

# Package ‘hierfstat’

February 14, 2012

**Version** 0.04-6

**Date** 2011-11-23

**Title** Estimation and tests of hierarchical F-statistics

**Author** Jerome Goudet <jerome.goudet@unil.ch>

**Maintainer** Jerome Goudet <jerome.goudet@unil.ch>

**Depends** gtools

**Suggests** adegenet,ape

**Description** This R package allows the estimation of hierarchical F-statistics from haploid or diploid genetic data with any numbers of levels in the hierarchy, following the algorithm of Yang (Evolution, 1998, 52(4):950-956). Functions are also given to test via randomisations the significance of each F and variance components, using the likelihood-ratio statistics G -see Goudet etal (Genetics, 1996, 144(4): 1933-1940)

**License** GPL (>= 2)

**URL** <http://www.r-project.org>, <http://www.unil.ch/popgen/software/hierfstat.htm>

**Repository** CRAN

**Date/Publication** 2011-11-24 08:20:34

## R topics documented:

allele.count . . . . .	2
allelic.richness . . . . .	3
basic.stats . . . . .	4
boot.ppfis . . . . .	6
boot.ppfst . . . . .	7
boot.vc . . . . .	8
cfe.dist . . . . .	9

da.dist . . . . .	10
eucl.dist . . . . .	10
eucl.dist.trait . . . . .	11
exhier . . . . .	12
g.stats . . . . .	12
g.stats.glob . . . . .	13
genot2al . . . . .	15
getal . . . . .	16
getal.b . . . . .	17
gtrunchier . . . . .	17
hierfstat . . . . .	18
ind.count . . . . .	19
mat2vec . . . . .	19
nb.alleles . . . . .	20
nei.dist . . . . .	21
pcoa . . . . .	21
pop.freq . . . . .	22
pp.fst . . . . .	23
pp.sigma.loc . . . . .	23
prepdata . . . . .	24
print.pp.fst . . . . .	25
read.fstat . . . . .	25
read.fstat.data . . . . .	26
samp.between . . . . .	27
samp.between.within . . . . .	28
samp.within . . . . .	29
sim.freq . . . . .	30
sim.genot . . . . .	30
test.between . . . . .	31
test.between.within . . . . .	32
test.g . . . . .	33
test.within . . . . .	33
varcomp . . . . .	34
varcomp.glob . . . . .	36
vec2mat . . . . .	37
wc . . . . .	37
yangex . . . . .	38

**Index** **39**

---

allele.count	<i>Allelic counts</i>
--------------	-----------------------

---

**Description**

Counts the number of copies of the different alleles at each locus and population

**Usage**

```
allele.count(data,diploid=TRUE)
```

**Arguments**

**data** A data frame containing the population of origin in the first column and the genotypes in the following ones

**diploid** Whether the data are from diploid individuals

**Value**

A list of tables, –each with np (number of populations) columns and nl (number of loci) rows– of the count of each allele

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

**Examples**

```
data(gtrunchier)
allele.count(gtrunchier[, -2])
```

---

allelic.richness      *Estimates allelic richness*

---

**Description**

Estimates allelic richness, the rarefied allelic counts, per locus and population

**Usage**

```
allelic.richness(data,min.n=NULL,diploid=TRUE)
```

**Arguments**

**data** A data frame, with as many rows as individuals. The first column contains the population to which the individual belongs, the following to the different loci

**min.n** The number of alleles down to which the number of alleles should be rarefied. The default is the minimum number of individuals genotyped (times 2 for diploids)

**diploid** a boolean specifying whether individuals are diploid (default) or haploid

**Value**

**min.all** The number of alleles used for rarefaction

**Ar** A table with as many rows as loci and columns as populations containing the rarefied allele counts

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

**References**

El Mousadik A. and Petit R.J. (1996) High level of genetic differentiation for allelic richness among populations of the argan tree *argania spinosa* skeys endemic to Morocco. *Theoretical and Applied Genetics*, 92:832-839

Hurlbert S.H. (1971) The nonconcept of species diversity: a critique and alternative parameters. *Ecology*, 52:577-586

Petit R.J., El Mousadik A. and Pons O. (1998) Identifying populations for conservation on the basis of genetic markers. *Conservation Biology*, 12:844-855

**Examples**

```
data(gtrunchier)
allelic.richness(gtrunchier[, -1])
```

---

basic.stats

*Basic statistics*

---

**Description**

Estimates individual counts, allelic frequencies, observed heterozygosities and genetic diversities per locus and population. Also Estimates mean observed heterozygosities, mean gene diversities within population  $H_s$ , Gene diversities overall  $H_t$  and corrected  $H_{tp}$ , and  $D_{st}$ ,  $D_{stp}$ . Finally, estimates  $F_{st}$  and  $F_{stp}$  as well as  $F_{is}$  following Nei (1987) per locus and overall loci

**Usage**

```
basic.stats(data, diploid=TRUE, digits=4)
```

**Arguments**

data	a data frame where the first column contains the population to which the different individuals belong, and the following columns contain the genotype of the individuals -one locus per column-
diploid	Whether individuals are diploids (default) or haploids
digits	how many digits to print out in the output (default is 4)

**Value**

n. ind. samp	A table –with np (number of populations) columns and nl (number of loci) rows– of genotype counts
pop. freq	A list containing allele frequencies. Each element of the list is one locus. For each locus, Populations are in rows and alleles in column
Ho	A table –with np (number of populations) columns and nl (number of loci) rows– of observed heterozygosities
Hs	A table –with np (number of populations) columns and nl (number of loci) rows– of observed gene diversities
Fis	A table –with np (number of populations) columns and nl (number of loci) rows– of observed Fis
perloc	A table –with as many rows as loci– containing basic statistics Ho, Hs, Ht, Dst, Ht', Dst', Fst, Fst', Fis, Dest
overall	Basic statistics averaged over loci

**Note**

For the perloc and overall tables (see value section), the following statistics, defined in eq.7.38–7.43 pp.164–5 of Nei (1987) are estimated:

The observed heterozygosity

$$H_o = 1 - \sum_k \sum_i P_{kii}/np,$$

where  $P_{kii}$  represents the proportion of homozygote  $i$  in sample  $k$  and  $np$  the number of samples.

