation from the prior of the spatial F-model

[simdata](#): Simulation of georeferenced genotypes under an IBD + barrier model

[show.simdata](#): Graphical display of data simulated by [simdata](#)

[MCMC](#): Full Bayesian Markov Chain Monte Carlo inference of parameters in the spatial F-model

[PostProcessChain](#): Post-processing of [MCMC](#) output for maps of posterior probability of populations subdomains

[PlotTessellation](#): Graphical display of inferred sub-domains

The following functions are very basic and are only intended to be an aid for those not familiar with R. Most probably you may want to use directly the output files of `MCMC` and `PostProcessChain` to print your own figures.

`PlotDrift`: Graphical display of drift factors along MCMC run

`PlotFreqA`: Graphical display of allele frequencies in the ancestral population along MCMC run

`PlotFreq`: Graphical display of allele frequencies in the present time population along MCMC run

`Plotnpop`: Graphical display of number of populations along MCMC run

Author(s)

Arnaud Estoup, Gilles Guillot, Filipe Santos

www.inapg.inra.fr/ens_rech/mathinfo/personnel/guillot/welcome.html

See also

www.inapg.inra.fr/ens_rech/mathinfo/personnel/guillot/Geneland.html

References

- G. Guillot, Estoup, A., Mortier, F. Cosson, J.F. A spatial statistical model for landscape genetics. *Genetics*, 170, 1261-1280, 2005.
- G. Guillot, Mortier, F., Estoup, A. Geneland : A program for landscape genetics. *Molecular Ecology Notes*, 5, 712-715, 2005.
- Gilles Guillot, Filipe Santos and Arnaud Estoup, Analysing georeferenced population genetics data with Geneland: a new algorithm to deal with null alleles and a friendly graphical user interface *Bioinformatics* 2008 24(11):1406-1407.
- G. Guillot. Inference of structure in subdivided populations at low levels of genetic differentiation. The correlated allele frequencies model revisited. *Bioinformatics*, 24:2222-2228, 2008
- G. Guillot and F. Santos A computer program to simulate multilocus genotype data with spatially auto-correlated allele frequencies. *Molecular Ecology Resources*, 2009
- G. Guillot, R. Leblois, A. Coulon, A. Frantz Statistical methods in spatial genetics, *Molecular Ecology*, 2009.

coordinates

Example of coordinate file

Description

Example of coordinate file

EstimateFreqNA *Estimate frequencies of null alleles.*

Description

Estimate frequencies of null alleles at each locus.

Usage

```
EstimateFreqNA (path.mcmc)
```

Arguments

path.mcmc path to directory containing stored MCMC computations

Value

A vector of length `nloc` (the number of loci) whose entries are estimated frequencies of null alleles.

Author(s)

Gilles Guillot

References

Gilles Guillot, Filipe Santos and Arnaud Estoup, Analysing georeferenced population genetics data with Geneland: a new algorithm to deal with null alleles and a friendly graphical user interface *Bioinformatics* 2008 24(11):1406-1407.

FormatGenotypes *Formatting file of genotype data.*

Description

Takes genotype data as a matrix with one line per individual and two columns per locus, with alleles coded by integers (number of replications for micro-satellites data). Build a new matrix with alleles codes as consecutive integers. If a locus has 7 alleles they will be coded as 1,2,...7. Since version 1.0.1, this function does not have to be called by users. It is called through MCMC.

Usage

```
FormatGenotypes (genotypes, ploidy)
```

Arguments

<code>genotypes</code>	A matrix with one line per individual and two columns per locus, with alleles coded by integers
<code>ploidy</code>	1 or 2

Value

A list with elements:

<code>genotypes</code>	a matrix with one line per individual and one or two columns per locus with alleles coded by integers
<code>allele.numbers</code>	a vector giving the number of possible alleles per locus

Author(s)

Gilles Guillot

<code>Fstat</code>	<i>Computes F statistics on a population genetics dataset given as R object</i>
--------------------	---

Description

Computes F statistics according to Weir and Cockerham's estimators. Missing values are allowed and accounting for in the computation of Fst. The presence of missing values involves a downward bias in the computation of Fis. This function should not be used on haploid data.

Usage

```
Fstat (genotypes, npop, pop.mbrship, ploidy)
```

Arguments

<code>genotypes</code>	Diploid codominant genotype data. A matrix with one line per individual and 2 columns per locus
<code>npop</code>	total number of population present in the dataset
<code>pop.mbrship</code>	Vector of integers giving the population membership for each individual
<code>ploidy</code>	Integer: 1 or 2 (default is 2) under development. Do not use for haploid data.

Value

A list with components

<code>Fis</code>	A vector of estimations of within-population Fis
<code>Fst</code>	A matrix of estimations the pairwise population Fst. (Only the upper triangular part is returned).

Author(s)

Arnaud Estoup for original code in Turbo Pascal. Translation in Fortran and interface with R by Gilles Guillot

References

Weir, B.S. and C.C. Cockerham, Estimating F-statistics for the analysis of population structure, *Evolution*, 1984, vol. 38, 1358-1370.

