Package ‘OriGen’

January 16, 2016

Type Package

Title Fast Spatial Ancestry via Flexible Allele Frequency Surfaces

Version 1.4.3

Author John Michael O Ranola, John Novembre, and Kenneth Lange

Depends maps, ggplot2

Maintainer John Michael O. Ranola <ranolaj@uw.edu>

Description Used primarily for estimates of allele frequency surfaces from point estimates.
It can also place individuals of unknown origin back onto the geographic map with great accuracy.
Additionally, it can place admixed individuals by estimating contributing fractions at each
location on a map. Lastly, it can rank SNPs by their ability to differentiate populations.
See ``Fast Spatial Ancestry via Flexible Allele Frequency Surfaces'' (John Michael Ranola, John
Novembre, Kenneth Lange) in Bioinformatics 2014 for more info.

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-01-16 09:10:00

R topics documented:

`OriGen-package` *Fast Spatial Ancestry via Flexible Allele Frequency Surfaces*

Description

This package primarily estimates allele frequency surfaces from point estimates. It can also place individuals of unknown origin back onto the map with great accuracy. Additionally, it can place admixed individuals by estimating contributing fractions at each location on a map. Lastly, it can rank SNPs by their ability to differentiate populations.

Details

Package:	OriGen
Type:	Package
Version:	0.1
Date:	2013-10-13
License:	GPL2

Index:

- `ConvertPEDData` This function converts Plink PED format files (PED/MAP) along with location files to the input required for OriGen.
- `ConvertUnknownPEDData` This function converts Plink PED format files (PED/MAP) along with location files to the input required for OriGen. This differs from ConvertPEDData by its additional PED formatted input which contains the genotype information for unknown individuals.
- `ConvertMicrosatData` This function converts Microsatellite data files into a format appropriate for analysis.
- `FitOriGenModel` Fits the OriGen model for SNPs and returns the allele frequency surfaces. These surfaces can be plotted with the function `PlotAlleleFrequencySurface`.
- `FitMultinomialModel` Fits the OriGen model for microsatellites and returns the allele frequency surfaces. These surfaces can be plotted with the function `PlotAlleleFrequencySurface`.

- `FitOriGenModelFindUnknowns` Fits the OriGen model for SNPs and places individuals of unknown origin onto the map. This returns probability heat maps for each unknown individual. These heat maps can be plotted with `PlotUnknownHeatMap`. For microsatellite analysis see `FitMultinomialModelFindUnknowns`.
- `FitMultinomialModelFindUnknowns` Fits the OriGen model for microsatellites and places individuals of unknown origin onto the map. This returns probability heat maps for each unknown individual. These heat maps can be plotted with `PlotUnknownHeatMap`. For SNP analysis see `FitOriGenModelFindUnknowns`.
- `FitAdmixedModelFindUnknowns` Fits the OriGen model for SNPs and places unknown individuals who may be admixed onto the map. Instead of returning a probability heat map for each individual, this returns admixture fractions at each location. Note that many locations are 0. This can be plotted with the function `PlotAdmixedSurface`.
- `RankSNPsLRT` This function takes a PED file along with a location file and outputs the likelihood ratio ranking of each SNP along with the LRT statistic and Rosenberg's informativeness for assignment.
- `PlotAlleleFrequencySurface` Plots a specified allele frequency surface from the output of `FitOriGenModel` or `FitMultinomialModel`. Note that all alleles can be plotted by setting AlleleNumber=0.
- `PlotUnknownHeatMap` Plots a specified unknown individuals heat map from the output of `FitOriGenModelFindUnknowns` or `FitMultinomialModelFindUnknowns`.
- `PlotAdmixedSurface` Plots the admixture fractions of a specified individual from the output of `FitAdmixedModelFindUnknowns`.

Author(s)

John Michael Ranola, John Novembre, and Kenneth Lange

Maintainer: John Michael Ranola <ranolaj@uw.edu>

References

Ranola J, Novembre J, and Lange K (2014) Fast Spatial Ancestry via Flexible Allele Frequency Surfaces. Bioinformatics 30(20):2915-22.

`10SNPs.map` *Plink sample PED data*

Description

This data set gives the genetic data in Plink format to be used for testing only. This is to be used with 10SNPs.ped and Locations.txt.

Format

A Plink PED format file.

`10SNPs.ped` *Plink sample PED data*

Description

This data set gives the genetic data in Plink format to be used for testing only. This is to be used with 10SNPs.map and Locations.txt.

Format

A Plink PED format file.

`CalcFractionsMultiLoglik` *Calculates the loglikelihood for placing a sample 100 percent back into its own sample site*

Description

This function takes the UnknownDataArray which contains allelelic information for individuals WITHIN a single sample site and calculates the resulting fraction loglikelihood for placing all individuals 100 percent back into their site

Usage

```
CalcFractionsMultiLoglik(UnknownDataArray,LambdaParameter=100)
```

Arguments

`UnknownDataArray`
An array showing the unknown individuals genetic data. It lists the two allele numbers of the unknown data. The dimension of this array is [NumberUnknowns,2,NumberLoci].

`LambdaParameter`
This is a real precision parameter weighting the admixture fractions algorithm. For the most part, this does not need to be changed as it seems to only affect the time to convergence. Default is 100.

Value

An array giving the penalized loglikelihood resulting from placing each unknown individual 100 percent back into his own sample site. The length of this array is [NumberUnknowns].

Author(s)

John Michael Ranola, John Novembre, and Kenneth Lange

References

Ranola J, Novembre J, Lange K (2014) Fast Spatial Ancestry via Flexible Allele Frequency Surfaces. Bioinformatics, in press.

See Also

`FitMultinomialAdmixedModelFindUnknowns` for getting loglikelihoods of unknown individuals placed into chosen regions.

Examples

```
#Data generation
NumberUnknowns = 50
NumberLoci = 10
TestUnknownDataArray=array(sample(1:5,2*NumberUnknowns*NumberLoci,replace=TRUE)
,dim=c(NumberUnknowns,2,NumberLoci))

CalcFractionsMultiLoglik(TestUnknownDataArray)
```

`ConvertMicrosatData` *Microsatellite file conversion for known and unknown data*

Description

This function converts two Microsatellite data files (one for the genotypes and one for locations) into the data format required for OriGen.

Usage

```
ConvertMicrosatData(DataFileName,LocationFileName)
```

Arguments

`DataFileName`	Name of file containing the genotypes of the various locations. The columns here would be LocationName, LocationNumber, Locus1, Locus2, etc. Each individual would take up 2 rows (one for each allele) with the same LocationName and LocationNumber. The value under Locus would be the length of the allele of that individual. Note that unknown individuals should have location number "-1".

`LocationFileName`
: Space or tab delimited text file with the location information for the individuals. The columns are LocationName, LocationNumber, Latitude, and Longitude. Note that the first two columns must be in the same order as the FileName.

Value

List with the following components:

`DataArray` An array giving the number alleles grouped by sample sites for each locus. The dimension of this array is [MaxAlleles,SampleSites,NumberSNPs].

`SampleCoordinates`
This is an array which gives the longitude and latitude of each of the found sample sites. The dimension of this array is [SampleSites,2], where the second dimension represents longitude and latitude respectively.

`AllelesAtLocus` This shows the integer vector of alleles found at each locus.

`MaxAlleles` This shows the maximum of AllelesAtLocus. The maximum number of alleles at all loci.

`SampleSites` This shows the integer number of sample sites found.

`NumberLoci` This shows the integer number of loci found.

`NumberUnknowns` This is an integer value showing the number of unknowns found.

`UnknownDataArray`
An array showing the unknown individuals genetic data. The dimension of this array is [NumberUnknowns,2,NumberLoci].

`LocationNames` This is a list of all the LocationNames (The first column of the input files).

`DataFileName` This shows the inputted DataFileName.

`LocationFileName`
This shows the inputted LocationFileName.

Author(s)

John Michael Ranola, John Novembre, and Kenneth Lange

References

Ranola J, Novembre J, Lange K (2014) Fast Spatial Ancestry via Flexible Allele Frequency Surfaces. Bioinformatics, in press.

See Also

`ConvertMicrosatData` for converting Microsatellite data files into a format appropriate for analysis,

`ConvertPEDData` for converting Plink PED files into a format appropriate for analysis,

`FitMultinomialModel` for fitting allele surfaces to the converted Microsatellite data,

`PlotAlleleFrequencySurface` for a quick way to plot the resulting allele frequency surfaces from `FitOriGenModel` or `FitMultinomialModel`,;

Examples

```
#Note that sample files MicrosatTrialDataSmall.txt and
#LocationTrialDataSmall.txt are included in data for formatting.
#Note that this was done to allow inclusion of the test data in the package.

## Not run: MicrosatDataSmall=ConvertMicrosatData("MicrosatTrialDataSmall.txt",
"LocationTrialDataSmall.txt")
## End(Not run)
## Not run: str(MicrosatDataSmall)
## Not run: MicrosatAnalysisSmall=FitMultinomialModel(MicrosatDataSmall$DataArray,
MicrosatDataSmall$SampleCoordinates,MaxGridLength=20)
## End(Not run)
## Not run: str(MicrosatAnalysisSmall)
## Not run: PlotAlleleFrequencySurface(MicrosatAnalysisSmall)
```

ConvertPEDData *Plink PED file conversion*

Description

This function converts a Plink PED/MAP file into the data format required for OriGen.

Usage

```
ConvertPEDData(PlinkFileName,LocationFileName)
```

Arguments

PlinkFileName Base name of Plink PED file (i.e. without ".ped" or ".map")

LocationFileName
Space or tab delimited text file with Longitude and Latitude coordinates for each individual listed in the 4th and 5th columns respectively. Note that rows should correspond to the individuals in the Plink File. Also, this file should have a header row.

Value

List with the following components:

DataArray An array giving the number of major/minor SNPs (defined as the most occuring in the dataset) grouped by sample sites for each SNP. The dimension of this array is [2,SampleSites,NumberSNPs].

