# Package 'Sunder'

February 19, 2015

**Title** Quantification of the effect of geographic versus environmental isolation on genetic differentiation

**Author** Filippo Botta, Casper Eriksen, Gilles Guillot

**Description**
Quantification of the effect of geographic versus environmental isolation on genetic differentiation

**Maintainer** Filippo Botta <filippo.botta@gmail.com>

**URL** http://www2.imm.dtu.dk/~gigu/Sunder

**License** GPL

**Version** 0.0.4

**Date** Mon Jan 12 19:54:04 2015

**Depends** mnormt

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2015-01-13 10:27:07

## R topics documented:

## Sunder-package *Inference and model selection for analysis of geographical genetic variation*

### Description

Data simulation, inference and model selection by cross-validation for the analysis of geographical genetic variation

### Details

| | |
|---|---|
| Package: | Sunder |
| Type: | Package |
| Version: | 0.0.4 |
| Date: | Mon Jan 12 19:54:04 2015 |
| License: | GPL |

Simulation, inference and model selection by cross-validation for the analysis of geographical genetic variation

### Author(s)

Filippo Botta, Casper Eriksen and Gilles Guilllot Maintainer: Filippo Botta <filippo.botta@gmail.com>

### References

Cf. `citation(Sunder)`

## D_E *Matrix of environmental distances*

### Description

Matrix of environmental distances

## D_G *Matrix of geographical distances*

### Description

Matrix of geographical distances

gen *An array of allele counts*

## Description

An array of allele counts in the format suitable for the inference functions

MCMCCV *Inference and model selection for analysis of geographical genetic variation*

## Description

MCMC inference and model selection by cross-validation for the analysis of geographical genetic variation

## Usage

```
MCMCCV(gen, D_G, D_E,
               nit, thinning, theta.max, theta.init,
               run, ud, n.validation.set,print.pct)
```

## Arguments

| | |
|---|---|
| gen | An array with dimensions (n,l,a) n: number of geographical locations, l: number of loci, a: max number of alleles |
| D_G | A matrix of geograpical distances |
| D_E | A matrix of environmental distances |
| nit | Number of iterations |
| thinning | Thinning of MCMC iterations |
| theta.max | Upper bounds for the vector of parameters in theta. Note that in theta, the parameters are assumed to be in this order: (alpha,beta_G, beta_E, gamma, delta) |
| theta.init | Initial state of theta |
| run | A vector of booleans of length 3 stating which sub-model is investigated among G+E,G,E (in this order). For example, the default value `run=c(TRUE,FALSE,FALSE)` means that only one MCMC run will be performed to estimate paremeters in the G+E model while e.g. `run=c(TRUE,FALSE,TRUE)` means that MCMC runs will be performed for models G+E and E. If `n.validation.set >0` likelihoods under these two models will be returned for model selection. |
| ud | A vector of booleans of length 5 stating which entries in theta=(alpha,beta_E,beta_G,gamma,delta) will be updated in the MCMC iterations. By default all parameters in theta will be updated. If one entry is not updated, the value used along the MCMC simulation for this parameter is the initial value. |

`n.validation.set`
: The number of pairs (sites x locus) used as validation set

`print.pct`
: A boolean stating whether Fortran prints percentage of computation achieved along each MCMC run.

## Value

A list with a component named mod.lik containing likelihoods on the validation set for the various models compared.

## Author(s)

Filippo Botta, Gilles Guillot

## Examples

```
## Not run:
data(toydata,package='Sunder')

     #### Computing options
     nit <- 10^2
     run  <- c(1,1,1)
     thinning  <- 1 # max(nit/10^3,1);
     ud   <- c(0,1,1,0,0)
     theta.init <- c(1,2,1,1,0.01)
     n.validation.set  <- dim(gen)[1]*dim(gen)[2]/10
     theta.max  <- c(10,10*max(D_G),10*max(D_E),1,0.01)

     plot  <- TRUE
     trace <- FALSE

     #### Call Sunder ####
     output <- MCMCCV(gen,D_G,D_E,
                              nit,thinning,
                              theta.max,
                              theta.init,
                              run,ud,n.validation.set)

     mod.lik <- output$mod.lik
     tvt <- output$theta


     ## plotting outputs
     upd=matrix(nrow=sum(run), ncol=length(theta.init), data=1)
     upd[2,3]=upd[3,2]=0

     plot(as.vector(D_G),as.vector(cor(t(gen[,,1]))),
        bg=colorRampPalette(c("blue", "red"))(dim(D_E)[1]^2)[order(order(as.vector(D_E)))],
          pch=21,
          xlab='Geographic distance',
          ylab='Empirical covariance genotypes')
```

```
      kol=c(4,2,3)
      xseq=seq(thinning,nit,thinning)
      ylab=c(expression(paste(alpha)),
             expression(paste(beta[D])),
             expression(paste(beta[E])),
             expression(paste(gamma)),
             expression(paste(delta)))

      par(mfrow=c(sum(run),length(theta.init)))
      for (j in 1:sum(run))
      {
        for(k in 1:length(theta.init))
        {
          if (sum(upd[,k]==1)>0)
          {
            if(upd[j, k]==1)
            {
              if(exists("theta"))
                ylim=c(min(tvt[k,,j],theta[k]),max(tvt[k,,j],theta[k])) else
                  ylim=c(min(tvt[k,,j]),max(tvt[k,,j]))
              plot(0, type="n",xlab="",ylab="", xlim=c(0,nit), ylim=ylim)
              lines(xseq,tvt[k,,j],col=kol[j],xlab="",ylab="")
              if(exists("theta")) abline(h=theta[k],lty=2)
              title(xlab="iterations")
              mtext(ylab[k], side=2, line=2.3,las=1)} else plot.new()
          }
        }
      }

print(mod.lik)
print(paste('The model achieving the highest likelihood on the validation set is:',
            names(mod.lik)[order(mod.lik,decreasing=TRUE)[1]]))
theta.GE <- apply(output$theta[,,1],1,mean)
print('Posterior mean theta under model G+E:')
print(theta.GE)

theta.G <- apply(output$theta[,,2],1,mean)
theta.G[3] <- NA
print('Posterior mean theta under model G:')
print(theta.G)

theta.E <- apply(output$theta[,,3],1,mean)
theta.E[2] <- NA
print('Posterior mean theta under model E:')
print(theta.E)


## End(Not run)
```