Fstat.output	<i>Computes F statistics on the output of an inference by MCMC simulation</i>
--------------	---

Description

Computes F statistics according to Weir and Cockerham's estimators. Missing values are allowed but as of today, the NA code is only treated as an extra allele which might bias the result. This function should not be used on haploid data.

Usage

```
Fstat.output(coordinates, genotypes, ploidy, burnin, path.mcmc)
```

Arguments

coordinates	Matrix with one line per individual and two columns
genotypes	Genotypes of individuals. A matrix with one line per individual and 2 columns per locus
burnin	Integer: number of saved iterations to discard.
ploidy	Integer: 1 or 2 (default is 2). Do not use for haploid data.
path.mcmc	Path to output files directory

Value

A list with components

Pairwise.Fis

A matrix of real numbers estimating the pairwise Fis

Pairwise.Fst

A matrix of real numbers estimating the pairwise Fst

Author(s)

Arnaud Estoup for original code in Turbo Pascal.

Translation in Fortran and interface with R by Gilles Guillot

References

Weir, B.S. and C.C. Cockerham, Estimating F-statistics for the analysis of population structure, *Evolution*, 1984, vol. 38, 1358-1370.

Geneland.GUI	<i>Graphical interface for package Geneland</i>
--------------	---

Description

Launch a menu driven interface to package Geneland

Usage

```
Geneland.GUI(lib.loc)
```

Arguments

<code>lib.loc</code>	A character string giving the path to the directory where Geneland is installed
----------------------	---

Author(s)

Filipe Santos

<code>genotypes</code>	<i>Example of coordinate file</i>
------------------------	-----------------------------------

Description

Example of coordinate file

 gl2gp

Geneland to Genepop conversion

Description

Takes the matrices of coordinates and genotypes in the Geneland format and writes it as an ascii file in the Genepop format

Usage

```
gl2gp(coordinates, genotypes, file)
```

Arguments

coordinates	Matrix of coordinates
genotypes	MATRIX of genotypes
file	Character string giving the path to the file where the data in Genepop format should be written

Value

An ascii file is written. This file may need to be process in order to convert eol character from unix style to dos style.

Author(s)

Gilles Guillot and Tanguy de l'Argentaye

MCMC

Markov Chain Monte-Carlo inference under various models

Description

Markov Chain Monte-Carlo inference under various models

Usage

```

MCMC (
## input data
coordinates=NULL, # spatial coordinates
geno.dip.codom=NULL, # diploid codominant markers
                    # one line per indiv.
                    # two column per marker
geno.dip.dom=NULL, # diploid dominant markers
                    # one line per indiv.
```

```

                                # one column per marker
geno.hap=NULL, # haploid
                                # one line per indiv.
                                # one column per marker
qtc, # quantitative continuous variables
qtd, # quantitative discrete variables
ql, # qualitative variables
## path to output directory
path.mcmc,
## hyper-prior parameters
rate.max,delta.coord=0,shapel=2,shape2=20,
npopmin=1,npopinit,npopmax,
## dimensions
nb.nuclei.max,
## mcmc computations options
nit,
thinning=1,
freq.model="Uncorrelated",
varnpop=TRUE,
spatial=TRUE,
jcf=TRUE,
filter.null.alleles=TRUE,
prop.update.cell=0.1,
## writing mcmc output files options
write.rate.Poisson.process=FALSE,
write.number.nuclei=TRUE,
write.number.pop=TRUE,
write.coord.nuclei=TRUE,
write.color.nuclei=TRUE,
write.freq=TRUE,
write.ancestral.freq=TRUE,
write.drifts=TRUE,
write.logposterior=TRUE,
write.loglikelihood=TRUE,
write.true.coord=TRUE,
write.size.pop=FALSE,
write.mean.quant=TRUE,
write.sd.quant=TRUE,
miss.loc=NULL)

```

Arguments

- `coordinates` Spatial coordinates of individuals. A matrix with 2 columns and one line per individual.
- `geno.dip.codom` Genotypes for diploid data with codominant markers. A matrix with one line per individual and two columns per locus.
- `geno.dip.dom` Genotypes for diploid data with dominant markers. A matrix with one line per individual and one column per locus. Presence/absence of a band should be