The within population gene diversity (sometimes misleadingly called expected heterozygosity):

$$H_s = \tilde{n}/(\tilde{n} - 1)[1 - \sum_i \bar{p}_i^2 - H_o/2\tilde{n}],$$

where  $\tilde{n} = np / \sum_k 1/n_k$  and  $\bar{p}_i^2 = \sum_k p_{ki}^2 / np$

The overall gene diversity

$$H_t = 1 - \sum_i \bar{p}_i^2 + H_s/(\tilde{n}np) - H_o/(2\tilde{n}np),$$

where  $\bar{p}_i = \sum_k p_{ki} / np$ .

The amount of gene diversity among samples  $Dst = H_t - H_s$

$$Dst' = np/(np - 1)Dst$$

$$Ht' = H_s + Dst'$$

$Fst = Dst/Ht'$ . (This is not the same as Nei's  $Gst$ , Nei's  $Gst$  is an estimator of  $Fst$  based on allele frequencies only)

$$Fst' = Dst'/Ht'$$

$$Fis = 1 - H_o/H_s$$

Last,  $Dest = np/(np - 1)(Ht' - Hs)/(1 - Hs)$  a measure of population differentiation as defined by Jost (2008) is also given

Here, the  $p_{ki}$  are unweighted by sample size. These statistics are estimated for each locus and an overall loci estimates is also given, as the unweighted average of the per locus estimates. In this way, monomorphic loci are accounted for (with estimated value of 0) in the overall estimates.

Note that the equations used here all rely on genotypic rather than allelic number and are corrected for heterozygosity.

### Author(s)

Jerome Goudet <jerome.goudet@unil.ch>

### References

- Nei M. (1987) Molecular Evolutionary Genetics. Columbia University Press
- Jost L (2008) GST and its relatives do not measure differentiation. *Molecular Ecology*, 17, 4015-4026.
- Nei M, Chesser R (1983) Estimation of fixation indexes and gene diversities. *Annals of Human Genetics*, 47, 253-259.

### See Also

[ind.count, pop.freq.](#)

### Examples

```
data(gtrunchier)
basic.stats(gtrunchier[, -1])
```

---

boot.ppfis	<i>Performs bootstrapping over loci of population's Fis</i>
------------	---

---

### Description

Performs bootstrapping over loci of population's Fis

### Usage

```
boot.ppfis(dat=dat, nboot=100, quant=c(0.025, 0.975), diploid=TRUE, dig=4, ...)
```

### Arguments

dat	a genetic data frame
nboot	number of bootstraps
quant	quantiles
diploid	whether diploid data
dig	digits to print
...	further arguments to pass to the function

**Value**

call	function call
fis.ci	Bootstrap ci of Fis per population

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

**Examples**

```
data(gtrunchier)
boot.ppfis(gtrunchier[, -2])
```

---

boot.ppfst	<i>Performs bootstrapping over loci of pairwise Fst</i>
------------	---

---

**Description**

Performs bootstrapping over loci of pairwise Fst

**Usage**

```
boot.ppfst(dat=dat, nboot=100, quant=c(0.025, 0.975), diploid=TRUE, dig=4, ...)
```

**Arguments**

dat	a genetic data frame
nboot	number of bootstraps
quant	the quantiles for bootstrapped ci
diploid	whether data are from diploid organisms
dig	numebr of digits to print
...	further arguments to pass to the function

**Value**

call	
ll	lower limit ci
ul	upper limit ci
vc.per.loc	for each pair of population, the variance components per locus

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

boot.vc

*Bootstrap confidence intervals for variance components***Description**

Provides a bootstrap confidence interval (over loci) for sums of the different variance components (equivalent to gene diversity estimates at the different levels), and the derived F-statistics, as suggested by Weir and Cockerham (1984). Will not run with less than 5 loci. Raymond and Rousset (199X) points out shortcomings of this method.

**Usage**

```
boot.vc(levels=levels, loci=loci, diploid=TRUE, nboot=1000, quant=c(0.025, 0.5, 0.975))
```

**Arguments**

levels	a data frame containing the different levels (factors) from the outermost (e.g. region) to the innermost before the individual
loci	a data frame containing the different loci
diploid	Specify whether the data are coming from diploid or haploid organisms (diploid is the default)
nboot	Specify the number of bootstrap to carry out. Default is 1000
quant	Specify which quantile to produce. Default is <code>c(0.025, 0.5, 0.975)</code> giving the percentile 95% CI and the median

**Value**

boot	a data frame with the bootstrapped variance components. Could be used for obtaining bootstrap ci of statistics not listed here.
res	a data frame with the bootstrap derived statistics. H stands for gene diversity, F for F-statistics
ci	Confidence interval for each statistic.

**References**

Raymond M and Rousset F, 1995. An exact test for population differentiation. *Evolution*. 49:1280-1283

Weir, B.S. (1996) *Genetic Data Analysis II*. Sinauer Associates.

Weir BS and Cockerham CC, 1984. Estimating F-statistics for the analysis of population structure. *Evolution* 38:1358-1370.

**See Also**

[varcomp.glob.](#)

**Examples**

```
#load data set
data(gtrunchier)
boot.vc(gtrunchier[,c(1:2)],gtrunchier[,~c(1:2)],nboot=100)
```

---

`cfe.dist`*Estimates Cavalli-Sforza & Edwards Chord distance*

---

**Description**

Estimates Cavalli-Sforza & Edwards Chord distance

**Usage**

```
cfe.dist(data,allic=FALSE,distance="cfe")
```

**Arguments**

<code>data</code>	a genetic data set, such as obtained from <code>read.fstat</code>
<code>allic</code>	whether to print out the distance for each locus
<code>distance</code>	not used yet. Goal is to have only one distance function

**Value**

either a matrix  $[np*(np-1)/2, nl]$  or a vector  $[np*(np-1)/2]$  of genetic distances among pairs 2-1, 3-1, 3-2, 4-1, etc

