Contents

List of Figures

List of Tables

Chapter 1

General introduction

At the onset of the century, the two scientists Hardy (1908) and Weinberg (1908) discovered independently the basic law, or principle, of population genetics, which predicts the fate of genotypic and allelic frequencies in idealised populations. This principle states that, in an infinite sized Mendelian population, allele and genotype frequencies stay constant over time and therefore, the population does not evolve.

This principle defined the basis of population genetics theory and opened the way for the first generation of theoretical population geneticists, Wright, Fisher and Haldane. While these three workers prepared the ground for many investigations of known and unknown evolutionary problems, their work was difficult to follow by biologists and was described as

> 'As technical a body of research as that in statistical mechanics, say, and requiring as detailed a study' Bartlett (1955)

and

> 'Brilliant intuitions, daring approximations, arguments set out so briefly that one was not always sure precisely what was being argued, however much diluted by passages of limpid lucidity, posed a formidable task for the reader.' Gale (1990)

Following this early work, scientists such as Kimura and Feller, in the fifties, started a systematic examination of the writings of the founders of the subject, and have clarified many arguments by setting them out in detail and discussing them in a more rigorous manner (Gale, 1990).

The problematic of population genetics at this time was the description and explanation of genetic variation within and among populations. It remains its problematic some forty years later (Lewontin, 1991).

Before the development of biochemical and molecular techniques, genetic variation was difficult to measure since genes could only be perceived through the conspicuous phenotype of the individuals. Despite these difficulties, Wright (1943) used theoretical predictions to explain the genetic polymorphism of flower colour in the desert snow *Linanthus parryae*, while others were focusing on the shell colour polymorphism of the land snail *Cepaea nemoralis* (Lamotte, 1951,1959; Cain & Sheppard, 1950, 1954).

The independent discovery of the application of protein gel electrophoresis to genetic studies by Harris (1966) and Lewontin & Hubby (1966) provided direct access to an astonishingly large quantity of variability (Lewontin, 1991). Whilst it opened passionate discussions about the evolutionary basis of this polymorphism (Lewontin, 1974; Kimura, 1983), it also provided robust data sets to which statistical methods could be applied.

The rapid progress made by molecular biologists at the onset of the eighties opened the doors to yet more information on the genetic make-up of species.

Genetic variation within and among populations can be described in terms of allele and genotypic frequencies. As Wright (1931) pointed out, the proportion of heterozygotes in the total population is a good indicator of this variation. He developed statistics, called fixation indices or F-statistics, that partition the proportion of heterozygotes into within and among population components. These quantities, however, need to be estimated, since they are only based on samples of the total population. Even if the whole set of populations were to be sampled, the genetic sampling of gametes would still be occurring every generation. The work that Nei (1973, 1975, 1977) and Cockerham (1969, 1973) have initiated on the estimation of these quantities is very useful but has laid a trap for the unwary because comparisons between quantities estimated by one or the other approach are not valid, as will be shown in Chapter 3.

Since describing the fate of genes within and among populations is the main area of interest, it would be of interest to define what a population is. This task was undertaken by Crawford (1984), but a definite answer was not found. Since a reference population is one where mating occurs at random, developing tools that detect such units, if they exist, could lead to dramatic improvements in the understanding of the

genetic structure of populations, as will be shown in Chapters 4 and 5.

While of great evolutionary interest, the description and understanding of the processes maintaining genetic variation remained for a long time the preoccupation of relatively few biologists. The increasing awareness in contemporary society regarding ethical questions posed by ecology and genetics makes the problem of interest to a much wider audience. The issues raised by conservation biology have helped to place the subject of genetic variation and its maintenance firmly in 'limelight'. Population geneticists are requested to help in understanding the risks associated with the release into the environment of Genetically Modified Organisms. Forensic science, particularily DNA finger-printing, is part of the apparatus used in courts of law to determine innocence or guilt. These three examples should be sufficient to emphasize how crucial it is that population geneticists state clearly what can be inferred from their studies, as well as to highlight areas in which they feel unable to make definite statements. To this end, it seems important to develop theoretical models of structured populations, to test their predictions with Monte-Carlo simulations using tools of an appropriate nature, and to apply these tools to biological models of relevance to the problem.

æ

Chapter 2

Population genetic models and analytical solutions

2.1 Introduction

The understanding of the genetical structure of natural populations has been greatly enhanced by the modelling of population structure. The pioneer of this approach was Wright (1931) with his island model of population. He considered a monoecious, diploid population with discrete (non-overlapping) generations, subdivided into an infinite number of finite sized islands (named sub-populations, gamodemes, demes or local populations). He focussed his attention on a one locus, two allele system. Each island exchanges migrants at a rate m, with migrants coming from any of the other islands. With an infinite number of islands, the allele frequencies of the total population do not change from one generation to the next and therefore, the allele frequencies in the migrant pool also stay constant. Migration could be haploid (gametes) or diploid (individuals). Nagylaki (1983) showed that the type of migration has little influence on the general outcome of the model. The island model has been enhanced by Latter (1973), who considered a finite set of finite sized islands. Slatkin (1985a) called this second version the n-island model. The main difference between the n-island model and Wright's version is that allele frequencies fluctuate in the former, leading to somewhat more complex analytical solutions than the latter.

Kimura (1955) introduced the first geographically structured model, the stepping-stone. Each deme can exchange migrants only with its closest neighbours. The number of neighbouring demes available for exchanges of migrants is called connectedness in

the rest of this work. A connectedness of two represents a one-dimensional stepping-stone model and would correspond to a species living in a linear habitat like a river bank, a sea shore, or a road edge. Increasing the connectedness leads to two- (connectedness 4) or three-dimensional (connectedness 6/8) stepping-stone models. The higher the connectedness, the closer the model is to an island model, which could be described as a stepping-stone of connectedness $(D-1)$, D being the number of islands. Restricting migration to the nearest demes is as unrealistic as hyper-connectedness, and an intermediate model is developed here, called a pseudo-neighbourhood, where the probability of migrants arriving at a deme is a decreasing function of distance from that deme.

There remains the possibility that, in reality, no truly panmictic unit (deme) may exist. To take account of this type of population structure, another set of models has been developed, in which no panmictic unit is assumed: the isolation by distance, or neighbourhood model, of Wright (1943). In this model, each individual disperses its genes according to a decreasing function of distance (the pseudo-neighbourhood described above could be defined as a model in which demes disperse their genes according to a decreasing function of distance). No discrete structure is assumed, but a useful device, the neighbourhood size, can be defined: it consists of the area from which the parents of the central individual could be considered as if drawn at random (Wright, 1943). This area is defined as a circle of radius 2σ centred on the individual under investigation, providing that the distribution of dispersed particles (gametes or individuals) is normal, where σ is the parent to offspring dispersal standard deviation. To implement this model, one must make very restrictive assumptions about growth rate (Poisson distribution of the numbers of offspring) and spatial distribution of individuals (if individuals are not constrained to occupy intersections of a lattice grid, the population will eventually collapse into a biological black-hole (Felsenstein, 1975)).

2.2 Necessary prerequisites.

The first step in modelling is the definition of the goals. The aim of this research is to understand the behaviour of F-statistics as a function of biological and genetical parameters. It is therefore useful to have a model with the maximum possible information. Maximum possible information in population genetics is given by the probability of identity by descent (Malécot, 1948). The probability of identity by descent is the probability that two alleles are descended from the same common ancestral allele. It is usually contrasted with the probability of identity in state, which is the probability that the two alleles cannot be distinguished by the observer. This last probability is dependent on the devices used to detect genetic variation (Cockerham, 1984; Cockerham & Weir, 1987).

2.2.1 Identity by descent versus identity in state.

To implement a computer program that will display the identity by descent, it is necessary and sufficient to have all the alleles in the starting generation with a different label. That is, if the diploid population consists of D demes and N individuals per deme, the starting generation will contain $2DN$ different alleles.

All alleles in subsequent generations bearing the same label will therefore be descended from the same unique allele of the starting generation. Mutation can be included in this model, providing that each new mutant allele possesses a new label. This model without mutation could be called the $2DN$ state model. With mutation it is the infinite allele model (e.g. Hartl & Clark, 1989). To relate this model to biological reality and quantify the disparity between identity by descent and identity in state, it is possible to implement a procedure that will reduce the number of alleles (labels) present in the starting generation. The procedure consists of assigning at random one of k allelic states to the $2DN$ allele array (Figure 2.1). An equivalence relation or mapping $R(k)$ (Figure 2.2) between the infinite and the k alleles is therefore defined. The mapping $R(k)$ is then applied to subsequent generations. It is worth noticing that this mapping is independent of population structure, migration pattern and selfing proportion.

2.2.2 Efficient procedure to build generations.

Drawing random numbers is time consuming and should be avoided if the quality of the results is not to be affected. Gliddon (pers. comm.) suggested that it is only necessary to apply the genetic sampling rules to the first generation. The subsequent generations can then be built from replicates of the first. The $2DN$ ordered labels of the starting generations can also be considered as location markers. The procedure can be described as follows (Figures 2.3 & 2.4) : the leftmost column of Figure 2.4

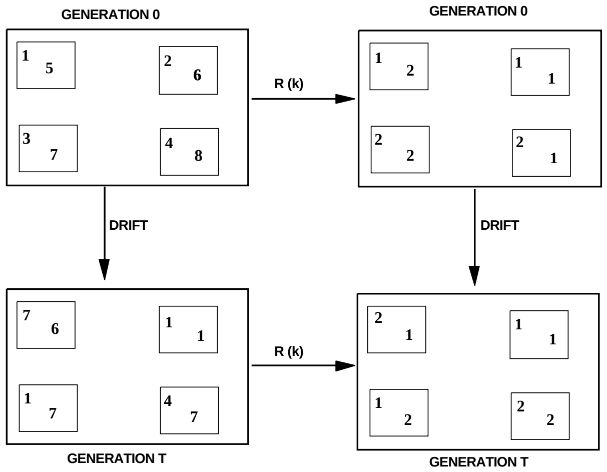

Figure 2.1: Reduction of the number of alleles

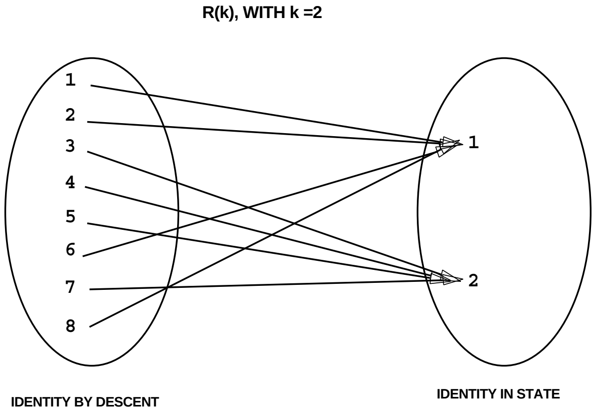

Figure 2.2: The equivalence relation between identity by descent and identity in state

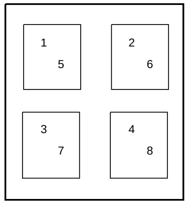

Figure 2.3: location of allele prior to mating

represents the genotypic array prior to mating. Each allele is positioned according to Figure 2.3. The middle column of Figure 2.4 represents the genetic sampling between generation $T-1$ and T. This genetic sampling is also equivalent to a transition matrix from a Markov chain (Hartl & Clark, 1989). Focussing on the top row of Figure 2.4, position 5 and 7 (position refers to number in Figure 2.3) of generation 1 (filled circles) are occupied by the allele in position 4 in generation 0. The allele in position 4 at generation 0 is D, hence the presence of D at position 5 and 7 in generation 1. The same process is applied to subsequent generations. The middle row of Figure 2.4 focusses on the outcomes of allele D: the transition matrix between generations 1 and 2 shows that this allele could be picked either via position 5 or position 7. Empty circles are for position 7 whilst black circles are for position 5. Mutation can be added to this procedure, after the genetic sampling stage. To model a multi-locus system with this procedure (with or without recombination), a position is given to individuals rather than alleles.

The obvious advantage of the procedure described is the avoidance of at least $2DN$ draws in a random number generator each generation (more if there is migration and selfing). A rough estimate of the number of possible states of the transition matrix is $2DN^{2DN}$ (each $2DN$ location can be occupied by one of $2DN$ alleles). How many transition matrix states one needs to generate to be sure of the reliability of the results

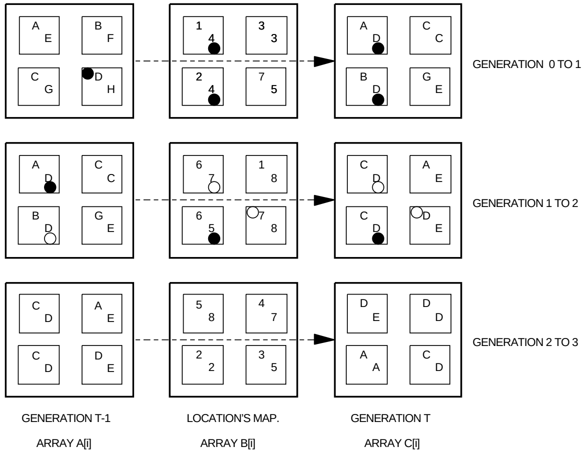

Figure 2.4: Equivalence between genetic sampling and transition matrix

is still unknown, but simulations with as few as 5 transition matrix states for an island model gave results similar after 100 generations to simulations where draws were made in the random number generator each generation. Caution is needed however for stepping-stone models: since migration is restricted to adjacent demes, if migration is low, it is quite likely that some pairs of demes will not exchange migrants if the number of replicates is too low. In the case of the one-dimensional stepping-stone model, if no migration occurs between two adjacent demes in all replicates, then the demes lying on each side of these two demes will be effectively completely isolated. The number of replicated first generations needs therefore to be higher for low migration and geographically structured populations. As a result of the consideration given above, the results displayed are based on data sets built from either 20, 50 or 100 replicates of the transition matrix.

2.2.3 Random number generators.

One of the stumbling-blocks of stochastic computer modelling is due to the generation of random numbers. In fact, it is nearly impossible to generate a true random sequence on a digital computer (one way would be connection to a truly random phenomenon such as the noise of an electronic diode, but the procedure is not simple (Ripley, 1987)). Random number generators are in fact pseudo random, that is, they are based on a deterministic equation that produces a random-like sequence as output. One of the equations that has proved to be fairly reliable is the Multiplicative Linear Congruential Generator (MLCG):

$$X_{n+1} = (aX_n + c) \bmod m, n \geq 0 \tag{2.1}$$

where m, the modulus, is a positive integer, a and c are both positive and less than m and X_0 is a positive integer between 0 and m (Knuth, 1981). Depending on the choice of a, c and m, one will obtain a more or less random sequence. Figures 2.5 to 2.8 give examples of such MLCGs extracted from the literature.

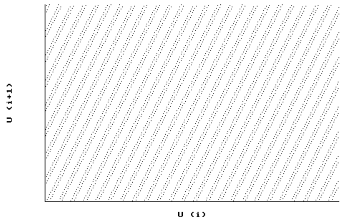

Figure 2.5: Example of a bad random number (Park & Miller, 1988). Phase plot U_i, U_{i+1}. X-axis from 0 to 1 (total phase space). This generator is a 16 bit version of the infamous RANDU

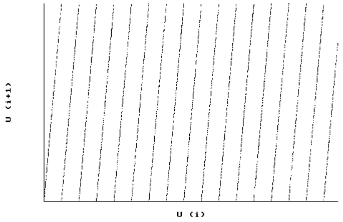

Figure 2.7: Minimum standard RND advocated by Park and Miller (1988). X-axis from 0 to 0.001. The period is long (of the order of the modulus), but the fine grain scale shows obvious interleaving.

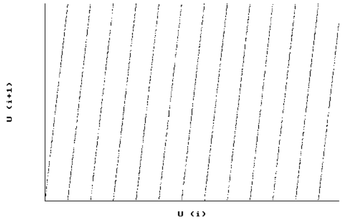

Figure 2.6: Random number generator of early version of Turbo Pascal (Park & Miller, 1988). Although not as bad as above, the lattice structure can easily be seen. X-axis from 0 to 0.1.

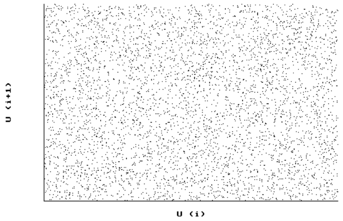

Figure 2.8: Random number generator proposed by L'Ecuyer (1988). X-axis from 0 to 0.001. This generator combines two of the best MLGC's known.

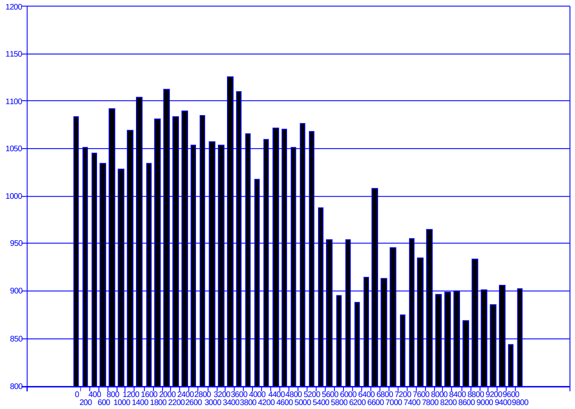

Figure 2.9: The result of 50000 draws in the integer random function of Turbo Pascal. A good random number generator?

Testing the quality of random number generators is still a matter of investigation and no less than 20 empirical tests exist. For a generator to be good, it should pass all the tests. Figures 2.5 to 2.8 provide an empirical picture of the quality of random number generators (L'Ecuyer, 1988). Only Figure 2.8 provides a random-like pattern. This generator consists of the combination of two of the best 16-bit MLCG according to L'Ecuyer (1988). The period of this generator is larger than 210^{18} and it passed all 21 empirical tests described in L'Ecuyer (1988). The Pascal code for it is found in L'Ecuyer (1988) and is reproduced in Annex A.1.1. The generator in Figure 2.7, advocated by Park and Miller (1988) as a minimal standard, scores quite badly in the spectral test (L'Ecuyer, 1988), shows a fairly coarse lattice structure and possesses a period of only 210^{10}.

Another warning needs to be made about random number generators provided by commercial packages: Figure 2.9 shows the results of 50000 draws in the integer random function of Turbo Pascal Version 6.00 using 10000 as a maximum. The results are sorted and stored in 50 classes (each of width 200). Figure 2.9 shows a strong bias favouring low integer values. This pattern, however, is not found when using the integer part of the real random number generator multiplied by 10000.

To generate normal and exponential deviates, the following algorithms were implemented (Ripley, 1987):

Exponential deviate:

1. Draw a random number between 0 and 1 from a uniform distribution.
2. The exponential deviate is the absolute value of the natural logarithm of the random number drawn in (1). If the exponential distribution has a mean λ different from one, multiply the result of (2) by λ.

Normal deviate

1. Draw two random numbers x_1 and x_2 between -1 and 1 from a uniform distribution.
2. Repeat step (1) until x_1 and x_2 belong to the unit circle, that is, until the sum of their squares s, is less than 1.
3. Let $l = \sqrt{-2\ln(s)/s}$
4. The two normal deviates are given by $x_1 l$ and $x_2 l$

The algorithm for the exponential deviate consists of an inversion of the exponential function, whereas the algorithm for the normal deviate consists of drawing two independent uniform deviates from the unit circle (polar algorithm). Although more efficient algorithms exist (e.g Marsaglia, 1964 for a random normal deviate), the two chosen prove satisfactory for our purposes.

2.2.4 Independence between migration and selfing

The two evolutionary forces to be dealt with in this research are migration and selfing. Migration is chosen to be gametic rather than zygotic, primarily because the early population genetic models of population structure were based on gametic migration. Gametic migration is found very often in nature, either in plants (via pollen dispersal) or in animals (via exchanges of one sex between herds as in monkeys (Chesser,1991) and whales (Amos, Barrett and Dover, 1991)).