	coded as 0/1 (0 for absence / 1 for presence). Dominant and codominant markers can be analyzed jointly by passing variables to arguments <code>geno.dip.codom</code> and <code>geno.dip.dom</code> . Haploid data and diploid dominant data can not be analyzed jointly in the current version.
<code>geno.hap</code>	Genotypes of haploid data. A matrix with one line per individual and one column per locus. Dominant diploid data and haploid data can be analyzed jointly (e.g. to analyse microsatellite data or SNP data together with mtDNA. Haploid data and diploid dominant data can not be analyzed jointly in the current version.
<code>qtc</code>	TODO
<code>qtd</code>	TODO
<code>ql</code>	TODO
<code>path.mcmc</code>	Path to output files directory. It seems that the path should be given in the Unix style even under Windows (use <code>V</code> instead of <code>\</code>). This path <i>has to</i> end with a slash (<code>V</code>) (e.g. <code>path.mcmc="/home/me/Geneland-stuffs/"</code>)
<code>rate.max</code>	Maximum rate of Poisson process (real number >0). Setting <code>rate.max</code> equal to the number of individuals in the dataset has proved to be efficient in many cases.
<code>delta.coord</code>	Parameter prescribing the amount of uncertainty attached to spatial coordinates. If <code>delta.coord=0</code> spatial coordinates are considered as true coordinates, if <code>delta.coord>0</code> it is assumed that observed coordinates are true coordinates blurred by an additive noise uniform on a square of side <code>delta.coord</code> centered on 0.
<code>shape1</code>	First parameter in the Beta(<code>shape1,shape2</code>) prior distribution of the drift parameters in the Correlated model.
<code>shape2</code>	Second parameter in the Beta(<code>shape1,shape2</code>) prior distribution of the drift parameters in the Correlated model.
<code>npopmin</code>	Minimum number of populations (integer ≥ 1)
<code>npopinit</code>	Initial number of populations (integer such that <code>npopmin</code> \leq <code>npopinit</code> \leq <code>npopmax</code>)
<code>npopmax</code>	Maximum number of populations (integer \geq <code>npopinit</code>). There is no obvious rule to select <code>npopmax</code> , it should be set to a value larger than any value that you can reasonably expect for your data.
<code>nb.nuclei.max</code>	Integer: Maximum number of nuclei in the Poisson-Voronoi tessellation. A good guess consists in setting this value equal to $3 * \text{rate.max}$. Lower values can also be used in order to speed up computations. The relevance of the value set can be checked by inspection of the MCMC run. The number of tiles should not go too close to <code>nb.nuclei.max</code> . If it does, you should re-run your chain with a larger value for <code>nb.nuclei.max</code> . In case of use of the option <code>SPATIAL=FALSE</code> , <code>nb.nuclei.max</code> should be set equal to the number of individuals.
<code>nit</code>	Number of MCMC iterations
<code>thinning</code>	Number of MCMC iterations between two writing steps (if <code>thinning=1</code> , all states are saved whereas if e.g. <code>thinning=10</code> only each 10 iteration is saved)

<code>freq.model</code>	Character: "Correlated" or "Uncorrelated" (model for frequencies). See also details in detail section of GeneLand help page.
<code>varnpop</code>	Logical: if TRUE the number of class is treated as unknown and will vary along the MCMC inference, if FALSE it will be fixed to the initial value <code>npopinit</code> . <code>varnpop = TRUE</code> *should not* be used in conjunction with <code>freq.model = "Correlated"</code> as in this case it seems that large numbers of populations are not penalized enough and there is a serious risk of inferring spurious sub-populations.
<code>spatial</code>	Logical: if TRUE the colored Poisson-Voronoi tessellation is used as a prior for the spatial organisation of populations. If FALSE, all clustering receive equal prior probability. In this case spatial information (i.e coordinates) are not used and the locations of the nuclei are initialized and kept fixed at the locations of individuals.
<code>jcf</code>	Logical: if true update of c and f are performed jointly
<code>filter.null.alleles</code>	Logical: if TRUE, tries to filter out null alleles. An extra fictive null allele is created at each locus coding for all putative null allele. Its frequency is estimated and can be viewed with function <code>PlotFreq</code> . This option is available only with <code>freq.model="Uncorrelated"</code> .
<code>prop.update.cell</code>	Integer between 0 and 1. Proportion of cell updated. For debugging only.
<code>write.rate.Poisson.process</code>	Logical: if TRUE (default) write rate of Poisson process simulated by MCMC
<code>write.number.nuclei</code>	Logical: if TRUE (default) write number of nuclei simulated by MCMC
<code>write.number.pop</code>	Logical: if TRUE (default) write number of populations simulated by MCMC
<code>write.coord.nuclei</code>	Logical: if TRUE (default) write coordinates of nuclei simulated by MCMC
<code>write.color.nuclei</code>	Logical: if TRUE (default) write color of nuclei simulated by MCMC
<code>write.freq</code>	Logical: if TRUE (default is FALSE) write allele frequencies simulated by MCMC
<code>write.ancestral.freq</code>	Logical: if TRUE (default is FALSE) write ancestral allele frequencies simulated by MCMC
<code>write.drifts</code>	Logical: if TRUE (default is FALSE) write drifts simulated by MCMC
<code>write.logposterior</code>	Logical: if TRUE (default is FALSE) write logposterior simulated by MCMC
<code>write.loglikelihood</code>	Logical: if TRUE (default is FALSE) write loglikelihood simulated by MCMC
<code>write.true.coord</code>	Logical: if TRUE (default is FALSE) write true spatial coordinates simulated by MCMC

<code>write.size.pop</code>	Logical: if TRUE (default is FALSE) write size of populations simulated by MCMC
<code>write.mean.quant</code>	Logical: if TRUE (default is FALSE) write means of quantitative variables in the various groups simulated by MCMC
<code>write.sd.quant</code>	Logical: if TRUE (default is FALSE) write standard deviations of quantitative variables in the various groups simulated by MCMC
<code>miss.loc</code>	A matrix with <code>nindiv</code> lines and <code>nloc</code> columns of 0 or 1. For each individual, at each locus it says if the locus is genuinely missing (no attempt to measure it). This info is used under the option <code>filterNA=TRUE</code> to decide how a double missing value should be treated (genuine missing data or double null allele).