`SampleCoordinates`
This is an array which gives the longitude and latitude of each of the found sample sites. The dimension of this array is [SampleSites,2], where the second dimension represents longitude and latitude respectively.

`PlinkFileName` This shows the inputted PlinkFileName with ".ped" attached.

`LocationFile` This shows the inputted LocationFileName.

`SampleSites` This shows the integer number of sample sites found.

`NumberSNPs` This shows the integer number of SNPs found.

Author(s)

John Michael Ranola, John Novembre, and Kenneth Lange

References

Ranola J, Novembre J, Lange K (2014) Fast Spatial Ancestry via Flexible Allele Frequency Surfaces. Bioinformatics, in press.

See Also

`FitOriGenModel` for fitting allele surfaces to the converted data,

`PlotAlleleFrequencySurface` for a quick way to plot the resulting allele frequency surfaces from `FitOriGenModel`

`ConvertUnknownPEDData` for converting a known and unknown PED files (2 separate files) into the format required for OriGen (Note that this is what you want if you want to place unknown individuals back on the map);

Examples

```
#Note that Plink files "10SNPs.ped", "10SNPs.map" and also "Locations.txt"
#are included in the data folder of the OriGen package with ".txt" appended to the Plink files.
#Please remove ".txt" and navigate to the appropriate location
#before testing the following commands.
#Note that this was done to allow inclusion of the test data in the package.

## Not run: trials=ConvertPEDData("10SNPs","Locations.txt")
## Not run: str(trials)
MaxGridLength=20
RhoParameter=10
## Not run: trials2=FitOriGenModel(trials$DataArray,trials$SampleCoordinates,
MaxGridLength,RhoParameter)
## End(Not run)
## Not run: PlotAlleleFrequencySurface(trials2)
```

ConvertUnknownPEDData *Plink PED file conversion for known and unknown data*

Description

This function converts two Plink PED/MAP files (one for the known samples and one with unknown locations) into the data format required for OriGen.

Usage

```
ConvertUnknownPEDData(PlinkFileName,LocationFileName,PlinkUnknownFileName)
```

Arguments

`PlinkFileName` Base name of Plink PED file (i.e. without ".ped" or ".map") containing the individuals with known locations.

`LocationFileName` Space or tab delimited text file with Longitude and Latitude coordinates for each individual listed in the 4th and 5th columns respectively. Note that rows should correspond to the individuals in the Plink File. Also, this file should have a header row.

`PlinkUnknownFileName` Base name of Plink PED file (i.e. without ".ped" or ".map") containing the individuals with unknown locations.

Value

List with the following components:

`DataArray` An array giving the number of major/minor SNPs (defined as the most occuring in the dataset) grouped by sample sites for each SNP. The dimension of this array is [2,SampleSites,NumberSNPs].

`SampleCoordinates` This is an array which gives the longitude and latitude of each of the found sample sites. The dimension of this array is [SampleSites,2], where the second dimension represents longitude and latitude respectively.

`PlinkFileName` This shows the inputted PlinkFileName with ".ped" attached.

`LocationFile` This shows the inputted LocationFileName.

`SampleSites` This shows the integer number of sample sites found.

`NumberSNPs` This shows the integer number of SNPs found.

`UnknownPEDFile` This shows the inputted PED file for the unknown individuals.

`NumberUnknowns` This is an integer value showing the number of unknowns found in the UnknownPEDFile.

UnknownData An array showing the unknown individuals genetic data. The dimension of this array is [NumberUnknowns,NumberSNPs].

Membership This is an integer valued vector showing the group number of each member of the inputted known group. The dimension of this array is [NumberKnown].

NumberKnown This is an integer value showing the number of known found in the PlinkFileName.

Author(s)

John Michael Ranola, John Novembre, and Kenneth Lange

References

Ranola J, Novembre J, Lange K (2014) Fast Spatial Ancestry via Flexible Allele Frequency Surfaces. Bioinformatics, in press.

See Also

ConvertUnknownPEDData for converting two Plink PED files (known and unknown)into a format appropriate for analysis,

FitOriGenModelFindUnknowns for fitting allele surfaces to the converted data and finding the locations of the given unknown individuals,

PlotUnknownHeatMap for a quick way to plot the resulting unknown heat map surfaces from FitOriGenModelFindUnknowns,;

Examples

```
#Note that Plink files "10SNPs.ped", "10SNPs.map" and also "Locations.txt"
#are included in the data folder of the OriGen package with ".txt" appended to the Plink files.
#Please remove ".txt" and navigate to the appropriate location
#before testing the following commands.
#Note that this was done to allow inclusion of the test data in the package.

## Not run: trials3=ConvertUnknownPEDData("10SNPs","Locations.txt",""10SNPs"")
## Not run: str(trials3)
MaxGridLength=30
RhoParameter=10
## Not run: trials4=FitOriGenModelFindUnknowns(trials3$DataArray,trials3$SampleCoordinates,
trials3$UnknownData[1:2,],MaxGridLength,RhoParameter)
## End(Not run)
## Not run: PlotUnknownHeatMap(trials4,UnknownNumber=1,MaskWater=TRUE)
```

FindRhoParameterCrossValidation

Finds the appropriate value of the Rho parameter via crossvalidation.

Description

This function finds the appropriate value of the tuning constant, RhoParameter, via a leave one sample site out cross validation.

Usage

```
FindRhoParameterCrossValidation(PlinkFileName,LocationFileName,MaxIts=6,MaxGridLength=20)
```

Arguments

`PlinkFileName` Base name of Plink PED file (i.e. without ".ped" or ".map")

`LocationFileName` Space or tab delimited text file with Longitude and Latitude coordinates for each individual listed in the 4th and 5th columns respectively. Note that rows should correspond to the individuals in the Plink File. Also, this file should have a header row.

`MaxIts` An integer giving the number of iterations before selecting the rho parameter. Note that this is a long process so it is best to start small.

`MaxGridLength` An integer giving the maximum number of boxes to fill the longer side of the region. Note that computation time increases quadratically as this number increases, but this number also should be high enough to separate different sample sites otherwise they will be binned together as a single site.

Value

List with the following components:

`PlinkFileName` This shows the inputted PlinkFileName with ".ped" attached.

`LocationFile` This shows the inputted LocationFileName.

`NumberSNPs` This shows the integer number of SNPs found.

`MaxIts` An integer giving the number of iterations before selecting the rho parameter. Note that this is a long process so it is best to start small. This number is inputted into the function.

`MaxGridLength` An integer giving the maximum number of boxes to fill the longer side of the region. Note that computation time increases quadratically as this number increases, but this number also should be high enough to separate different sample sites otherwise they will be binned together as a single site. This number was part of the inputs.

RhoVector An array giving the tested values of RhoParameter along with the resulting cross validation results where lower is better.

GridLength An array giving the number of longitudinal and latitudinal divisions. The dimension of this array is [2], where the first number is longitude and the second is latitude.

RhoParameter A real value showing the best RhoParameter value found.

SampleSites This shows the integer number of sample sites found.

Author(s)

John Michael Ranola, John Novembre, and Kenneth Lange

References

Ranola J, Novembre J, Lange K (2014) Fast Spatial Ancestry via Flexible Allele Frequency Surfaces. Bioinformatics, in press.

See Also

ConvertPEDData for converting Plink PED files into a format appropriate for analysis,

FitOriGenModel for fitting allele surfaces to the converted data,

PlotAlleleFrequencySurface for a quick way to plot the resulting allele frequency surfaces from FitOriGenModel,

ConvertUnknownPEDData for converting two Plink PED files (known and unknown)into a format appropriate for analysis,

FitOriGenModelFindUnknowns for fitting allele surfaces to the converted data and finding the locations of the given unknown individuals,

PlotUnknownHeatMap for a quick way to plot the resulting unknown heat map surfaces from FitOriGenModelFindUnknowns,;

FitAdmixedModelFindUnknowns for fitting allele surfaces to the converted data and finding the locations of the given unknown individuals who may be admixed,

PlotAdmixedSurface for a quick way to plot the resulting admixture surfaces from FitAdmixedFindUnknowns,

RankSNPsLRT for reducing the number of SNPs using a likelihood ratio test criteria or informativeness for assignment,

Examples

```
#Note that Plink files "10SNPs.ped", "10SNPs.map" and also "Locations.txt"
#are included in the data folder of the OriGen package.
#Please navigate to the appropriate location before testing
#the following commands.

## Not run: trials5=FindRhoParameterCrossValidation("10SNPs","Locations.txt",
MaxIts=4,MaxGridLength=20)
## End(Not run)
## Not run: trials5
```

FitAdmixedModelFindUnknowns

Fit the OriGen model and place unknown individuals who may be admixed

Description

This function fits the OriGen model and places individuals of unknown origins who may be admixed. This function estimates admixture fractions at each location rather than the probability of coming from each location.

Usage

```
FitAdmixedModelFindUnknowns(DataArray,SampleCoordinates,UnknownData,
MaxGridLength=20,RhoParameter=10,LambdaParameter=100,MaskWater=TRUE)
```

Arguments

`DataArray` An array giving the number of major/minor SNPs (defined as the most occuring in the dataset) grouped by sample sites for each SNP. The dimension of this array is [2,SampleSites,NumberSNPs].

`SampleCoordinates` This is an array which gives the longitude and latitude of each of the found sample sites. The dimension of this array is [SampleSites,2], where the second dimension represents longitude and latitude respectively.

`UnknownData` An array showing the unknown individuals genetic data. The dimension of this array is [NumberUnknowns,NumberSNPs].

`MaxGridLength` An integer giving the maximum number of boxes to fill the longer side of the region. Note that computation time increases quadratically as this number increases, but this number also should be high enough to separate different sample sites otherwise they will be binned together as a single site.

`RhoParameter` This is a real precision parameter weighting the amount of smoothing in the alllele frequency surface. A higher value flattens out the surface while a lower value allows for more fluctuations. The default value of 10 was used in our analysis and should prove a good starting point. To choose a value by crossvalidation please see `FindRhoParameterCrossValidation`

`LambdaParameter` This is a real precision parameter weighting the admixture fractions algorithm. For the most part, this does not need to be changed as it seems to only affect the time to convergence.