Gametic migration means that dispersal takes place prior to mating. Under this condition, selfed individuals could only be non-migrants, a feature that would introduce dependence between selfing proportion and migration proportion in the model.

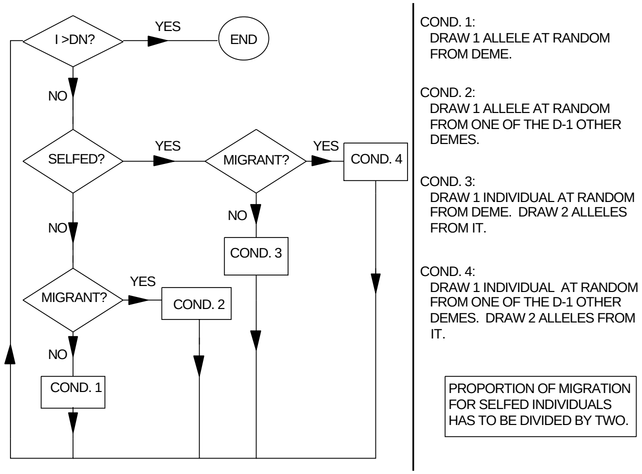
I >DN?
YES
END
NO
SELFED?
YES
MIGRANT?
YES
COND. 4
NO
NO
COND. 3
MIGRANT?
YES
COND. 2
NO
COND. 1
COND. 1:
DRAW 1 ALLELE AT RANDOM FROM DEME.
COND. 2:
DRAW 1 ALLELE AT RANDOM FROM ONE OF THE D-1 OTHER DEMES.
COND. 3:
DRAW 1 INDIVIDUAL AT RANDOM FROM DEME. DRAW 2 ALLELES FROM IT.
COND. 4:
DRAW 1 INDIVIDUAL AT RANDOM FROM ONE OF THE D-1 OTHER DEMES. DRAW 2 ALLELES FROM IT.
PROPORTION OF MIGRATION FOR SELFED INDIVIDUALS HAS TO BE DIVIDED BY TWO.

Figure 2.10: Algorithm for independence between migration and selfing

The approach is therefore to implement a mixed mating model, in which reproduction occurs first if the individual is selfed, with a probability of migration of $m/2$ (because 2 alleles instead of 1 will migrate), and dispersal occurs first if the individual is not selfed. Figure 2.10 summarises the algorithm.

2.3 Implementation of the models

The goal of this section is to describe a computer program, **MODEL42**, that was developed during this research. This program integrates the different gene-flow patterns described in the introduction and more in a single package (option 1 from the main menu). Once the gene-flow patterns have been built and saved into a file, a number of further options exist:

- Estimation of the average dispersal distance between parents and offspring. This option is useful for checking that the neighbourhood size is as intended. It will then calculate the one-way dispersal variance and give the distribution of the different distances of dispersal. More details about this option are given in the section on isolation by distance.

- Building of generations following the procedure described in the previous section. A graphical output of the number of alleles left in the population is also given.
- Graph of the distribution of allelic frequencies per generation. Up to 5 different generations can be pictured on the screen, the x-axis representing the frequency class and the y-axis the number of alleles in each class. This option could be used in conjunction with the Ewens-Watterson test for neutrality (Ewens, 1972; Watterson, 1978) to test for the effect of subdivision and geographical structuring.
- Reduction of the number of alleles (identity by descent → identity in state), using the procedure described in the previous section.
- Sampling at random the modelled population.
- Estimation of Wright's F-statistics on the samples or the total population.
- Visualisation of the genotypic composition of the population after the number of alleles has been reduced to 2 (with more alleles, the number of colours necessary to distinguish between genotypes becomes too large, as the number of genotypes for k alleles is $k(k+1)/2$).

As the name of the program suggests, it intends to do almost anything that can be done in the framework of neutral models in population genetics. Hopefully, the answers to the questions will not be as obscure as that given by the computer in *The Hitchhiker's Guide to the Galaxy*, but **MODEL42** lacks flexibility. It is therefore a useful tool for demonstration or teaching purposes, but it has to be 'disintegrated' for research purposes. In particular, the sampling procedure and the estimation of F-statistics are better used as standalone programs.

2.3.1 Constants and variables required to construct the first generation.

The first item that needs defining is the size of the total population. It will be a constant over the whole program and is called `MaxInd` for maximum number of individuals. The equivalent for the number of alleles is `DMaxInd`, twice the number of individuals. `MaxInd` has to be a power of 4. In most simulations it will be 4096, 4^6. Individual genotypes are stored in a two-dimensional array of size `MaxInd*2` called a field of genotypes. Two Boolean arrays also need defining, one for the migrants and one for the selfed individuals. These arrays will be one-dimensional and of size `DMaxInd` and `MaxInd` respectively. Population size then needs to be entered and is called `PopSize`. `MaxInd` divided by `PopSize` is the number of subpopulations `NumbSp`. `PopSize` is a power of 4, with its exponent between 0 and 6. Other parameters that need to be entered are the migration proportion `MigProp` and the selfing proportion `SelfProp`, both real numbers between 0 and 1. Once all these parameters have been entered, the field of genotypes for generation 0 needs to be initialised according to Figure 2.3. Initialisation of the two Boolean arrays for migration and selfing is then achieved by comparing the outcome of a draw in the random number generator to the input value of either migration or selfing. If the random number is less than `MigProp` or `SelfProp`, the corresponding Boolean value in the array is set to true, otherwise it is set to false. The following steps depend on the gene-flow patterns.

2.3.2 The island model

Two different forms of the island model of populations (Figure 2.11) can be modelled. The first is the infinite-size-continent island model, the second the gametic-cloud island model. The difference between the two lies in the migration pattern: in the infinite-size-continent migrants come from all the islands, including the recipient, whereas in the gametic-cloud, migrants come from all islands but the recipient. Both are finite island models because the number of islands is finite.

These two types of island models are compared in Takahata & Nei (1984): the migration proportion in the infinite-sized-continent island model is related to the gametic cloud as follows:

$$m_{isc} = \frac{D}{D-1} m_{gc}$$

where D is the number of demes, m_{isc} is the migration proportion in the infinite-sized-continent and m_{gc} is the migration proportion in the gametic cloud. We can see readily that, with a large number of demes, these two proportions will be essentially the same.

In terms of programming, the difference between the two models is that in the infinite-sized-continent, a random number between 1 and `MaxInd` is drawn if the individual is a migrant, but we repeat the procedure of drawing a random number until it does not belong to the original deme in the gametic-cloud island model. For non-migrants, a random number between 1 and `PopSize` is drawn and if i is the identifier

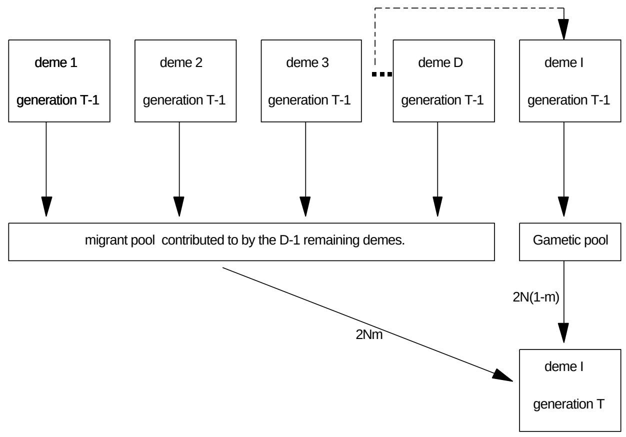

Figure 2.11: The gametic-cloud finite island model of population structure

of the deme, we add to this random number $(i-1)*$`PopSize`. The Pascal code for the gametic-cloud island model is given below:

```
x:=0;
for i:=1 to NumbSp Do
For k:=1 to PopSize Do
Begin
     x:=x+1;
     Temp1:=Grandom(PopSize)+1;
     Temp2:=Grandom(PopSize)+1;
     If Not Sel^[x]
     Then begin
               If Mig^[x]
               Then Begin
                         Repeat
                               Where1:=Grandom(MaxInd)+1
                         Until ((Where1<=(i-1)*PopSize)
                             or (Where1>i*Popsize));
                    End
               Else Where1:=(i-1)*PopSize+Temp1;
               If Mig^[x+MaxInd]
               Then Begin
                         Repeat
                               Where2:=Grandom(MaxInd)+1
                         Until ((Where2<=(i-1)*PopSize)
                             or (Where2>i*Popsize));
                    End
               Else Where2:=(i-1)*PopSize+Temp2;
```

```
        End
    Else Begin
            If Mig^[x]
            Then Begin
                    Repeat
                         Where1:=Grandom(MaxInd)+1
                    Until ((Where1<=(i-1)*PopSize)
                       or (Where1>i*Popsize));
                End
            Else Where1:=(i-1)*PopSize+Temp1;
            Where2:=Where1;
        End;
    Champ2^[x,1]:=ParChamp2^[Where1,Grandom(2)+1];
    Champ2^[x,2]:=ParChamp2^[Where2,Grandom(2)+1];
end;
```

in which GRandom is the random number generator, Mig^ and Sel^ are the Boolean arrays of migrants and selfers, ParChamp^ is the field of genotypes at generation 0 and Champ^ is the field of genotypes at generation 1. The complete code for these two procedures can be found in appendices A.4 and A.5.

The infinite-size-continent island model is intended to mimic constant allelic frequencies over generations in the migrant pool. However, this is in conflict with the fast procedure to build generations: as each allele in the offspring field is determined by a random location in the parent field and allele frequencies fluctuate over time, the allele frequencies in the migrant pool will also fluctuate. A way to implement the infinite-sized-continent island model would be to replace migration by mutation: each generation, a proportion PropMig of the DMaxInd alleles mutates (migrates) to one of the DMaxInd possible allelic states. This will ensure constancy of allelic frequencies in the migrant pool.

2.3.3 The stepping-stone model

As we have seen in the introduction, migration occurs only between adjacent demes in the true stepping-stone model. A graphical representation of a one-dimensional stepping-stone is given in Figure 2.12 and of a two-dimensional stepping-stone in Figure 2.13. The initialisation procedure, as well as filling the Boolean arrays for migration and selfing, is done as for the island model. The difference lies in the provenance of migrants. In a 2-dimensional stepping-stone model, we need to lay the field of genotypes on a 2-dimensional surface. This is done by specifying a number of rows and columns as a function of NumbSp:

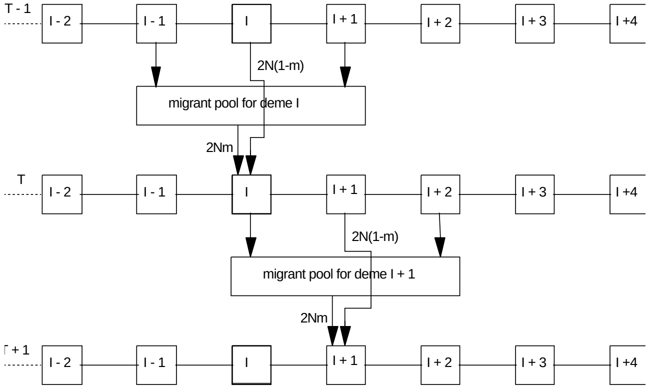

Figure 2.12: 1-dimensional stepping-stone model

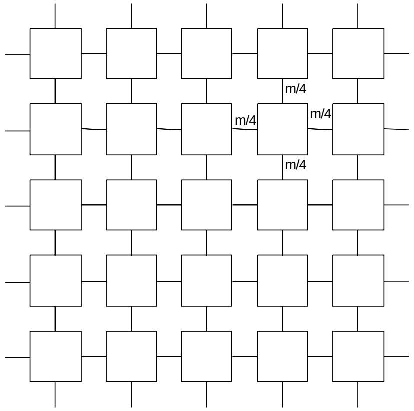

Figure 2.13: 2-dimensional stepping-stone model

```
    Case NumbSp of
             4096 :begin numbrow:=64;numbcol:=64;end;
             1024 :begin numbrow:=32;numbcol:=32;end;
             256  :begin NumbRow:=16;NumbCol:=16;end;
             64   :begin NumbRow:=8;NumbCol:=8;end;
             16   :begin NumbRow:=4;NumbCol:=4;end;
             4    :begin NumbRow:=2;NumbCol:=2;end;
             1    :begin NumbRow:=1;NumbCol:=1;end;
    end;
```

The i loop in the island model needs to be replaced by two nested loops, corresponding to the number of rows and columns respectively. If the gamete is a migrant, we call a procedure that randomly picks one of the 4 possible provenances:

```
Procedure GetOfset(Var OfsVer,OfsHor:Shortint);
var
temp, dist:byte;
begin
     OfsHor:=0;
     OfsVer:=0;
     temp:=GRandom(4);
     case temp of
          0 : OfsHor:=-1;
          1 : OfsHor:=1;
          2 : OfsVer:=-1;
          3 : OfsVer:=1;
     end;
end; {Of Proc GetOfs}
```

However, a problem arises if the deme under consideration is on one of the field sides and the offset causes the migrant to come from outside the field. One solution would be to make the 2-dimensional surface a torus, so that there are no edges. Alternatively, one could decide that if the migrant is coming from outside the field, it is not a migrant, which will reduce the migration proportion for demes on the edges. An option in **MODEL42** lets us choose between these two options and assign the value true or false to the Boolean `Tor`. The function `GetNewCoord` then returns the appropriate horizontal and vertical coordinates:

```
Function GetNewCoord(Tor:boolean;a,Ni:integer;Ofsa:ShortInt):integer;
var Res : integer;
Begin
     If Tor
     Then Res:=(a+Ni-1+Ofsa) mod Ni
     Else Begin
               If ((a+Ofsa)<1)
               Then Res:=GRandom(a)
               Else If ((a+Ofsa)>Ni)
                    Then Res:=a+Grandom(Ni-a)-1
                    Else Res:=a+Ofsa-1;
          end;
     GetNewCoord:=Res;
End; {Of Function GetNewCoord}
```

where `a` is the row or column identifier of the recipient deme, `Ni` is the number of rows or columns and `Ofsa` is the offset obtained from the previous procedure. The complete code for this procedure is found in appendix A.6.

Migration need not to be restricted to the nearest deme. Indeed, it is more realistic to consider that the distribution of migrants is some decreasing function of distance such as a negative exponential. Then, for a given proportion of migration, the largest proportion will come from the nearest neighbour, the next largest from the second nearest and so on so forth. The shape of the distribution can be altered by use of the mean for the negative exponential. To allow migration to be a decreasing function of distance, two extra parameters are required: the average of the negative exponential `Aver` and the maximum distance of dispersal `Dist`. If we want to model a true stepping-stone model, it is sufficient to input a large average dispersal distance and to set `Dist` to 1. **MODEL42** implements a 1-dimensional stepping-stone with a negative exponential distribution of migrants, as well as a 3-dimensional stepping-stone (migrants can come from 8 directions) with either a half-normal distribution or a negative exponential. The procedure for 3-dimensions, while slightly more complicated, uses the same logic as in two-dimensions. Pascal code for these procedures can be found in appendix A.7,A.8 and A.9.

2.3.4 The isolation by distance model (IBD)

In the models of population structure considered so far, individuals have been packaged in discrete structures called demes, within which mating occurred at random apart from a defined proportion `SelfProp` of selfing. It is likely, however, that individuals are distributed in a continuum, with a dispersal of gametes following some decreasing function of distance. This is the isolation by distance, or neighbourhood model in which the parameters to be specified are the male and female standard deviations of the dispersal distance, the distribution in use being the normal distribution. The field of parents at generation 0 is initialised in the same way as for the other geneflow patterns. Different variants are then considered: a 'true Wright' Neighbourhood model, in which coordinates of the male and female gametes are picked from the location of the offspring, or a plant neighbourhood model, where the coordinates of the female gamete is picked from the location of the offspring and the coordinate of the male gamete is picked from the location of the female gamete. Selfing could be random, that is, a function of the dispersal distance, or fixed. If it is fixed, a female gamete is picked at random and the male gamete is drawn from the same location if a random number is less than the proportion of fixed selfing. The continuum can be on a toroidal or a flat surface. To avoid the biological black-hole phenomenon (Felsenstein, 1975), individuals are located at a fixed position on the intersection of a grid. Clumping can also be avoided with density dependence: the denser the surrounding, the less likely it is that a seed can germinate. Obviously, fixing individuals on the intersection of a grid is a form of density dependence, but this limits the number of parameters required by the model. The Pascal code for this gene-flow pattern is found in appendix A.10.

2.3.5 The pseudo-neighbourhood model

Rather than having one single individual at the intersection of a grid, we could have a deme. This model, intermediate between the 3-dimensional stepping-stone model with migration following a decreasing function of distance and the IBD model, has been named a pseudo-neighbourhood model. It is equivalent to the addition of an extra parameter, density, to the IBD model: the larger the deme, the denser the population. As with the IBD model, migration is not a parameter, but is deduced from the underlying dispersal distribution of gametes: with a negative exponential distribution of dispersal distance $f(x) = \lambda \exp(-\lambda x)$, the proportion of gametes migrating is $1 - \exp(-\lambda)$.

This gene-flow pattern can be implemented through sub-option 6 of option 1 in **MODEL42**. The Pascal code can be found in appendix A.11.

2.4 Analytic solutions

2.4.1 The Island model of a population

Consider an infinite set of finite sized islands each composed of N diploid monoecious individuals. Individuals within each of these islands breed at random, apart from a proportion m of migrants drawn at random from the whole (Wright, 1943). The

Table 2.1: Genotype frequencies in subdivided populations

	Genotypes		
	AA	Aa	aa
population 1	p_1^2	$2p_1q_1$	q_1^2
population 2	p_2^2	$2p_2q_2$	q_2^2
$\vdots$	$\vdots$	$\vdots$	$\vdots$
population i	p_i^2	$2p_iq_i$	q_i^2
$\vdots$	$\vdots$	$\vdots$	$\vdots$
population k	p_k^2	$2p_kq_k$	q_k^2
pooled	$\frac{\sum_{i=1}^k p_i^2}{k}$	$\frac{\sum_{i=1}^k 2p_iq_i}{k}$	$\frac{\sum_{i=1}^k q_i^2}{k}$
letting $\overline{p} = \frac{\sum_{i=1}^k p_i}{k}$ and adding the variable F leads to:			
pooled	$\overline{p}^2(1-F) + \overline{p}F$	$2\overline{p}\,\overline{q}(1-F)$	$\overline{q}(1-F) + \overline{q}F$

number of islands being infinite, the overall allele frequency is constant, generation after generation, as is the allele frequency in the migrant pool. On the other hand, the allele frequency in each island will be dictated by the opposing effects of genetic drift and migration.

The overall effect of this structuring will be an alteration of the composition of the genotypic array leading to an apparent deficit of heterozygotes in the whole population (Walhund, 1928): consider k isolated populations of diploid, monoecious individuals. Focussing our attention on one locus with two allelic states a and A, it is possible to derive the expected frequencies of the three genotypes in each subpopulation (Table 2.1).

Now suppose that the k populations are grouped together and that individuals mate at random. One generation is sufficient to restore panmixia and therefore, the expected frequency of each genotype is $\overline{p}^2$, $2\overline{p}\,\overline{q}$ and $\overline{q}^2$ for AA, Aa and aa respectively. The ratio of the two heterozygote proportions before and after pooling is $(1 - F)$. If F is equal to zero, there is no difference in Aa genotype frequencies between the 2 generations, if F is positive, the population before pooling showed a deficit in heterozygotes. The expression for F can be derived from the above table by taking, for example, the frequency of AA genotypes:

$$\frac{\sum_{i=1}^{k} p_i^2}{k} = \overline{p}^2(1-F) + \overline{p}F \tag{2.2}$$

solving for F leads to:

$$F = \frac{\sigma^2}{\overline{p}\,\overline{q}} \tag{2.3}$$

F is therefore always positive and zero only when $p_1 = p_2 = \ldots = p_i = \ldots = p_k$. As soon as two or more isolated populations show unequal allele frequencies, the populations considered as a whole will present a deficit in heterozygotes.