Value

Successive states of all blocks of parameters are written in files contained in `path.mcmc` and named after the type of parameters they contain.

Storage format

All parameters processed by function `MCMC` are written in the directory specified by `'path.mcmc'` as follows:

- File `'population.numbers.txt'` contains values of the number of populations (`nit` lines, one line per iteration of the MCMC algorithm).
- File `'population.numbers.txt'` contains values of the number of populations (`nit` lines, one line per iteration of the MCMC algorithm).
- File `'nuclei.numbers.txt'` contains the number of points in the Poisson point process generating the Voronoi tessellation.
- File `'color.nuclei.txt'` contains vectors of integers of length `nb.nuclei.max` coding the class membership of each Voronoi tile. Vectors of class membership for successive states of the chain are concatenated in one column. Some entries of the vector containing class membership for a current state may have missing values as the actual number of polygon may be smaller than the maximum number allowed `nb.nuclei.max`. This file has `nb.nuclei.max*chain/thinning` lines.
- File `'coord.nuclei.txt'` contains coordinates of points in the Poisson point process generating the Voronoi tessellation. It has `nb.nuclei.max*chain/thinning` lines and two columns (hor. and vert. coordinates).
- File `'drifts.txt'` contains the drift factors for each population, (one column per population).
- File `'ancestral.frequencies.txt'` contains allele frequencies in ancestral population. Each line contains all frequencies of the current state. The file has `nit` lines. In each line, values of allele frequencies are stored by increasing allele index and locus index (allele index varying first).
- File `'frequencies.txt'` contains allele frequencies of present time populations. Column `xx` contains frequencies of population number `xx`. In each column values of allele frequencies are stored by increasing allele index and locus index (allele index varying first), and values

of successive iterations are pasted. The file has `nallmax*nloc*nit/thinning` lines where `nallmax` is the maximum number of alleles over all loci.

- File ‘`Poisson.process.rate.txt`’ contains rates of Poisson process.
- File ‘`hidden.coord.txt`’ contains the coordinates of each individual as updated along the chain if those given as input are not considered as exact coordinates (which is specified by `delta.coord` to a non zero value).
- File ‘`log.likelihood.txt`’ contains log-likelihood of data for the current state of parameters of the Markov chain.
- File ‘`log.posterior.density.txt`’ contains log of posterior probability (up to marginal density of data) of the current state of parameters in the Markov chain.

Author(s)

Gilles Guillot

References

- G. Guillot, Estoup, A., Mortier, F. Cosson, J.F. A spatial statistical model for landscape genetics. *Genetics*, 170, 1261-1280, 2005.
- G. Guillot, Mortier, F., Estoup, A. Geneland : A program for landscape genetics. *Molecular Ecology Notes*, 5, 712-715, 2005.
- Gilles Guillot, Filipe Santos and Arnaud Estoup, Analysing georeferenced population genetics data with Geneland: a new algorithm to deal with null alleles and a friendly graphical user interface *Bioinformatics* 2008 24(11):1406-1407.
- G. Guillot. Inference of structure in subdivided populations at low levels of genetic differentiation. The correlated allele frequencies model revisited. *Bioinformatics*, 24:2222-2228, 2008
- G. Guillot and F. Santos A computer program to simulate multilocus genotype data with spatially auto-correlated allele frequencies. *Molecular Ecology Resources*, 2009
- G. Guillot, R. Leblois, A. Coulon, A. Frantz Statistical methods in spatial genetics, *Molecular Ecology*, 2009.

See Also

[simFmodel](#)

nullify

Simulates null alleles

Description

Simulates null alleles

Usage

```
nullify(genotypes, nall.null = 1, nloc.null)
```

Arguments

<code>genotypes</code>	a matrix of genotypes as produced by <code>simFmodel</code> and <code>simIBD</code>
<code>nall.null</code>	number of null alleles on each locus
<code>nloc.null</code>	number of loci carrying null alleles

Details

For diploid data only

Value

A list with component:

<code>genotypes</code>	The new genotypes after alteration.
------------------------	-------------------------------------

Author(s)

Gilles Guillot

PlotDrift

Plot the drift factors

Description

Gives a plot of the trace of the drift factors along the MCMC run

Usage

```
PlotDrift(path.mcmc, printit=FALSE, file)
```

Arguments

<code>path.mcmc</code>	Path to output files directory
<code>printit</code>	Logical : if TRUE, figures are also printed
<code>file</code>	Character : Path to file where figures should be printed

Author(s)

Gilles Guillot

PlotFreq	<i>Plot the trace of allele frequencies in present time population and optionally print it.</i>
----------	---

Description

Plot of allele frequencies in present time populations number `ipop`, for allele `iall` of locus `iloc`