`MaskWater` Logical value that if true removes water from the plotted regions.

Value

List with the following components:

`AdmixtureFractions`
An array giving the admixture fraction from the given location. In other words this is the fractional contribution of the location to the unknown individuals genetic data. The dimension of this array is [NumberLongitudeDivisions, NumberLatitudeDivisions, NumberUnknowns], where either NumberLongitudeDivisions or NumberLatitudeDivisions is equal to MaxGridLength(an input to this function) and the other is scaled so that the geodesic distance between points horizontally and vertically is equal.

`DataArray` An array giving the number of major/minor SNPs (defined as the most occuring in the dataset) grouped by sample sites for each SNP. The dimension of this array is [2, SampleSites, NumberSNPs].

`NumberSNPs` This shows the integer number of SNPs found.

`GridLength` An array giving the number of longitudinal and latitudinal divisions. The dimension of this array is [2], where the first number is longitude and the second is latitude.

`RhoParameter` A real value showing the inputted RhoParameter value.

`SampleSites` This shows the integer number of sample sites found.

`MaxGridLength` An integer giving the maximum number of boxes to fill the longer side of the region. Note that computation time increases quadratically as this number increases, but this number also should be high enough to separate different sample sites otherwise they will be binned together as a single site. This number was part of the inputs.

`SampleCoordinates`
This is an array which gives the longitude and latitude of each of the found sample sites. The dimension of this array is [SampleSites,2], where the second dimension represents longitude and latitude respectively.

`GridCoordinates`
An array showing the corresponding coordinates for each longitude and latitude division. The dimension of this array is [2,MaxGridLength], with longitude coordinates coming first and latitude second. Note that one of these rows may not be filled entirely. The associated output GridLength should be used to find the lengths of the two rows. Rows not filled in entirely will contain zeroes at the end.

`NumberUnknowns` This is an integer value showing the number of unknowns found in the UnknownPEDFile.

`UnknownData` An array showing the unknown individuals genetic data. The dimension of this array is [NumberUnknowns,NumberSNPs].

`IsLand` This is a logical valued array that is TRUE when the given coordinates are over land and FALSE when over water. The dimension of this array is [GridLength[1],GridLength[2]].

Author(s)

John Michael Ranola, John Novembre, and Kenneth Lange

References

Ranola J, Novembre J, Lange K (2014) Fast Spatial Ancestry via Flexible Allele Frequency Surfaces. Bioinformatics, in press.

See Also

`ConvertUnknownPEDData` for converting two Plink PED files (known and unknown)into a format appropriate for analysis,

`FitOriGenModelFindUnknowns` for fitting allele surfaces to the converted data and finding the locations of the given unknown individuals,

`PlotUnknownHeatMap` for a quick way to plot the resulting unknown heat map surfaces from `FitOriGenModelFindUnknowns`,;

`FitAdmixedModelFindUnknowns` for fitting allele surfaces to the converted data and finding the locations of the given unknown individuals who may be admixed,

`PlotAdmixedSurface` for a quick way to plot the resulting admixture surfaces from `FitAdmixedFindUnknowns`,

Examples

```
#this example not run because it takes longer than 5 secs
#note - type example(FunctionName, run.dontrun=TRUE) to run the example where FunctionName is
#the name of the function
## Not run:

#Data generation
SampleSites=10
NumberSNPs=4
TestData=array(sample(2*(1:30),2*SampleSites*NumberSNPs,replace=TRUE),
dim=c(2,SampleSites,NumberSNPs))
#Europe is about -9 to 38 and 34 to 60
TestCoordinates=array(0,dim=c(SampleSites,2))
TestCoordinates[,1]=runif(SampleSites,-9,38)
TestCoordinates[,2]=runif(SampleSites,34,60)

#This code simulates the number of major alleles the unknown individuals have.
NumberUnknowns=2
TestUnknowns=array(sample(0:2,NumberUnknowns*NumberSNPs,replace=TRUE),
dim=c(NumberUnknowns,NumberSNPs))

#Fitting the admixed model
#MaxGridLength is the maximum number of boxes allowed to span the region in either direction
#Note that MaxGridLength is reduced here to allow the example to run in less than 5 secs
#RhoParameter is a tuning constant
print("MaxGridLength is intentionally set really low for fast examples.
Meaningful results will most likely require a higher value.")
trials6=FitAdmixedModelFindUnknowns(TestData,TestCoordinates,
TestUnknowns,MaxGridLength=8,RhoParameter=10)

#Plots the admixed surface disregarding fractions less than 0.01
PlotAdmixedSurface(trials6)

## End(Not run)
```

`FitMultinomialAdmixedModelFindUnknowns`

Fit the multinomial OriGen model and place unknown individuals who may be admixed

---

Description

This function fits the multinomial OriGen model and places individuals of unknown origins who may be admixed. This function estimates admixture fractions at each location rather than the probability of coming from each location.

Usage

```
FitMultinomialAdmixedModelFindUnknowns(DataArray,SampleCoordinates,UnknownDataArray,
MaxGridLength=20,RhoParameter=10,LambdaParameter=100,MaskWater=TRUE,NumberLoci=-1)
```

Arguments

`DataArray` An array giving the number alleles grouped by sample sites for each locus. The dimension of this array is [MaxAlleles,SampleSites,NumberSNPs].

`SampleCoordinates` This is an array which gives the longitude and latitude of each of the found sample sites. The dimension of this array is [SampleSites,2], where the second dimension represents longitude and latitude respectively.

`UnknownDataArray` This is an array which gives the alleles for the individuals of unknown origin. The dimension of this array is [NumberUnknowns,2,NumberLoci], where 2 represents to 2 alleles each individual has at each locus. Note that these should not be allele lengths but rather the allele number matching the dimension in DataArray. Note that 0 or negative values here indicate unknown alleles and it is assumed that both are either known or unknown.

`MaxGridLength` An integer giving the maximum number of boxes to fill the longer side of the region. Note that computation time increases quadratically as this number increases, but this number also should be high enough to separate different sample sites otherwise they will be binned together as a single site.

`RhoParameter` This is a real precision parameter weighting the amount of smoothing in the alllele frequency surface. A higher value flattens out the surface while a lower value allows for more fluctuations. The default value of 10 was used in our analysis and should prove a good starting point. To choose a value by crossvalidation please see `FindRhoParameterCrossValidation`

`LambdaParameter`
: This is a real precision parameter weighting the admixture fractions algorithm. For the most part, this does not need to be changed as it seems to only affect the time to convergence.

`MaskWater` If TRUE, this logical parameter restricts the heat maps to land areas only.

`NumberLoci` An integer value giving the number of loci to use in the analysis. If set to -1, which is the default, it uses all loci.

Value

List with the following components:

`AdmixtureFractions`
: An array giving the admixture fraction from the given location. In other words this is the fractional contribution of the location to the unknown individuals genetic data. The dimension of this array is [NumberLongitudeDivisions, NumberLatitudeDivisions, NumberUnknowns], where either NumberLongitudeDivisions or NumberLatitudeDivisions is equal to MaxGridLength(an input to this function) and the other is scaled so that the geodesic distance between points horizontally and vertically is equal.

`DataArray` An array giving the number alleles grouped by sample sites for each locus. The dimension of this array is [MaxAlleles,SampleSites,NumberSNPs].

`NumberLoci` This shows the integer number of loci found.

`GridLength` An array giving the number of longitudinal and latitudinal divisions. The dimension of this array is [2], where the first number is longitude and the second is latitude.

`RhoParameter` A real value showing the inputted RhoParameter value.

`SampleSites` This shows the integer number of sample sites found.

`MaxGridLength` An integer giving the maximum number of boxes to fill the longer side of the region. Note that computation time increases quadratically as this number increases, but this number also should be high enough to separate different sample sites otherwise they will be binned together as a single site. This number was part of the inputs.

`SampleCoordinates`
: This is an array which gives the longitude and latitude of each of the found sample sites. The dimension of this array is [SampleSites,2], where the second dimension represents longitude and latitude respectively.

`GridCoordinates`
: An array showing the corresponding coordinates for each longitude and latitude division. The dimension of this array is [2,MaxGridLength], with longitude coordinates coming first and latitude second. Note that one of these rows may not be filled entirely. The associated output GridLength should be used to find the lengths of the two rows. Rows not filled in entirely will contain zeroes at the end.

`NumberUnknowns` This is an integer value showing the number of unknowns found.

UnknownDataArray
: This is an array which gives the alleles for the individuals of unknown origin. The dimension of this array is [NumberUnknowns,2,NumberLoci], where 2 represents to 2 alleles each individual has at each locus. Note that these should not be allele lengths but rather the allele number matching the dimension in DataArray.

IsLand
: This is a logical valued array that is TRUE when the given coordinates are over land and FALSE when over water. The dimension of this array is [GridLength[1],GridLength[2]].

MaxAlleles
: An integer giving the maximum number of alleles across all loci.

Author(s)

John Michael Ranola, John Novembre, and Kenneth Lange

References

Ranola J, Novembre J, Lange K (2014) Fast Spatial Ancestry via Flexible Allele Frequency Surfaces. Bioinformatics, in press.

