Available online at www.sciencedirect.com
ScienceDirect

ELSEVIER

Infection, Genetics and Evolution 7 (2007) 731–735

Infection, Genetics and Evolution

www.elsevier.com/locate/meegid

Short communication

A step-by-step tutorial to use HierFstat to analyse populations hierarchically structured at multiple levels

Thierry de Meeûs [a,*], Jérôme Goudet [b]

[a] *Génétique et Evolution des Maladies Infectieuses, Unité Mixte de Recherche 2724, Institute de Recherche pour le Développement, Centre National de la Recherche Scientifique, Centre IRD, 911 Av d'Agropolis, BP 64501, 34394 Montpellier Cedex 5, France*
[b] *Department of Ecology & Evolution, Biophore Building, UNIL, CH-1015 Lausanne, Switzerland*

Received 19 March 2007; received in revised form 11 July 2007; accepted 12 July 2007
Available online 19 July 2007

Abstract

The populations of parasites and infectious agents are most of the time structured in complex hierarchy that lies beyond the classical nested design described by Wright's F-statistics (F_{IS}, F_{ST} and F_{IT}). In this note we propose a user-friendly step-by-step notice for using recent software (HierFstat) that computes and test fixation indices for any hierarchical structure. We add some tricks and tips for some special data kind (haploid, single locus), some other procedure (bootstrap over loci) and how to handle crossed factors.

Keywords: Fixation indices; Population structure; Hierarchy

1. Introduction

Population biologists, and among them those studying host populations, their pathogens and their vectors are interested in studying natural populations through molecular markers. This is particularly true for molecular epidemiologists because this represents the sole (or nearly so) way to study the populations they are interested in (e.g. De Meeûs et al., 2004). The most widely used parameters to infer population structure are the so-called F-statistics (Wright, 1951; Nagylaki, 1998) and their unbiased estimators (Weir and Cockerham, 1984). Classically, these parameters are defined for three hierarchical levels. The F_{IS} measures the identity (or homozygosity) of alleles within individuals within sub-populations as compared to Hardy–Weinberg expectations, it is thus a measure of deviation from local panmixia (random union of gametes producing zygotes). F_{ST} measures identity of individuals within sub-populations as compared to individuals from other sub-populations within the total population, or the total homozygosity due to the Wahlund effect. It is thus a measure of differentiation between sub-populations. Finally, F_{IT} is a measure of homozygosity of individuals in the total population and thus measures the deviation from Hardy–Weinberg due to local deviation from panmixia and Wahlund effect. The three indices are connected by the famous relationship: $(1 - F_{IT}) = (1 - F_{IS})(1 - F_{ST})$. Note that F_{ST} can be computed for haploids but of course not F_{IS} or F_{IT}. This can be analysed by many different free downloadable software (see Goudet, 2005). However, the population of pathogenic agents might not be well described with these three levels. In particular, several individuals (infra-population) of a pathogenic agent can colonise an individual host (e.g. a patient), different individual hosts may group into different villages themselves belonging to particular counties, states, countries, continent, etc.... In such cases, a global analysis requires another algorithm (and software implementing it).

Recently, Goudet (2005) developed a package for R (R Development Core Team, 2007) based on Yang's (1998) algorithm, which provides a convenient way to compute and test the significance of hierarchical F-statistics for any number of hierarchical levels, that he called HierFstat. However, the use of this package requires some knowledge of the R language. Now, many molecular epidemiologists are not very familiar with R and this could seriously limit the use of HierFstat and all the benefits that can come from a global analysis of such subdivided data (see Nébavi et al., 2006 for a good example).

[*] Corresponding author. Tel.: +33 467 4163 10; fax: +33 67 4162 99.
E-mail address: demeeus@mpl.ird.fr (T. de Meeûs).

1567-1348/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.meegid.2007.07.005

While other softwares (Arlequin, GDA, TFPGA, reviewed in Excoffier and Heckel, 2006) offer the possibility to handle up to four hierarchical levels, HierFstat is the only program allowing for an unlimited number of levels, F-estimate and randomisation testing. There may also be other kind of subdividing factors such as date of sampling, sex of the host or the cohort it belongs to (age class), which are not hierarchical but crossed factors and will require special care. This is why in this note we propose a step-by-step and user-friendly tutorial to implement any kind of analysis with HierFstat, with special recommendations on data structure, a special interest to haploid data, how to handle single locus analyses, how to obtain bootstrap confidence intervals of the different F measured at different levels and how to handle crossed factors.