Usage

```
PlotFreq(path.mcmc, ipop, iloc, iall, printit=FALSE, path)
```

Arguments

<code>path.mcmc</code>	Path to output files directory
<code>ipop</code>	Integer number : index of population
<code>iloc</code>	Integer number : index of locus
<code>iall</code>	Integer number : index of allele. If MCMC was launched with option <code>filter.null.alleles=TRUE</code> , an extra fictive allele standing for putative null alleles is created. Its estimated frequency can be also plotted. If there is say, 5 alleles at a locus, the estimated frequency of null alleles can be seen invoking PlotFreq with <code>iall=6</code> .
<code>printit</code>	Logical : if TRUE, figures are also printed
<code>path</code>	Character : Path to directory where figures should be printed

Author(s)

Gilles Guillot

PlotFreqA	<i>Plots allele frequencies in ancestral population along the MCMC run and optionally prints it</i>
-----------	---

Description

Plot frequency of allele `iall` of locus `iloc` in ancestral population

Usage

```
PlotFreqA(path.mcmc, iloc, iall, printit=FALSE, path)
```

Arguments

<code>path.mcmc</code>	Path to output files directory
<code>iloc</code>	Integer number : index of locus
<code>iall</code>	Integer number : index of allele
<code>printit</code>	Logical : if TRUE, figures are also printed
<code>path</code>	Character : Path to directory where figures should be printed

Author(s)

Gilles Guillot

Plotnpop

Plot of number of populations along the MCMC run

Description

Gives a plot of the number of populations along the MCMC run

Usage

```
Plotnpop(path.mcmc, burnin, printit=FALSE, file, format="pdf")
```

Arguments

<code>path.mcmc</code>	Path to output files directory
<code>printit</code>	Logical : if TRUE, figures are also printed
<code>file</code>	Character : Path to file where figures should be printed
<code>format</code>	format of the output file, should be either "ps" or "pdf"
<code>burnin</code>	An integer: number of saved iterations to discard for the representation of the histogram of the chain

Author(s)

Gilles Guillot

References

A spatial statistical model for landscape genetics, Guillot, Estoup, Mortier, Cosson, *Genetics*, 2005
 Guillot, Mortier, Estoup, Geneland : A program for landscape genetics. *Molecular Ecology Notes*, 2005.

Plotntile *Plot of number of tiles along the MCMC run*

Description

Plot of number of tiles in the Poisson-Voronoi tessellation along the MCMC run

Usage

```
Plotntile(path.mcmc, burnin, printit=FALSE, file)
```

Arguments

path.mcmc	Path to output files directory
printit	Logical : if TRUE, figures are also printed
file	Character : Path to file where figures should be printed
burnin	An integer: number of saved iterations to discard for the representation of the histogram of the chain

Author(s)

Gilles Guillot

PlotTessDyn *Plot tessellation along an MCMC run.*

Description

Plot tessellation along an MCMC run.

Usage

```
PlotTessDyn(coordinates, path.mcmc, nxgrid, nygrid)
```

Arguments

coordinates	coordinates
path.mcmc	Path to output files directory
nxgrid	nb of pixels in the x direction
nygrid	nb of pixels in the y direction

Author(s)

Gilles Guillot

PlotTessellation *Maps of posterior probability of membership*

Description

Plots maps of posterior probabilities of population membership for each population

Usage

```
PlotTessellation(coordinates, path.mcmc, printit, path)
```

Arguments

<code>coordinates</code>	Spatial coordinates of individuals. A matrix with 2 columns and one line per individual.
<code>path.mcmc</code>	Character : Path to output files directory
<code>printit</code>	Logical : if TRUE, figures are also printed
<code>path</code>	Character : Path to directory where figures should be printed

Author(s)

Gilles Guillot

See Also

[PostProcessChain](#)

PosteriorMode *Map of mode of posterior distribution of population membership*

Description

Plots a map giving the modal population as a color code for each pixel.

Usage

```
PosteriorMode(coordinates, path.mcmc, plotit=TRUE, format="pdf",  
              printit=FALSE, file, main.title)
```

Arguments

<code>coordinates</code>	Spatial coordinates of individuals. A matrix with 2 columns and one line per individual.
<code>path.mcmc</code>	Path to output files directory
<code>plotit</code>	Logical: if TRUE the map is plotted
<code>printit</code>	Logical : if TRUE, figures are also printed
<code>format</code>	"ps" or "pdf"
<code>file</code>	Character : Path to file where figures should be printed
<code>main.title</code>	Character : Title to appear on top of the graphic

Author(s)

Gilles Guillot

References

- G. Guillot, Estoup, A., Mortier, F. Cosson, J.F. A spatial statistical model for landscape genetics. *Genetics*, 170, 1261-1280, 2005.
- G. Guillot, Mortier, F., Estoup, A. Geneland : A program for landscape genetics. *Molecular Ecology Notes*, 5, 712-715, 2005.

PostProcessChain *Computation for maps of posterior probability of population membership*

Description

Computes posterior probabilities of population membership for each pixel of the spatial domain.