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

**Examples**

```
data(gtrunchier)
nei.dist(gtrunchier[,~1])
```

---

da.dist *Estimates Nei's DA distance*

---

**Description**

Estimates Nei's DA distance

**Usage**

```
da.dist(data,allic=FALSE,distance="da")
```

**Arguments**

data            a genetic data set, such as obtained from read.fstat  
allic            whether to print out the distance for each locus  
distance        not used yet. Goal is to have only one distance function

**Value**

a vector of genetic distances among pairs 2-1, 3-1, 3-2,4-1,etc

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

**Examples**

```
data(gtrunchier)
nei.dist(gtrunchier[,-1])
```

---

eucl.dist *Estimates euclidian distances*

---

**Description**

Estimates euclidian distances among pairs of samples from a genetic data set

**Usage**

```
eucl.dist(data,allic=FALSE,distance="eucl")
```

**Arguments**

data            a genetic data set, such as obtained from read.fstat  
allic            whether to print out the distance for each locus  
distance        not used yet. Goal is to have only one distance function

**Value**

either a matrix  $[np*(np-1)/2, nl]$  or a vector  $[np*(np-1)/2]$  of genetic distances among pairs 2-1, 3-1, 3-2, 4-1, etc

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

**Examples**

```
data(gtrunchier)
nei.dist(gtrunchier[, -1])
```

---

eucl.dist.trait      *calculates euclidian distance among populations for a trait*

---

**Description**

calculates euclidian distance among populations for a trait

**Usage**

```
eucl.dist.trait(data)
```

**Arguments**

data      a 2 columns data frame, the first column identifying the population of origin, the second the value for the trait

**Value**

a vector of euclidian distances 2-1, 3-1, 3-2 etc

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

---

 exhier

*Example data set with 4 levels, one diploid and one haploid locus*


---

**Description**

Example data set with 4 levels, one diploid and one haploid locus

**Usage**

```
data(exhier)
```

**Value**

lev1	outermost level
lev2	level 2
lev3	Level 3
lev4	Level 4
diplo	Diploid locus
haplo	Haploid locus

**Examples**

```
data(exhier)
varcomp(exhier[,1:5])
varcomp(exhier[,c(1:4,6)],diploid=FALSE)
```

---

 g.stats

*Calculates likelihood-ratio G-statistic on contingency table*


---

**Description**

Calculates the likelihood ratio G-statistic on a contingency table of alleles at one locus X sampling unit. The sampling unit could be any hierarchical level

**Usage**

```
g.stats(data,diploid=TRUE)
```

**Arguments**

data	a two-column data frame. The first column contains the sampling unit, the second the genotypes
diploid	Whether the data are from diploid (default) organisms

**Value**

obs	Observed contingency table
exp	Expected number of allelic observations
$\chi$ .squared	The chi-squared statistics, $\sum \frac{(O-E)^2}{E}$
g.stats	The likelihood ratio statistics, $2 \sum (O \log(\frac{O}{E}))$

**Author(s)**

Jerome Goudet, DEE, UNIL, CH-1015 Lausanne Switzerland  
<jerome.goudet@unil.ch>

**References**

- Goudet J., Raymond, M., DeMeeus, T. and Rousset F. (1996) Testing differentiation in diploid populations. *Genetics*. 144: 1933-1940
- Goudet J. (2005). Hierfstat, a package for R to compute and test variance components and F-statistics. *Molecular Ecology Notes*. 5:184-186
- Petit E., Balloux F. and Goudet J.(2001) Sex-biased dispersal in a migratory bat: A characterization using sex-specific demographic parameters. *Evolution* 55: 635-640.

**See Also**

[g.stats.glob](#).

**Examples**

```
data(gtrunchier)
attach(gtrunchier)
g.stats(data.frame(Patch,L21.V))
```

---

g.stats.glob

*Likelihood ratio G-statistic over loci*

---

**Description**

Calculates the likelihood ratio G-statistic on a contingency table of alleles at one locus X sampling unit, and sums this statistic over the loci provided. The sampling unit could be any hierarchical level (patch, locality, region,...). By default, diploid data are assumed

**Usage**

```
g.stats.glob(data,diploid=TRUE)
```

**Arguments**

<code>data</code>	a data frame made of $n_l+1$ column, $n_l$ being the number of loci. The first column contains the sampling unit, the others the multi-locus genotype. Only complete multi-locus genotypes are kept for calculation
<code>diploid</code>	Whether the data are from diploid (default) organisms

**Value**

<code>g.stats.l</code>	Per locus likelihood ratio statistic
<code>g.stats</code>	Overall loci likelihood ratio statistic

**Author(s)**

Jerome Goudet, DEE, UNIL, CH-1015 Lausanne Switzerland  
 <jerome.goudet@unil.ch>

**References**

- Goudet J. (2005). Hierfstat, a package for R to compute and test variance components and F-statistics. *Molecular Ecology Notes*. 5:184-186
- Goudet J., Raymond, M., DeMeeus, T. and Rousset F. (1996) Testing differentiation in diploid populations. *Genetics*. 144: 1933-1940
- Petit E., Balloux F. and Goudet J.(2001) Sex-biased dispersal in a migratory bat: A characterization using sex-specific demographic parameters. *Evolution* 55: 635-640.

**See Also**

[g.stats](#), [samp.within](#), [samp.between](#).