2. Data structure

For the following, the data should have the same format as the example file examplehier.txt (see the file as supplementary material available at http://gemi.mpl.ird.fr/SiteSGASS/deMeeus/ExampleFilesHierFstat.html) for three factor levels and five loci. Each column is separated by a tabulation, lev1, lev2 and lev3 represent different levels of population structure, lev1 being the most inclusive one but itself included in the total data set and lev3 the innermost one, but itself containing individuals. This means that individuals are grouped into different clusters of lev3, themselves included in different meta-clusters defined by lev2, which are themselves included in the partition defined by lev3. There are thus here two supplementary levels at each extreme of the hierarchy: the total population and the individuals (corresponding to F_{IT} and F_{IS}). Loc1, Loc2 ... Loc5 are the data obtained for five different loci. There may of course be more than five loci (actually five loci is a minimum for obtaining confidence intervals by bootstrap) and the number of hierarchical levels is not limited. The data file must be in text mode only. It is best if the labels used to define the state of each level are numbered sequentially, not repeating the labels (e.g. 1 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3; not 2 2 2 2 1 1 1 1 3 3 3 3) in the relevant column. In the same way, a sequence like 1 1 1 2 2 2 1 1 1 2 2 2 1 1 1 2 2 2 should be avoided. Thus, a labelling like the one presented in Table 1 is ideal.

It is easier if missing data are coded as “NA” (upper cases as R differentiate it from lower cases). If “0” are to be used for missing data, the user needs to specify it when the file is read into R, using the option of the `read.table` command `na.string = “0”`.

3. Estimating and testing hierarchical F-statistics

It is now assumed that you have downloaded and installed R in your computer (from http://www.r-project.org/) and the HierFstat package into it (from the menu “Package” click on “Install from a zip file” and browse where you copied the software). A good and gentle introduction to R can be found in Dalgaard (2002). Several tutorials and quick start guides can be found from R homepage at http://www.r-project.org/. And help for the different R commands may be obtained by typing the name of the command preceded by a question mark (e.g. `?library`). In the following, we also assume a Windows platform.

Table 1
Example of labels for factor levels

lev1	lev2	lev3
1	1	1
1	1	1
1	1	1
1	1	1
1	1	2
1	1	2
1	1	2
1	2	3
1	2	3
1	2	3
1	2	4
1	2	4
1	2	4
2	3	5
2	3	5
2	3	5
2	3	5
2	3	6
2	3	6
2	3	6
2	4	7
2	4	7

Launch R. From the R menu load HierFstat. You just need to click in the Menu “Package”, to click on “Load Package” and on “HierFstat” (or type the command `library(hierfstat)`). You then need to go to the directory where the data to analyse are present. In the R Menu “File” Click on “Change Dir ...” and browse to the directory where the data file is present (or type `setwd(‘‘mydir’’)`, using / -not \- between folders, e.g. `setwd(‘‘c:/myfolder/hierfstat/’’)`.

You need now to load the data in R. We will use the data from the file examplehier.txt available at http://gemi.mpl.ird.fr/SiteSGASS/deMeeus/ExampleFilesHierFstat.html. This is done by typing the following command:

```
data<-read.table("examplehier.txt",header=TRUE)

attach(data)
```

This instructs R that your data file should be read and stored in the R object named data. The option “`header = TRUE’’` means that you have named each column. Do respect capitalisation as the language behind R is case sensitive. The command `attach(data)` allows accessing directly the variable names. The file examplehier.txt is made of eight columns, the first three corresponding to the different hierarchical levels and the next five to the different loci (see `?read.table` for help).