Usage

```
PostProcessChain(coordinates,
path.mcmc, nxdom, nydom, burnin)
```

Arguments

<code>coordinates</code>	Spatial coordinates of individuals. A matrix with 2 columns and one line per individual.
<code>path.mcmc</code>	Path to output files directory
<code>nxdom</code>	Number of pixel for discretization of the spatial domain in the horizontal direction
<code>nydom</code>	Number of pixel for discretization of the spatial domain in the vertical direction

`burnin` Number of iterations of the chain to throw away. **WARNING** : this argument should be given the number of stored iterations (and not the number of computed iterations which differ if `burnin != 1`). If you have `nit=100000` and `thinning=100`, then only 1000 iterations are stored. Then `burnin=10` will throw away 10 stored iterations, namely 100*10 computed iterations.

Value

Posterior probability of population membership for each pixel:
They are written in an ascii file called 'proba.pop.membership.txt'. Two first columns are coordinates of pixels then one column per population (thus `npopmax` values are computed for each pixel). Images in each column of 'proba.pop.membership.txt' are stored starting from the bottom left pixel. First line of 'proba.pop.membership.txt' = bottom left pixel , second line of 'proba.pop.membership.txt' = upward neighbor of the previous pixel, etc...

Posterior probability of population membership for each individual:
They are written in a file named 'proba.pop.membership.indiv.txt'. One line per individuals and $2 + \text{npopmax}$ columns per individual. Two first columns are spatial coordinates.

Label of modal population for pixels and individuals:
They are written in files named 'modal.pop.txt' and 'modal.pop.indiv.txt' respectively. See the example section of function [MCMC](#) to see how they can be added in a plot.

Author(s)

Gilles Guillot

References

G. Guillot, Estoup, A., Mortier, F. Cosson, J.F. A spatial statistical model for landscape genetics. *Genetics*, 170, 1261-1280, 2005.

G. Guillot, Mortier, F., Estoup, A. Geneland : A program for landscape genetics. *Molecular Ecology Notes*, 5, 712-715, 2005.

See Also

[PlotTessellation](#)

PostProcessMultChain

Computation for maps of posterior probability of population membership

Description

Computes posterior probabilities of population membership for each pixel of the spatial domain.

Usage

```
PostProcessMultChain(coordinates, genotypes,
  path.all, nrun, nxdom, nydom, burnin)
```

Arguments

<code>coordinates</code>	Spatial coordinates of individuals. A matrix with 2 columns and one line per individual.
<code>genotypes</code>	Genotypes of individuals. A matrix with one line per individual and 2 columns per locus
<code>path.all</code>	Path to output files directory
<code>nrun</code>	Number of runs
<code>nxdom</code>	Number of pixel for discretization of the spatial domain in the horizontal direction
<code>nydom</code>	Number of pixel for discretization of the spatial domain in the vertical direction
<code>burnin</code>	Number of iterations of the chain to throw away. WARNING : this argument should be given the number of stored iterations (and not the number of computed iterations which differ if <code>burnin != 1</code>). If you have <code>nit=100000</code> and <code>thinning=100</code> , then only 1000 iterations are stored. Then <code>burnin=10</code> will throw away 10 stored iterations, namely 100*10 computed iterations.

Value

Posterior probability of population membership for each pixel:
 They are written in an ascii file called 'proba.pop.membership.txt' (one column per population, `npopmax` values are computed for each pixel. Images in each column of 'proba.pop.membership.txt' are stored column-wise starting from the bottom left pixel. First line of 'proba.pop.membership.txt' = bottom left pixel, second line of 'proba.pop.membership.txt' = upward neighbor of the previous pixel, etc...)

Another file called 'proba.pop.membership.perm.txt' tries to get rid of label switching issues by labelling the population according to a fixed constraint. (This has proved to be usefull with a small number of loci, (e.g. `nloc=3`), for well differentiated populations.)

Posterior probability of population membership for each individual:
 They are written in a file named 'proba.pop.membership.indiv.txt'.

Label of modal population for pixels and individuals:
 They are written in files named 'modal.pop.txt' and 'modal.pop.indiv.txt' respectively. See the example section of function [MCMC](#) to see how they can be added in a plot.

Author(s)

Gilles Guillot

References

G. Guillot, Estoup, A., Mortier, F. Cosson, J.F. A spatial statistical model for landscape genetics. *Genetics*, 170, 1261-1280, 2005.

G. Guillot, Mortier, F., Estoup, A. Geneland : A program for landscape genetics. *Molecular Ecology Notes*, 5, 712-715, 2005.