Measures of gene diversity often found in the literature are f_0 and f_1, the identity by descent within and among populations respectively (Nei, 1973; Felsenstein, 1976; Slatkin, 1985a; Slatkin 1993). The overall identity by descent, $\overline{f}$ is then defined as $\frac{1}{D}f_0 + (1-\frac{1}{D})f_1$, where D is the number of demes in the population. If there is random mating within populations, the expression for f_0 is simply:

$$\sum_{j=1}^{D}\sum_{i=1}^{k} p_{ji}^2 = \overline{p^2} \tag{2.4}$$

whereas the expression for $\overline{f}$ is:

$$\sum_{i=1}^{k}(\sum_{j=1}^{D} p_{ji})^2 = \overline{p}^2 \tag{2.5}$$

combining these results in (2.3) leads to:

$$F = \frac{f_0 - \overline{f}}{1 - \overline{f}}$$

Structuring leads to heterozygote deficit, but in a different way from selfing. To quantify the extent of these two deficits, the F-statistics described in Chapter 3 are often used. Focussing first on the within-population deficit due to selfing, the expected value of F_{is} at generation t can be expressed as a function of the proportion of selfing and the value of F_{is} at generation $(t-1)$ as follow:

$$F_{is_t} = \frac{s}{2} + \frac{s}{2}F_{is_{t-1}} + (1-s)0 \tag{2.6}$$

that is, a proportion $s/2$ of individuals will carry two alleles descending from the same allele of the preceding generation, a proportion $s/2$ will carry two alleles descending from different alleles of the preceding generation but which where copies of a single allele in a previous generation and a proportion $(1-s)$ will remain in random mating proportions (one generation of random mating restores Hardy-Weinberg equilibrium). Equation 2.6 can be rewritten:

$$F_{is_t} = \frac{s}{2}(1 + F_{is_{t-1}}) \tag{2.7}$$

at equilibrium, $F_{is_t} = F_{is_{t-1}}$ and therefore

$$\hat{F_{is}} = \frac{s}{2-s} \tag{2.8}$$

(Crow & Kimura, 1970).

One important feature of the above equations is that they are independent of population size, providing that $\hat{F_{is}}$ is unbiased. Therefore, selfing affects the breeding structure of the population at the genotypic level rather than the allelic level.

I shall turn now to the between-population heterozygote deficit which is due to genetic drift. This can be expressed as a function of the variance effective size of the sub-population, N_e (see next section), the migration proportion, m and the heterozygote deficit of the preceding generation, F_{st}, as follows:

$$F_{st_t} = (1-m)^2\left(\frac{1}{2N_e} + (1 - \frac{1}{2N_e})F_{st_{t-1}}\right) \tag{2.9}$$

(Wright, 1943). Contributions to F_{st} come only from non-migrants. A proportion $1/2N_e$ of individuals will carry two alleles descending from the same ancestral allele of the preceding generation, whereas a proportion $(2N_e - 1)/2N_e$ will be descended from different alleles of the preceding generation, but which are copies of a single allele in a previous generation in proportion $F_{st_{t-1}}$. An interesting feature of this last expression appears for $N_e = 1$. Substituting 1 for N_e in (2.9) leads to:

$$\begin{aligned} F_{st_t} &= (1-m)^2\left(\tfrac{1}{2} + (1-\tfrac{1}{2})F_{st_{t-1}}\right) \\ &= \tfrac{(1-m)^2}{2}(1 + F_{st_{t-1}}) \end{aligned}$$

In comparing this with (2.7), it can be seen that $(1-m)^2 \equiv s$, at least formally. Indeed, $(1-m)^2$ is the proportion of gametes ‘staying’ in the population and if the population contains only one individual, this is the proportion of selfing.

At equilibrium, $F_{st_t} = F_{st_{t-1}}$, which leads to:

$$\hat{F_{st}} = \frac{(1-m)^2}{2N_e - (2N_e - 1)(1-m)^2} \tag{2.10}$$

(Wright, 1943; Crow & Kimura, 1970; Hartl & Clark, 1989). If terms involving m, m^2 and $N_e m^2$ are considered to be small, (2.10) reduces to:

$$\hat{F_{st}} = \frac{1}{4N_e m + 1} \tag{2.11}$$

The above approximation was made by Wright (1951) at a time when computers were not commonplace. However, the simplicity of the expression of this approximation made it very popular and it has been widely adopted by population geneticists.

The concept of population effective size, N_e

The notion of effective population size traces back to Wright (1931). This is a very useful concept for comparative purposes and a practical necessity when dealing with natural populations (Wright, 1969, p 211). Indeed, when one wishes to compare two populations, it is necessary to define an ‘idealised’system, in which both populations could be compared. This system is defined as follows (Hartl and Clark, 1989, p64):

1. diploid organism
2. sexual reproduction
3. non-overlapping generations
4. many independent sub-populations, each of constant size N
5. random mating within each sub-population
6. no migration between sub-populations
7. no mutation
8. no selection.

Any departure from these very restrictive hypotheses will lead to different expectations for the rate of changes in homozygosity and/or the rate of allele frequency drift. The effective size of a population is then defined as the size of an idealised population that would show the same changes in homozygosity (inbreeding effective size, N_e^i), or the same changes in allele frequencies (variance effective size, N_e^v) as the population under investigation. A third effective size has been defined by Haldane (1939) and Ewens (1979), the eigenvalue effective size (N_e^e), where N_e^e is defined as a function of the largest non-unit eigenvalue of the transition matrix of the Wright-Fisher model (Crow and Denniston, 1988).

The three effective sizes are equivalent most of the time (Crow and Kimura, 1970, carry out an extensive comparative analysis of N_e^i and N_e^v and show that most of the time they lead to the same estimate of N_e). The inbreeding effective size of a population is defined as whatever must be substituted for N in the following formula:

$$1 - F_t = (1 - \frac{1}{2N})(1 - F_{t-1}) \tag{2.12}$$

where F_t is the rate of change in homozygosity of the population at time t. However, this formula leads to an indetermination (0/0) when the rate of change of homozygosity reaches 0 ($F = 1$), when the population is at equilibrium. This equation could be rewritten in term of heterozygosity:

$$H_t = \frac{1}{2N} H_{t-1}$$

with the same problem ($H = 0$ at equilibrium with no mutation nor migration).

A more useful formula, that of the variance effective size of a population, is defined as whatever must be substituted for N in the following formula:

$$\sigma^2_{\Delta p} = \frac{p(1-p)}{2N} \tag{2.13}$$

where $\sigma^2_{\Delta p}$ is the sampling variance of gametes over generations of the population (Wright, 1969, p 211). Replacing N by N_e^v in the previous equation and rearranging leads to:

$$N_e^v = \frac{\sigma^2_{\Delta p}}{2p(1-p)} \tag{2.14}$$

This result is valid for a one generation time interval. For more than one generation, multiplying the right hand side of this last equation by the time interval in generations, t, has been suggested (Nei & Tajima 1981; Pollak, 1983; Waples, 1989). Waples (1989) also defined parameters, F_c and F_k, that lead to appropriate (unbiased) measurements of N_e^v for two sampling schemes, corresponding to sampling before and after reproduction. The estimate of F_c given by Waples (1989) is:

$$\hat{F}_c = \frac{1}{K} \sum_{i=1}^{K} \frac{(x_i - y_i)^2}{(x_i + y_i)/2 - x_i y_i}$$

where x_i and y_i represents estimates of the frequency of allele i in the two generations and K the number of segregating alleles. It has been found that a better estimate of F_c is:

$$\hat{F_{cw}} = \frac{\sum_{i=1}^{K} (x_i - y_i)^2}{\sum_{i=1}^{K} ((x_i + y_i)/2 - x_i y_i)}$$

where F_{cw} stands for weighted F_c. When all the population is sampled, an estimate of N_e^v is:

$$N_e^v = \frac{t}{2\hat{F_{cw}}} \tag{2.15}$$

because the sampling correction in both generations cancels the covariance term (c.f. Waples, 1989, equation 12, p 382). Computer simulations showed that this last formula is in very close agreement with (2.14), whereas the non-weighted F_c often leads to negative estimate (Infinite effective size), as mentioned by Waples (1989).

Expression for N_e under specific breeding systems can now be sought.

The sampling variance in an inbred population through selfing could be expressed as follows:

$$\sigma^2_{\Delta p} = \frac{p(1-p)}{2N}(1-F) + \frac{p(1-p)}{N}F \tag{2.16}$$

That is, the inbred population is divided in two components, one non inbred, with a sampling variance of $p(1-p)/2N$ and the other, inbred, with a sampling variance of $p(1-p)/N$. Equation 2.13 reduces to:

$$\sigma^2_{\Delta p} = \frac{p(1-p)(1+F)}{2N} \tag{2.17}$$

We can now use these two definitions to derive the variance effective size of a partially self-fertilising population. The same result could be obtained from a inbreeding effective size perspective (e.g Pollak, 1987, 1988). Substituting (2.17) into (2.13) leads to:

$$N_e^v = N_e^i = \frac{N}{1+F} \tag{2.18}$$

(Wright, 1943; Li, 1955, p323) where F is an F_{is}, the within-population heterozygote deficit. Under a 100% self-fertilisation regime, F_{is} will be equal to 1 and the effective population size will be half the real population size, as expected.

Inbreeding could be due to mating between relatives rather than selfing. This problem has been researched by Pollak (1987,1988) and Caballero & Hill (1992). They arrived at the following results:

If partial full-sib mating occurs:

$$N_e^v = N_e^i = \frac{2D}{1+3F} \tag{2.19}$$

where D is the number of families.

If partial half-sib mating occurs:

$$N_e^v = N_e^i = \frac{4D}{1+7F} \tag{2.20}$$

The preceding results could be generalised to:

$$N_e^v = N_e^i = \frac{DN}{1+(2N-1)F} \tag{2.21}$$

where N is the family size and D the number of families. This result was given in a slightly different form in Pollak (1988). That is, if groups are considered as families (this is not true for the first few generations in an island model where individuals at generation 0 are unrelated, as in the model described in the preceding section, but this becomes true as time goes on), the effective size of a population made of D families of size N is given by (2.21).

The effective size of a subdivided population can also be derived, following Wright (1943) and Li (1955). If a population is subdivided into D breeding groups of equal size N, the sampling variance in each group is $p_i(1-p_i)/2N$, where p_i is the allele frequency in group i. The average value of $\sigma^2_{\Delta p}$ over the D groups is:

$$\sigma^2_{\Delta p} = \frac{1}{D}\sum_{i=1}^{D}\frac{p_i(1-p_i)}{2N} = \frac{2\sum_{i=1}^{D} p_i(1-p_i)}{4DN} \tag{2.22}$$

Now, $2\sum_{i=1}^{D} p_i(1-p_i)/D$ is the proportion of heterozygotes in the total population (providing that there is random mating within each subpopulation) and is thus equal to $2\overline{p}\,\overline{q}(1-F)$, where F is F_{st}. Substituting into the last equation leads to:

$$\sigma^2_{\Delta p} = \frac{\overline{p}\,\overline{q}(1-F)}{2N} \tag{2.23}$$

substituting (2.23) into (2.13) and remembering that the population is made of DN individuals leads to:

$$N_e^v = \frac{DN}{1-F} \tag{2.24}$$

This last formula, however, does not hold true from an inbreeding effective size perspective. The inbreeding effective size of a subdivided population is given by (2.21).

Combination of these results in a single formula seems to be a daunting task: which one is appropriate to specific cases, how to combine them to obtain the effective size of a population undergoing inbreeding through selfing, together with mating with relatives and subdivisions, with the added complication that the number of successful gametes may not have a Poisson (binomial) distribution and the sex ratio may be different from 1 : 1.

Suggestions on how to deal with the last points (Poisson distribution of successful gametes and sex-ratio) are made in Pollak (1987, 1988). N in the previous expressions should be replaced with N', where N' is defined as the reciprocal of the probability that two gametes contributing to random separate adults come from the same parent. Namely, in the case of unequal sex ratio, N should be replaced by $4N_mN_f/(N_m+N_f)$ where N_m and N_f are respectively the number of males and females in the population. In the case of non-Poisson distribution of the number of successful gametes, N becomes $\frac{4N-2}{\sigma_k^2+2}$ where σ_k^2 is the variance of the number of successful gametes (e.g. Li, 1955, p 322).

The effective size of a selfed, subdivided population can now be considered. The effective size of each group is given by (2.18) and the global effective size is given by (2.24). Combining the two leads to:

$$N_{eg} = \frac{D\frac{N}{1+F_{is}}}{1-F_{st}} \tag{2.25}$$

which can be rearranged into:

$$N_{eg} = \frac{DN}{(1+F_{is})(1-F_{st})} \tag{2.26}$$

This formula is valid if (i) all the different levels of structuring have been properly identified, (ii) the effect of mutation is considered negligible and (iii) there is territoriality. Such a situation may occur in conservation, where all the members of the species are sampled and the time scale does not exceed a few generations, therefore allowing mutation to be neglected.

On the other hand, if only a small range of the species has been surveyed, or mutation is considered important, or the aim of the study is to compare two (subdivided) populations of the same species at different locations, or there is no correlation between the parent and offspring spatial location (no territoriality), or we are interested in inbreeding effective size rather than variance effective size to quantify the effect of inbreeding depression, (2.21) should be used instead of (2.24) and we obtain:

$$N_{epg} = \frac{D\frac{N}{1+F_{is}}}{1+(2\frac{N}{1+F_{is}}-1)F_{st}} \tag{2.27}$$

which rearranges to:

$$N_{epg} = \frac{DN}{2NF_{st}+(1+F_{is})(1-F_{st})} \tag{2.28}$$

One can readily see that levels can be added, providing that conditions of validity are met and that each new effect (sex-ratio, unequal number of successful gametes, selfing, inbreeding, subdivisions) is incorporated at the appropriate level. The combined effect of unequal number of successful gametes with inbreeding due to mating with relatives is dealt with in Caballero & Hill (1992). More on the topic of effective size can be found in Chapter 3.

Mixed breeding patterns

An application of the results of the preceding section permits the derivation of the equilibrium for $\hat{F_{st}}$ in an infinite island model, when there is partial selfing. Equation 2.10 can be rewritten:

$$\hat{F_{st}} = \frac{(1+\hat{F_{is}})(1-m)^2}{2N-(2N-1-\hat{F_{is}})(1-m)^2} \tag{2.29}$$

combining with (2.8) leads to:

$$\hat{F_{st}} = \frac{(1-m)^2}{N(2-s)-(N(2-s)-1)(1-m)^2} \tag{2.30}$$

that is, the magnitude of drift changes from $1/2N$ to $1/2N(1-s)$.

Biological inference

Now the equilibrium values of both F_{is} and F_{st} have been established, the problem can be reversed: given F_{is} and F_{st}, is it possible to infer what the proportion of selfing and migration in the population under investigation are? If the assumptions of the infinite island model stand, it is sufficient to reverse the results of equation 2.8 and equation 2.10, which leads to:

$$\hat{s} = \frac{2\hat{F_{is}}}{1+\hat{F_{is}}} \tag{2.31}$$

and

$$\hat{m} = 1-\sqrt{\frac{2N_e\hat{F_{st}}}{(2N_e-1)\hat{F_{st}}+1}} \tag{2.32}$$

The utility of this equation will be discussed in chapter 4, but it should be noted that N_e refers to the effective sample size. The effective population size follows only if the whole population is sampled, or if experiments, such as mark-release-recapture, lead to estimates of the census local population size. If (2.11) is used instead of (2.10), one can extract the product N_em:

$$\widehat{N_em} = \frac{1-\hat{F_{st}}}{4\hat{F_{st}}} \tag{2.33}$$

(e.g. Slatkin, 1985a). That is, the effective number of migrants per local population can be extracted. Providing that the conditions leading to equation 2.11 are met and that the estimate of F_{st} is independent of both sample size and number of demes sampled, this expected number of migrants will estimate the actual number of migrants, regardless of the sampling strategy.

Non-equilibrium situation

It may be of interest to predict values of F_{st} and F_{is} in situations where equilibrium is not reached, either because the process has not been going on for long enough, or because a disturbance has modified the conditions. It has been shown that it is only necessary to derive equations for F_{st}, solutions for F_{is} can be readily found by replacing N by 1 and $(1-m)^2$ by s. Consider equation 2.9, it is possible to express F_{st} as a function of time and F_0:

$$\begin{aligned} F_1 &= A + BF_0 \\ F_2 &= A + BF_1 = A + AB + B^2 F_0 \\ &\vdots \\ F_t &= B^t F_0 + A \overbrace{\sum_{i=1}^{t} B^{t-1}} \end{aligned} \tag{2.34}$$

where $A = \frac{(1-m)^2}{2N}$ and $B = (1-m)^2(1-\frac{1}{2N})$ ($A = B = s/2$ for F_{is}). The over-braced sum in the last equation can be rewritten:

$$\sum_{i=1}^{t} B^{t-1} = \frac{B^t - 1}{B - 1} \tag{2.35}$$

leading to:

$$F_t = \overbrace{B^t(F_0 - \frac{A}{1-B})} + \frac{A}{1-B} \tag{2.36}$$

It is worth noticing that $\frac{A}{1-B}$ is the equilibrium value of F_{st} (F_{is}). The over-braced part of last equation tends to 0 as t tends to ∞, because B is less than 1. The larger B is, the longer it will take for F_{st} to reach its equilibrium value. For B to be large, N needs to be large and m small, conditions necessary to apply equation 2.11 for the estimation of the product $N_e m_e$. This means that the approximation will only be useful in cases where the equilibrium value takes a long time to be reached and is, therefore, unlikely to ever be attained. Time to equilibrium can be assessed with the following treatment: consider the time t it will take for F_{st} to reach x% of its equilibrium value:

$$\begin{aligned} B^t(F_0 - \frac{A}{1-B}) + \frac{A}{1-B} &= x\frac{A}{(1-B)} \\ B^t &= \frac{(x-1)A}{F_0(1-B) - A} \\ t &= \frac{\ln\left(\frac{(x-1)A}{F_0(1-B)-A}\right)}{\ln(B)} \end{aligned} \tag{2.37}$$

If F_0 is 0, this last expression reduces to:

$$t = \frac{\ln(1-x)}{\ln(B)} \tag{2.38}$$

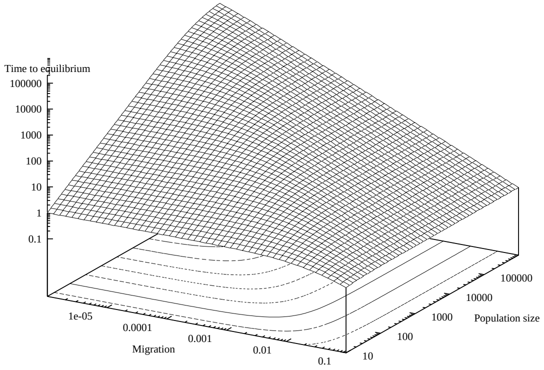

Figure 2.14: Number of generations before F_{st} reaches 95% of its equilibrium value

As x tends to 1^-, the numerator tends to $-\infty$. If both $m \ll 1$ and $N \gg 1$, the denominator will be close to 0^- and the population will take a very long time to reach an equilibrium. For F_{is}, the cases of interest are for large s and we therefore see that equilibrium will be reached very quickly.

Figure 2.14 shows the time it takes for F_{st} to reach 95% of its equilibrium under different combinations of migration and local population sizes, F_0 being set to 0. We can see that what determines time to equilibrium is the greater of m or $1/N$. The larger they are, the faster equilibrium is reached. As few as 20 generations are sufficient for equilibrium to be reached if m is close to 0.1 or N is close to 10.