**Examples**

```
data(gtrunchier)
attach(gtrunchier)
nperm<-99
nobs<-length(Patch)
gglobs.o<-vector(length=(nperm+1))
gglobs.p<-vector(length=(nperm+1))
gglobs.l<-vector(length=(nperm+1))

gglobs.o[nperm+1]<-g.stats.glob(data.frame(Patch,gtrunchier[,-c(1,2)]))$g.stats
gglobs.p[nperm+1]<-g.stats.glob(data.frame(Patch,gtrunchier[,-c(1,2)]))$g.stats
gglobs.l[nperm+1]<-g.stats.glob(data.frame(Locality,gtrunchier[,-c(1,2)]))$g.stats

for (i in 1:nperm) #careful, might take a while
{
  gglobs.o[i]<-g.stats.glob(data.frame(Patch,gtrunchier[sample(Patch),-c(1,2)]))$g.stats
  gglobs.p[i]<-g.stats.glob(data.frame(Patch,gtrunchier[samp.within(Locality),-c(1,2)]))$g.stats
  gglobs.l[i]<-g.stats.glob(data.frame(Locality,gtrunchier[samp.between(Patch),-c(1,2)]))$g.stats
}
```

```

#p-value of first test (among patches)
p.globs.o<-sum(gglobs.o>=gglobs.o[nperm+1])/(nperm+1)

#p-value of second test (among patches within localities)
p.globs.p<-sum(gglobs.p>=gglobs.p[nperm+1])/(nperm+1)

#p-value of third test (among localities)
p.globs.l<-sum(gglobs.l>=gglobs.l[nperm+1])/(nperm+1)

#Are alleles associated at random among patches
p.globs.o

#Are alleles associated at random among patches within localities?
#Tests differentiation among patches within localities
p.globs.p

#Are alleles associated at random among localities, keeping patches as one unit?
#Tests differentiation among localities
p.globs.l

```

---

genot2al

*Separates diploid genotypes in its constituent alleles*


---

### Description

Separates the input vector of diploid genotypes in two vectors each containing one allele, and returns a vector of length  $2 \times \text{length}(y)$  with the second part being the second allele

### Usage

```
genot2al(y)
```

### Arguments

`y` the diploid genotypes at one locus

### Value

returns a vector of length  $2 \times \text{length}(y)$ , with the second half of the vector containing the second alleles

### Author(s)

Jerome Goudet, DEE, UNIL, CH-1015 Lausanne Switzerland  
<jerome.goudet@unil.ch>

### References

Goudet J. (2004). A library for R to compute and test variance components and F-statistics. In Prep

**See Also**

[varcomp](#).

**Examples**

```
data(gtrunchier)
genot2al(gtrunchier[,4])
```

---

getal	<i>Converts diploid genotypic data into allelic data</i>
-------	--

---

**Description**

Converts diploid genotypic data into allelic data

**Usage**

```
getal(data)
```

**Arguments**

data	a data frame where the first column contains the population to which the different individuals belong, and the following columns contain the genotype of the individuals -one locus per column-
------	---

**Value**

data.al	a new data frame, with twice as many row as the input data frame and one extra column. each row of the first half of the data frame contains the first allele for each locus, and each row of the second half of the data frame contains the second allele at the locus. The extra column in second position corresponds to the identifier of the individual to which the allele belongs
---------	--

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

**Examples**

```
data(gtrunchier)
getal(data.frame(gtrunchier[, -2]))
```

---

getal.b	<i>Converts diploid genotypic data into allelic data</i>
---------	--

---

**Description**

Converts a data frame of genotypic diploid data with as many lines as individuals (ni) and as many columns as loci (nl) into an array [ni,nl,2] of allelic data

**Usage**

```
getal.b(data)
```

**Arguments**

data	a data frame with ni rows and nl columns. Each line encodes one individual, each column contains the genotype at one locus of the individual
------	--

**Value**

an array [ni,nl,2] of alleles. The two alleles are stored in the third dimension of the array

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

**Examples**

```
data(gtrunchier)
#multilocus diploid genotype of the first individual
gtrunchier[1,-c(1:2)]
#the diploid genotype splitted in its two constituent alleles
getal.b(gtrunchier[,-c(1:2)])[1,,]
```

---

gtrunchier	<i>Genotypes at 6 microsatellite loci of Galba truncatula from different patches in Western Switzerland</i>
------------	---

---

**Description**

Data set consisting of the microsatellite genotypes of 370 Galba truncatula, a tiny freshwater snail, collecting from different localities and several patches within localities in Western Switzerland.

**Usage**

```
data(gtrunchier)
```

**Value**

Locality	Identifier of the locality of origin
Patch	Identifier of the patch of origin
L21.V	Genotype at locus L21.V. For instance the first individual carries allele 2 and 2 at this locus  gtrunchier\$L21.V[1]
L37.J	Genotype at locus L37.J
L20.B	Genotype at locus L20.B
L29.V	Genotype at locus L29.V
L36.B	Genotype at locus L36.B
L16.J	Genotype at locus L16.J

**References**

- Trouve S., L. Degen et al. (2000) Microsatellites in the hermaphroditic snail, *Lymnaea truncatula*, intermediate host of the liver fluke, *Fasciola hepatica*. *Molecular Ecology* 9: 1662-1664.
- Trouve S., Degen L. and Goudet J. (2005) Ecological components and evolution of selfing in the freshwater snail *Galba truncatula*. *Journal of Evolutionary Biology*. 18, 358-370

---

hierfstat

*General information on the hierfstat package*

---

**Description**

This package contains functions to estimate hierarchical F-statistics for any number of hierarchical levels using the method described in Yang (1998). It also contains functions allowing to test the significance of population differentiation at any given level using the likelihood ratio G-statistic, showed previously to be the most powerful statistic to test for differentiation (Goudet et al, 1996). The difficulty in a hierarchical design is to identify which units should be permuted. Functions `samp.within` and `samp.between` give permutations of a sequence that allows reordering of the observations in the original data frame. An exemple of application is given in the help page for function `g.stats.glob`.

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

**References**

- Goudet J. (2005). Hierfstat, a package for R to compute and test variance components and F-statistics. *Molecular Ecology Notes*. 5:184-186
- Goudet J., Raymond, M., DeMeeus, T. and Rousset F. (1996) Testing differentiation in diploid populations. *Genetics*. 144: 1933-1940
- Weir, B.S. (1996) *Genetic Data Analysis II*. Sinauer Associates.
- Yang, R.C. (1998). Estimating hierarchical F-statistics. *Evolution* 52(4):950-956

---

ind.count	<i>individual counts</i>
-----------	--------------------------

---

**Description**

Counts the number of individual genotyped per locus and population

**Usage**

```
ind.count(data)
```

**Arguments**

data            a data frame containing the population of origin in the first column and the genotypes in the following ones

**Value**

A table –with np (number of populations) columns and nl (number of loci) rows– of genotype counts

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

**Examples**

```
data(gtrunchier)
ind.count(gtrunchier[,-2])
```

---

mat2vec	<i>Rewrite a matrix as a vecor</i>
---------	------------------------------------