It is convenient to define and name a data frame in R format that contains only loci (genetic) information. This is done by typing the following command:

```
loci<-data.frame(loc1,loc2,loc3,loc4,loc5)
```

or

```
loci<-data[,c(4:8)]
```

(this instructs R to store in a data frame the last five columns of the data set, those that contain the genotypic information). The genetic data are then in the data frame named loci. Similarly, a data frame containing only the hierarchical levels can be created:

```
levels<-data.frame(lev1,lev2,lev3)
```

or more simply:

```
levels<-data[,c(1:3)]
```

In order to estimate hierarchical F-statistics from these data, the following command can now be typed:

```
varcomp.glob(levels,loci)
```

This will produce the estimation of the variance components for each locus and overall, as well as the matrix of hierarchical F-statistics (in the output, this is the table that follows after the sign $F), which reads as in Table 2. In fact, for each column, the most interesting value are found in the last line before the zeros (always of the form $F_{\mathrm{lev}(i)/\mathrm{lev}(i-1)}$. These are here: $F_{\mathrm{Ind/lev3}}$ (F_{IS} equivalent), $F_{\mathrm{lev3/lev2}}$ (differentiation between levels of rank 3 within each level 2), $F_{\mathrm{lev2/lev1}}$ (differentiation between levels of rank 2 within each level 1) and $F_{\mathrm{lev1/Total}}$ (differentiation between levels of rank 1 within the total).

Table 2
Example of hierarchical F-statistics computed and presented by HierFstat

	lev1	lev2	lev3	Ind.
Total	$\boldsymbol{F}_{\mathbf{lev1/Total}}$	$F_{\mathrm{lev2/Total}}$	$F_{\mathrm{lev3/Total}}$	$F_{\mathrm{Ind/Total}}$
lev1	0	$\boldsymbol{F}_{\mathbf{lev2/lev1}}$	$F_{\mathrm{lev3/lev1}}$	$F_{\mathrm{Ind/lev1}}$
lev2	0	0	$\boldsymbol{F}_{\mathbf{lev3/lev2}}$	$F_{\mathrm{levInd/lev2}}$
lev3	0	0	0	$\boldsymbol{F}_{\mathbf{Ind/lev3}}$

"Ind." stands for individuals and "/" means "within". Most interesting measures are in bold.

Then you will probably want to test the significance of genetic differentiation at the different levels, controlling for the effects at the other levels. Note that it is straightforward to test $F_{\mathrm{Ind/lev3}}$ in Fstat (Goudet, 1995) keeping the labels for lev3 as sub-population names. HierFstat does not contain a routine that randomise alleles within sub-populations as in Fstat. The command to test the innermost level is `test.within`, help on which can be obtained by typing `?test.within`. For instance, to test the effect of lev3 in our example (the lowest in the hierarchy) the command to type is:

```
test.within(loci,test=lev3,within=lev2,nperm=1000)
```

This command means that you want to carry out 1000 permutations of individuals between units defined by lev3, but keeping them within units defined by lev2. Of course you can set `nperm` to a higher value (e.g. 10,000). Similarly, to test the outermost level, the command is `test.between` (see `?test.between` for help). In our example, we would type:

```
test.between(loci,rand.unit=lev2,test=lev1,
     nperm=1000)
```

Here, whole units of lev2 will be permuted among units defined by lev1 a 1000 times. To perform tests for all other levels, the command to use is `test.between.within` (`?test.between.within` for help). In our example, to test the effect of level 2, type:

```
test.between.within(loci,within=lev1,
     rand.unit=lev3,test=lev2, nperm=1000)
```

This means that you want to carry out 1000 permutations of units defined by lev3 between units defined by lev2, but keeping them within units defined by lev1 (units of rank 2 are permuted