2.4.2 The stepping-stone model of population

The island model of population we have just been investigating is the simplest among the models dealing with subdivision of populations because it does not have any geographical structure. The stepping-stone model, introduced by Kimura (1955) is a half way house between the very realistic but intractable isolation by distance, or neighbourhood model (Felsenstein, 1975) and the island model of populations. Solutions for the island model are straightforward because of the equal relationship between each deme. As soon as geographical structure is added to the model, the mathematical treatment has to be different. In particular, the correlation of gametes belonging to different sub-populations has to be expressed as some function of the distance between these sub-populations.

Approximate solutions for the correlation of gene frequencies of populations k steps apart are given in Kimura and Weiss (1964) and Weiss and Kimura (1965) for the one- two- and three-dimensional stepping-stone. For the infinite one-, two- and three-dimensional stepping-stone, the correlation between populations k steps apart is:

$$r(k) = \exp\left(-\sqrt{\frac{2m_\infty}{m_1}}\,k\right) \tag{2.39}$$

$$r(\rho) = \frac{\exp\left(-\sqrt{\frac{4m_\infty}{m_1}}\,\rho\right)}{\sqrt{\rho}} \tag{2.40}$$

$$r(\rho) = \frac{1}{\pi}\frac{\exp\left(-\sqrt{\frac{6m_\infty}{m_1}}\,\rho\right)}{\rho} \tag{2.41}$$

(Kimura & Weiss, 1964) where m_1 is the short range migration and m_∞ is the long range migration, ρ is defined as $\sqrt{k_1^2 + k_2^2}$ in two dimensions and $\sqrt{k_1^2 + k_2^2 + k_3^2}$ in three.

We can see that the correlation of gene frequencies falls off more rapidly in three dimensions than in two dimensions, which in turn falls off quicker than in a one-dimensional stepping stone model. Figure 2.15 displays the changes in the correlation as a function of distance, with $m_\infty = 10^{-6}$ and $m_1 = 0.1$. A peculiarity of the three-dimensional stepping-stone is that, even if the long range migration (or mutation) is 0, the correlation does not go to 1, but its maximum is $\frac{1}{\pi}$.

Subsequent work on stepping-stone models has been done by Marayuma (1970, 1971a,b,c, 1972a,b,c, 1974). An interesting finding is that a quantity akin to F_{st}, the ratio $(1 - f_k)/(1 - f_0)$,where f_k is the coefficient of inbreeding between genes drawn from individuals k colonies apart and f_0 is the coefficient of inbreeding between genes drawn from individuals in the same colony, tends to stabilise even thought the individual f_k's approach 1.

A computer simulation of stepping-stone models is described in Kimura and Marayuma (1971). They investigated a toroidal two-dimensional stepping-stone, and a circular one-dimensional stepping-stone. They found that if the product $N_e m$ is larger than 4, no local differentiation occurs and the whole population behaves as if panmictic. On the other hand, when $N_e m$ is less than 1, marked differentiation between random breeding units occurs. They also show that in a one-dimensional stepping-stone model, differentiation occurs for higher values of $N_e m$ than in the two-dimensional

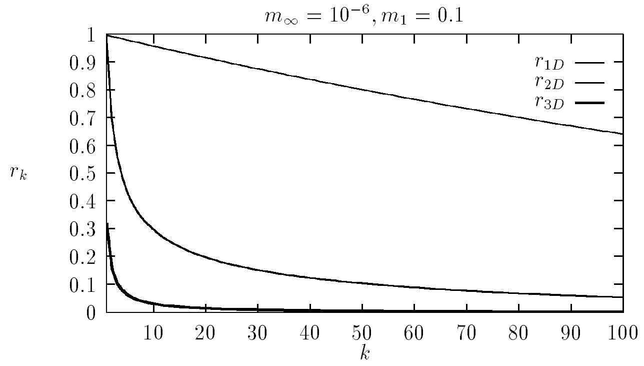

Figure 2.15: Correlation between populations k steps apart in 1-, 2- and 3-dimensions.

case, which is to be expected just by looking at the prediction of the correlation of gene frequencies with distance.

Another point stressed in this paper is that the appropriate measurement of genetic differentiation will be dependent on the level of mutation or, more accurately, on the product of the total population size and the mutation rate $N_t u$. If $N_t u$ is small, then $(1-f_0)/(1-\overline{f})$, where $\overline{f}$ is the average of the different f_k's, is the appropriate measure of genetic differentiation. However, if this product is large (larger than two) then the appropriate measurement is $f_0/\overline{f}$ because both f_0 and $\overline{f}$ will be small.

Crow and Aoki (1984) derived exact solutions for $G_{st} = \frac{f_0 - \overline{f}}{1 - \overline{f}}$ in an island-model under migration and mutation. They showed that in both an island and a stepping-stone model, the equilibrium value of G_{st} is independent of the mutation rate, but that G_{st} is linearly related to the logarithm of the number of demes in a stepping-stone. They also showed that the shape of the habitat (the connectedness) has a large influence on the equilibrium values of G_{st}, a result to be expected given the findings of Kimura & Weiss (1964) (Figure 2.15). Slatkin (1993) considered non-equilibrium situations, and showed that in some cases, structuring can be detected. However, no analytical treatment of the time to equilibrium of the different Fixation indices in a stepping-stone model seems to exist. As Felsenstein (1976) puts it:

> 'Wright's quantities are of great biological interest and hopefully future work will resume their use.'

2.4.3 Isolation by distance.

Malécot (1948) and Wright (1943) pioneered analytical work on the isolation by distance, or neighbourhood, model. As was pointed out by Felsenstein, Malécot's results are wrong because of an incompatibity between the assumptions of the model:

1. Random distribution of individuals
2. Poisson distribution of the number of offspring
3. Independence of migration among offspring

Felsenstein (1975) showed that assumption (1) is incompatible with assumptions (2) and (3): (2) & (3) lead to clumping of individuals.

Although his results are in general agreement with other models, Wright's isolation by distance model involves a complex set of assumptions, most of which are inexplicit (Felsenstein, 1976).

To compare isolation by distance with other discrete models, there is a need to define the equivalent of a random mating unit. This is the so-called neighbourhood of Wright, presented in the introduction of this chapter. The size of the neighbourhood will be dependent on the mating systems and the distribution of parent-offspring dispersal among others things. Formulae for this unit are given in Wright (1969) and a review paper by Crawford (1984). If individuals are distributed along a linear habitat and the parent-offspring dispersal distribution is normal,the neighbourhood length, N_L is defined as:

$$N_L = 2\sqrt{\pi}\sigma$$

where σ is the standard deviation of the parent-offspring dispersal distances.

In a two-dimensional habitat, with the dispersal distances following a bivariate, zero-mean, normal distribution with equal variances σ^2 along two orthogonal axes, the neighbourhood area, N_A, is defined as:

$$N_A = 4\pi\sigma^2$$

and is circular. As we are interested in the number of individuals within the neighbourhood area, it is sufficient to multiply the neighbourhood area with a density parameter, d (in **MODEL42** the density is kept constant at 1, making the number of individuals in the neighbourhood area equal to the neighbourhood area itself). Providing everything else is kept constant, the expression for σ in the above equations

will change as a function of the mating system. For a true Wright model, this is just the average of the male and female dispersal, $\frac{\sigma_m+\sigma_f}{2}$, but for a plant model, the neighbourhood area is (Crawford, 1984):

$$N_A^{Plant} = 4\pi(\frac{\sigma_m^2}{2} + \sigma_f^2)$$

If selfing occurs, the male dispersal variance needs multiplying by $(1-s)$. Extra-components can be added to take account, for example, of the effect of vegetative growth (Gliddon & Saleem, 1985).

When measuring these quantities in nature, data are usually projected on one axis to give the axial dispersal variance, which is half the absolute dispersal variance. This projection leads to a change in the underlying distribution of dispersal. If it is a circular bivariate normal, then its projection on one dimension gives a Rayleigh distribution (Parzen, 1960, p 320):

$$f(x) = \frac{x}{\sigma^2}\exp\left(\frac{-1}{2}(\frac{x}{\sigma})^2\right)$$

For bivariate distributions other than the normal, there is no such simple solution. Whittaker (pers. comm.) gives a solution for distributions of the form

$$y = \frac{K}{\sigma}\exp\left(\frac{-1}{2}(\frac{x}{\sigma})^{\frac{2}{1+\beta}}\right)$$

known as the exponential power family distribution, defined for $-1 < \beta \leq 1$ (this reduces to the bivariate normal if $\beta = 0$). The solution for the projection of these distributions on one axis takes the form:

$$f(x) = \frac{x}{\sigma^2 2^\beta \Gamma(2+\beta)}\exp\left(\frac{-1}{2}(\frac{x}{\sigma})^{\frac{2}{1+\beta}}\right)$$

and it can be readily seen that for $\beta = 0$, this expression reduces to the Rayleigh distribution.

Wright devoted thirty pages in volume 2 of his masterpiece (Wright, 1969, pp 295–324) to the expectation of F-statistics in a continuum. Consider an individual, I. Consider a circular area of radius r (r is equal to twice the standard deviation of the dispersal distance of parents to offspring, considered to be normal) centred on I, containing N *uniformly distributed, with a density of 1 per area unit,* individuals, all equally likely to be the parents of individual I. The area is $\pi r^2 = 4\pi\sigma^2$. There is therefore $N = 4\pi\sigma^2$ equally likely parents. If the grandparents are considered, the variance will be twice as large and the standard deviation $\sqrt{2}$ times as large. The area from which grandparents could be considered as if drawn at random is therefore $\pi(2\sqrt{2}\sigma)^2 = 8\pi\sigma^2$. As the distribution of individuals is uniform, the number of equally likely grandparents is $2N$. For any generation K in the past, the area of equally likely ancestors has a radius of $\sqrt{K}r$ and effective population size KN. The inbreeding of individual I relative to an area SN can be expressed as:

$$\left\{\begin{array}{lcl} F_{1s} & = & \frac{1}{N}(\frac{1+F'_{1s}}{2}) + \frac{N-1}{N}F'_{2s} \\ \\ F'_{2s} & = & \frac{1s}{2N}(\frac{1+F''_{1s}}{2}) + \frac{2N-1}{2N}F''_{3s} \\ \\ F''_{3s} & = & \frac{1s}{3N}(\frac{1+F'''_{1s}}{2}) + \frac{3N-1}{3N}F'''_{4s} \\ \\ \vdots & & \end{array}\right. \tag{2.42}$$

where $'$ denotes number of generations in the past and subscripts the size of the area concerned. It is tempting to consider F_{1s} in the previous system to be an F_{is}, because it is the inbreeding of one individual relative to a subset SN of the global population, which is supposed to be infinite. I suggest however that this is a peculiar type of F_{st} and will delay the discussion and justification of this statement until later.

System (2.42) has $(S-1)$ equations and S unknown $F_{.s}$ if the primes are dropped. However, F_{ss} can be considered as zero. By sequentially replacing the different $F_{.s}$ and letting the process go on for long enough so that the primes can be dropped, the following expression emerges:

$$F_{1s} = \frac{\sum_{k=1}^{S-1} t_k}{2 - \sum_{k=1}^{S-1} t_k},\ t_k = \frac{(k-1)N-1}{kN} t_{(k-1)} \tag{2.43}$$

(Wright, 1943). It has been shown that:

$$\sum_{k=1}^{S-1} t_k = 1 - S\,N\,t_S,\ S\,N\,t_S = \prod_{k=1}^{S-1}(1 - \frac{1}{kN})$$

(Wright, 1969, p 297, equation 12.24 with $N_X = XN$, as is the case for an area).

The product can be written:

$$\prod_{k=1}^{S-1}(1 - \frac{1}{kN}) = \frac{\Gamma(S - \frac{1}{N})}{\Gamma(S)\Gamma(1 - \frac{1}{N})}$$

where $\Gamma(x)$ is Euler's gamma function. The expression for F_{1s} becomes:

$$F_{1s} = \frac{\Gamma(1 - \frac{1}{N})\Gamma(S) - \Gamma(S - \frac{1}{N})}{\Gamma(1 - \frac{1}{N})\Gamma(S) + \Gamma(S - \frac{1}{N})} \tag{2.44}$$

and the expression for the total inbreeding, F_{1t} can be found by taking the limit as $S \rightarrow \infty$.

Calculations of this expression are tedious, with the numerator and denominator growing to huge quantities as the number of neighbourhoods S increases. The time to equilibrium is very long, of the order of tens of thousands of neighbourhoods (and therefore generations).

Wright proposed approximating the sum $\sum_{k=1}^{S-1} t_k$ by an integral. He suggested using (2.43) for the first 40 or 50 terms, for there is a large discrepancy between the exact value of F_{1s} and the continuous approximation and then to use the continuous approximation (Wright, 1969, equation 12.33, p300).

The same treatment has been applied to populations located on a linear continuum. The number of individuals to consider this time after S generations is $\sqrt{S}N$ and t_k is expressed as:

$$t_k = \frac{\sqrt{(k-1)}N - 1}{\sqrt{k}N} t_{(k-1)} \tag{2.45}$$

The expression for F_{1s} becomes:

$$F_{1s} = \frac{1 - \prod_{k=1}^{S-1}(1 - \frac{1}{\sqrt{k}N})}{1 + \prod_{k=1}^{S-1}(1 - \frac{1}{\sqrt{k}N})} \tag{2.46}$$

For estimation of this quantity, the same procedure as for area is followed (approximation of the sum with an integral), but the number of times (2.45) needs to be calculated has to be much larger than for an area. Wright then provides an approximation for higher terms (equations 12.28 and 12.29 p298).

The basic conclusion of this work is that F_{1s} increases as the area surveyed increases. The increase is faster when the original neighbourhood size is small. This effect is stronger in one-dimension than in two dimensions, as can be seen from Figures 12.2 & 12.3 (p 299 & 301 respectively in Wright, 1969). Indeed, with an original neighbourhood size of 5 in two-dimensions, even with 10^7 neighbourhoods, F_{1s} has not reached 1.

Wright points out that a more interesting quantity is the amount of differentiation among areas of any given effective population number within some constant large total considered to be infinite, F_{st}. Although it cannot be calculated directly, making use of the relationship between the different F's and providing that F_{1t} can be calculated, it is easy to derive F_{st}. Behaviour of F_{st} in the one-dimensional case shows some very interesting features (Fig 12.2 in Wright, 1969). Even when the length compared to the total contains many neighbourhoods, F_{st} stays constant (for $N = 10^3$, F_{st} starts decreasing for a length corresponding to 300 neighbourhoods, for a total of 3000). [It should be pointed out here that there is a misprint in the book: F_{it} in the third line of text, p 299, should read F_{is}]. On the other hand, in the two-dimensional case, F_{st} decreases from the start.

Formulae are also given for non-normal distribution of parent-offspring dispersal distances (an effect similar to increasing the neighbourhood size, Figures 12.5 & 12.7), long range migration (it lowers F_{st}, Figure 12.9). The effect of selfing leads to equations 12.57 & 12.58 (Wright, 1969, p312).

Wright suggested that the quantity F_{1s} described above is akin to an F_{is}. I have to disagree, at least partially, with him there. First of all, there do not seem to be any conditions that will lead to a negative value of F_{1s} in (2.43), since t_k is always positive and less than 1. One could say that selfing does not lead to negative values either, but avoidance of mating with relatives could be pictured as a negative rate of selfing, or a rate of outcrossing larger than 1. Furthermore, we saw in the subsection on the island model that F_{is} is the heterozygote deficit due to evolutionary pressures independent of population size. The expressions derived above (2.42) are typically dependent on population size. In his treatment of selfing in populations distributed in a continuum (p 311, 1969), Wright had to introduce another quantity, that he called E_{1s}, for the correlation between random gametes from neighbourhoods relative to an array of S neighbourhoods. The definition he gives for F_{1s} (equation 12.55, p 312, 1969) is essentially the same as (2.6).

In fact, if we consider the second equation in system (1) of Pollak (1987),

$$\left\{\begin{array}{lcl} F_{(t+1)} & = & (1-m)^2\left[\frac{s}{2}F_t + (1-\frac{1}{s})\theta_t\right] \\ \\ \theta_{(t+1)} & = & (1-m)^2\left[\frac{1}{2N}(1+F_t) + (1-\frac{1}{N})\theta_t\right] \end{array}\right. \tag{2.47}$$

the coefficients of F and θ in the second equation are the same as those of F'_{1s} and F'_{2s} in (2.42).

The solution of (2.47) can be easily found for equilibrium ($F_{(t+1)} = F_t = F$ and $\theta_{(t+1)} = \theta_t = \theta$):

$$\left\{\begin{array}{lcl} F & = & \frac{(1\overbrace{-2m+m^2})(1+2mNs\overbrace{-2m+m^2-m^2Ns}}{1+4mN-2mNs\underbrace{-4m^2+4m^3-m^4-2m^2N+5m^2Ns-4m^3Ns+m^4Ns}} \\ \\ \theta & = & \frac{(1\overbrace{-2m+m^2})}{1+4mN-2mNs\underbrace{-4m^2+4m^3-m^4-2m^2N+5m^2Ns-4m^3Ns+m^4Ns}} \end{array}\right. \tag{2.48}$$

On the other hand, the following system,which we have encountered in the subsection for an island model (this is the system leading to equation 2.30) can be written:

$$\left\{\begin{array}{lcl} f_{(t+1)} & = & (1+f_t)\frac{s}{2} \\ \\ \theta_{(t+1)} & = & (1-m)^2(\frac{1+f_t}{2N} + (1-\frac{1+f_t}{2N})\theta_t) \end{array}\right. \tag{2.49}$$

When equilibrium is reached, we have:

$$\left\{\begin{array}{lcl} \theta & = & \frac{(1\overbrace{-2m+m^2})}{1+4mN-2mNs\underbrace{-2m+m^2-2m^2N+m^2Ns}} \\ \\ f & = & \frac{s}{2-s} \end{array}\right. \tag{2.50}$$

under the usual simplifying conditions, over- and under-braced elements of the solutions (2.48) and (2.50) can be neglected leading respectively to:

$$\left\{\begin{array}{lcl} F & \simeq & \frac{1+2mNs}{1+4mN-2mNs} \\ \\ \theta & \simeq & \frac{1}{1+4mN-2mNs} \end{array}\right. \tag{2.51}$$

and:

$$\left\{\begin{array}{lcl} \theta & \simeq & \frac{1}{1+4mN-2mNs} \\ \\ f & = & \frac{s}{2-s} \end{array}\right. \tag{2.52}$$

that is, θ is the same in both cases, confirming the view that F_{1s} in (2.43) is akin to F_{st}. However, f and F are different since it can be shown that F is an F_{it}, whereas f in the second system is an F_{is}.

On the other hand, it is true that F_{1s} in equation 2.43 is the lowest possible in the hierarchy of F's. It would therefore be useful (following Wright's notation) to keep the notation F_{1s}, rather than associating with either F_{is} or F_{st}. The problem may seem semantic, but we will see in the next two chapters that there are fundamental differences between F_{is} and F_{st}, in terms of biases and of variance effective sizes.

2.5 Comparison of the different models.

2.5.1 Materials and methods

In order to compare the effects of different gene-flow patterns on genetic variability, **MODEL42** was used. Three levels of migration and two population sizes were used with the gametic cloud island model, and the 1-, 2- and 3-dimensional stepping stone model. For the 1- and 3-dimensional stepping stone, dispersal was limited to the nearest neighbours. Ten replicates were run over 10000 generations and F_{st} was used as a measure of the level of genetic variability. The three levels of migration were 0.1%, 1% and 10% for deme sizes of 16 and 64 and a total number of individuals of 4096. F_{st} was calculated every generation for the first 100, then every 10 until the 1000th generation and every 100 generations after that. Curvilinear regressions were applied to each of the 24 sets of parameters (3 levels of migration, 2 deme sizes and 4 gene-flow patterns) using the statistical package Genstat. Equation (2.36), with F_0 set to 0 gives:

$$F_t = \frac{(1-m)^2}{2N-(2N-1)(1-m)^2}(1-(1-m)^{2t}(1-\frac{1}{2N})^t)$$

and was used for the curvilinear regression. This equation should fit well with the data from the island model. Discrepancies with other gene-flow patterns should give some insights into how geographical structuring affects the genetic drift process.