See Also

ConvertUnknownPEDData for converting two Plink PED files (known and unknown)into a format appropriate for analysis,

FitOriGenModelFindUnknowns for fitting allele surfaces to the converted data and finding the locations of the given unknown individuals,

PlotUnknownHeatMap for a quick way to plot the resulting unknown heat map surfaces from FitOriGenModelFindUnknowns,;

FitMultinomialAdmixedModelFindUnknowns for fitting allele surfaces to the converted data and finding the locations of the given unknown individuals who may be admixed,

PlotAdmixedSurface for a quick way to plot the resulting admixture surfaces from FitAdmixedFindUnknowns,

Examples

```
#this example not run because it takes longer than 5 secs
#note - type example(FunctionName, run.dontrun=TRUE) to run the example where FunctionName is
#the name of the function

## Not run:

##Data generation
SampleSites=5
NumberLoci=3
MaxAlleles=2
if(MaxAlleles==2){
NumberAllelesAtEachLocus=rep(2,NumberLoci)
}else{
```

```
NumberAllelesAtEachLocus=sample(2:MaxAlleles,NumberLoci,replace=TRUE)
}
TestData=array(0,dim=c(MaxAlleles,SampleSites,NumberLoci))
for(i in 1:NumberLoci){
for(j in 1:NumberAllelesAtEachLocus[i]){
TestData[j,,i]=sample(1:10,SampleSites,replace=TRUE)
}
}
##This data is simulated in Europe which is around Longitude -9 to 38 and Latitude 34 to 60
TestCoordinates=array(0,dim=c(SampleSites,2))
TestCoordinates[,1]=runif(SampleSites,-9,38)
TestCoordinates[,2]=runif(SampleSites,34,60)

##This simulates the unknown data
NumberUnknowns=2
UnknownData=array(0,dim=c(NumberUnknowns,2,NumberLoci))
for(i in 1:NumberUnknowns){
for(j in 1:NumberLoci){
UnknownData[i,,j]=sample(1:NumberAllelesAtEachLocus[j],2)
}
}

##MaxGridLength is the maximum number of boxes allowed
##to span the region in either direction
##Note that this number was reduced to allow the example to run in less than 5 secs
##RhoParameter is a tuning constant
print("MaxGridLength is intentionally set really low for fast examples.
Meaningful results will most likely require a higher value.")

##Fits the allele frequency surfaces only
#SurfaceTrials=FitMultinomialModel(TestData,TestCoordinates,
#MaxGridLength=20,RhoParameter=10)
#str(SurfaceTrials)
##Plotting the model
#PlotAlleleFrequencySurface(SurfaceTrials,LocusNumber=1,AlleleNumber=1,
# MaskWater=TRUE,Scale=FALSE)

##You can generate heatmaps of unknown individual's placements from with the allele
##surfaces using GenerateHeatMaps or use FitMultinomialModelFindUnknowns
#HeatMapTrials=GenerateHeatMaps(SurfaceTrials,UnknownData,NumberLoci=NumberLoci)
##Plotting the unknown heat map
#PlotUnknownHeatMap(HeatMapTrials,UnknownNumber=1,MaskWater=TRUE)

##Fitting the model and finding the unknown locations
#UnknownTrials=FitMultinomialModelFindUnknowns(TestData,TestCoordinates,
# UnknownData,MaxGridLength=20,RhoParameter=10)
#str(UnknownTrials)
##Plotting the unknown heat map
#PlotUnknownHeatMap(UnknownTrials,UnknownNumber=1,MaskWater=TRUE)

##Fitting the admixed model
##Note that MaxGridLength is intentionally set unusably low so that the example
##runs in under 5 seconds.  The default value of 20 is more reasonable in general
```

```
AdmixedTrials=FitMultinomialAdmixedModelFindUnknowns(TestData,TestCoordinates,
UnknownData,MaxGridLength=8,RhoParameter=10,MaskWater=TRUE)

##Plots the admixed surface disregarding fractions less than 0.01
PlotAdmixedSurface(AdmixedTrials,UnknownNumber=1)


## End(Not run)
```

FitMultinomialModel *Fit OriGen allele frequency surfaces*

Description

This function fits allele frequency surfaces to microsatellite data.

Usage

```
FitMultinomialModel(DataArray,SampleCoordinates,MaxGridLength=20,RhoParameter=10)
```

Arguments

`DataArray` An array giving the number of alleles grouped by sample sites for each SNP. The dimension of this array is [MaxAlleles,SampleSites,NumberLoci].

`SampleCoordinates` This is an array which gives the longitude and latitude of each of the found sample sites. The dimension of this array is [SampleSites,2], where the second dimension represents longitude and latitude respectively.

`MaxGridLength` An integer giving the maximum number of boxes to fill the longer side of the region. Note that computation time increases quadratically as this number increases, but this number also should be high enough to separate different sample sites otherwise they will be binned together as a single site.

`RhoParameter` This is a real precision parameter weighting the amount of smoothing. A higher value flattens out the surface while a lower value allows for more fluctuations. The default value of 10 was used in our analysis and should prove a good starting point. To choose a value by crossvalidation please see `FindRhoParameterCrossValidation`

Value

List with the following components:

`AlleleFrequencySurfaces` An array giving the allele frequency for each allele, each coordinate, and each SNP. The dimension of this array is [MaxAlleles, NumberLoci, NumberLongitudeDivisions, NumberLatitudeDivisions], where either NumberLongitudeDivisions or NumberLatitudeDivisions is equal to MaxGridLength(an input to this function) and the other is scaled so that the geodesic distance between points horizontally and vertically is equal.

`DataArray` An array giving the number alleles grouped by sample sites for each locus. The dimension of this array is [MaxAlleles,SampleSites,NumberSNPs].

`RhoParameter` A real value showing the inputted RhoParameter value.

`SampleSites` This shows the integer number of sample sites found.

`GridLength` An array giving the number of longitudinal and latitudinal divisions. The dimension of this array is [2], where the first number is longitude and the second is latitude.

`MaxGridLength` An integer giving the maximum number of boxes to fill the longer side of the region. Note that computation time increases quadratically as this number increases, but this number also should be high enough to separate different sample sites otherwise they will be binned together as a single site. This number was part of the inputs.

`MaxAlleles` This shows the maximum of AllelesAtLocus. The maximum number of alleles at all loci.

`NumberLoci` This shows the integer number of loci found.

`SampleCoordinates`

This is an array which gives the longitude and latitude of each of the found sample sites. The dimension of this array is [SampleSites,2], where the second dimension represents longitude and latitude respectively.

`AllelesAtLocus` This shows the integer vector of alleles found at each locus.

`GridCoordinates`

An array showing the corresponding coordinates for each longitude and latitude division. The dimension of this array is [2,MaxGridLength], with longitude coordinates coming first and latitude second. Note that one of these rows may not be filled entirely. The associated output GridLength should be used to find the lengths of the two rows. Rows not filled in entirely will contain zeroes at the end.

Author(s)

John Michael Ranola, John Novembre, and Kenneth Lange

References

Ranola J, Novembre J, Lange K (2014) Fast Spatial Ancestry via Flexible Allele Frequency Surfaces. Bioinformatics, in press.

See Also

`ConvertMicrosatData` for converting Microsatellite data files into a format appropriate for analysis,

`ConvertPEDData` for converting Plink PED files into a format appropriate for analysis,

`FitOriGenModel` for fitting allele surfaces to the converted SNP data, FitMultinomialModel for fitting allele surfaces to the converted Microsatellite data,

PlotAlleleFrequencySurface for a quick way to plot the resulting allele frequency surfaces from FitOriGenModel or FitMultinomialModel,;

Examples

```
#These examples are not run because they take a little more than 5 secs
#note - type example(FunctionName, run.dontrun=TRUE) to run the example where FunctionName is
#the name of the function

## Not run:

##Data generation
SampleSites=10
NumberLoci=4
MaxAlleles=4
if(MaxAlleles==2){
NumberAllelesAtEachLocus=rep(2,NumberLoci)
}else{
NumberAllelesAtEachLocus=sample(2:MaxAlleles,NumberLoci,replace=TRUE)
}
TestData=array(0,dim=c(MaxAlleles,SampleSites,NumberLoci))
for(i in 1:NumberLoci){
for(j in 1:NumberAllelesAtEachLocus[i]){
TestData[j,,i]=sample(1:10,SampleSites,replace=TRUE)
}
}
##This data is simulated in Europe which is around Longitude -9 to 38 and Latitude 34 to 60
TestCoordinates=array(0,dim=c(SampleSites,2))
TestCoordinates[,1]=runif(SampleSites,-9,38)
TestCoordinates[,2]=runif(SampleSites,34,60)

##This simulates the unknown data
NumberUnknowns=2
UnknownData=array(0,dim=c(NumberUnknowns,2,NumberLoci))
for(i in 1:NumberUnknowns){
for(j in 1:NumberLoci){
UnknownData[i,,j]=sample(1:NumberAllelesAtEachLocus[j],2)
}
}

##MaxGridLength is the maximum number of boxes allowed
##to span the region in either direction
##Note that this number was reduced to allow the example to run in less than 5 secs
##RhoParameter is a tuning constant
print("MaxGridLength is intentionally set really low for fast examples.
Meaningful results will most likely require a higher value.")

##Fits the allele frequency surfaces only
SurfaceTrials=FitMultinomialModel(TestData,TestCoordinates,
MaxGridLength=20,RhoParameter=10)
```

```
str(SurfaceTrials)
##Plotting the model
PlotAlleleFrequencySurface(SurfaceTrials,LocusNumber=1,AlleleNumber=1,
MaskWater=TRUE,Scale=FALSE)

##You can generate heatmaps of unknown individual's placements from with the allele
##surfaces using GenerateHeatMaps or use FitMultinomialModelFindUnknowns
#HeatMapTrials=GenerateHeatMaps(SurfaceTrials,UnknownData,NumberLoci=NumberLoci)
##Plotting the unknown heat map
#PlotUnknownHeatMap(HeatMapTrials,UnknownNumber=1,MaskWater=TRUE)

##Fitting the model and finding the unknown locations
#UnknownTrials=FitMultinomialModelFindUnknowns(TestData,TestCoordinates,
# UnknownData,MaxGridLength=20,RhoParameter=10)
#str(UnknownTrials)
##Plotting the unknown heat map
#PlotUnknownHeatMap(UnknownTrials,UnknownNumber=1,MaskWater=TRUE)

##Fitting the admixed model
#AdmixedTrials=FitMultinomialAdmixedModelFindUnknowns(TestData,TestCoordinates,
# UnknownData,MaxGridLength=10,RhoParameter=10)
##Plots the admixed surface disregarding fractions less than 0.01
#PlotAdmixedSurface(AdmixedTrials,UnknownNumber=1)


## End(Not run)
```

FitMultinomialModelFindUnknowns

Fit OriGen microsatellite allele frequency surfaces

Description

This function fits allele frequency surfaces to microsatellite data and then finds locations for unknown individuals..