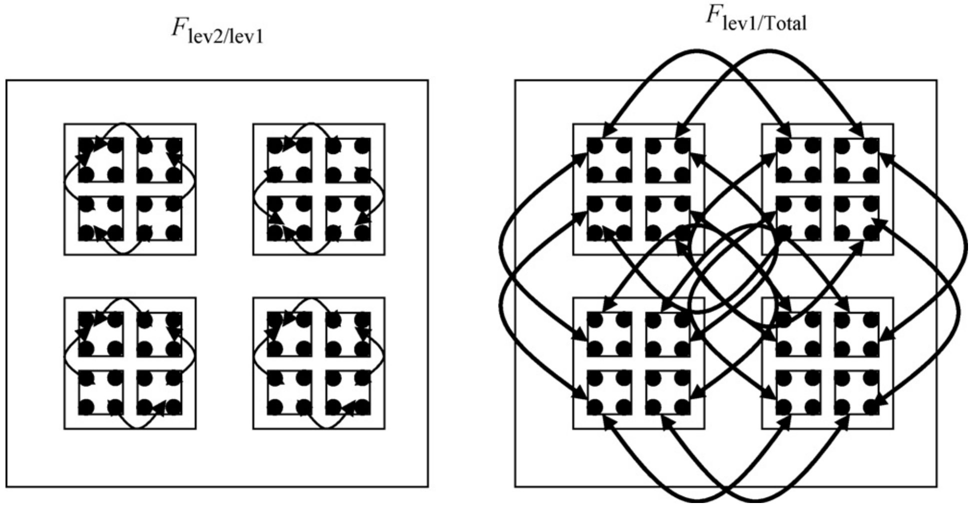

Fig. 1. Schematic representation of how permutations are handled under HierFstat to test for different levels of population structure. To test the significance of differentiation due to level 2 (smallest squares) $F_{\mathrm{lev2/lev1}}$, individuals (black dots) are randomly permuted across units of lev2 within each entity defined by level 1 (medium squares). To test for the effect of level 1 $F_{\mathrm{lev1/Total}}$, all individuals of each entity defined by level 2 (smallest squares) are randomly permuted together across entities defined by level 1. For each randomisation, a new F is computed (corresponding to a possible F under the null hypothesis of no differentiation at the level tested). The P-value of the test thus corresponds to the proportion of times a value as large or larger was obtained during the permutations (under the null hypothesis).

between units of rank 1). Fig. 1 illustrates the process for two factor levels.

While analysing the data file examplehier.txt, you will see that no level appears significant except the second one lev3, which displays a differentiation of $F_{lev3/lev2} = 0.026$ with an associated *P*-value of 0.003. Note that you might obtain a slightly different *P*-value, since this is estimated via permutations.

4. Testing one locus

It may be desirable to check if the same trend is followed by all or most loci. We thus need to estimate and test hierarchical *F*'s for each locus separately. This can be done using the following commands (example given for Loc1 and two levels, note that the commands vary a bit):

```
varcomp(data.frame(lev1,lev2,Loc1))

test.within(data.frame(Loc1), test=lev2,
     within=lev1, nperm=1000)

test.between(data.frame(Loc1),rand.unit=lev2,
     test=lev1,nperm=1000)
```

5. Special case: one haploid locus

If you have one haploid locus (mitochondrial, multilocus genotype), named here Haplo, you will specify the option `diploid = FALSE` to the command `varcomp` (again, for help on this function, use `?varcomp`):

```
varcomp(data.frame(lev1,lev2,Haplo),diploid=FALSE)

test.within(data.frame(Haplo), test=lev2, within=lev1, nperm=1000,

diploid=FALSE)

test.between((data.frame(Haplo),rand.unit=lev2,test=lev1, nperm=1000,

diploid=FALSE)
```

6. Bootstrapping over loci

Bootstrap is a convenient method to obtain confidence intervals that is widely used in population genetics. In HierFstat, this is done with the function `boot.vc` (`?boot.vc` for help), which is used with the same syntax as `varcomp.glob`. The bootstrap confidence intervals are obtained by typing:

```
boot.vc(data.frame(lev1,lev2,lev3),loci)
```

Note that an error message will appear each time you try bootstrapping over less than five polymorphic loci.

7. Special recommendations about the sampling design

Hierarchical *F*-statistics, and their associated tests, can and should only be applied, by definition, to hierarchical and thus nested designs. As an example, the different individual parasites can be contained within different individual hosts, themselves contained within different geographical locations which may be themselves contained within different continents. This design is nested because one individual parasite cannot be met in more than one individual host, location, and continent. But other kind of factors can be met that can influence population genetics of parasites such as the sex of the host (e.g. Caillaud et al., 2006), the year of sampling or the species of host. For these factors, the same rank can be found in different units of another level (for instance, male and female hosts are present in village 1 and in village 2). Thus, these factors are not nested but crossed and cannot be used as nested ones. An easy way to handle such a factor is to compute and test its contribution to the partition of genetic diversity (its *F*) independently within each nested factor so that the crossed factor is the only one remaining. For instance, if one aims at measuring differentiation of parasite infra-populations from two different host species in different sites, then the differentiation due to host species differences should be measured as $F_{\mathrm{SP/Site}\text{-}i}$ independently in each site. Note that the effect of individual hosts can still be controlled for as within each site one parasite cannot be present in two different species of host (infra-populations are nested in host species), in that case the significance of $F_{\mathrm{SP/Site}\text{-}i}$ in site *i* is tested randomising infra-populations between host species in site *i*. This procedure leads to as many *F* estimates and corresponding *P*-values as there are sites sampled (say *n*) were the tests are undertaken. A convenient way to obtain a global test is to combine these *n P*-values with a Fisher procedure (Fisher, 1970). The expression $-2\sum_{i=1}^{n}\log_e(P_i)$, where P_i are the different *P*-values obtained, follows a Chi square distribution with 2*n* degrees of freedom. The Fisher procedure may be difficult to apply in particular cases. Care must be taken when the distribution of the *P*-values are U shaped (many values close to 0 and/or 1), which should be rarely encountered but is still possible. For a discussion on such issues the readers are invited to read the article from Goudet (1999).