2.5.2 Results

Figures 2.16 & 2.17 display the results. Each point on these graphs represents the average of F_{st} over the 10 replicates for the given generation. The generations are displayed on a logarithmic scale.

F_{st} increases with time, as expected and reaches higher values with low migration than with high migration. Equilibrium for F_{st} is reached in most of the cases after ten thousand generations. Time to equilibrium for F_{st} in the island model is determined by which of m or $1/N$ is the largest (Figure 2.14). In the last graph of Figures 2.16 & 2.17,

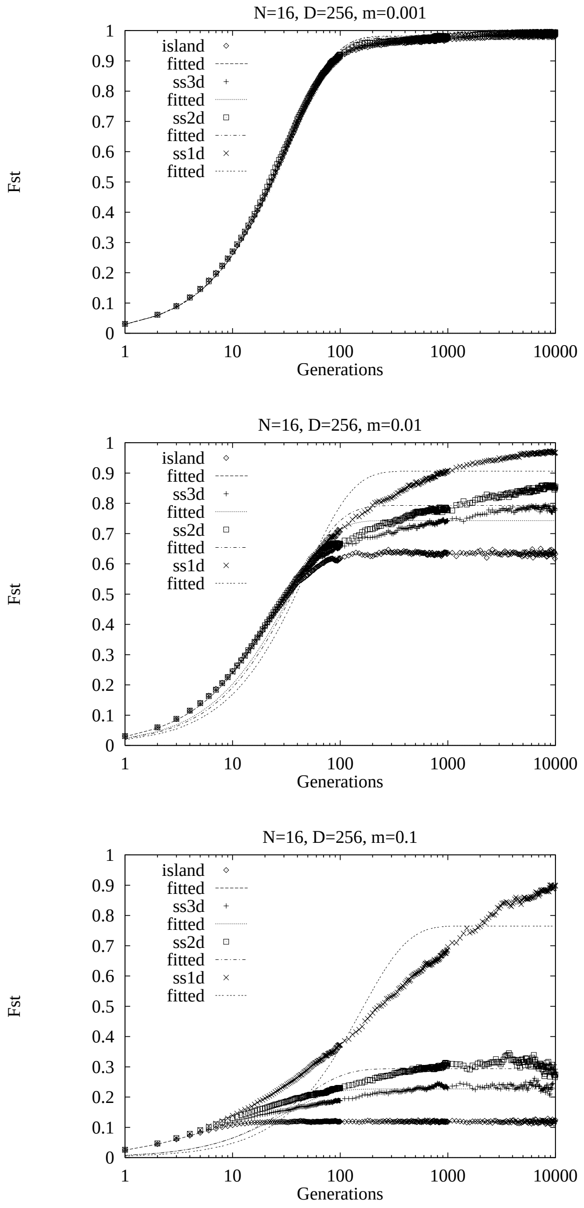

Figure 2.16: Changes in F_{st} over generations for different gene-flow patterns with the same sets of parameters

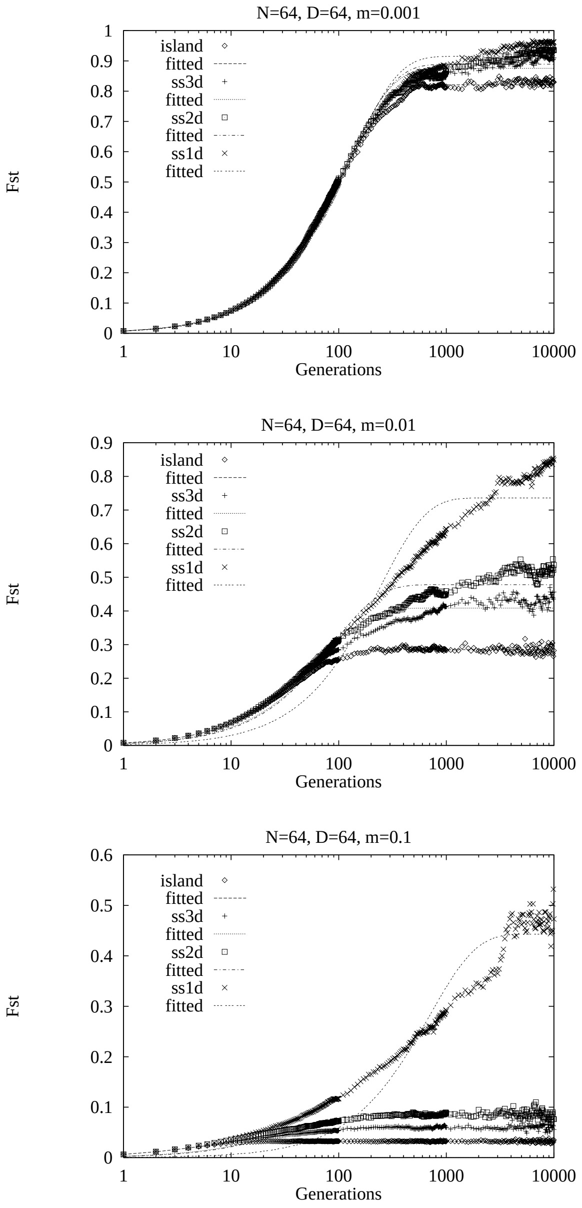

Figure 2.17: Changes in F_{st} over generations for different gene-flow patterns with the same sets of parameters

Table 2.2: Estimated m and N from curvilinear regression.Standard error in parenthesis.

		$m = 0.001$	$m = 0.01$	$m = 0.1$
Island, $N = 64$				
	$\hat{m}(\times 100)$	$0.084(0.0007)$	$0.922(0.009)$	$10.2(0.36)$
	$\hat{N}$	$65.9(0.3)$	$68.3(0.6)$	$70.0(2.5)$
$N = 16$				
	$\hat{m}(\times 100)$	$0.1(0.0)$	$0.9(0.0)$	$10.7(0.5)$
	$\hat{N}$	$18(0.097)$	$16(0.15)$	$17(0.86)$
Step. sto. $3D, N = 64$				
	$\hat{m}(\times 100)$	$0.055(0.001)$	$0.42(0.009)$	$2.28(0.08)$
	$\hat{N}$	$66.5(0.5)$	$87(1.6)$	$175(5.7)$
$N = 16$				
	$\hat{m}(\times 100)$	$0.1(0.0)$	$0.4(0.0)$	$1.5(0.1)$
	$\hat{N}$	$18(0.1)$	$23(0.5)$	$57(3.5)$
Step. sto. $2D, N = 64$				
	$\hat{m}(\times 100)$	$0.047(0.001)$	$0.294(0.008)$	$1.34(0.04)$
	$\hat{N}$	$68.8(0.7)$	$93.4(2.1)$	$202.6(5.8)$
$N = 16$				
	$\hat{m}(\times 100)$	$0.1(0.0)$	$0.3(0.0)$	$0.8(0.0)$
	$\hat{N}$	$17(0.1)$	$24(0.5)$	$70(3.3)$
Step. sto. $1D, N = 64$				
	$\hat{m}(\times 100)$	$0.034(0.001)$	$0.056(0.003)$	$0.040(0.002)$
	$\hat{N}$	$72.2(0.8)$	$161.9(6.0)$	$784.3(24.9)$
$N = 16$				
	$\hat{m}(\times 100)$	$0(0)$	$0.1(0)$	$0.1(0)$
	$\hat{N}$	$17(0.1)$	$26(0.5)$	$116(4.8)$

time to equilibrium for the island model is essentially the same, because m in both cases is larger than $1/N$.

The effect of connectedness is also as expected: as connectedness decreases, F_{st} increases, *but only after a certain number of generations.* This behaviour, as far as I am aware, has never been observed and is best seen in the middle graph of Figure 2.16. We can see that F_{st} up to around the 50th generation is the same for all gene-flow patterns and diverges thereafter. The same observation can be made for all the figures, with divergence time occurring earlier (as in the case for high migration), or later (as in the case for low migration). F_{st} increases at the same rate in stepping-stone as in island models until it levels off in the island model, whilst still increasing in stepping-stones. Two- and three-dimensional stepping stones seem to reach a plateau after some time (top graph of Figure 2.17 and middle graph of Figure 2.16 are exceptions), whereas 1-dimensional stepping stone models never seem to plateau before reaching 1 (bottom graph of Figure 2.17 is an exception, but I suspect that this is due to the small number of demes).

The first graph in Figure 2.16 shows that, when migration is very low (1 migrant every 60 generations) there is no difference between the gene-flow patterns. When migration is so low that the plateau occurs at very high value of F_{st}, there is virtually no effect of connectedness. This is not surprising, since very low rates of migration ensure that even neighbouring demes will display large variances in allelic frequencies (low correlation). If there is no migration at all, the different populations are not connected and all the models behave in the same manner. A rule of the thumb could be that with $Nm < 0.025$, the effect of geographical structuring is virtually non-existent. As migration increases, the different gene-flow patterns start to differentiate. The difference between 2- and 3-dimensional stepping-stones being much smaller than the difference between 2- and 1-dimensional. It seems that for values of Nm larger than 5, the disparities between island models and 2- and 3-dimensional stepping-stones lessen, but this may be due to the small number of demes in the population.

Estimated migration and deme size from the curvilinear regressions are given in Table 2.2. Estimates of both N and m for the island model are very close to the input parameters, whereas m is always lower and N larger for stepping-stones. Within the stepping-stones, the 1-dimensional always leads to the lowest estimates of migration and largest estimates of deme sizes, followed by the 2-dimensional and the 3-dimensional stepping-stone. The fit of the curvilinear regression to the data is however rather bad, showing a tendency to underestimate F_{st} in the early phases and to overestimate it in the late phases (Figures 2.16 & 2.17). Therefore, unless migration is so low that the equilibrium value of F_{st} is very close to 1 (top graph in Figure 2.16), equation 2.36 is a bad predictor of F_{st} in stepping stone models.

We can, however, gain some information on what the equation should be like for stepping-stone models from the Figures: it is only when F_{st} reaches a plateau in the island model that its value diverges for stepping-stone models. This is the time necessary for correlation of allelic frequencies between adjacent groups to develop. As these correlations developed, it makes the panmictic unit larger and decreases the migration, because these larger units of random mating exchange, on average, less migrants than the smaller units of the early process. By making N and m dependent on time, it should be possible to get a better fit of the curvilinear regression to the data in a stepping-stone model.

2.6 Discussion and conclusion

2.6.1 Pros and cons of MODEL42.

MODEL42 was developed on a DOS platform, using Borland's Turbo-Pascal version IV, V and VI. In these implementations of a Pascal compiler under MS-DOS, the maximum size of an array is 64 kilobytes, allowing for a maximum of 16384 individuals if identity by descent is to be measured. Moving to a UNIX platform, or using the new version of the Pascal compiler from Borland, Borland-Pascal VII, would solve this problem. Translation of the code into C is also currently being done.

Only single locus systems can be modelled. Replicates are often considered as equivalent to different independent loci (Slatkin, 1985a), but they are not, since the pedigrees of independent loci from the same individual are the same, whereas the pedigree of independent replicates are different. This may have some important consequences in term of variance between loci. Indeed, Feldman & Christiansen (1975) have shown that migration among a set of semi-isolated populations could result in a cline of linkage disequilibria and Ohta (1982) proposed to measure the extent of isolation between populations with D_{st}, an F_{st}-like statistic based on linkage disequilibrium. A solution to this problem has already been suggested. Rather than giving a location to alleles in the initial generation, locations are specified for individuals and each generation, after the sampling of parents, the sampling of alleles within parents is carried out as many times as there is loci. This will also allow the modelling of certain types of selection, such as meiotic drive. Other types of selection, however, seem much more complicated to implement.

2.6.2 State of analytical work.

As we have seen, the theory for island models of populations is rather accurate and the discrepancies between the different types of island models are very small. Predictions of the values of F_{st} at equilibrium, as well as in non-equilibrium situations can be made, and seem accurate (Figures 2.16 & 2.17). The time to equilibrium is dependent on which of the two quantities m or $1/N$ is the largest and value at equilibrium is essentially dependent on the effective number of migrants, Nm. Although results are not shown, predictions in an island model with a proportion s of selfing are also accurate for equilibrium as well as non-equilibrium situations.

Equations for prediction of the effective sizes of both local and global population are also given. The effect of subdivisions and of selfing, although similar in terms of F are opposite in terms of variance effective sizes: subdivision leads to larger effective than census sizes, whereas selfing leads to smaller effective than census sizes.

The situation is quite different for stepping-stone models. The only quantity that has been derived analytically is the correlation of allele frequencies at equilibrium for demes k steps apart, when the number of demes is infinite and with a proportion of long range migration ensuring that variability is still present in the total array. Relating the correlations of allele frequencies to F_{st} has yet to be done. Wright derived equilibrium values for F_{is} in a model of randomly distributed clusters, which bears only some resemblances to the stepping-stone models (Wright, 1969, p 320–323). The finding of Crow & Aoki (1984) that equilibrium values of G_{st} in stepping-stone models are dependent on the shape of the habitat are confirmed here. It would, however, be of interest to obtain a relation similar to (2.36) for stepping-stone models. I suggest that this could be achieved if N and m are made time dependent. Expressions for $N(t)$ and $m(t)$ have yet to be found for the relation:

$$F(t+1) = \frac{(1-m(\infty))^2}{2N(\infty) - (2N(\infty)-1)\,(1-m(\infty))^2}\left(1-(1-m(t))^{2t}(1-\frac{1}{2N(t)})^t\right)$$

However, they should be constant until equilibrium is reached in an island model and

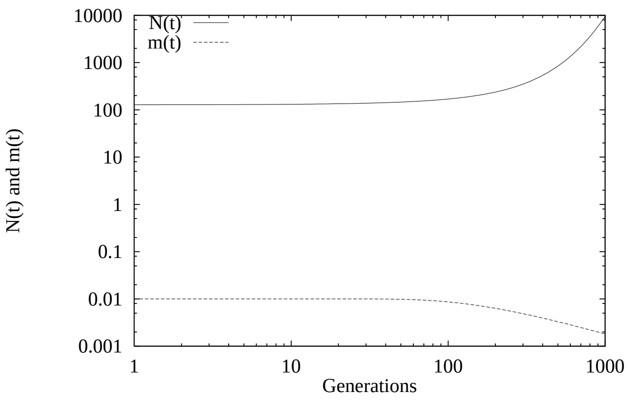

Figure 2.18: Typical behaviour of the functions N(t) and m(t).

then $N(t)$ should increase while $m(t)$ should decrease. A possible form could be:

$$\begin{aligned} N(t) &= N(1 + A1\exp(t/C1)),\\ m(t) &= m(1 - A2\exp(-C2/t)) \end{aligned}$$

where $C1$ and $C2$ are constants of the order of the number of generations it takes for equilibrium to be reached in an island model and $A1$ and $A2$ are coefficients taking into account the dimension of the habitat. Figure 2.18 shows what the typical behaviour of these functions should be.

The situation is even more complex for the isolation by distance model. As mentioned before, the neighbourhood size is supposed to be equivalent to a random mating unit and depends on the distribution of parent-offspring dispersal. The assumption that individuals within a neighbourhood mate at random is, however, a gross simplification: individuals at the centre of the neighbourhood are more likely to be the parents of the central individual than individuals at the edges. Furthermore, the proportion of selfing in the population is given by the distribution of parent-offspring dispersal distances since it is the proportion of the parent-offspring dispersal distances that fall in the unit square. The smaller the neighbourhood, the larger this quantity (which is always larger than $1/N$, c.f. table 1 in Rolhf & Schnell, 1971, p 297). Rolhf & Schnell (1971) looked at the effect of the dispersal distribution of parent-offspring and showed that a uniform rather than a conical (mimicking a normal) distribution decreases F

drastically (Figure 14, p 316). They then attempted to derive exact solutions for F's under different parent-offspring dispersal patterns, obtained good agreement of the theory with their simulated data set and found discrepancies with Wright's results for the uniform distribution. However,they could calculate F only up to the second generation. As they put it:

> 'Unfortunately, it does not appear feasible to work out the expected F's for later generations using our approach.'

They also showed that, as time goes on and even with a uniform distribution of parent-offspring dispersal, the distribution of ancestors after only 2 generations is no longer uniform (Table 2,p 313).

In this work, no attempt was made to follow the change of F_{st} in isolation by distance models for the reasons stated above. Furthermore, we will see in the next chapters that it is not possible to detect a random breeding unit in isolation by distance models.

Chapter 3

F-Statistics

3.1 Introduction

Prior to 1966, the amount and distribution of genetic variability within species was largely unknown. The phenotypic markers available were frequently under polygenic control or were likely to be unrepresentative of the whole genome (eg. lethal alleles). The discovery of protein gel electrophoresis independently by Harris (Harris, 1966) and Lewontin & Hubby (Hubby & Lewontin, 1966; Lewontin & Hubby, 1966) initiated twenty five years of intensive investigation of protein variation in natural populations by hundreds of laboratories (Lewontin, 1991). This polymorphism was shown to segregate in a Mendelian manner. The amount of variability detected was astonishing: about one third of the loci surveyed over a wide range of species (Nevo *et al.* (1984) carried out a literature survey and found studies of intraspecific variation in 1111 species, with an average of 23 loci and 200 individuals per species examined) were polymorphic and the average heterozygosity per individual was 10 percent (Lewontin, 1991). These observations generated a debate between the adherents of a selectionist (balancing) view led by Mayr, Cain and Dobzhansky on the one hand and the adherents of a neutralist (neoclassical) view, led by Kimura on the other. Although still unresolved, the tenants of both schools are now aware that nothing is as clear cut as they first thought, due mainly to the fact that their hypotheses were based on very simplistic population genetic models with no population structuring or specific reproductive system. Fortunately, another outcome of the discovery of gel electrophoresis was the possibility of using this information to get a better understanding of the genetic structure of natural populations, providing that the loci under scrutiny are not undergoing too strong a selection. Measurements of gene-flow are discussed in the next sections and the statistical robustness of these measurements are discussed.

3.2 Measuring gene-flow

Gene-flow is a collective term that embodies all mechanisms resulting in the movement of genes from one group of individuals to another (Slatkin, 1985a). The word population will not be used here, because, as we will see in Chapter 4, they are extremely difficult to characterise. Measuring gene-flow implies estimating a quantity that will provide information about movement of genes. This may be achieved by using two distinct approaches:

1. Dispersal of individuals or of gametes. These types of measurements will give some information about gene dispersal, providing that the individual reproduces or that the gamete is successful in producing an individual. *These types of measurements are direct methods of estimating gene-flow.*

2. Inferences of gene-flow by observing the frequency distribution of alleles and genotypes. *These are indirect methods of estimating gene-flow.* The actual movement of genes will not be observed, but the distribution of allele and genotype frequencies should give us some indication of how much gene-flow occurs in the surveyed species.

The respective advantages and inconveniences of both types of measurement are discussed below.

3.2.1 Direct methods

The principle behind this technique is to identify visible (conspicuous) markers and to follow their movement. The first study to answer an evolutionary question using this type of marker is that of Dobzhansky & Wright (1943). They used an orange marker gene on *Drosophila pseudoobscura* that was known to have no effect on dispersal and released homozygous individuals from a source point. They collected the flies in traps 10 or 20 metres apart laid out in a crossed pattern. They used the information obtained to get a temporal estimate of the neighbourhood effective size, which was of the order of 500–1000 individuals. However, another study by Crumpacker & Williams (1973) gave an estimate of the order of 10000, an order of magnitude larger than Dobzhansky & Wright's estimate. Further studies showed that the condition under which the flies are released will have a drastic influence on the neighbourhood size estimate. A study by Jones *et al.* (1981) was carried out in Death Valley, California, where flies can only be found in discrete oases separated by several kilometres. The marked flies were released in a clearly unsuitable habitat. The following day, flies were trapped not only in the closest oasis, 2 km away, but also in the farthest one, 8 km away. The average dispersal distance was found to be 400-500m, three times as much as that found by Dobzhansky and Wright. It was shown (Coyne *et al.*, 1982) that, even where the flies were released in an oasis, some were trapped as far as 5 km away in the desert and at an oasis 14.6 km away.