---

**Description**

transform lower triangular matrix in vector 1.2,1.3,2.3,1.4,2.4,3.4

**Usage**

```
mat2vec(mat)
```

**Arguments**

mat            a (square) matrix

**Value**

x a vector

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

**Examples**

```
mat2vec(matrix(1:16,nrow=4))
```

---

nb.alleles	<i>Number of different alleles</i>
------------	------------------------------------

---

**Description**

Counts the number of different alleles at each locus and population

**Usage**

```
nb.alleles(data,diploid=TRUE)
```

**Arguments**

data	A data frame containing the population of origin in the first column and the genotypes in the following ones
diploid	whether individuals are diploid

**Value**

A table, –with np (number of populations) columns and nl (number of loci) rows– of the number of different alleles

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

**Examples**

```
data(gtrunchier)
nb.alleles(gtrunchier[,-2])
```

---

nei.dist	<i>Estimates Nei's genetic distance</i>
----------	---

---

**Description**

Estimates Nei's unbiased genetic distance

**Usage**

```
nei.dist(data)
```

**Arguments**

data            a genetic data set, such as obtained from read.fstat

**Value**

a vector of genetic distances among pairs 2-1, 3-1, 3-2,4-1,etc

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

**Examples**

```
data(gtrunchier)
nei.dist(gtrunchier[, -1])
```

---

pcoa	<i>Principal coordinate analysis</i>
------	--------------------------------------

---

**Description**

principal coordinates analysis as described in Legendre & Legendre Numerical Ecology

**Usage**

```
pcoa(mat, plotit=TRUE, ...)
```

**Arguments**

mat            a distance matrix  
plotit        Whether to produce a plot of the pcoa  
...            further arguments (graphical for instance) to pass to the function

**Value**

valp            the eigen values of the pcoa  
vecp            the eigen vectors of the pcoa (the coordinates of observations)  
eucl            The cumulative euclidian distances among observations,

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

---

pop.freq            *Allelic frequencies*

---

**Description**

Estimates allelic frequencies for each population and locus

**Usage**

```
pop.freq(data,diploid=TRUE)
```

**Arguments**

data            a data frame where the first column contains the population to which the different individuals belong, and the following columns contain the genotype of the individuals -one locus per column-

diploid        specify whether the data set consists of diploid (default) or haploid data

**Value**

A list containing allele frequencies. Each element of the list is one locus. For each locus, Populations are in rows and alleles in column

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

**Examples**

```
data(gtrunchier)  
pop.freq(gtrunchier[,-2])
```

---

pp.fst                      *fst per pair*

---

**Description**

fst per pair

**Usage**

```
pp.fst(dat=dat,diploid=TRUE,...)
```

**Arguments**

dat	a genetic data frame
diploid	whether data from diploid organism
...	further arguments to pass to the function

**Value**

call	function call
fst.pp	pairwise Fsts
vc.per.loc	for each pair of population, the variance components per locus

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

---

pp.sigma.loc                      *wrapper to return per locus variance components*

---

**Description**

wrapper to return per locus variance components between pairs of samples x & y

**Usage**

```
pp.sigma.loc(x,y,dat=dat,diploid=TRUE,...)
```

**Arguments**

x,y	samples 1 and 2
dat	a genetic data set
diploid	whether dats are diploid
...	further arguments to pass to the function

**Value**

sigma.loc          variance components per locus

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

---

prepdata                      *Sort and renumber and recode levels for the hierarchical analysis if necessary*

---

**Description**

Called by varcomp to rearrange data. Should not need to be called directly

**Usage**

```
prepdata(data)
```

**Arguments**

data                      takes as input the different factor from the outermost (e.g. region) to the innermost (e.g. individual). This function must be applied before transformation of genotypic to allelic data

**Value**

a reordered,renumbered and sorted data frame

**References**

Goudet J. (2005). Hierfstat, a package for R to compute and test variance components and F-statistics. Molecular Ecology Notes. 5:184-186

**See Also**

[varcomp](#).

**Examples**

```
f1<-rep(c("B","C","A"),each=10)
f2<-rep(c("d","i","f","g","h","e"),each=5)
prepdata(data.frame(loc=f1,patch=f2,ind=1:30))
```

---

```
print.pp.fst          print function for pp.fst
```

---

**Description**

print function for pp.fst

**Usage**

```
## S3 method for class 'pp.fst'
print(x,...)
```

**Arguments**

```
x          an object of class pp.fst
...        further arguments to pass to the function
```

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

---

```
read.fstat          Reads data from a FSTAT file
```

---

**Description**

Imports a *FSTAT* data file into R. The data frame created is made of  $n_l+1$  columns,  $n_l$  being the number of loci. The first column corresponds to the Population identifier, the following columns contains the genotypes of the individuals.

**Usage**

```
read.fstat(fname, na.s = c("0", "00", "000", "0000", "00000", "000000", "NA"))
```

**Arguments**

```
fname          a file in the FSTAT format (http://www.unil.ch/popgen/software/fstat.htm): The file must have the following format:
The first line contains 4 numbers: the number of samples,  $n_p$ , the number of loci,  $n_l$ , the highest number used to label an allele,  $n_u$ , and a 1 if the code for alleles is a one digit number (1-9), a 2 if code for alleles is a 2 digit number (01-99) or a 3 if code for alleles is a 3 digit number (001-999). These 4 numbers need to be separated by any number of spaces.
The first line is immediately followed by  $n_l$  lines, each containing the name of a locus, in the order they will appear in the rest of the file.
On line  $n_l+2$ , a series of numbers as follow:
```

```
1      0102  0103  0101  0203          0      0303
```

The first number identifies the sample to which the individual belongs, the second is the genotype of the individual at the first locus, coded with a 2 digits number for each allele, the third is the genotype at the second locus, until locus `nl` is entered (in the example above, `nl=6`). Missing genotypes are encoded with 0, 00, 0000, 000000 or NA. Note that 0001 or 0100 are not a valid format, as both alleles at a locus have to be known, otherwise, the genotype is considered as missing. No empty lines are needed between samples.