8. Concluding remarks

The possibility to analyse globally the effect of an unlimited number hierarchical levels brings a significant new degree of freedom to population biologists analysing natural populations through molecular markers, in particular for parasites and infectious agents that often arrange sub-populations into such designs (individual hosts, host populations, etc.. . .). The users must be warned that because all levels are taken into account, the genetic variance is partitioned hierarchically into all these levels and the corresponding F will not be simply connected to a number of migrants. Lower levels may concentrate most of the genetic variation letting little degree of freedom to higher levels. A very small and not significant F may thus not necessarily mean free migration between units defined by the corresponding level of population structure but simply that most of the variation is found at lower levels. A generalisation of the procedure that would allow analysing nested and crossed factors together is still lacking and would help escape the caveats of combining procedures. The model was explicitly written (Johannesson and Tatarenkov, 1997) but software is still needed.

Acknowledgements

The author would like to thank two anonymous referees whose comments considerably helped improve the manuscript. This article was partly inspired by a workshop financed by the ''Réseau Ecologie des Intéractions Durables''. T. de Meeûs is financed by the CNRS and IRD.

References

Caillaud, D., Prugnolle, F., Durand, P., Théron, A., De Meeûs, T., 2006. Host sex and parasite genetic diversity. Microbes Infect. 8, 2477–2483.

De Meeûs, T., Humair, P.F., Delaye, C., Grunau, C., Renaud, F., 2004. Non-Mendelian transmission of alleles at microsatellite loci: an example in *Ixodes ricinus*, the vector of Lyme disease. Int. J. Parasitol. 34, 943–950.

Dalgaard, P., 2002. Introductory Statistics with R. Springer, New York.

Excoffier, L., Heckel, G., 2006. Computer programs for population genetics data analysis: a survival guide. Nat. Rev. Genet. 7, 745–758.

Fisher, R.A., 1970. Statistical Methods for Research Workers, 14th ed. Oliver and Boyd, Edinburgh.

Goudet, J., 1995. FSTAT (vers. 1. 2): a computer program to calculate F-statistics. J. Hered. 86, 485–486.

Goudet, J., 1999. An improved procedure for testing the effects of key innovations on rate of speciation. Am. Nat. 153, 550–555.

Goudet, J., 2005. HIERFSTAT, a package for R to compute and test hierarchical F-statistics. Mol. Ecol. Notes 5, 184–186.

Johannesson, K., Tatarenkov, A., 1997. Allozyme variation in a snail (*Littorina saxatilis*)—deconfounding the effects of microhabitat and gene flow. Evolution 51, 402–409.

Nagylaki, T., 1998. Fixation indices in subdivided populations. Genetics 148, 1325–1332.

Nébavi, F., Ayala, F.J., Renaud, F., Bertout, S., Eholié, S., Koné, M., Mallié, M., De Meeûs, T., 2006. Clonal population structure and genetic diversity of *Candida albicans* in AIDS patients from Abidjan (Côte d'Ivoire). Proc. Natl. Acad. Sci. U.S.A. 103, 3663–3668.

R Development Core Team, 2007. R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org, ISBN 3-900051-07-0.

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

Wright, S., 1951. The genetical structure of populations. Ann. Eugen. 15, 323–354.

Yang, R.C., 1998. Estimating hierarchical F-statistics. Evolution 52, 950–956.