Other examples of direct measurement of gene-flow can be found in Endler (1977), Wright (1978, chapter 2) and Slatkin (1985a) but the above example is sufficient to indicate what can and what cannot be done using this type of measurement. At best, direct measurement can only indicate the gene-flow occurring under the conditions when the experiment is conducted. Experiments tend to be carried out under normal or natural environmental conditions. If the sampling strategy does not disturb the dispersal pattern (but see Johnston & Heed, 1975) and is adequate, direct measurement will provide good estimates of common movements. In a suitable environment, *D. pseudoobscura* has an average daily displacement of 200 metres but, when conditions get more difficult, the average daily displacement increases. Unfortunately, one aspect of the dynamics of movement is not taken into account by direct measurement: its stochasticity. Although flies may move an average of 200 meters a day, a drought, or the local extinction of a population, may lead to drastic shift in the dispersal pattern over a very short period of time. These type of movement are very unlikely to be recorded by direct measurement, but will affect the genetic make-up of the population.

3.2.2 Indirect methods

> 'An indirect method is one that uses observed spatial distribution of alleles, chromosomal segment or phenotypic traits to draw inferences about the level or pattern of gene-flow and other mechanism of genetic evolution.' (Slatkin, 1985a)

The methods for making these inferences has been discussed in the previous chapter. It is sufficient to say here that the allelic and genotypic distribution in a population are a function of the evolutionary forces acting on it, such as selfing and migration. If many independent loci show a similar pattern of allelic and genotypic distributions, it is possible to relate these to distributions obtained from population genetic models. Many different techniques have been developed, and are reviewed below.

The 'private allele' method of Slatkin(1985b)

The 'private allele' method of estimating gene-flow is one of the most intuitive ways of approaching the problem. When surveying a population by gel electrophoresis, one obtains a distribution of allele frequencies. If the survey encompasses many samples, the distribution of allele frequencies can be obtained for each sample. The idea behind the 'private allele' technique is that if an allele is present in only one of the samples, then its frequency in this sample will be some function of the migration rate. However, when private alleles are at very low frequencies, they could come from a newly arisen mutant in the population, but, if they reach a high frequency in only one of the samples, then this suggests that very little genetic material is exchanged between the different samples. The inference of migration level can then be done by running computer simulations and relating the frequencies of private alleles to the migration proportion. Slatkin (1985b) provides the relationship between 'private allele' frequencies and the effective number of migrants, $N_e m_e$:

$$\log \overline{p_1} = a \log N_e m_e + b, \tag{3.1}$$

where a and b are found by computer simulation and depend on the sample size and the number of demes sampled (Slatkin, 1985a; Slatkin & Takahata, 1986). Applications of this technique to natural data is found in Slatkin (1985a) and gives an estimate of $N_e m_e$ consistent with other indirect methods. As pointed out in the paper, more work is needed to identify sources of bias, but the technique seems to be quite insensitive to weak selection (Slatkin, 1985a) and mutation rate (Barton & Slatkin, 1986).

Although appealing in its simplicity, this technique has several drawbacks. Slatkin & Barton (1989) showed that the 'private allele' technique is not as robust as other existing techniques, such as those based on F-statistics. Indeed, most of the information collected in an experiment is not used, because only few alleles will be private. Another drawback of the technique is that the genotypic composition of the population is ignored and it is therefore impossible to know if each sample belongs to one or more breeding units.

Spatial autocorrelation analysis

Spatial autocorrelation is a technique derived from the field of ecology. If locations close to each other tend to exchange more genetic material than locations further apart, then calculating the correlation coefficients of allelic frequencies with distance should give a good overview of the amount of gene-flow occurring in the population. For each allele, the correlation coefficient is calculated over all pairs of locations that are a specified distance apart and used to generate a correlogram. To date, this technique only describes the genetic correlations between samples, without making any genetical inferences from it (Slatkin, 1985a). To carry out a spatial autocorrelation analysis, samples from different locations need to be taken. It is then necessary to find a measure of the distance between these locations, either geographic distance, or nearest neighbour (Slatkin & Arter, 1991). Figure 1 in Slatkin & Arter (1991) shows a variety of such possible distances measures, depending on the assumptions made *a priori* about dispersal pathways. The final step is to compute the spatial autocorrelation for each variable in each distance class. Moran's I (Cliff & Ord, 1981) is often used:

$$I_k = \frac{n \sum_{i \neq j} w_{ij}^{(k)} (x_i - \overline{x})(x_j - \overline{x})}{\sum_{i \neq j} w_{ij}^{(k)} \sum_i (x_i - \overline{x})^2}, \tag{3.2}$$

where n is the number of locations sampled, $\overline{x}$ is the average value of x_i, the sample frequency of the allele under scrutiny, k is the distance class and $w_{ij}^{(k)} = 1$ if i and j are both in the same class and 0 otherwise. For each class of distance, I_k can be estimated. This set of values can then be plotted on a correlogram. A flat correlogram would be an indication that geographical structure is non existent, as in an island model, whereas a decaying one would indicate restricted gene-flow between locations. The statistical significance can then be tested using techniques such as those described in Cliff & Ord (1981).

With reference to what biological inferences can be made using this method, there is a vigourous debate between Slatkin and co-workers on the one hand (Slatkin & Arter, 1991) and Sokal and co-workers on the other (Sokal & Oden, 1991). As mentioned already, drawing inferences about levels of gene-flow using this technique seems difficult because a genetical theory to support these estimates does not exist. As with the 'private allele' technique, only allelic frequencies are used and all the information contained by genotype distributions is lost. On the other hand, this is the only technique that explicitly takes into account geographical distance and it may be the best method of analysing populations living on a continuum, as pointed out by Heywood (1991).

Lethal alleles

The first estimates of $N_e m_e$ were based on the frequency of lethal alleles. Because of the strong selection against the allele, any two lethals could be assumed to have descended from the same mutation in the recent past. Simple models predict the way in which effective population size, heterozygote fitness and immigration rate affect the probability of allelism of lethal from the same and different populations (Slatkin, 1985a). This technique, however, has been used mainly to infer population size and heterozygote fitness, but should be mentioned as one of the first methods to attempt to estimate gene-flow.

Dobzhansky & Wright(1941) used observations of lethal alleles in *D. pseudoobscura* from isolated populations to measure several populations parameters and found $N_e m_e$ to be 54. In their discussion they tempered the estimate down due to biases, but concluded that $N_e m_e$ is certainly larger than 5 in this species, indicating that genetic drift is not a strong enough force to allow differentiation between populations of this species.

Wallace(1966) used lethal alleles to measure the decrease in frequency with geographical distance and found that in *D. melanogaster*, the frequency of lethal alleles would decrease approximately 50% at a distance of 150 meters.

Genetic distances

Genetic distances were first used in population genetics to provide a single quantitative measure of differences in two or more sets of allele frequencies. Differences in gene frequencies between populations provide such a measure, although there exist other ways of estimating genetic distances, such as differences in a quantitative character, or number of nucleotide substitutions.

The first study involving genetic distances as measured with differential gene frequencies is that of Cavalli-Sforza & Edwards (1967), where the authors obtained an evolutionary tree of human races.

Nei (1987) distinguishes two classes of genetic distances: the first is used for population classification and includes Pearson's coefficient of racial likeness (Pearson, 1926), Rogers' distance (Rogers, 1972) and Mahalanobis' D^2 statistic (Mahalanobis, 1936). They are geometric distances, in the sense that the population could be represented as a point in a v-dimensional space on the basis of the frequencies of the v alleles at a locus. All populations then lie on the hyper-plane defined by $\sum_{u=1}^{v} p_u = 1$. With two populations, X and Y, with respective allele frequencies p_1 and $p_2 = 1 - p_1$ and q_1, q_2, the Euclidean distance between the populations, based on that diallelic locus, is

$$d_{XY} = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2} \tag{3.3}$$

and, if there are v alleles

$$d_{XY} = \sqrt{\sum_{u=1}^{v} (p_u - q_u)^2} \tag{3.4}$$

Geometric distances based on the square-root of allelic frequencies have also been used, so that instead of lying on the hyper-plane, populations lie on a hyper-sphere with radius 1. A measure of the genetic distance is then simply the angle between the two radii joining the centre of the hyper-sphere to the location of the populations on its surface. It can be shown that the distance between two populations can be expressed as

$$\psi^2 = \frac{1}{2} \sum_u \left[\frac{(p_u - q_u)^2}{p_u + q_u} \right], \tag{3.5}$$

where ψ is the angle between the radius of the two populations (Weir, 1990). Cavalli-Sforza & Bodmer (1971) used the chord length, d, between population X and Y as a measure of genetic distance, where

$$d = \sqrt{[2 - 2\cos(\psi)]}. \tag{3.6}$$

A second class of genetic distances is used for evolutionary studies and can be related to Wright's F-statistics. This second class includes F_{st} itself (cf. next subsection) and Nei's standard genetic distance, D, (Nei, 1972):

$$D = -\ln\left(\frac{J_{XY}}{\sqrt{J_x J_Y}}\right) \tag{3.7}$$

where $J_X = \sum p_u^2$, $J_Y = \sum q_u^2$ and $J_{XY} = \sum p_u q_u$. This last equation is used very often to assess genetic distances.

Slatkin (1985a) proposed another classification following Latter's idea (Latter, 1973), using genetic distances based on heterozygosity, and another based on homozygosity. He pointed out that the type of information extracted from each class is quite different. While $N_e m_e$ can be extracted from F_{st} (cf. Chapter 2), it seems possible to extract m_e directly from D as proposed by Nei (1975, p. 194).

Although promising, the statistical problems associated with estimations and inferences of this last category of distances remain largely unexplored.

The F-statistics of population structure

The F-statistics are tools devised by Wright(1921,1951) that measure the heterozygote deficit relative to its expectation under the Hardy-Weinberg equilibrium (H.W.E.). Although reminiscent of the 'beanbag genetics' of Mayr (1959), the Hardy-Weinberg equilibrium remains the reproductive regime of reference for two reasons: it is the best understood and, whatever the genotypic make up of the population, one generation of panmixia restores the equilibrium.

A measure of the heterozygote deficit is simply the ratio of the difference between expected and observed heterozygosity to the expected heterozygosity:

$$F = \frac{H_{Exp} - H_{Obs}}{H_{Exp}} = 1 - \frac{H_{Obs}}{H_{Exp}} \tag{3.8}$$

The symbol F stands for Fixation index. If individuals are to be more homozygous than predicted by H.W.E., F will be positive, with a maximum of 1, when all individuals are homozygous. On the other hand, if individuals tend to be less homozygous than predicted by H.W.E., then F will be negative, with a minimum of -1. A very nice feature of this parameter is that it can be related to both the inbreeding coefficient and the probability of identity by descent as shown in Chapter 2. Wright preferred to define F as the correlation of the presence or absence of an allele in uniting gametes, because a probability cannot be negative. As he puts it (Wright, 1969):

> 'In a panmictic population, there is no correlation between homologous genes of uniting gametes relative to the gene frequencies in the whole population. On splitting up into small lines which breed within themselves, a correlation between uniting gametes is to be expected. This suggests a description of population structure in general and the effects of inbreeding in particular by means of the correlations expected under Mendelian heredity. The concept of correlation of homologous genes of a certain class is required from the broader standpoint of a group of parameters useful for the description of population structure in general.'

First, let us demonstrate that F is a correlation coefficient by considering a diploid population with a single, multi-allelic, segregating locus. If allele A_i is opposed to all others, frequency of homozygote A_iA_i is given by $(1-F)p_i^2 + Fp_i$ where p_i is the frequency of allele A_i in the population and the frequency of heterozygote $A_iA_j, j \neq i$ is $2p_i(1-p_i)F$. To show that the fixation index F is the same as the correlation between uniting gametes, let $V_1, V_2, \ldots, V_k$ be arbitrary values assigned to the alleles and w, with suitable subscripts, describes the proportional frequencies of the different alleles and genotypes as shown in Table 3.1. The formula of Pearson's correlation coefficient's , ρ, is

Table 3.1: Proportional frequencies of the different genotypes in the case of multiple alleles under any reproductive regime

	A_1	A_2	$\ldots$	A_k	Total
A_1	$(1-F)p_1^2 + Fp_1$	$(1-F)p_1p_2$	$\ldots$	$(1-F)p_1p_k$	p_1
A_2	$(1-F)p_2p_1$	$(1-F)p_2^2 + Fp_2$	$\ldots$	$(1-F)p_2p_k$	p_2
$\vdots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
A_k	$(1-F)p_kp_1$	$(1-F)p_kp_2$	$\ldots$	$(1-F)p_k^2 + Fp_k$	p_k
Total	p_1	p_2	$\ldots$	p_k	1

$$\rho = \frac{Cov(x,y)}{\sqrt{\sigma_x^2\sigma_y^2}} = \frac{\overline{xy} - \overline{x}\,\overline{y}}{\sqrt{(\overline{x^2} - \overline{x}^2)(\overline{y^2} - \overline{y}^2)}}.$$

The different components of this expression are defined, in the context of correlation between alleles as follows: the mean genetic value of the population and the variance are given by

$$\overline{x} = \sum_i^k V_i w_i = \sum_i^k V_i p_i$$

and

$$\sigma_x^2 = \sum_i^k V_i^2 p_i - (\sum_i^k V_i p_i)^2,$$

which leads to

$$\rho = \frac{\overbrace{\sum_j^k \sum_i^k V_i V_j w_{ij}} - (\sum_i^k V_i p_i)^2}{\sigma_x^2}. \tag{3.9}$$

The w_{ij} are derived from Table 3.1 with: $w_{ij} = (1-F)p_ip_j$ if $i \neq j$ and $w_{ij} = (1-F)p_i^2 + Fp_i$ if $i = j$. Thus the over-braced part of (3.9) can be written

$$\sum_j^k \sum_i^k V_i V_j w_{ij} = \sum_i^k \overbrace{V_i^2[(1-F)p_i^2} + Fp_i] + \sum_i^k \overbrace{V_i(1-F)p_i \sum_{j\neq i}^k V_j p_j} \tag{3.10}$$

Using Newton's expansion on the over-braced part of (3.10) and rearranging, we obtain

$$(1-F)\sum_i^k [V_i^2 p_i^2 + V_i p_i \sum_{j\neq i}^k V_j p_j] = (1-F)(\sum_i^k V_i p_i)^2 \tag{3.11}$$

This is valid only if F is the same for all alleles and their combinations, such as when there is no selection affecting the locus under scrutiny. This problem is not mentioned in Wright (1969) but has been pointed out by different authors (Roughgarden, 1979; Golding & Strobeck, 1983). In the fourth volume of his masterpiece, Wright (1978, p. 60) came back to this problem by highlighting the differences between the inbreeding coefficient, f, to which the demonstration applies and F-statistics, to which the demonstration applies only under the hypothesis that the locus under scrutiny is neutral. Bearing in mind that the evolutionary forces we are interested in affect alleles in the same way, this 'mathematical trick' should not affect the proof and equation 3.10 can now be rewritten

$$\sum_j^k \sum_i^k V_i V_j w_{ij} = F \sum_i^k V_i^2 p_i - F(\sum_i^k V_i p_i)^2 + (\sum_i^k V_i p_i)^2 \qquad (3.12)$$

Substituting (3.12) into (3.9) leads to:

$$\rho = \frac{F\sigma_x^2}{\sigma_x^2} = F \qquad (3.13)$$

This completes the proof that the fixation index F is the same as the correlation coefficient between uniting gametes and that it is independent of the genetic values assigned to the different alleles.

The success of fixation indices compared to other indirect methods come from their ability to partition the heterozygote deficit into two components (which could be extended to many more, e.g. Wright, 1978). If we sample randomly in natural populations, samples are taken from different locations. These samples may or may not belong to the same panmictic unit. Fixation indices allow the measurement of the heterozygote deficit within sampled locations and provide an estimate of F due to evolutionary forces such as selfing. This fixation index is called F_{is}, i for individual and s for subpopulation. The fixation index could also be measured for the whole sample, leading to F_{it}, t for total population. If F_{is} and F_{it} differ, then another source of heterozygote deficit must exist: it is known as the Walhund effect (Walhund, 1928) and is quantified by F_{st}. In terms of correlation we define:

- F_{it} as the correlation between gametes that unite to produce the individuals relative to the gametes of the total population
- F_{is} as the average over all subdivisions of the correlation between uniting gametes relative to those of their own subdivisions

- F_{st} as the correlation between random gametes within subdivisions, relative to gametes of the total population.

The relationship between these three F's can now be derived. Within a single population, s, the heterozygote deficit, F_{is_i}, for allele A_i can be written as a function of the allele frequency in that subpopulation, p_{s_i} and the observed number of heterozygotes:

$$H_{Obs_{s_i}} = 2p_{s_i}(1 - p_{s_i})(1 - F_{is_i}) \qquad (3.14)$$

where $H_{Obs_{s_i}}$ is the observed heterozygosity in the sth subpopulation. F_{is} is then defined as the average over all subpopulations of F_{is_i}. This average will not be an unweighted average, because allelic frequencies between subpopulations will differ and therefore, the contribution of each subpopulation to the global F_{is} will be different. If the average was not weighted, the contribution to F_{is} of a subpopulation with only one copy of allele A_i would be the same as the contribution of a subpopulation with N copies of allele A_i. This point has been stressed by some authors (eg. Nei, 1973), but did not appear in the demonstration of Wright (1969, pp. 294– 295), although it is explicitly taken into account in Wright (1978, p. 80). The expression of the weight is simply $p_s(1-p_s)$ [dropping the subscript i for simplicity], leading to:

$$\overline{F_{is}} = \frac{\sum_s^D p_s(1-p_s)F_{is}}{\sum_s^D p_s(1-p_s)} \qquad (3.15)$$

where D is the number of subpopulations sampled. The total heterozygosity can now be written

$$H_{Obs_t} = \frac{2}{D}\left[\sum_s^D p_s(1-p_s)\left(1 - \frac{\sum_s^D p_s(1-p_s)F_{is}}{\sum_s^D p_s(1-p_s)}\right)\right], \qquad (3.16)$$

which leads to

$$H_{Obs_t} = \frac{2}{D}\sum_s^D p_s(1-ps)(1-\overline{F_{is}}). \qquad (3.17)$$

Since $\overline{F_{is}}$ is the same for all subpopulations, it can therefore be taken out of the summation sign. Rearranging then leads to

$$H_{Obs_t} = 2(1-\overline{F_{is}})\left(p_t - \frac{1}{D}\sum_s^D p_s^2\right), \qquad (3.18)$$

where p_t is the frequency of A_i over all subpopulations.

Bearing in mind that the variance of the frequency of A_i, $\sigma_{p_s}^2$, over subpopulation is $1/D\sum_s^D p_s^2 - p_t^2$ we obtain

$$H_{Obs_t} = 2(1-\overline{F_{is}})\left[p_t(1-p_t) - \sigma_{p_s}^2\right]. \qquad (3.19)$$

Table 3.2: Values of the different F's under extreme reproductive regimes. Cases 1 and 2 affect all loci equally, whereas cases 3 and 4 affect only the loci tightly linked to the locus undergoing disassortive mating.