`na.s` The strings that correspond to the missing value. *You should note have to change this*

### Value

a data frame containing the desired data, in a format adequate to pass to `varcomp`

### References

Goudet J. (1995). FSTAT (Version 1.2): A computer program to calculate F- statistics. *Journal of Heredity* 86:485-486

Goudet J. (2005). Hierfstat, a package for R to compute and test variance components and F-statistics. *Molecular Ecology Notes*. 5:184-186

### Examples

```
read.fstat(paste(.path.package("hierfstat"), "/extdata/diploid.dat", sep="", collapse=""))
```

---

<code>read.fstat.data</code>	<i>Reads data from a FSTAT file</i>
------------------------------	-------------------------------------

---

### Description

Imports a *FSTAT* data file into R. The data frame created is made of `nl+1` columns, `nl` being the number of loci. The first column corresponds to the Population identifier, the following columns contains the genotypes of the individuals.

### Usage

```
read.fstat.data(fname, na.s = c("0", "00", "000", "0000", "00000", "000000", "NA"))
```

### Arguments

`fname` a file in the FSTAT format (<http://www.unil.ch/popgen/software/fstat.htm>): The file must have the following format:  
The first line contains 4 numbers: the number of samples, `np`, the number of loci, `nl`, the highest number used to label an allele, `nu`, and a 1 if the code for alleles is a one digit number (1-9), a 2 if code for alleles is a 2 digit number

(01-99) or a 3 if code for alleles is a 3 digit number (001-999). These 4 numbers need to be separated by any number of spaces.

The first line is immediately followed by `n1` lines, each containing the name of a locus, in the order they will appear in the rest of the file.

On line `n1+2`, a series of numbers as follow:

```
1      0102  0103  0101  0203      0      0303
```

The first number identifies the sample to which the individual belongs, the second is the genotype of the individual at the first locus, coded with a 2 digits number for each allele, the third is the genotype at the second locus, until locus `n1` is entered (in the example above, `n1=6`). Missing genotypes are encoded with 0, 00, 0000, 000000 or NA. Note that 0001 or 0100 are not a valid format, as both alleles at a locus have to be known, otherwise, the genotype is considered as missing. No empty lines are needed between samples.

`na.s` The strings that correspond to the missing value. *You should note have to change this*

### Value

a data frame containing the desired data, in a format adequate to pass to `varcomp`

### References

Goudet J. (1995). FSTAT (Version 1.2): A computer program to calculate F- statistics. *Journal of Heredity* 86:485-486

Goudet J. (2005). Hierfstat, a package for R to compute and test variance components and F-statistics. *Molecular Ecology Notes*. 5:184-186

### Examples

```
read.fstat.data(paste(.path.package("hierfstat"), "/extdata/diploid.dat", sep="", collapse=""))
```

---

samp.between

*Shuffles a sequence among groups defined by the input vector*

---

### Description

Used to generate a permutation of a sequence `1:length(lev)`. blocks of observations are permuted, according to the vector `lev` passed to the function.

### Usage

```
samp.between(lev)
```

### Arguments

`lev` a vector containing the groups to be permuted.

**Value**

a vector `1:length(lev)` (with blocks defined by data) randomly permuted. Usually, one passes the result to reorder observations in a data set in order to carry out permutation-based tests

**Author(s)**

Jerome Goudet, DEE, UNIL, CH-1015 Lausanne Switzerland  
<jerome.goudet@unil.ch>

**References**

Goudet J. (2005). Hierfstat, a package for R to compute and test variance components and F-statistics. *Molecular Ecology Notes*. 5:184-186

**See Also**

[samp.within](#), [g.stats.glob](#).

**Examples**

```
samp.between(rep(1:4, each=4))  
#for an application see example in g.stats.glob
```

---

`samp.between.within`    *Shuffles a sequence*

---

**Description**

Used to generate a permutation of a sequence `1:length(inner.lev)`. blocks of observations defined by `inner.lev` are permuted within blocks defined by `outer.lev`

**Usage**

```
samp.between.within(inner.lev, outer.lev)
```

**Arguments**

<code>inner.lev</code>	a vector containing the groups to be permuted.
<code>outer.lev</code>	a vector containing the blocks within which observations are to be kept.

**Value**

a vector `1:length(lev)` (with blocks defined by data) randomly permuted. Usually, one passes the result to reorder observations in a data set in order to carry out permutation-based tests

**See Also**

[test.between.within](#).

---

`samp.within`*Shuffles a sequence within groups defined by the input vector*

---

**Description**

Used to generate a permutation of a sequence `1:length(lev)`. observations are permuted within blocks, according to the vector `lev` passed to the function.

**Usage**

```
samp.within(lev)
```

**Arguments**

`lev` a vector containing the group to which belongs the observations to be permuted.

**Value**

a vector `1:length(lev)` (with blocks defined by

`lev`

) randomly permuted. Usually, one passes the result to reorder observations in a data set in order to carry out permutation-based tests.

**Author(s)**

Jerome Goudet, DEE, UNIL, CH-1015 Lausanne Switzerland

<jerome.goudet@unil.ch>

**References**

Goudet J. (2005). Hierfstat, a package for R to compute and test variance components and F-statistics. *Molecular Ecology Notes*. 5:184-186

**See Also**

[samp.between.g.stats.glob.](#)

**Examples**

```
samp.within(rep(1:4,each=4))  
#for an application see example in g.stats.glob
```

---

sim.freq	<i>Simulates frequencies, for internal use only</i>
----------	---

---

**Description**

Simulates frequencies, for internal use only

---

sim.genot	<i>Simulates genotypes in an island model at equilibrium</i>
-----------	--

---

**Description**

Simulates genotypes from several individuals in several populations at several loci in an island model at equilibrium

**Usage**

```
sim.genot(size=20,nbal=4,nbloc=2,nbpop=3,N=1000,mig=0.001,mut=0.001,f=0)
```

**Arguments**

size	The number of individuals to sample per populations
nbal	The maximum number of alleles present at a locus
nbloc	The number of loci to simulate
nbpop	The number of populations to simulate
N	The population size of an island
mig	the proportion of migration among islands
mut	The loci mutation rate
f	the inbreeding coefficient