Usage

```
FitMultinomialModelFindUnknowns(DataArray,SampleCoordinates,UnknownDataArray,
MaxGridLength=20,RhoParameter=10,MaskWater=TRUE)
```

Arguments

`DataArray`	An array giving the number of alleles grouped by sample sites for each SNP. The dimension of this array is [MaxAlleles,SampleSites,NumberLoci].

`SampleCoordinates`
This is an array which gives the longitude and latitude of each of the found sample sites. The dimension of this array is [SampleSites,2], where the second dimension represents longitude and latitude respectively.

`UnknownDataArray`
This is an array which gives the alleles for the individuals of unknown origin. The dimension of this array is [NumberUnknowns,2,NumberLoci], where 2 represents to 2 alleles each individual has at each locus. Note that these should not be allele lengths but rather the allele number matching the dimension in DataArray. Note that 0 or negative values here indicate unknown alleles and it is assumed that both are either known or unknown.

`MaxGridLength` An integer giving the maximum number of boxes to fill the longer side of the region. Note that computation time increases quadratically as this number increases, but this number also should be high enough to separate different sample sites otherwise they will be binned together as a single site.

`RhoParameter` This is a real precision parameter weighting the amount of smoothing. A higher value flattens out the surface while a lower value allows for more fluctuations. The default value of 10 was used in our analysis and should prove a good starting point. To choose a value by crossvalidation please see `FindRhoParameterCrossValidation`

`MaskWater` If TRUE, this logical parameter restricts the heat maps to land areas only.

Value

List with the following components:

`AlleleFrequencySurfaces`
An array giving the allele frequency for each allele, each coordinate, and each SNP. The dimension of this array is [MaxAlleles, NumberLoci, NumberLongitudeDivisions, NumberLatitudeDivisions], where either NumberLongitudeDivisions or NumberLatitudeDivisions is equal to MaxGridLength(an input to this function) and the other is scaled so that the geodesic distance between points horizontally and vertically is equal.

`UnknownGrids` An array giving the probability that an unknown individual comes from the given location. The dimension of this array is [NumberLongitudeDivisions, NumberLatitudeDivisions, NumberUnknowns], where either NumberLongitudeDivisions or NumberLatitudeDivisions is equal to MaxGridLength(an input to this function) and the other is scaled so that the geodesic distance between points horizontally and vertically is equal.

`DataArray` An array giving the number alleles grouped by sample sites for each locus. The dimension of this array is [MaxAlleles,SampleSites,NumberSNPs].

`RhoParameter` A real value showing the inputted RhoParameter value.

`SampleSites` This shows the integer number of sample sites found.

`GridLength` An array giving the number of longitudinal and latitudinal divisions. The dimension of this array is [2], where the first number is longitude and the second is latitude.

`MaxGridLength` An integer giving the maximum number of boxes to fill the longer side of the region. Note that computation time increases quadratically as this number increases, but this number also should be high enough to separate different sample sites otherwise they will be binned together as a single site. This number was part of the inputs.

`MaxAlleles` This shows the maximum of AllelesAtLocus. The maximum number of alleles at all loci.

`NumberLoci` This shows the integer number of loci found.

`SampleCoordinates`
This is an array which gives the longitude and latitude of each of the found sample sites. The dimension of this array is [SampleSites,2], where the second dimension represents longitude and latitude respectively.

`GridCoordinates`
An array showing the corresponding coordinates for each longitude and latitude division. The dimension of this array is [2,MaxGridLength], with longitude coordinates coming first and latitude second. Note that one of these rows may not be filled entirely. The associated output GridLength should be used to find the lengths of the two rows. Rows not filled in entirely will contain zeroes at the end.

`AllelesAtLocus` This shows the integer vector of alleles found at each locus.

`NumberUnknowns` Integer number of unknown individuals found.

`UnknownDataArray`
This is an array which gives the alleles for the individuals of unknown origin. The dimension of this array is [NumberUnknowns,2,NumberLoci], where 2 represents to 2 alleles each individual has at each locus. Note that these should not be allele lengths but rather the allele number matching the dimension in DataArray.

Author(s)

John Michael Ranola, John Novembre, and Kenneth Lange

References

Ranola J, Novembre J, Lange K (2014) Fast Spatial Ancestry via Flexible Allele Frequency Surfaces. Bioinformatics, in press.

See Also

`ConvertMicrosatData` for converting Microsatellite data files into a format appropriate for analysis,

`ConvertPEDData` for converting Plink PED files into a format appropriate for analysis,

`FitOriGenModel` for fitting allele surfaces to the converted SNP data,

`FitMultinomialModelFindUnknowns` for fitting allele surfaces to the converted Microsatellite data,

`PlotAlleleFrequencySurface` for a quick way to plot the resulting allele frequency surfaces from `FitOriGenModel` or `FitMultinomialModelFindUnknowns`,;

Examples

```
#this example not run because it takes longer than 5 secs
#note - type example(FunctionName, run.dontrun=TRUE) to run the example where FunctionName is
#the name of the function

## Not run:

##Data generation
SampleSites=10
NumberLoci=4
MaxAlleles=4
if(MaxAlleles==2){
NumberAllelesAtEachLocus=rep(2,NumberLoci)
}else{
NumberAllelesAtEachLocus=sample(2:MaxAlleles,NumberLoci,replace=TRUE)
}
TestData=array(0,dim=c(MaxAlleles,SampleSites,NumberLoci))
for(i in 1:NumberLoci){
for(j in 1:NumberAllelesAtEachLocus[i]){
TestData[j,,i]=sample(1:10,SampleSites,replace=TRUE)
}
}
##This data is simulated in Europe which is around Longitude -9 to 38 and Latitude 34 to 60
TestCoordinates=array(0,dim=c(SampleSites,2))
TestCoordinates[,1]=runif(SampleSites,-9,38)
TestCoordinates[,2]=runif(SampleSites,34,60)

##This simulates the unknown data
NumberUnknowns=2
UnknownData=array(0,dim=c(NumberUnknowns,2,NumberLoci))
for(i in 1:NumberUnknowns){
for(j in 1:NumberLoci){
UnknownData[i,,j]=sample(1:NumberAllelesAtEachLocus[j],2)
}
}

##MaxGridLength is the maximum number of boxes allowed
##to span the region in either direction
##Note that this number was reduced to allow the example to run in less than 5 secs
##RhoParameter is a tuning constant
print("MaxGridLength is intentionally set really low for fast examples.
Meaningful results will most likely require a higher value.")

##Fits the allele frequency surfaces only
#SurfaceTrials=FitMultinomialModel(TestData,TestCoordinates,
#MaxGridLength=20,RhoParameter=10)
#str(SurfaceTrials)
##Plotting the model
#PlotAlleleFrequencySurface(SurfaceTrials,LocusNumber=1,AlleleNumber=1,
# MaskWater=TRUE,Scale=FALSE)
```

```
##You can generate heatmaps of unknown individual's placements from with the allele
##surfaces using GenerateHeatMaps or use FitMultinomialModelFindUnknowns
#HeatMapTrials=GenerateHeatMaps(SurfaceTrials,UnknownData,NumberLoci=NumberLoci)
##Plotting the unknown heat map
#PlotUnknownHeatMap(HeatMapTrials,UnknownNumber=1,MaskWater=TRUE)

##Fitting the model and finding the unknown locations
UnknownTrials=FitMultinomialModelFindUnknowns(TestData,TestCoordinates,
UnknownData,MaxGridLength=20,RhoParameter=10)
str(UnknownTrials)
##Plotting the unknown heat map
PlotUnknownHeatMap(UnknownTrials,UnknownNumber=1,MaskWater=TRUE)

##Fitting the admixed model
#AdmixedTrials=FitMultinomialAdmixedModelFindUnknowns(TestData,TestCoordinates,
# UnknownData,MaxGridLength=10,RhoParameter=10)
##Plots the admixed surface disregarding fractions less than 0.01
#PlotAdmixedSurface(AdmixedTrials,UnknownNumber=1)


## End(Not run)
```

FitOriGenModel *Fit OriGen allele frequency surfaces*

Description

This function fits allele frequency surfaces to the data.

Usage

```
FitOriGenModel(DataArray,SampleCoordinates,MaxGridLength=20,RhoParameter=10)
```

Arguments

`DataArray` An array giving the number of major/minor SNPs (defined as the most occuring in the dataset) grouped by sample sites for each SNP. The dimension of this array is [2,SampleSites,NumberSNPs].

`SampleCoordinates` This is an array which gives the longitude and latitude of each of the found sample sites. The dimension of this array is [SampleSites,2], where the second dimension represents longitude and latitude respectively.

`MaxGridLength` An integer giving the maximum number of boxes to fill the longer side of the region. Note that computation time increases quadratically as this number increases, but this number also should be high enough to separate different sample sites otherwise they will be binned together as a single site.

`RhoParameter` This is a real precision parameter weighting the amount of smoothing. A higher value flattens out the surface while a lower value allows for more fluctuations. The default value of 10 was used in our analysis and should prove a good starting point. To choose a value by crossvalidation please see `FindRhoParameterCrossValidation`

Value

List with the following components:

`AlleleFrequencySurfaces` An array giving the allele frequency for each coordinate and each SNP. The dimension of this array is [NumberSNPs, NumberLongitudeDivisions, NumberLatitudeDivisions], where either NumberLongitudeDivisions or NumberLatitudeDivisions is equal to MaxGridLength(an input to this function) and the other is scaled so that the geodesic distance between points horizontally and vertically is equal.

`DataArray` An array giving the number of major/minor SNPs (defined as the most occuring in the dataset) grouped by sample sites for each SNP. The dimension of this array is [2,SampleSites,NumberSNPs].

`NumberSNPs` This shows the integer number of SNPs found.

`GridLength` An array giving the number of longitudinal and latitudinal divisions. The dimension of this array is [2], where the first number is longitude and the second is latitude.