Breeding system	F_{is}	F_{st}	F_{it}
1. Random mating			
• a. large migration between subpopulations	0	0	0
• b. no migration between subpopulations	0	1	1
2. Total selfing			
• a. large migration between subpopulations	1	0	1
• b. no migration between subpopulations	1	1	1
3. Disassortative mating, 2 alleles			
• a. large migration between subpopulations	-1	<0	-1
• b. no migration between subpopulations	-1	<0	-1
4. Disassortative mating, large number of alleles			
• a. large migration between subpopulations	-1	?	-1
• b. no migration between subpopulations	-1	?	-1

Dividing both sides of the equation by $2p_t(1-p_t)$ leads to

$$\frac{H_{Obs_t}}{2p_t(1-p_t)} = (1-\overline{F_{is}}) - \frac{\sigma^2_{p_s}}{p_t(1-p_t)}(1-\overline{F_{is}}). \tag{3.20}$$

In the left-hand side of the equation, we can recognise the expression $1-F_{it}$ (e.g, equation 3.8). Rewriting (3.20) then gives

$$1-F_{it} = (1-\overline{F_{is}})(1-\overbrace{\frac{\sigma^2_{p_s}}{p_t(1-p_t)}}). \tag{3.21}$$

The over-braced part of the last equation is the formula for F_{st}. Therefore the general relationship between the three F's is:

$$(\mathbf{1}-\mathbf{F_{it}}) = (\mathbf{1}-\overline{\mathbf{F_{is}}})(\mathbf{1}-\mathbf{F_{st}}) \tag{3.22}$$

Now that the relation between the three F's is established, we need to focus on their respective meanings. Table 3.2 provides examples of the values taken by these three statistics under different extreme reproductive regimes. The first thing to notice is that different reproductive regimes lead to the same value for F_{it}, as shown by cases 1b, 2a and 2b. The outcome of these reproductive regimes is that no heterozygotes are left in the population, but the statistic F_{it} is unable to discriminate between the

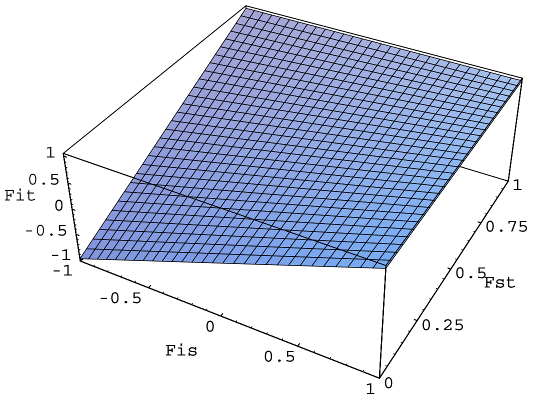

Figure 3.1: F_{it} as a function of F_{is} and F_{st}. The same F_{it} value can arise from different combination of F_{is} and F_{st}.

forces that caused this deficit. The same conclusions apply to any value of the statistic F_{it} as shown in figure 3.1.

On the other hand, F_{is} and F_{st} quantify the respective contributions of inbreeding and structuring to the heterozygote deficit, providing that the sampling strategy is adequate.

F_{st} can be interpreted as a measure of the amount of differentiation among sub-populations, relative to the limiting amount under complete fixation within each subpopulation, in contrast to $\sigma^2_{p_s}$ which measures this differentiation in absolute terms (Wright, 1978). Indeed, the denominator of F_{st}, p_tq_t, is the expression of the maximum possible variance in allelic frequency in the population, which would occur if a proportion p_t of the populations were fixed for one allele and the remaining populations for the other, in the simple case of two alleles segregating at a locus. Wright(1978) pointed out that observations of F_{st} alone could be misleading, because different patterns of allelic frequencies could lead to the same F_{st}. For example, consider a sample of twenty populations. In one instance, one population is fixed for allele A_1 and the remaining nineteen fixed for A_2. Another set of twenty populations has ten populations fixed for A_1 and the remainder for A_2. The F_{st} obtained from both samples will be the same, but the numerator and denominator will be different. The extent of differentiation seems larger in the second than in the first case and, if many loci where to show the same pattern, one could suspect that the nineteen populations fixed for the same allele are isolated from the twentieth, but not isolated from each other. On the other hand, if such a pattern is displayed by only one locus, the hypothesis that the pattern had arisen by chance would be difficult to reject.

Another interesting example is one where only two populations are sampled and the locus under scrutiny has four alleles, with alleles A_1 and A_2 equi-frequent in population 1 and allele A_3 and A_4 equi-frequent in population 2. Although it seems that little if any genetic exchanges occur between these populations, F_{st} will only be 1/3, because complete fixation of four alleles could not occur in a sample of only two populations. *F_{st} therefore measures the extent to which the process of fixation has gone toward completion* (Wright, 1978).

To overcome the problems highlighted in the above examples, Wright advocates the use of not only F_{st}, but also of $\sigma^2_{p_s}$ and $p_t q_t$ to assess population differentiation. If a great deal of allele replacement seems to have occurred, so that the populations under scrutiny seem very differentiated, as in different species within a genus, the quantity of interest will be $\sigma^2_{p_s}$, whereas if there is little differentiation, F_{st} will be of interest in assessing population structure.

3.3 On estimating F-statistics

In the preceding section, F-statistics were defined in terms of population gene and genotype frequencies. These, however, cannot be readily obtained, even if the whole population is sampled, due to two sources of error: the first due to genetic sampling occurring in each generation (sampling of gametes from the parental array to produce the next generation); and the second, statistical sampling. Another source of sampling error exists, namely the parametric sampling, due to different mutation rate at different loci (Slatkin & Arter, 1991). In the model developed here, however, mutation is not of interest, because of the time scale at which we are working. Information about this third source of variation and its influence on the estimators of F-statistics can be found in Cockerham & Weir (1987) and Weir & Basten (1990).

Estimation of the population gene and genotype frequencies traces back to Levene

(1949). Since then, a lot of progress has been made and I will present two families of estimators that are widely used in the literature. The first was developed by Cockerham (1969, 1973) and Weir & Cockerham (1984) and takes account of the two sources of biases mentioned in the previous paragraph. The other, developed by Nei (1973, 1978) and Nei & Chesser (1983) takes only account of the statistical sampling. If our interest lies only on the population from which the sample was taken, Nei's approach could be justified, but if the intention is to use the F- statistics to compare structuring in the sample with other populations of the same species or with different species, the first method is to be preferred.

Measuring F-statistics is of little interest if no population parameter can be extracted. A third method for seeking appropriate estimators is developed which leads to an estimate of both the local and global effective size of the samples. This is an interesting parameter which may help in coming to an understanding of the level at which selection is acting in the framework of Wright's shifting balance theory (Wright, 1977, Chapter 13).

3.3.1 Cockerham's method (1969, 1973)

Cockerham(1969, 1973) approaches the problem of the estimation of the different F-statistics by mean of a hierarchical analysis of variance (ANOVA). The observational unit used is the gene (each and every gene, Cockerham, 1973). Let a_{kij} index the jth allele in the ith individual in the kth population. x_{kij} is defined as a measure of frequency such that $x_{kij} = 1$ if $a_{kij} = A$, and $x_{kij} = 0$ if $a_{kij} = \overline{A} \neq A$ (1969). Let the population frequency of A be $P(a_{kij} = A) = p$. The following model can be written (Cockerham, 1969)

$$x_{kij} = p + a_k + b_{ki} + w_{kij} \tag{3.23}$$

where the effects, all random, are a for groups, b for individuals and w for within individuals and have variances σ^2_a, σ^2_b and σ^2_w, respectively. The expectations of quadratics over classes of genes are:

$$\begin{aligned} E(x_{kij}x_{k'i'j'}) \; = & \\ & p^2 + p(1-p) = p^2 + \sigma^2 && \text{if} \quad k = k', i = i', j = j' \\ & p^2 + \overline{F}p(1-p) = p^2 + Cov_{ab} && \text{if} \quad k = k', i = i', j \neq j' \\ & p^2 + \overline{\theta}p(1-p) = p^2 + Cov_a && \text{if} \quad k = k', i \neq i' \\ & p^2 + Cov_g && \text{if} \quad k \neq k'. \end{aligned}$$

Therefore, $\overline{F}$ and θ are simply defined as a function of the covariances. f is the ratio $(\overline{F}-\theta)/(1-\theta)$. For uncorrelated groups, $Cov_g = 0$. Otherwise, it is the covariance between the least related genes, in the sense that they are furthest apart in the hierarchy (Cockerham, 1973). If this correlation is not zero, all the estimated statistics will be relative to it and could be redefined as $p'q' = (1-\theta_g)pq = \sigma^2$, $\theta' = (\theta-\theta_g)/(1-\theta_g)$ and $F' = (F-\theta_g)/(1-\theta_g)$. *This point stresses the importance of properly identifying the different level of structuring.* A discussion of the problem is found in Cockerham(1973) and is developed further in Chapter 4. It is sufficient to say here that, without modification of the basic model (equation 3.23), if there are isolates within subpopulations, F and θ can be estimated, but not f, whereas if there are subpopulations within areas and areas within populations, F and θ cannot be estimated but f can. Parametrically we have in terms of correlations (Cockerham,1973)

$$\begin{aligned}
\sigma^2 &= p(1-p)\\
Cov_{ab} &= \overline{F}p(1-p)\\
Cov_a &= \overline{\theta}p(1-p).
\end{aligned}$$

The correlations are related to the components of variance as follows:

$$\begin{aligned}
(1-\overline{F})p(1-p) &= \sigma_w^2\\
(\overline{F}-\overline{\theta})p(1-p) &= \sigma_b^2\\
\overline{\theta}p(1-p) &= \sigma_a^2
\end{aligned}$$

and

$$\sigma^2 = \sigma_w^2 + \sigma_b^2 + \sigma_a^2.$$

It is now necessary to estimate $F_{it}(F)$, $F_{st}(\theta)$ and $F_{is}(f)$ respectively from a hierarchical analysis of variance design. To do this, we simply need to construct the different sums of squares for the analysis of variance (e.g. Sokal and Rohlf,1982):

$$\begin{aligned}
SS_0 &= \textstyle\sum_{k,i,j} x_{kij}^2 &&= \textstyle\sum_k (2N_2+N_1) &&= 2DN\overline{p_k}\\
SS_1 &= \frac{\sum_{k,i}(\sum_j x_{kij})^2}{2} &&= \frac{\sum_k (4N_2+N_1)}{2} &&= 2DN\overline{p_k} - \frac{\sum_k N_1}{2}\\
SS_2 &= \frac{\sum_k(\sum_{i,j} x_{kij})^2}{2N} &&= \frac{\sum_k (2N_2+N_1)^2}{2N} &&= 2DN\overline{p_k^2}\\
SS_3 &= \frac{(\sum_{k,i,j} x_{kij})^2}{2DN} && &&= 2DN\overline{p_k}^2
\end{aligned}$$

where N_2 stands for the number of homozygotes AA and N_1 stands for the number of heterozygotes $A\overline{A}$. The differences of sums of squares follow is given by:

$$SS_{0-3} = 2DN\overline{p_k}\,\overline{q_k}$$

adding and subtracting $2DN\overline{p_k}^2$ to SS_{0-2} leads to:

$$\begin{aligned}
SS_{0-2} &= 2DN(\overline{p_k}\,\overline{q_k} - \sigma_{p_k}^2)\\
SS_{0-1} &= \frac{\sum_k N_1}{2}\\
SS_{1-2} &= 2DN(\overline{p_k}\,\overline{q_k} - \sigma_{p_k}^2) - \frac{\sum_k N_1}{2}\\
SS_{2-3} &= 2DN\sigma_{p_k}^2.
\end{aligned}$$

where σ_{pk}^2 is the *population variance* of the allele frequency. This parameterisation of Cockerham's equations will allow to find the relation between Wright's F-statistics and Cockerham's estimators.

The set of equations leads to Table 3.3 if for simplicity, we rewrite p for $\overline{p_k}$ and σ^2 for $\sigma_{p_k}^2$. The expressions for $\hat{F}$, $\hat{\theta}$ and $\hat{f}$ can be readily extracted from this table and give

$$\hat{F} = \frac{\sigma_a^2+\sigma_b^2}{\sigma_a^2+\sigma_b^2+\sigma_w^2} \tag{3.24}$$

$$\hat{\theta} = \frac{\sigma_a^2}{\sigma_a^2+\sigma_b^2+\sigma_w^2} \tag{3.25}$$

$$\hat{f} = \frac{\sigma_b^2}{\sigma_b^2+\sigma_w^2}. \tag{3.26}$$

On the other hand, if only allelic frequencies are available, one has to make the assumption that each of the samples are in Hardy-Weinberg equilibrium, and the only statistic that can be estimated is $\hat{\theta}$. The layout of the analysis of variance for this case is shown in Table 3.4. It is also the table that would be used in the case of data from haploids with $N_{hap} = 2N_{dip}$. In this case the expression for $\hat{\theta}$ will be :

$$\hat{\theta} = \frac{\sigma_b^2}{\sigma_{w2}^2+\sigma_b^2}. \tag{3.27}$$

These two tables are for 1 allele only. To get an estimate over alleles at a locus and over loci, one simply sums numerators and denominators.

If u indexes alleles and r loci:

$$\hat{F} = \frac{\sum_{r,u}(\sigma_{b_{ru}}^2+\sigma_{a_{ru}}^2)}{\sum_{r,u}\sigma_{\cdot ru}^2}, \tag{3.28}$$

$$\hat{\theta} = \frac{\sum_{r,u}\sigma_{a_{ru}}^2}{\sum_{r,u}\sigma_{\cdot ru}^2}, \tag{3.29}$$

Table 3.3: Hierarchical analysis of variance on allele frequencies when genotypic frequencies are available

Source of variation	Degrees of freedom	Sum of Squares	Mean Squares	Expected Mean Squares
Among Demes	$D-1$	$2DN\sigma^2$	$\frac{2DN\sigma^2}{D-1}$	$\sigma_w^2 + 2\sigma_b^2 + 2N\sigma_a^2$
Among individuals within demes	$D(N-1)$	$2DN(pq-\sigma^2)$ $-\frac{N_1}{2}$	$\frac{2DN(pq-\sigma^2)}{D(N-1)}$ $-\frac{N_1}{2D(N-1)}$	$\sigma_w^2 + 2\sigma_b^2$
Within individuals	DN	$\frac{N_1}{2}$	$\frac{N_1}{2DN}$	σ_w^2
Total	$2DN-1$	$2DNpq$	$\frac{2DNpq}{2DN-1}$	

where $\sigma^2_{\cdot ru}$ stands for the sum of the three variance components,

$$\hat{f} = \frac{\sum_{r,u} \sigma^2_{b_{ru}}}{\sum_{r,u}(\sigma^2_{b_{ru}} + \sigma^2_{w_{ru}})}. \tag{3.30}$$

These estimates will be unbiased in the sense that they are ratios of unbiased estimators. Other methods of averaging over alleles and loci have been investigated, but gave worse results than this simple weighted average (Weir & Cockerham, 1984). It should be noted that, although the relation between the three Fs stands for each allele, it does not for the combined estimators.

Confidence intervals for these estimates can be readily obtained by means of resampling techniques such as Jackknife and Bootstrap (Weir, 1990, pp. 137–143). The advantage of such techniques is that they do not depend on the assumption of normality and are easily implemented on a computer (cf. algorithms and Fortran source code in Weir, 1990). Randomisation tests (permutations of alleles within samples, between samples and permutations of multi-locus genotypes) can be carried out to test if f, F and θ respectively are significantly different from zero. These tests also allow to generate an empirical distribution of the different estimators under the null hypothesis.

Table 3.4: Analysis of variance on allele frequencies when genotypic frequencies are not available

Source of variation	Degrees of freedom	Sum of Squares	Mean Squares	Expected Mean Squares
Among Demes	$D-1$	$2DN\sigma^2$	$\frac{2DN\sigma^2}{D-1}$	$\sigma_{w2}^2 + 2N\sigma_a^2$
Within demes	$D(2N-1)$	$2DN(pq-\sigma^2)$	$\frac{2DN(pq-\sigma^2)}{D(2N-1)}$	σ_{w2}^2
Total	$2DN-1$	$2DNpq$	$\frac{2DNpq}{2DN-1}$	

Weir's program (1990) has been translated in Pascal and C and the code is given in appendix D. These randomisation tests will be further discussed in Chapter 4

Long (1986) refined Cockerham's approach by extending the diallelic system to a multi-allelic system by mean of multiple analysis of variance (MANOVA) rather than ANOVA and proposes an approximate test based on Wilk's Λ distribution.

3.3.2 Nei & Chesser's methods (1983)

Let p_{ki} be the frequency of allele A_i in the kth population and P_{kij} be the frequency of genotype A_iA_j in the kth population. Nei (1977) defined the fixation indices in the following way:

$$\begin{aligned} F_{is} &= 1 - H_o/H_s, & (3.31)\\ F_{it} &= 1 - H_o/H_t, & (3.32)\\ F_{st} &= 1 - H_s/H_t, & (3.33) \end{aligned}$$

where $H_o = 1 - \sum_i P_{ii}$, $H_s = 1 - \sum_i \overline{p_i^2}$ and $H_t = 1 - \sum_i \overline{p_i}^2$. P_{ii}, $\overline{p_i^2}$ and $\overline{p_i}^2$ are the respective weighted averages over populations if they are of different sizes. The problem is then to estimate H_o, H_s and H_t from the samples (Nei & Chesser, 1983). Let x_{ki} and X_{kii} be the sample frequency of allele A_i and genotype A_iA_i respectively. *It should be stressed that —although it is not mentioned anywhere in Nei & Chesser (1983)— even if the total population is sampled, it is still only one of the possible* *states of the genotypic array and therefore, estimators are to be used.* An account of the question of fixed versus random effect can be found in Weir (1990, pp. 136, 145).

An unbiased estimate $\hat{H}_o$ of H_o is just the number of homozygotes:

$$\hat{H}_o = 1 - \sum_{k,i} X_{kii}/D. \tag{3.34}$$

where D is the number of samples. x_{ki}^2, however, is not an unbiased estimate of $\overline{p_{ki}^2}$. Under the multinomial sampling of genotypes, we have (subscript k is dropped for brevity):

$$E(x_i^2) = Var(x_i) + [E(x_i)]^2 = E[X_{ii}^2 + X_{ii}(\sum_{i \neq j} X_{ij}) + (\sum_{i \neq j} X_{ij}/2)^2] \tag{3.35}$$

because

$$x_i^2 = (X_{ii} + \sum_{i \neq j} X_{ij}/2)^2$$

Equation 3.35 becomes:

$$\begin{aligned} E(x_i^2) &= P_{ii}^2 + P_{ii}(1 - P_{ii})/N \\ &\quad + P_{ii}(\textstyle\sum_{i \neq j} P_{ij}) - P_{ii}(\textstyle\sum_{i \neq j} P_{ij})/N \\ &\quad + (\textstyle\sum_{i \neq j} P_{ij})^2/4 + \textstyle\sum_{i \neq j} P_{ij}/4N - (\textstyle\sum_{i \neq j} P_{ij})^2/4N \\ &= p_i^2 + P_{ii}/N + \textstyle\sum_{i \neq j} P_{ij}/4N - p_i^2/N. \end{aligned}$$

where N is the sample size, presumed constant over samples. This leads to:

$$\hat{H}_s = \frac{N}{N-1}[1 - \sum_i \overline{x_i^2} - \frac{\hat{H}_o}{2N}] \tag{3.36}$$

where $\hat{H}_o$ is given by (3.34).

Similarly for the estimate of H_t, we get:

$$\hat{H}_t = 1 - \sum_i \overline{x_i}^2 + \frac{\hat{H}_s}{ND} - \frac{\hat{H}_o}{2ND} \tag{3.37}$$

where $\hat{H}_s$ and $\hat{H}_o$ are given by (3.36) and (3.34) respectively.

Now the different estimates of the fixation indices are:

$$\hat{F_{is}} = 1 - \hat{H}_o/\hat{H}_s \tag{3.38}$$

$$\hat{F_{st}} = 1 - \hat{H}_s/\hat{H}_t \tag{3.39}$$

$$\hat{F_{it}} = 1 - \hat{H}_o/\hat{H}_t. \tag{3.40}$$

3.3.3 A population genetics view

So far, we have been dealing with a statistical approach of the problem of estimating F-statistics. Biologists, however, often find it difficult to understand statistical papers, and I will attempt here to derive unbiased estimators of the F-statistics using concepts more familiar to the population geneticist and, more generally, to the population biologist.