See Also

[PlotTessellation](#)

<code>show.simdata</code>	<i>Graphical display of data simulated by simdata</i>
---------------------------	---

Description

Graphical display of data simulated by simdata

Usage

```
show.simdata(dataset,
plot.coord = FALSE,
file.plot.coord,
plot.tess = FALSE,
file.plot.tess,
plot.freq.grid = FALSE,
file.plot.freq.grid,
loc.grid = 1,
plot.freq.indiv = FALSE,
file.plot.freq.indiv,
loc.indiv=1,
zlim.freq=c(0,1),
plot.gen = FALSE,
file.plot.gen)
```

Arguments

<code>dataset</code>	An R object produced by function <code>simdata</code>
<code>plot.coord</code>	Logical
<code>file.plot.coord</code>	Character string. Path to the file where the graphic should be stored
<code>plot.tess</code>	Logical
<code>file.plot.tess</code>	Character string. Path to the file where the graphic should be stored
<code>plot.freq.grid</code>	Logical

file.plot.freq.grid	Character string. Path to the file where the graphic should be stored
loc.grid	Vector of integers giving indices of loci for which the map of frequencies is required
plot.freq.indiv	Logical
file.plot.freq.indiv	Character string. Path to the file where the graphic should be stored
loc.indiv	Vector of integers giving indices of loci for which the plot of frequencies at sites of individuals is required
plot.gen	Logical
file.plot.gen	Character string. Path to the file where the graphic should be stored
zlim.freq	A vector of two integers giving the limit of the values for the image of fields of frequencies

Value

No value returned

Author(s)

Gilles Guillot

References

Guillot G. and Estoup A.

simdata

Simulation of georeferenced genotypes under an IBD + barrier model

Description

Simulates coordinates and genotypes for a `npop` populations. Each population is supposed to be under an Isolation by Distance model and different populations are supposed to be separated by impermeable barriers. The barriers are given by a Poisson-Voronoi tessellation.

Usage

```
simdata(nindiv,
        coord.indiv,
        coord.lim=c(0, 1, 0, 1),
        npop,
        rate,
        number.nuclei,
        coord.nuclei,
```

```

color.nuclei,
allele.numbers,
sim.gen=FALSE,
IBD=TRUE,
model="stable",
alpha=1,
beta=1,
gamma=1.8,
sim.quantile=FALSE,
nquantile.var,
mean.quantile,
sd.quantile,
seed.coord,
seed.tess,
seed.freq,
give.tess.grid=FALSE,
give.freq.grid=FALSE,
npix,
comp.Fst=FALSE,
comp.Dsigma2=FALSE,
comp.diff=FALSE,
width,
plot.pairs.borders=FALSE)

```

Arguments

<code>nindiv</code>	Number of individuals
<code>coord.indiv</code>	Coordinates of the individuals
<code>coord.lim</code>	Limits of the geographical domain. The domain is supposed to be rectangular and the limits are given as (abs min, abs max, ord min, ord max)
<code>npop</code>	Number of Populations
<code>rate</code>	Rate of the Poisson process governing the hidden tessellation
<code>number.nuclei</code>	Number of nuclei in the tessellation (if given, then <code>rate</code> is ignored)
<code>coord.nuclei</code>	Coordinates of the nuclei (the number of coordinates of the nuclei given here as a matrix has to comply with <code>number.nuclei</code>)
<code>color.nuclei</code>	Population membership of the nuclei: a vector of integer of length <code>number of nuclei</code> whose values are between 1 and <code>npop</code>
<code>sim.gen</code>	Logical to say whether genetic data should be simulated
<code>allele.numbers</code>	A vector giving the number of alleles observed at each locus
<code>IBD</code>	Logical. If TRUE, then the allele frequencies are simulated according to an IBD model. If FALSE, panmixia is assumed.
<code>model</code>	Model of spatial covariance function used for the underlying Gaussian fields (see documentation of package <code>RandomFields</code> for details)

<code>alpha</code>	Parameter of the spatial Dirichlet vector field of frequencies (a positive real)
<code>beta</code>	Scale parameter of the spatial covariance function used for the underlying Gaussian fields. A positive real number (see documentation of package <code>RandomFields</code> for details)
<code>gamma</code>	Smoothing parameter of spatial covariance function used for the underlying Gaussian fields. (see documentation of package <code>RandomFields</code> for details)
<code>sim.quant</code>	Logical to say whether quantitative data should be simulated
<code>nquant.var</code>	Number of quantitative variables to be simulated
<code>mean.quant</code>	Mean of the quantitative variables in the various groups. A matrix with <code>npop</code> lines and <code>nvar.quant</code> columns
<code>sd.quant</code>	Standard deviation of the quantitative variables in the various groups. A matrix with <code>npop</code> lines and <code>nvar.quant</code> columns
<code>seed.coord</code>	Random seed to initialise the simulation of the coordinates (mostly for debugging)
<code>seed.tess</code>	Random seed to initialise the simulation of the tessellation (mostly for debugging)
<code>seed.freq</code>	Random seed to initialise the simulation of the frequencies (mostly for debugging)
<code>give.freq.grid</code>	Logical to tell whether frequencies on a grid are also returned
<code>give.tess.grid</code>	Logical to tell whether population memberships of pixels on a grid are also returned
<code>npix</code>	A vector of two integers telling how many horizontal and vertical pixel should contain the grid for the graphical representations
<code>comp.Fst</code>	Logical to tell whether <code>Fst</code> , <code>Fis</code> and <code>Fit</code> should be computed
<code>comp.Dsigma2</code>	Logical to tell whether IBD index <code>Dsigma2</code> should be computed
<code>comp.diff</code>	Logical to tell whether the local differentiation across the barriers should be computed
<code>width</code>	Real number specifying the width around the barrier in the computation of its local differentiation
<code>plot.pairs.borders</code>	Logical to tell whether the pairs of individuals coming into the computation of the differentiation of the barriers should be plotted

Value

A list whose components can be seen using `summary`

Author(s)

Arnaud Estoup, Gilles Guillot, Filipe Santos

References

G. Guillot, F. Santos, A. Estoup. Inference in population genetics with Geneland: a sensitivity analysis to spatial sampling scheme, null alleles and isolation by distance. Submitted.