**Value**

a data frame with nbpop\*size lines and nbloc+1 columns. Individuals are in rows and genotypes in columns, the first column being the population identifier

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

**Examples**

```
dat<-sim.genot(nbpop=10,nbal=20,nbloc=10,mig=0.001,mut=0.0001,f=0.5)
basic.stats(dat)
```

---

test.between	<i>Tests the significance of the effect of test.lev on genetic differentiation</i>
--------------	--

---

**Description**

Tests the significance of the effect of test.lev on genetic differentiation

**Usage**

```
test.between(data, test.lev, rand.unit, nperm, ...)
```

**Arguments**

data	a data frame containing the genotypes for the different loci
test.lev	A vector containing the units from which to construct the contingency tables
rand.unit	A vector containing the assignment of each observation to the units to be permuted
nperm	The number of permutations to carry out for the test
...	Mainly here to allow passing diploid=FALSE if necessary

**Value**

g.star	A vector containing all the generated g-statistics, the last one being the observed
p.val	The p-value associated with the test

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

**Examples**

```
data(gtrunchier)
attach(gtrunchier)
#test whether the locality level has a significant effect on genetic structuring
test.between(gtrunchier[,-c(1,2)], test.lev=Locality, rand.unit=Patch)
```

---

`test.between.within`    *Tests the significance of the effect of test.lev on genetic differentiation*

---

### Description

Tests, using permutations of `rand.unit` within units defined by the vector `within` the significance of the contingency tables allele X (levels of `test.lev`)

### Usage

```
test.between.within(data, within, test.lev, rand.unit, nperm, ...)
```

### Arguments

<code>data</code>	a data frame containing the genotypes for the different loci
<code>within</code>	A vector containing the units in which to keep the observations
<code>test.lev</code>	A vector containing the units from which to construct the contingency tables
<code>rand.unit</code>	A vector containing the assignment of each observation to the units to be permuted
<code>nperm</code>	The number of permutations to carry out for the test
<code>...</code>	Mainly here to allow passing <code>diploid=FALSE</code> if necessary

### Value

<code>g.star</code>	A vector containing all the generated g-statistics, the last one being the observed
<code>p.val</code>	The p-value associated with the test

### Author(s)

Jerome Goudet <jerome.goudet@unil.ch>

### Examples

```
data(yangex)
attach(yangex)
#tests for the effect of spop on genetic structure
test.between.within(data.frame(genot), within=pop, test=spop, rand=sspop)
```

---

test.g	<i>Tests the significance of the effect of level on genetic differentiation</i>
--------	---

---

**Description**

Tests the significance of the effect of level on genetic differentiation

**Usage**

```
test.g(data = data, level, nperm = 100, ...)
```

**Arguments**

data	a data frame containing the genotypes for the different loci
level	A vector containing the assignment of each observation to its level
nperm	The number of permutations to carry out for the test
...	Mainly here to allow passing diploid=FALSE if necessary

**Value**

g.star	A vector containing all the generated g-statistics, the last one being the observed
p.val	The p-value associated with the test

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

**Examples**

```
data(gtrunchier)
attach(gtrunchier)
test.g(gtrunchier[,-c(1,2)], Locality)
```

---

test.within	<i>Tests the significance of the effect of inner.level on genetic differentiation within blocks defined by outer.level</i>
-------------	--

---

**Description**

Tests the significance of the effect of inner.level on genetic differentiation within blocks defined by outer.level

**Usage**

```
test.within(data, within, test.lev, nperm, ...)
```

**Arguments**

<code>data</code>	a data frame containing the genotypes for the different loci
<code>within</code>	A vector containing the units in which to keep the observations
<code>test.lev</code>	A vector containing the units from which to construct the contingency tables
<code>nperm</code>	The number of permutations to carry out for the test
<code>...</code>	Mainly here to allow passing <code>diploid=FALSE</code> if necessary

**Value**

<code>g.star</code>	A vector containing all the generated g-statistics, the last one being the observed
<code>p.val</code>	The p-value associated with the test

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

**Examples**

```
data(gtrunchier)
attach(gtrunchier)
#tests whether the patch level has a significant effect on genetic structure
test.within(gtrunchier[,-c(1,2)],within=Locality,test.lev=Patch)
```

---

varcomp

*Estimates variance components for each allele of a locus*

---

**Description**

Estimates variance components for each allele for a (fully) hierarchical random design defined by all but the last column of the data frame `data`, the last column containing the genetic data to analyse. Columns for the hierarchical design should be given from the outermost to the innermost before the individual (e.g. continent, region, population, patch,...)

**Usage**

```
varcomp(data,diploid=TRUE)
```

**Arguments**

<code>data</code>	a data frame that contains the different factors from the outermost (e.g. region) to the innermost before the individual. the last column of the data frame 'data' contains the locus to analyse, which can be multiallelic. Missing data are allowed.
<code>diploid</code>	a boolean stating whether the data come from diploid (TRUE=default) or haploid (FALSE) organisms

**Details**

The format for genotypes is simply the code for the 2 alleles put one behind the other, without space in between. For instance if allele 1 at the locus has code 23 and allele 2 39, the genotype format is 2339.

**Value**

df	the degrees of freedom for each level
k	the k matrix, the coefficients associated with the variance components
res	the variance components for each allele
overall	the variance components summed over alleles
F	a matrix of hierarchical F-statistics type-coefficients with the first line corresponding to $F_{(n-1)/n}, F_{(n-2)/n} \dots F_{i/n}$ and the diagonal corresponding to $F_{(n-1)/n}, F_{(n-2)/(n-1)}, F_{i/2}$

**Author(s)**

Jerome Goudet, DEE, UNIL, CH-1015 Lausanne Switzerland

<jerome.goudet@unil.ch>

<http://www.unil.ch/popgen/people/jerome.htm>

**References**

Goudet J. (2005). Hierfstat, a package for R to compute and test variance components and F-statistics. *Molecular Ecology Notes*. 5:184-186

Weir, B.S. (1996) *Genetic Data Analysis II*. Sinauer Associates.