MLCVGauss *Inference and model selection under the assumption of Gaussian distribution of allele counts*

## Description

Inference and model selection for analysis of geographical genetic variation under the assumption of Gaussian distribution of allele counts for bi-allelic loci. Parameter estimation by maximization of the likelihood.

## Usage

```
MLCVGauss(gen, D_G, D_E,theta.max,theta.min,ntrain,nresamp)
```

## Arguments

| | |
|---|---|
| gen | A matrix with dimensions (n,l) n: number of geographical locations, l: number of loci. |
| D_G | A matrix of geograpical distances |
| D_E | A matrix of environmental distances |
| theta.max | Upper bounds for the vector of parameters in theta. Note that in theta, the parameters are assumed to be in this order: (alpha,beta_G, beta_E, gamma, delta) |
| theta.min | Lower bounds for the vector of parameters in theta. Note that in theta, the parameters are assumed to be in this order: (alpha,beta_G, beta_E, gamma, delta) |
| ntrain | Number of sites used for training. An integer smaller than `nrow(gen)`. If `ntrain` is equal to the number of sampling sites, the function estimates parameters on the whole dataset and does not perform cross-validation. |
| nresamp | Number of resamplings. An integer larger than 1. |

## Value

A list with either a component named mod.lik (containing likelihoods on the validation set for the various models compared) or a vector of estimated parameters (if `ntrain` is equal to the number of sampling sites).

## Author(s)

Gilles Guillot

## Examples

```
## Not run:
nsite <- 200
nloc <- 1000
hap.pop.size <- 100
theta <- c(runif(n=1,.5,10),
           runif(n=1,.01,10),
           runif(n=1,.01,10),
```

```
            runif(n=1,.5,1),
            runif(n=1,.01,.1)
            )
mod <- 'G+E'
dat <- SimSunderData(mod=mod,
                     theta=theta,
                     nsite=nsite,
                     nloc=nloc,
                     hap.pop.size=hap.pop.size,
                     nalM=2,nalm=2, #bi-allelic loci
                     var.par=1,
                     scale.par=3)
gen <- dat$gen[,,1]
D_G <- dat$D_G
D_E <- dat$D_E

res <- MLCVGauss(gen,D_G,D_E,
                 ntrain=nrow(gen)/2,
                 nresamp=3)

which.max(res$mod.lik)

## End(Not run)
```

---

`Reformat23` *Reformat data*

---

## Description

Reformat data from the 2-way table format (containing allelic states) to the 3-way table format (containing counts of the various alellic states)

## Usage

```
Reformat23(x)
```

## Arguments

| | |
|---|---|
| `x` | A 2-way table containing allelic states |

## Value

A 3-way table with dimensions (n,l,a) n: number of geographical locations, l: number of loci, a: max number of alleles.

## Author(s)

Casper Eriksen

## Reformat32 *Reformat data*

### Description

Reformat data from the 3-way table format (containing counts of the various alellic states) to the 2-way table format (containing allelic states)

### Usage

```
Reformat32(g)
```

### Arguments

| | |
|---|---|
| g | An array in the 3-way table format |

### Value

A matrix in the 2-way table format (containing allelic states)

### Author(s)

Casper Eriksen

## SimSunderData *Simulation of data*

### Description

Simulate data under the model assumed by the MCMC inference program

### Usage

```
SimSunderData(mod, theta, nsite, nloc, hap.pop.size, nalM, nalm, var.par, scale.par)
```

### Arguments

| | |
|---|---|
| mod | A character string equal to "G+E", "G" or "E" |
| theta | A vector of length 5 containing values for (alpha,beta_G,beta_E,gamma,delta) |
| nsite | Number of geographical locations |
| nloc | Number of loci |
| hap.pop.size | Haploid population size at each combination geographical location x locus. In the main inference function of the package (SunderInference), this number can vary across combinations of geographical locations and loci. Currently the present simulation function accepts only a single number as input. |

| | |
|---|---|
| nalM | Maximum number of alleles over the various loci |
| nalm | Minimum number of alleles over the various loci |
| var.par | Variance of the random field model for the environmental variable |
| scale.par | Scale parameter in the exponential covariance of the random field model for the environmental variable |

## Value

A list containing genotypes, geographical distance, environmental distances, model and parameters.

## Author(s)

Filippo Botta

## Examples

```
data(toydata,package='Sunder')
```

# Index