See Also

Function `show.simdata` to make graphical display of simulated data.

Examples

```
## Not run:
dataset <- simdata(nindiv=100,
                  number.nuclei=10,
                  allele.numbers=rep(5,3),
                  model="stable",
                  IBD=TRUE,
                  alpha=1,
                  beta=1,
                  gamma=1,
                  npop=3,
                  give.tess.grid=TRUE,
                  give.freq.grid=TRUE,
                  npix=c(10,10),
                  comp.Fst=TRUE,
                  comp.Dsigma2=TRUE,
                  comp.diff=TRUE,
                  width=0.1,
                  plot.pairs.borders=TRUE)

## End(Not run)
```

simFmodel

Simulation of multi-locus genetic data from the spatial F-model

Description

Simulates multi-locus genotypes and spatial coordinates for individuals belonging to some spatially organised populations.

Usage

```
simFmodel(nindiv, coordinates, coord.lim, number.nuclei,
          coord.nuclei, color.nuclei, nall, npop, freq.model="Uncorrelated", drift, dominance="C
```

Arguments

<code>nindiv</code>	Integer: Number of individuals
<code>coordinates</code>	Matrix (2 rows, <code>nindiv</code> columns) of spatial coordinates of individuals
<code>coord.lim</code>	Vector of limits of spatial domain to be considered (x min, x max, y min, y max)
<code>number.nuclei</code>	Integer: number of nuclei in the Voronoi tessellation
<code>coord.nuclei</code>	Coordinates of nuclei of Voronoi tessellation
<code>color.nuclei</code>	Population labels of the nuclei (vector of integers of size <code>number.nuclei</code>)
<code>nall</code>	Vector of integers giving number of alleles at each locus
<code>npop</code>	Number of populations
<code>freq.model</code>	model for frequencies: "Correlated" or "Uncorrelated"
<code>drift</code>	Vector (of size <code>npop</code>) of drift factors between 0 and 1 (only for the Correlated model)
<code>dominance</code>	A character string "Codominant" or "Dominant". If "Dominant" is chosen, the first allele is treated as a recessive allele and all heterozigous are converted into homozigous for the dominant allele. The presence of the "dominant" allele is coded as 1, the absence of the "dominant" allele is coded as 0.
<code>plots</code>	Logical: if TRUE, spatial coordinates are plotted
<code>ploth</code>	Logical: if TRUE, barplots for allele frequencies are plotted

Details

`number.nuclei` uniform i.i.d points are randomly spread on the rectangular domain. These points generates the so called Voronoi tessellation of the domain in `number.nuclei` polygonal sub-domains. Each polygon is given a color uniformly on $\{1, \text{npop}\}$. The union of polygons of the color `k` gives the domain of population `k`. Then `nindiv` uniform i.i.d points are randomly spread on the domain and stand for the locations of individuals. Allele frequencies in the ancestral population are sampled from independent Dirichlet $D(1, \dots, 1)$. Allele frequencies in the present time population are drawn from Dirichlet distribution whose parameters depend on drift factors `drift` and allele frequencies in the ancestral population. Individual genotypes in each population are drawn from the allele frequencies of the corresponding population assuming Hardy-Weinberg equilibrium and linkage equilibrium.

Value

A list of variables involved in the simulation. The elements of this list are: `coordinates`, `genotypes`, `allele.numbers`, `number.nuclei`, `coord.nuclei`, `color.nuclei`, `frequencies`, `ancestral.frequencies`, `drifts`, `index.nearest.nucleus`

Author(s)

Gilles Guillot

Index

coordinates, [3](#)

EstimateFreqNA, [3](#)

FormatGenotypes, [4](#)

Fstat, [4](#)

Fstat.output, [5](#)

Geneland, [10](#)

Geneland (*Geneland-package*), [1](#)

Geneland-package, [1](#)

Geneland.GUI, [6](#)

genotypes, [6](#)

gl2gp, [7](#)

MCMC, [1](#), [2](#), [7](#), [11](#), [19](#), [20](#)

nullify, [12](#)

PlotDrift, [2](#), [13](#)

PlotFreq, [2](#), [14](#)

PlotFreqA, [2](#), [14](#)

Plotnpop, [2](#), [15](#)

Plotntile, [16](#)

PlotTessDyn, [16](#)

PlotTessellation, [2](#), [17](#), [19](#), [21](#)

PosteriorMode, [17](#)

PostProcessChain, [2](#), [17](#), [18](#)

PostProcessMultChain, [19](#)

show.simdata, [2](#), [21](#)

simdata, [2](#), [22](#)

simFmodel, [2](#), [12](#), [25](#)