Yang, R.C. (1998). Estimating hierarchical F-statistics. *Evolution* 52(4):950-956

**See Also**

[varcomp.glob.](#)

**Examples**

```
#load data set
data(gtrunchier)
attach(gtrunchier)
#
varcomp(data.frame(Locality,Patch,L21.V))
```

---

varcomp.glob	<i>Estimate variance components and hierarchical F-statistics over all loci</i>
--------------	---

---

### Description

Return multilocus estimators of variance components and F-statistics

### Usage

```
varcomp.glob(levels=levels, loci=loci, diploid=TRUE)
```

### Arguments

levels	a data frame containing the different levels (factors) from the outermost (e.g. region) to the innermost before the individual
loci	a data frame containing the different loci
diploid	Specify whether the data are coming from diploid or haploid organisms (diploid is the default)

### Value

loc	The variance components for each locus
overall	The variance components summed over all loci
F	a matrix of hierarchical F-statistics type-coefficients with the first line corresponding to $F_{(n-1)/n}, F_{(n-2)/n}, \dots, F_{i/n}$ and the diagonal corresponding to $F_{(n-1)/n}, F_{(n-2)/(n-1)}, F_{i/2}$

### Author(s)

Jerome Goudet DEE, UNIL, CH-1015 Lausanne Switzerland  
<jerome.goudet@unil.ch>

### References

Weir, B.S. (1996) Genetic Data Analysis II. Sinauer Associates.  
 Yang, R.C. (1998). Estimating hierarchical F-statistics. *Evolution* 52(4):950-956  
 Goudet J. (2005). Hierfstat, a package for R to compute and test variance components and F-statistics. *Molecular Ecology Notes*. 5:184-186

### See Also

[varcomp.](#)

**Examples**

```
#load data set
data(gtrunchier)
attach(gtrunchier)
varcomp.glob(data.frame(Locality,Patch),gtrunchier[,-c(1,2)])
```

---

vec2mat                      *Reads a vector into a matrix*

---

**Description**

Fills a lower triangular matrix from a vector and copy it to upper triangle

**Usage**

```
vec2mat(x)
```

**Arguments**

x                      a vector

**Value**

mat                    a matrix

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

---

wc                      *Computes Weir and Cockrham estimates of Fstatistics*

---

**Description**

Computes Weir and Cockrham estimates of Fstatistics

**Usage**

```
wc(ndat,diploid=TRUE,pol=0.0)
```

**Arguments**

ndat                    data frame with first column indicating population of origin and following representing loci

diploid                Whether data are diploid

pol                    level of polymorphism requested for inclusion. Note used for now

**Value**

sigma	variance components of allele frequencies for each allele, in the order among populations, among individuals within populations and within individuals
sigma.loc	variance components per locus
per.al	FST and FIS per allele
per.loc	FST and FIS per locus
FST	FST overall loci
FIS	FIS overall loci

**Author(s)**

Jerome Goudet <jerome.goudet@unil.ch>

**Examples**

```
data(gtrunchier)
wc(gtrunchier[, -1])
```

---

yangex

*Example data set from Yang (1998) appendix*

---

**Description**

Reproduce the example data set used in Yang's paper appendix. The genotype (column genot) is invented

**Usage**

```
data(exhier)
```

**Value**

pop	outermost level
spop	sub pop level
sspop	sub sub pop level
genot	dummy diploid genotype

**References**

Yang, R.C. (1998). Estimating hierarchical F-statistics. *Evolution* 52(4):950-956

**Examples**

```
data(yangex)
varcomp(yangex)
#the k matrix should be the same as matrix (A2) in Yang's appendix, p. 956
```

# Index

- \*Topic **datasets**
  - exhier, 12
  - gtrunchier, 17
  - yangex, 38
- \*Topic **manip**
  - genot2al, 15
  - getal, 16
  - getal.b, 17
  - prepdata, 24
  - read.fstat, 25
  - read.fstat.data, 26
  - samp.between, 27
  - samp.between.within, 28
  - samp.within, 29
- \*Topic **misc**
  - hierfstat, 18
- \*Topic **nonparametric**
  - test.between, 31
  - test.between.within, 32
  - test.g, 33
  - test.within, 33
- \*Topic **univar**
  - allele.count, 2
  - allelic.richness, 3
  - basic.stats, 4
  - boot.ppfis, 6
  - boot.vc, 8
  - cfe.dist, 9
  - da.dist, 10
  - eucl.dist, 10
  - g.stats, 12
  - g.stats.glob, 13
  - ind.count, 19
  - nb.alleles, 20
  - nei.dist, 21
  - pop.freq, 22
  - pp.fst, 23
  - pp.sigma.loc, 23
  - print.pp.fst, 25
  - varcomp, 34
  - varcomp.glob, 36
  - vec2mat, 37
  - wc, 37
- allele.count, 2
- allelic.richness, 3
- basic.stats, 4
- boot.ppfis, 6
- boot.ppfst, 7
- boot.vc, 8
- cfe.dist, 9
- da.dist, 10
- eucl.dist, 10
- eucl.dist.trait, 11
- exhier, 12
- g.stats, 12, 14
- g.stats.glob, 13, 13, 28, 29
- genot2al, 15
- getal, 16
- getal.b, 17
- gtrunchier, 17
- hierfstat, 18
- ind.count, 6, 19
- mat2vec, 19
- nb.alleles, 20
- nei.dist, 21
- pcoa, 21
- pop.freq, 6, 22
- pp.fst, 23
- pp.sigma.loc, 23
- prepdata, 24

print.pp.fst, [25](#)  
read.fstat, [25](#)  
read.fstat.data, [26](#)  
  
samp.between, [14](#), [27](#), [29](#)  
samp.between.within, [28](#)  
samp.within, [14](#), [28](#), [29](#)  
sim.freq, [30](#)  
sim.genot, [30](#)  
  
test.between, [31](#)  
test.between.within, [28](#), [32](#)  
test.g, [33](#)  
test.within, [33](#)  
  
varcomp, [16](#), [24](#), [34](#), [36](#)  
varcomp.glob, [8](#), [35](#), [36](#)  
vec2mat, [37](#)  
  
wc, [37](#)  
  
yangex, [38](#)