`RhoParameter` A real value showing the inputted RhoParameter value.

`SampleSites` This shows the integer number of sample sites found.

`MaxGridLength` An integer giving the maximum number of boxes to fill the longer side of the region. Note that computation time increases quadratically as this number increases, but this number also should be high enough to separate different sample sites otherwise they will be binned together as a single site. This number was part of the inputs.

`SampleCoordinates` This is an array which gives the longitude and latitude of each of the found sample sites. The dimension of this array is [SampleSites,2], where the second dimension represents longitude and latitude respectively.

`GridCoordinates` An array showing the corresponding coordinates for each longitude and latitude division. The dimension of this array is [2,MaxGridLength], with longitude coordinates coming first and latitude second. Note that one of these rows may not be filled entirely. The associated output GridLength should be used to find the lengths of the two rows. Rows not filled in entirely will contain zeroes at the end.

Author(s)

John Michael Ranola, John Novembre, and Kenneth Lange

References

Ranola J, Novembre J, Lange K (2014) Fast Spatial Ancestry via Flexible Allele Frequency Surfaces. Bioinformatics, in press.

See Also

`ConvertPEDData` for converting Plink PED files into a format appropriate for analysis,

`FitOriGenModel` for fitting allele surfaces to the converted data,

`PlotAlleleFrequencySurface` for a quick way to plot the resulting allele frequency surfaces from `FitOriGenModel`,;

Examples

```
#this example not run because it takes slightly longer than 5 secs
#note - type example(FunctionName, run.dontrun=TRUE) to run the example where FunctionName is
#the name of the function

## Not run:

#Note see the help files for ConvertPEDData and ConvertUnknownPEDData if you have Plink PED files

#Data generation
SampleSites=10
NumberSNPs=5
TestData=array(sample(2*(1:30),2*SampleSites*NumberSNPs,
replace=TRUE),dim=c(2,SampleSites,NumberSNPs))
#Europe is about -9 to 38 and 34 to 60
TestCoordinates=array(0,dim=c(SampleSites,2))
TestCoordinates[,1]=runif(SampleSites,-9,38)
TestCoordinates[,2]=runif(SampleSites,34,60)

#Fitting the model
#MaxGridLength is the maximum number of boxes allowed to span the region in either direction
#RhoParameter is a tuning constant
trials2=FitOriGenModel(TestData,TestCoordinates,MaxGridLength=20,RhoParameter=10)
str(trials2)

#Plotting the model
PlotAlleleFrequencySurface(trials2)


## End(Not run)
```

FitOriGenModelFindUnknowns

Fit the OriGen model and place unknown individuals

Description

This function fits the OriGen model and places individuals of unknown origins.

Usage

```
FitOriGenModelFindUnknowns(DataArray,SampleCoordinates,
UnknownData,MaxGridLength=20,RhoParameter=10)
```

Arguments

`DataArray` An array giving the number of major/minor SNPs (defined as the most occuring in the dataset) grouped by sample sites for each SNP. The dimension of this array is [2,SampleSites,NumberSNPs].

`SampleCoordinates` This is an array which gives the longitude and latitude of each of the found sample sites. The dimension of this array is [SampleSites,2], where the second dimension represents longitude and latitude respectively.

`UnknownData` An array showing the unknown individuals genetic data. The dimension of this array is [NumberUnknowns,NumberSNPs].

`MaxGridLength` An integer giving the maximum number of boxes to fill the longer side of the region. Note that computation time increases quadratically as this number increases, but this number also should be high enough to separate different sample sites otherwise they will be binned together as a single site.

`RhoParameter` This is a real precision parameter weighting the amount of smoothing. A higher value flattens out the surface while a lower value allows for more fluctuations. The default value of 10 was used in our analysis and should prove a good starting point. To choose a value by crossvalidation please see `FindRhoParameterCrossValidation`

Value

List with the following components:

`UnknownGrids` An array giving the probability that an unknown individual comes from the given location. The dimension of this array is [NumberLongitudeDivisions, NumberLatitudeDivisions, NumberUnknowns], where either NumberLongitudeDivisions or NumberLatitudeDivisions is equal to MaxGridLength(an input to this function) and the other is scaled so that the geodesic distance between points horizontally and vertically is equal.

`DataArray` An array giving the number of major/minor SNPs (defined as the most occuring in the dataset) grouped by sample sites for each SNP. The dimension of this array is [2,SampleSites,NumberSNPs].

`NumberSNPs` This shows the integer number of SNPs found.

`GridLength` An array giving the number of longitudinal and latitudinal divisions. The dimension of this array is [2], where the first number is longitude and the second is latitude.

`RhoParameter` A real value showing the inputted RhoParameter value.

`SampleSites` This shows the integer number of sample sites found.

`MaxGridLength` An integer giving the maximum number of boxes to fill the longer side of the region. Note that computation time increases quadratically as this number increases, but this number also should be high enough to separate different sample sites otherwise they will be binned together as a single site. This number was part of the inputs.

`SampleCoordinates`
This is an array which gives the longitude and latitude of each of the found sample sites. The dimension of this array is [SampleSites,2], where the second dimension represents longitude and latitude respectively.

`NumberUnknowns` This is an integer value showing the number of unknowns found in the UnknownPEDFile.

`UnknownData` An array showing the unknown individuals genetic data. The dimension of this array is [NumberUnknowns,NumberSNPs].

`GridCoordinates`
An array showing the corresponding coordinates for each longitude and latitude division. The dimension of this array is [2,MaxGridLength], with longitude coordinates coming first and latitude second. Note that one of these rows may not be filled entirely. The associated output GridLength should be used to find the lengths of the two rows. Rows not filled in entirely will contain zeroes at the end.

Author(s)

John Michael Ranola, John Novembre, and Kenneth Lange

References

Ranola J, Novembre J, Lange K (2014) Fast Spatial Ancestry via Flexible Allele Frequency Surfaces. Bioinformatics, in press.

See Also

`ConvertUnknownPEDData` for converting two Plink PED files (known and unknown)into a format appropriate for analysis,

`FitOriGenModelFindUnknowns` for fitting allele surfaces to the converted data and finding the locations of the given unknown individuals,

`PlotUnknownHeatMap` for a quick way to plot the resulting unknown heat map surfaces from `FitOriGenModelFindUnknowns`,;

`FitAdmixedModelFindUnknowns` for fitting allele surfaces to the converted data and finding the locations of the given unknown individuals who may be admixed,

`PlotAdmixedSurface` for a quick way to plot the resulting admixture surfaces from `FitAdmixedFindUnknowns`,

Examples

```
#this example not run because it takes slightly longer than 5 secs
#note - type example(FunctionName, run.dontrun=TRUE) to run the example where FunctionName is
#the name of the function

## Not run:

#Data generation
SampleSites=10
NumberSNPs=5
TestData=array(sample(2*(1:30),2*SampleSites*NumberSNPs,
replace=TRUE),dim=c(2,SampleSites,NumberSNPs))
#Europe is about -9 to 38 and 34 to 60
TestCoordinates=array(0,dim=c(SampleSites,2))
TestCoordinates[,1]=runif(SampleSites,-9,38)
TestCoordinates[,2]=runif(SampleSites,34,60)

#This code simulates the number of major alleles the unknown individuals have.
NumberUnknowns=2
TestUnknowns=array(sample(0:2,NumberUnknowns*NumberSNPs,
replace=TRUE),dim=c(NumberUnknowns,NumberSNPs))

#Fitting the model
#MaxGridLength is the maximum number of boxes allowed to span the region in either direction
#RhoParameter is a tuning constant
trials4=FitOriGenModelFindUnknowns(TestData,TestCoordinates,
TestUnknowns,MaxGridLength=20,RhoParameter=10)
str(trials4)

#Plotting the unknown heat map
PlotUnknownHeatMap(trials4,UnknownNumber=1,MaskWater=TRUE)


## End(Not run)
```

GenerateHeatMaps *Fit OriGen microsatellite allele frequency surfaces*

Description

This function generates heat maps from OriGen microsatellite data output and then finds locations for unknown individuals..

Usage

```
GenerateHeatMaps(FitModelOutput,UnknownDataArray,NumberLoci,MaskWater=TRUE)
```

Arguments

`FitModelOutput` This is the output from `FitMultinomialModel`.

`UnknownDataArray`
This is an array which gives the alleles for the individuals of unknown origin. The dimension of this array is [NumberUnknowns,2,NumberLoci], where 2 represents to 2 alleles each individual has at each locus. Note that these should not be allele lengths but rather the allele number matching the dimension in DataArray. Note that 0 or negative values here indicate unknown alleles and it is assumed that both are either known or unknown.

`NumberLoci` This integer value gives the number of loci to include when generating the heat maps. This is useful when generating heatmaps with multiple numbers of loci.

`MaskWater` If TRUE, this logical parameter restricts the heat maps to land areas only.

Value

List with the following components:

`AlleleFrequencySurfaces`
An array giving the allele frequency for each allele, each coordinate, and each SNP. The dimension of this array is [MaxAlleles, NumberLoci, NumberLongitudeDivisions, NumberLatitudeDivisions], where either NumberLongitudeDivisions or NumberLatitudeDivisions is equal to MaxGridLength(an input to this function) and the other is scaled so that the geodesic distance between points horizontally and vertically is equal.

`UnknownGrids` An array giving the probability that an unknown individual comes from the given location. The dimension of this array is [NumberLongitudeDivisions, NumberLatitudeDivisions, NumberUnknowns], where either NumberLongitudeDivisions or NumberLatitudeDivisions is equal to MaxGridLength(an input to this function) and the other is scaled so that the geodesic distance between points horizontally and vertically is equal.

`DataArray` An array giving the number alleles grouped by sample sites for each locus. The dimension of this array is [MaxAlleles,SampleSites,NumberSNPs].

`RhoParameter` A real value showing the inputted RhoParameter value.

`SampleSites` This shows the integer number of sample sites found.

`GridLength` An array giving the number of longitudinal and latitudinal divisions. The dimension of this array is [2], where the first number is longitude and the second is latitude.

`MaxGridLength` An integer giving the maximum number of boxes to fill the longer side of the region. Note that computation time increases quadratically as this number increases, but this number also should be high enough to separate different sample sites otherwise they will be binned together as a single site. This number was part of the inputs.

`MaxAlleles` This shows the maximum of AllelesAtLocus. The maximum number of alleles at all loci.

`NumberLoci` This shows the integer number of loci found.

`SampleCoordinates`
This is an array which gives the longitude and latitude of each of the found sample sites. The dimension of this array is [SampleSites,2], where the second dimension represents longitude and latitude respectively.

`GridCoordinates`
An array showing the corresponding coordinates for each longitude and latitude division. The dimension of this array is [2,MaxGridLength], with longitude coordinates coming first and latitude second. Note that one of these rows may not be filled entirely. The associated output GridLength should be used to find the lengths of the two rows. Rows not filled in entirely will contain zeroes at the end.

`AllelesAtLocus` This shows the integer vector of alleles found at each locus.

`NumberUnknowns` Integer number of unknown individuals found.

`UnknownDataArray`
This is an array which gives the alleles for the individuals of unknown origin. The dimension of this array is [NumberUnknowns,2,NumberLoci], where 2 represents to 2 alleles each individual has at each locus. Note that these should not be allele lengths but rather the allele number matching the dimension in DataArray.

Author(s)

John Michael Ranola, John Novembre, and Kenneth Lange

References

Ranola J, Novembre J, Lange K (2014) Fast Spatial Ancestry via Flexible Allele Frequency Surfaces. Bioinformatics, in press.

See Also

`ConvertMicrosatData` for converting Microsatellite data files into a format appropriate for analysis,

`ConvertPEDData` for converting Plink PED files into a format appropriate for analysis,

`FitMultinomialModel` for fitting allele surfaces to the converted microsatellite data,

`FitMultinomialModelFindUnknowns` for fitting allele surfaces to the converted Microsatellite data,

`PlotAlleleFrequencySurface` for a quick way to plot the resulting allele frequency surfaces from `FitOriGenModel` or `GenerateHeatMaps`,;

Examples

```
#this example not run because it takes longer than 5 secs
#note - type example(FunctionName, run.dontrun=TRUE) to run the example where FunctionName is
#the name of the function

## Not run:
```

```
##Data generation
SampleSites=10
NumberLoci=4
MaxAlleles=4
NumberAllelesAtEachLocus=sample(2:MaxAlleles,NumberLoci,replace=TRUE)
TestData=array(0,dim=c(MaxAlleles,SampleSites,NumberLoci))
for(i in 1:NumberLoci){
for(j in 1:NumberAllelesAtEachLocus[i]){
TestData[j,,i]=sample(1:10,SampleSites,replace=TRUE)
}
}
##This data is simulated in Europe which is around Longitude -9 to 38 and Latitude 34 to 60
TestCoordinates=array(0,dim=c(SampleSites,2))
TestCoordinates[,1]=runif(SampleSites,-9,38)
TestCoordinates[,2]=runif(SampleSites,34,60)

##This simulates the unknown data
NumberUnknowns=2
UnknownData=array(0,dim=c(NumberUnknowns,2,NumberLoci))
for(i in 1:NumberUnknowns){
for(j in 1:NumberLoci){
UnknownData[i,,j]=sample(1:NumberAllelesAtEachLocus[j],2)
}
}

##MaxGridLength is the maximum number of boxes allowed
##to span the region in either direction
##Note that this number was reduced to allow the example to run in less than 5 secs
##RhoParameter is a tuning constant
print("MaxGridLength is intentionally set really low for fast examples.
Meaningful results will most likely require a higher value.")

##Fits the allele frequency surfaces only
SurfaceTrials=FitMultinomialModel(TestData,TestCoordinates,
MaxGridLength=20,RhoParameter=10)
str(SurfaceTrials)
##Plotting the model
PlotAlleleFrequencySurface(SurfaceTrials,LocusNumber=1,AlleleNumber=1,
MaskWater=TRUE,Scale=FALSE)

##You can generate heatmaps of unknown individual's placements from with the allele
##surfaces using GenerateHeatMaps or use FitMultinomialModelFindUnknowns
HeatMapTrials=GenerateHeatMaps(SurfaceTrials,UnknownData,NumberLoci=NumberLoci)
##Plotting the unknown heat map
PlotUnknownHeatMap(HeatMapTrials,UnknownNumber=1,MaskWater=TRUE)

##Fitting the model and finding the unknown locations
#UnknownTrials=FitMultinomialModelFindUnknowns(TestData,TestCoordinates,
# UnknownData,MaxGridLength=20,RhoParameter=10)
#str(UnknownTrials)
##Plotting the unknown heat map
#PlotUnknownHeatMap(UnknownTrials,UnknownNumber=1,MaskWater=TRUE)
```

```
##Fitting the admixed model
#AdmixedTrials=FitMultinomialAdmixedModelFindUnknowns(TestData,TestCoordinates,
# UnknownData,MaxGridLength=10,RhoParameter=10)
##Plots the admixed surface disregarding fractions less than 0.01
#PlotAdmixedSurface(AdmixedTrials,UnknownNumber=1)


## End(Not run)
```

`Locations` *Locations of individuals in 10SNPs*

Description

This data set gives the locations of individuals in the Plink file 10SNPs to be used as a test data only.

Usage

```
Locations
```

Format

A matrix containing names and locations.

`LocationsTrialDataSmall` *Locations of individuals in MicrosatTrialDataSmall*

Description

This data set gives the locations of individuals in the file MicrosatTrialDataSmall to be used as a test data only. Space or tab delimited text file with the location information for the individuals. The columns are LocationName, LocationNumber, Latitude, and Longitude. Note that the first two columns must be in the same order as the MicrosatTrialDataSmall.

Usage

```
LocationsTrialDataSmall
```

Format

A text file containing names and locations.

`MicrosatTrialDataSmall` *Genotypes of individuals in located at LocationsTrialDataSmall*

Description

This data set gives the genotypes of individuals located in the file LocationsTrialDataSmall to be used as a test data only. The columns here would be LocationName, LocationNumber, Locus1, Locus2, etc. Each individual would take up 2 rows (one for each allele) with the same LocationName and LocationNumber. The value under Locus would be the length of the allele of that individual. Note that unknown individuals should have location number "-1".

Usage

```
MicrosatTrialDataSmall
```

Format

A text file containing names and locations.

`PlotAdmixedSurface` *Plots admixture fraction results*

Description

This function plots the admixture results from FitAdmixedModelFindUnknowns. These numbers represent the fractional contribution each location has to the individuals genetic data. In other words, an individual with unmixed parents from two different locations should have a fraction of 0.5 from each of those locations with enough data.

Usage

```
PlotAdmixedSurface(AdmixedOutput,UnknownNumber=1,Percent=FALSE,Title=NULL,MaskWater=TRUE)
```

Arguments

`AdmixedOutput`	The output of `FitAdmixedModelFindUnknowns`
`UnknownNumber`	Integer indicating the unknown individual heat map number to plot.
`Percent`	A logical value that will display percentages instead of fractions if TRUE.
`Title`	A string giving the title of the plot. If NULL, a default title is used.
`MaskWater`	Logical value that if true removes water from the plotted regions.

Value

This outputs a plot of the admixture fractions, the contribution of each location, for a particular unknown individual.

Author(s)

John Michael Ranola, John Novembre, and Kenneth Lange

References

Ranola J, Novembre J, Lange K (2014) Fast Spatial Ancestry via Flexible Allele Frequency Surfaces. Bioinformatics, in press.

See Also

ConvertUnknownPEDData for converting two Plink PED files (known and unknown)into a format appropriate for analysis,

FitOriGenModelFindUnknowns for fitting allele surfaces to the converted data and finding the locations of the given unknown individuals,

PlotUnknownHeatMap for a quick way to plot the resulting unknown heat map surfaces from FitOriGenModelFindUnknowns,

FitAdmixedModelFindUnknowns for fitting allele surfaces to the converted data and finding the locations of the given unknown individuals who may be admixed,

PlotAdmixedSurface for a quick way to plot the resulting admixture surfaces from FitAdmixedFindUnknowns,

Examples

```
#this example not run because it takes longer than 5 secs
#note - type example(FunctionName, run.dontrun=TRUE) to run the example where FunctionName is
#the name of the function

## Not run:

##Data generation
SampleSites=10
NumberSNPs=4
TestData=array(sample(2*(1:30),2*SampleSites*NumberSNPs,replace=TRUE),
dim=c(2,SampleSites,NumberSNPs))
##This data is simulated in Europe which is around Longitude -9 to 38 and Latitude 34 to 60
TestCoordinates=array(0,dim=c(SampleSites,2))
TestCoordinates[,1]=runif(SampleSites,-9,38)
TestCoordinates[,2]=runif(SampleSites,34,60)

##This code simulates the number of major alleles the unknown individuals have.
NumberUnknowns=2
TestUnknowns=array(sample(0:2,NumberUnknowns*NumberSNPs,
replace=TRUE),dim=c(NumberUnknowns,NumberSNPs))
```

```
##MaxGridLength is the maximum number of boxes allowed
##to span the region in either direction
##Note that this number was reduced to allow the example to run in less than 5 secs
##RhoParameter is a tuning constant
print("MaxGridLength is intentionally set really low for fast examples.
Meaningful results will most likely require a higher value.")

##Fitting the admixed model
##Note that MaxGridLength is intentionally set unusably low so that the example
##runs in under 5 seconds.  The default value of 20 is more reasonable in general
AdmixedTrials=FitAdmixedModelFindUnknowns(TestData,TestCoordinates,
TestUnknowns,MaxGridLength=8,RhoParameter=10)
##Plots the admixed surface disregarding fractions less than 0.01
PlotAdmixedSurface(AdmixedTrials,UnknownNumber=1)


## End(Not run)
```

PlotAlleleFrequencySurface

Plots an OriGen fitted allele frequency surface

Description

This function plots an allele frequency surface outputted by FitOriGenModel and FitMultinomialModel.

Usage

```
PlotAlleleFrequencySurface(AlleleSurfaceOutput,LocusNumber=1,
AlleleNumber=1,MaskWater=TRUE,Scale=FALSE)
```

Arguments

`AlleleSurfaceOutput`
The output of `FitOriGenModel` or `FitMultinomialModel`

`LocusNumber` Integer indicating the Locus number to plot.

`AlleleNumber` Integer indicating which allele to plot. If using microsatellites and AlleleNumber = 0, then this plots all the allele frequency surfaces in a grid.

`MaskWater` Logical value that if true removes water from the plotted regions.

`Scale` Logical value that if TRUE will scale the colors to (0,max(Frequency)) instead of (0,1).

Value

This outputs a plot (using ggplot) of the allele frequency surface on a map.

Author(s)

John Michael Ranola, John Novembre, and Kenneth Lange

References

Ranola J, Novembre J, Lange K (2014) Fast Spatial Ancestry via Flexible Allele Frequency Surfaces. Bioinformatics, in press.

See Also

ConvertMicrosatData for converting Microsatellite data files into a format appropriate for analysis,

ConvertPEDData for converting Plink PED files into a format appropriate for analysis,

FitOriGenModel for fitting allele surfaces to the converted SNP data,

FitMultinomialModel for fitting allele surfaces to the converted Microsatellite data,

PlotAlleleFrequencySurface for a quick way to plot the resulting allele frequency surfaces from FitOriGenModel or FitMultinomialModel,;

Examples

```
#this example not run because it takes a little longer than 5 secs
#note - type example(FunctionName, run.dontrun=TRUE) to run the example where FunctionName is
#the name of the function
## Not run:
#Data generation
SampleSites=10
NumberLoci=4
MaxAlleles=4
NumberAllelesAtEachLocus=sample(2:MaxAlleles,NumberLoci,replace=TRUE)

TestData=array(0,dim=c(MaxAlleles,SampleSites,NumberLoci))
for(i in 1:NumberLoci){
for(j in 1:NumberAllelesAtEachLocus[i]){
TestData[j,,i]=sample(1:10,SampleSites,replace=TRUE)
}
}
#Europe is about -9 to 38 and 34 to 60
TestCoordinates=array(0,dim=c(SampleSites,2))
TestCoordinates[,1]=runif(SampleSites,-9,38)
TestCoordinates[,2]=runif(SampleSites,34,60)

#Fitting the model
#MaxGridLength is the maximum number of boxes allowed to span the region in either direction
#RhoParameter is a tuning constant
trials2=FitMultinomialModel(TestData,TestCoordinates,MaxGridLength=20,RhoParameter=10)
str(trials2)

#Plotting the model
PlotAlleleFrequencySurface(trials2)
```

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

`PlotAlleleFrequencySurfaceOld`
Plots an OriGen fitted allele frequency surface

Description

This function plots an allele frequency surface outputted by FitOriGenModel.

Usage

```
PlotAlleleFrequencySurfaceOld(AlleleSurfaceOutput,SNPNumber=1,MaskWater=TRUE)
```

Arguments

`AlleleSurfaceOutput`
The output of `FitOriGenModel`

`SNPNumber` Integer indicating the SNP number to plot.

`MaskWater` Logical value that if true removes water from the plotted regions.

Value

This outputs a plot of the allele frequency surface on a map.

Author(s)

John Michael Ranola, John Novembre, and Kenneth Lange

References

Ranola J, Novembre J, Lange K (2014) Fast Spatial Ancestry via Flexible Allele Frequency Surfaces. Bioinformatics, in press.

See Also

`ConvertPEDData` for converting Plink PED files into a format appropriate for analysis,

`FitOriGenModel` for fitting allele surfaces to the converted data,

`PlotAlleleFrequencySurfaceOld` for a quick way to plot the resulting allele frequency surfaces from `FitOriGenModel`,;

Examples

```
#this example not run because it takes slightly longer than 5 secs
#note - type example(FunctionName, run.dontrun=TRUE) to run the example where FunctionName is
#the name of the function

## Not run:

#Data generation
SampleSites=10
NumberSNPs=5
TestData=array(sample(2*(1:30),2*SampleSites*NumberSNPs,replace=TRUE),
dim=c(2,SampleSites,NumberSNPs))
#Europe is about -9 to 38 and 34 to 60
TestCoordinates=array(0,dim=c(SampleSites,2))
TestCoordinates[,1]=runif(SampleSites,-9,38)
TestCoordinates[,2]=runif(SampleSites,34,60)

#Fitting the model
#MaxGridLength is the maximum number of boxes allowed to span the region in either direction
#RhoParameter is a tuning constant
trials2=FitOriGenModel(TestData,TestCoordinates,MaxGridLength=20,RhoParameter=10)
str(trials2)

#Plotting the model
PlotAlleleFrequencySurfaceOld(trials2)


## End(Not run)
```

PlotUnknownHeatMap — *Plots a heat map depicting the probability an unknown individual comes from each block*

Description

This function plots a probability heat map surface outputted by FitOriGenModelFindUnknowns.

Usage

```
PlotUnknownHeatMap(HeatMapOutput,UnknownNumber=1,MaskWater=TRUE)
```

Arguments

`HeatMapOutput`	The output of `FitOriGenModelFindUnknowns`
`UnknownNumber`	Integer indicating the unknown individual heat map number to plot.
`MaskWater`	Logical value that if true removes water from the plotted regions.

Value

This outputs a plot of the probability heat map for a particular unknown individual.

Author(s)

John Michael Ranola, John Novembre, and Kenneth Lange

References

Ranola J, Novembre J, Lange K (2014) Fast Spatial Ancestry via Flexible Allele Frequency Surfaces. Bioinformatics, in press.

See Also

`ConvertUnknownPEDData` for converting two Plink PED files (known and unknown)into a format appropriate for analysis,

`FitOriGenModelFindUnknowns` for fitting allele surfaces to the converted data and finding the locations of the given unknown individuals,

`PlotUnknownHeatMap` for a quick way to plot the resulting unknown heat map surfaces from `FitOriGenModelFindUnknowns`,

`FitAdmixedModelFindUnknowns` for fitting allele surfaces to the converted data and finding the locations of the given unknown individuals who may be admixed,

`PlotAdmixedSurface` for a quick way to plot the resulting admixture surfaces from `FitAdmixedFindUnknowns`,

Examples

```
#this example not run because it takes slightly longer than 5 secs
#note - type example(FunctionName, run.dontrun=TRUE) to run the example where FunctionName is
#the name of the function

## Not run:

#Data generation
SampleSites=10
NumberSNPs=5
TestData=array(sample(2*(1:30),2*SampleSites*NumberSNPs,replace=TRUE),
dim=c(2,SampleSites,NumberSNPs))
#Europe is about -9 to 38 and 34 to 60
TestCoordinates=array(0,dim=c(SampleSites,2))
TestCoordinates[,1]=runif(SampleSites,-9,38)
TestCoordinates[,2]=runif(SampleSites,34,60)

#This code simulates the number of major alleles the unknown individuals have.
NumberUnknowns=2
TestUnknowns=array(sample(0:2,NumberUnknowns*NumberSNPs,replace=TRUE),
dim=c(NumberUnknowns,NumberSNPs))

#Fitting the model
```

```
#MaxGridLength is the maximum number of boxes allowed to span the region in either direction
#RhoParameter is a tuning constant
trials4=FitOriGenModelFindUnknowns(TestData,TestCoordinates,TestUnknowns,
MaxGridLength=20,RhoParameter=10)
str(trials4)

#Plotting the unknown heat map
PlotUnknownHeatMap(trials4,UnknownNumber=1,MaskWater=TRUE)


## End(Not run)
```

RankSNPsLRT *Rank the SNPs based on the likelihood ratio test.*

Description

This function ranks the SNPs based on the likelihood ratio test comparing the data grouped into the different sample sites as inputted vs one large sample including all of the sites. To convert the data see `ConvertPEDData`.

Usage

```
RankSNPsLRT(DataArray)
```

Arguments

`DataArray` An array giving the number of major/minor SNPs (defined as the most occuring in the dataset) grouped by sample sites for each SNP. The dimension of this array is [2,SampleSites,NumberSNPs].

Value

List with the following components:

`DataArray` An array giving the number of major/minor SNPs (defined as the most occuring in the dataset) grouped by sample sites for each SNP. The dimension of this array is [2,SampleSites,NumberSNPs].

`SampleSites` This shows the integer number of sample sites found.

`NumberSNPs` This shows the integer number of SNPs found.

`Rankings` An integer valued vector giving the LRT based ranking of each SNP. This can be used to reduce the number of SNPs to use for assignment if analysis takes too long.

`LRT` This is a real valued array giving the Likelihood Ratio test statistic and the informativeness for assignment(Rosenberg) for each SNP. The dimension of this array is [2,NumberSNPs].

Author(s)

John Michael Ranola, John Novembre, and Kenneth Lange

References

Ranola J, Novembre J, Lange K (2014) Fast Spatial Ancestry via Flexible Allele Frequency Surfaces. Bioinformatics, in press.

See Also

`ConvertPEDData` for converting Plink PED files into a format appropriate for analysis,

Examples

```
#Data generation
SampleSites=25
NumberSNPs=10
TestData=array(sample(2*(1:30),2*SampleSites*NumberSNPs,replace=TRUE),
dim=c(2,SampleSites,NumberSNPs))
#Europe is about -9 to 38 and 34 to 60
TestCoordinates=array(0,dim=c(SampleSites,2))
TestCoordinates[,1]=runif(SampleSites,-9,38)
TestCoordinates[,2]=runif(SampleSites,34,60)

#This code simulates the number of major alleles the unknown individuals have.
NumberUnknowns=2
TestUnknowns=array(sample(0:2,NumberUnknowns*NumberSNPs,replace=TRUE),
dim=c(NumberUnknowns,NumberSNPs))

#Rank the SNPs
trials7=RankSNPsLRT(TestData)
trials7
```

Index