When a sample is taken from a natural population (and I stress again here that the sample could consist of the whole population under investigation), two measures of genetic variability are to be estimated: the allelic frequency and the genotypic frequency. These two measures will be estimated from the same sample. For example, either allozyme methods or RFLP techniques are used to obtain genotypic frequencies, from which allelic frequencies will be inferred. This corresponds to sampling without replacement, that is, once one allele of the individual under scrutiny is known, there are only $2N - 1$ possible alleles for the second if we are dealing with a diploid population. This simple fact means that instead of sampling from a binomial distribution, we are sampling from a hyper-geometric distribution. This point was stressed by Levene(1949), Haldane (1954) and Gouyon (pers. comm.). Consider a sample of N individuals and, therefore, $2N$ alleles, where p is the frequency of allele A and $q = 1 - p$ is the frequency of allele $\overline{A}$. The sample will consist of $2Np$ copies of allele A and $2Nq$ of allele $\overline{A}$. The probability of a heterozygote is the probability of obtaining A once and only once from two draws:

$$Prob(A\overline{A}) = Prob(X = 1) = \frac{\binom{1}{2Np}\binom{1}{2Nq}}{\binom{2}{2N}}.$$

This can be rewritten:

$$Prob(X = 1) = \frac{4N^2pq}{N(2N-1)} = \frac{4Npq}{2N-1}.$$

Replacing Het_{obs} with this last quantity into the equation of F_{is} leads to:

$$F_{is} = 1 - \frac{4Npq}{(2N-1)2pq} = \frac{-1}{2N-1}. \tag{3.41}$$

That is, the expected value of F_{is} is negative, a point already stressed by Kirby(1973) and Cockerham(1973). To obtain an unbiased estimate, we need to subtract this quantity from the definition of F_{is} to give:

$$F_{is_g} = \frac{1 - \frac{Het_{obs}}{2Npq} + \left(\frac{1}{2N-1}\right)}{1 + \left(\frac{1}{2N-1}\right)}. \tag{3.42}$$

The denominator of this last equation is necessary to standardise the estimate over the range -1 to $+1$. If we consider that the uncorrected F_{is}, named F_{is_0} in the following, consists of two components, F_{is_g} (the value toward which F_{is_0} converges when N tends to ∞) and $-1/(2N-1)$, the expected value of F_{is} when N is finite. This leads to:

$$(1 - F_{is_0}) = (1 - F_{is_g})(1 + \frac{1}{2N-1}) \tag{3.43}$$

which can be rearranged to give equation 3.42.

This correction would be sufficient if the sample size N was to be equal to the sample effective size N_e (the size of an idealised population that would lose heterozygote or drift at the same rate as the observed one). If it is not the case, N needs to be replaced by N_e.

We have seen in Chapter 2 that the effective size of a population undergoing partial selfing is $N/(1+F_{is})$. Replacing N by this last expression leads to a converging recursive formula:

$$F_{is_t} = 1 - \frac{1 - F_{is_0}}{1 + \frac{1}{\frac{2N}{1+F_{is_{(t-1)}}} - 1}}. \tag{3.44}$$

Putting $X_t = -1/(\frac{2N}{1+F_{is_t}} - 1)$ leads to the following sets of equations:

$$\begin{aligned}
F_{is_0} &= 1 - \frac{Het_{obs}}{2Npq} \\
F_{is_1} &= \frac{F_{is_0} - X_0}{1 - X_0} \\
F_{is_t} &= \frac{F_{is_0} - X_{(t-1)}}{1 - X_{(t-1)}},
\end{aligned}$$

At equilibrium $F_{is_t} = F_{is_{t-1}}$ and we obtain:

$$\hat{F_{is}} = \frac{F_{is_0} - \hat{X}}{1 - \hat{X}}.$$

Replacing $\hat{X}$ by its value and rearranging leads to:

$$\hat{F_{is}} = \frac{(2N-1)F_{is_0} + 1}{2N - 1 + F_{is_0}}. \tag{3.45}$$

If $N = 1$, $F_{is_0} = -1$ unless there is no heterozygotes in the population, when it is undefined. Substituting 1 for N and -1 for F_{is_0} into the expression for $\hat{F_{is}}$ leads to an undefined expression.

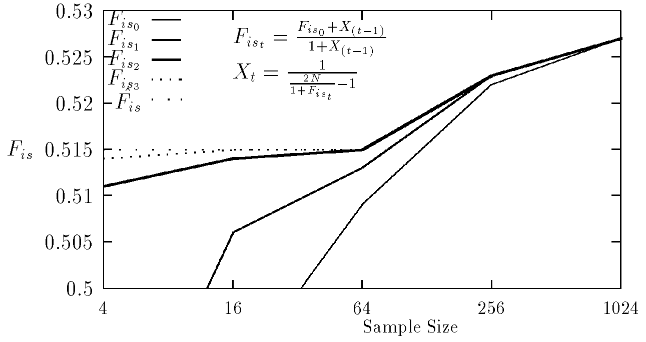

Figure 3.2: Behaviour of the family of F_{is}. Equilibrium is reached after only 3 iterations of the recursion. See text for details.

Equation 3.45 is the same as both Cockerham and Nei's estimator. It leads to the effective local sample size (both inbreeding and variance), which can be expressed as:

$$N_{el} = N_{eg}^{Selfv} = \frac{N}{1 + \hat{F_{is}}} \tag{3.46}$$

as was found in Chapter 2, equation 2.18. As expected, if there is 100% selfing in the population, N_{el} will be half the census size, because the rate of allele frequency drift, as well as the rate of loss of heterozygosity, will be twice as large as in a random mating population. In the absence of homozygotes, as in the case of overdominance with homozygotes being lethal, the local effective size will be infinite, a result to be expected, since there is no loss of heterozygotes or changes in allelic frequencies over generations.

The effects of the recursive correction are shown in Figure 3.2. Samples of different sizes were taken from a 2-dimensional stepping stone model composed of 64 demes of size 64,with 20% migration and 70% selfing. The different F_{is} are calculated for each sample size.

The effect of the corrections is obvious. F_{is_0} within a deme (for sample sizes below 64) increases as sample size increases, whereas $\hat{F_{is}}$ stays constant, with the other estimates being intermediate. This plotting technique will be used in Chapter 4 as a way of inferring the level of population structure.

When estimating F_{st}, as was the case for F_{is}, the sampling distribution of the variance of allele frequencies over populations will follow a hyper-geometric and not a binomial distribution. Once again two parameters will be estimated from the samples, the local allelic frequencies, p_k where k refers to the kth sample and $\overline{p_k}$, the global allelic frequency. The distribution will be the outcome of $2N$ draws without replacement in a sample of size $2DN$. The variance of such a distribution is given by the following equation:

$$Var(\mathcal{H}(2DN, 2N, 2DNp)) = \frac{D-1}{2DN-1}pq. \tag{3.47}$$

Replacing the numerator in the definition of F_{st} by this last expression leads to:

$$F_{st} = \frac{D-1}{2DN-1}. \tag{3.48}$$

An unbiased measure of F_{st}, *if the subdivisions were to be arbitrary subdivisions of a single panmictic unit* is given by:

$$F_{st_g} = \frac{\frac{\sigma^2_{p_k}}{\overline{\overline{p_k}\,\overline{q_k}}} - \left(\frac{D-1}{2DN-1}\right)}{1 - \left(\frac{D-1}{2DN-1}\right)}. \tag{3.49}$$

However, if they are not, it is necessary once again to correct the sample sizes to obtain an unbiased estimator. But we face a new problem because, under partial selfing, rates of allele frequencies drift and loss of heterozygosity are the same, providing that selfers are not territorial (the location in space of offspring is uncorrelated with that of the parents). However, in a subdivided population, these two parameters are different since subdivisions will lead to a faster rate of loss of heterozygosity, because individuals within populations are more related than individuals from the total, but to a slower rate of allele frequency drift, as is shown in Chapter 2 (equations (2.21) and (2.24)). We therefore need to consider these two approaches and we will see that the outcomes lead respectively to θ and G_{st}.

Correction for rate of loss of heterozygosity

We can calculate the global inbreeding effective size using (2.21) given in Chapter 2 (the global variance effective size if there is no territoriality). What needs to be corrected, however, is not N but D, the number of families (demes). As D appears twice in the expectation of F_{st} (3.48), we will have, putting

$$X_t = \frac{\frac{D}{1+(2N_{el}-1)F_{st_t}} - 1}{\frac{2DN_{el}}{1+(2N_{el}-1)F_{st_t}} - 1}$$

the following expressions:

$$\begin{aligned} F_{st_0} &= \frac{\sigma^2_{p_k}}{\overline{\overline{p_k}\,\overline{q_k}}} \\ F_{st_1} &= \frac{F_{st_0} - X_0}{1 - X_0} \\ F_{st_t} &= \frac{F_{st_0} - X_{(t-1)}}{1 - X_{(t-1)}}. \end{aligned}$$

At equilibrium, $F_{st_t} = F_{st_{(t-1)}}$, leading to:

$$\hat{F_{st}} = \frac{F_{st_0} - \hat{X}}{1 - \hat{X}}.$$

Replacing $\hat{X}$ by its value and rearranging leads to:

$$\hat{F_{st}} = \frac{(2DN_{el}-1)F_{st_0} - (D-1)}{(D-1)(2N_{el}-1) + (2N_{el}-1)F_{st_0}} \tag{3.50}$$

which is the same as Cockerham's θ. This leads to the global inbreeding effective size of the population, N^i_{eg}, as well as the global variance effective size if families (demes) are not territorial:

$$N^{Subd^i}_{eg} = N^{Rel^v}_{e} = \frac{DN_{el}}{1 + (2DN_{el} - 1)\hat{F_{st}}} = \frac{DN}{2N\hat{F_{st}} + (1 + \hat{F_{is}})(1 - \hat{F_{st}})} \tag{3.51}$$

which is also expression (2.28). When both $\hat{F_{is}}$ and $\hat{F_{st}}$ are 0, N^i_{eg} reduces to DN, whereas when $\hat{F_{st}} = 1$, $N^i_{eg} = \frac{D}{2}$. We can see here the analogy with F_{is}: when $N = 1$, $\hat{F_{is}}$ is undefined and the last expression reduces to the expression of the effective size when there is partial selfing.

Correction for rate of allelic frequency drift

In this case, the global variance effective size is used to correct the expected value of F_{st}. Using (2.24) of Chapter 2, applying it to both occurrences of D in (3.48) and using the now usual substitution:

$$X_t = \frac{\frac{D}{1-F_{st}} - 1}{\frac{2DN_{el}}{1-F_{st}} - 1}$$

we get:

$$\begin{aligned} F_{st_0} &= \frac{\sigma^2_{p_k}}{\overline{\overline{p_k}\,\overline{q_k}}} \\ F_{st_1} &= \frac{F_{st_0} - X_0}{1 - X_0} \\ F_{st_t} &= \frac{F_{st_0} - X_{(t-1)}}{1 - X_{(t-1)}}. \end{aligned}$$

At equilibrium, $F_{st_t} = F_{st_{(t-1)}}$, leading to:

$$\hat{F_{st}} = \frac{F_{st_0} - \hat{X}}{1 - \hat{X}}.$$

Replacing $\hat{X}$ by its value and rearranging leads to:

$$\hat{F_{st}} = \frac{(2DN_{el} - 1)F_{st_0} - (D-1)}{D(2N_{el} - 1) + (1 - F_{st_0})} \qquad (3.52)$$

which is the same as Nei's (1983) estimate, but different from that of Cockerham (1973). The expression of the global variance effective size can now be written:

$$N_{eg}^{Subd^v} = \frac{DN}{(1 + \hat{F_{is}})(1 - \hat{F_{st}})} \qquad (3.53)$$

which is the same as (2.26).

This last formula allows us to compare two systems that will lead to a similar genotypic composition of the total population, that is no heterozygotes: the first is 100% selfing in a single, non subdivided population of size N, the second is a 'random mating subdivided population', with each sub–population of size 1 and no migration between them. In the first case the effective global population size will be $N/2$, in the second, it will be infinite. This could be understood in terms of the variance of number of successful gametes: in the first case, the number of successful gametes is Poisson-distributed, whereas in the second case, this variance is 0, that is, every individual has one and only one offspring.

Figure 3.3 describes a similar situation. The different curves were obtained as follows:

- $N_e^{Self^v}$ is the average variance effective size estimated over 50 replicates —using (2.15)— of a population with the following parameters: $D = 1, N = 4096, s = 0.9$.

- $N_e^{Subd^v}$ is the average variance effective size estimated over 50 replicates of a population with the following parameters: $D = 4096, N = 1, m = 0.05$.

- $N_e^{Self}(x) = \frac{4096}{1+F_{st}(x)}$.

- $N_e^{Subd}(x) = \frac{4096}{1-F_{st}(x)}$.

$F_{st}(x)$ is calculated using:

$$F_{st}(x) = (1-m)^2 \left(\frac{1}{2N} + (1 - \frac{1}{2N})F_{st}(x-1) \right)$$

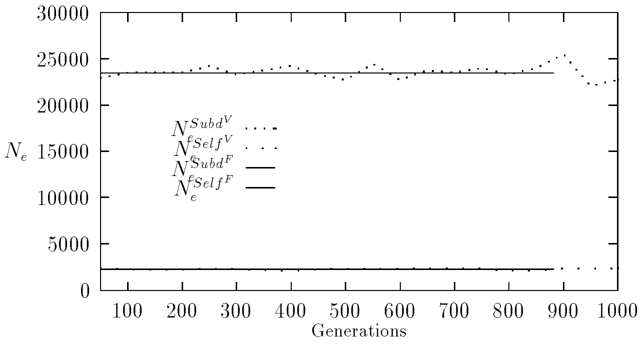

Figure 3.3: Comparison of selfing and subdivisions

with $N = 1$ and $m = 0.05$. $F_{is}(x)$ could have been used instead of $F_{st}(x)$, with

$$F_{is}(x) = \frac{s}{2}(1 + F_{is}(x-1))$$

and $s = 0.9$.

3.3.4 Summary of estimation procedures

Three different methods to obtain estimators of Wright's fixation indices have been derived, one based on an analysis of variance design, one based on the expectation of variances and one which gives a population genetics interpretation to the bias, leading to the effective size of both the local and global population. We need now to compare the two estimates of F_{st}, G_{st} and θ and find out if they are independent of both sample size and number of samples.

Before doing so, it will be useful to have formulae relating these different estimators to each other. From now on, Cockerham's estimates will be called respectively f and θ , Nei's G_{is} and G_{st}. The formula for F_{st} before recursion (equation (3.49)) will be called F_{st_g} because three of us, P.H. Gouyon, C.J. Gliddon and myself, originated it. Wright's basic formulae will keep their names. Table 3.5 summarises the results.

For F_{is}:

$$f = G_{is} = \frac{(2N-1)F_{is} + 1}{2N - 1 + F_{is}} \qquad (3.54)$$

For more than 2 alleles, F_{st} is the weighted average of the different F_{st_u}, where the

Table 3.5: Estimation procedures.

	F_{is}	F_{st}	
Infinite pop.	$F_{is_0} = 1 - \frac{Het_{obs}}{2pq}$	$F_{st_0} = \frac{\sigma_p^2}{\bar{p}\,\bar{q}}$	
Finite pop. R.M., no subd.	$F_{is_g} = \frac{F_{is_0} + \frac{1}{2N-1}}{1+\frac{1}{2N-1}}$	$F_{st_g} = \frac{F_{st_0} - \frac{D-1}{2DN-1}}{1-\frac{D-1}{2DN-1}}$	
No R.M., subd.	N (D) needs to be replaced by $N_e(D_e)$		
		No territoriality	Territoriality
Eff. size	$N_{el} = \frac{N}{1+\hat{F}_{is}}$	$D_e^{NT} = \frac{D}{1+(2N_{el}-1)\hat{F}_{st}}$	$D_e^T = \frac{D}{1-\hat{F}_{st}}$
$\hat{X}$	$\hat{X}_{is} = -\frac{1}{2N_{el}-1}$	$X_{st}^{\hat{N}T} = \frac{D_e^{NT}-1}{2D_e^{NT}N_{el}-1}$	$\hat{X}_{st}^T = \frac{D_e^T-1}{2D_e^TN_{el}-1}$
$\hat{F}$	$\hat{F}_{is} = \frac{F_{is_0}-\hat{X}_{is}}{1-\hat{X}_{is}}$	$F_{st}^{\hat{N}T} = \frac{F_{st_0}-X_{st}^{\hat{N}T}}{1-X_{st}^{\hat{N}T}}$	$\hat{F}_{st}^T = \frac{F_{st_0}-\hat{X}_{st}^T}{1-\hat{X}_{st}^T}$
	$\hat{F}_{is} = \frac{(2N-1)F_{is_0}+1}{2N-1+F_{is_0}}$	$\theta = \frac{(2DN_{el}-1)F_{st_0}-(D-1)}{(D-1+F_{st_0})(2N_{el-1})}$	$G_{st} = \frac{(2DN_{el}-1)F_{st_0}-(D-1)}{D(2N_{el-1})+1-F_{st_0}}$
Global eff. size	$N_{el} = \frac{N}{1+\hat{F}_{is}}$	$N_{eg}^{NT} = \frac{DN}{2N\theta+(1+\hat{F}_{is})(1-\theta)}$	$N_{eg}^T = \frac{DN}{(1+\hat{F}_{is})(1-G_{st})}$

Table 3.6: Functional analysis of $\hat{F}_{is}, \theta$ and G_{st}

	$\hat{F}_{is}$	θ	G_{st}
$\mathcal{D}$	For $x \in [-1,1], N > 1$ $F(x) \in [-1,1]$	For $x \in [0,1], D > 1$ $F(x) \in [\frac{-1}{2N-1}, 1]$	For $x \in [0,1], D > 1$ $F(x) \in [\frac{-(D-1)}{D(2N-1)+1}, 1]$
$F'(x)$	$\frac{(2N-1)^2-1}{(2N-1+x)^2}, > 0 \,\forall\, x$	$\frac{2DN(D-1)}{(D-1+x)^2(2N-1)}, > 0 \,\forall\, x$	$\frac{2D^2N(2N-1)}{(D(2N-1)+1-x)^2}, > 0 \,\forall\, x$
Sign	$- \Rightarrow x < \frac{-1}{2N-1}$ $0 \Rightarrow x = \frac{-1}{2N-1}$ $+ \Rightarrow x > \frac{-1}{2N-1}$	$- \Rightarrow x < \frac{D-1}{2DN-1}$ $0 \Rightarrow x = \frac{D-1}{2DN-1}$ $+ \Rightarrow x > \frac{D-1}{2DN-1}$	
$\lim_{N\to\infty}$	x	$\frac{Dx}{D-1+x}$	x
$\lim_{D\to\infty}$		$\frac{2Nx-1}{2N-1}$	$x - \frac{1}{2N}$

weight is $p_u q_u$.

$$F_{st_g} = \frac{(2DN_{el}-1)F_{st}-(D-1)}{D(2N_{el}-1)} \tag{3.55}$$

$$G_{st} = \frac{(2DN_{el}-1)F_{st}-(D-1)}{D(2N_{el}-1)+(1-F_{st})} \tag{3.56}$$

$$\theta = \frac{(2DN_{el}-1)F_{st}-(D-1)}{(D-1)(2N_{el}-1)+(2N_{el}-1)F_{st}} \tag{3.57}$$

where N_{el} is defined as $N/(1+f)$. The expression of each as a function of each other is also of interest:

$$\theta = F_{st_g}\frac{D}{D-1+F_{st}} \tag{3.58}$$

$$\theta = G_{st}\frac{D}{D-1+G_{st}} \tag{3.59}$$

$$G_{st} = \frac{F_{st_g}(D-1)}{D-1+F_{st}-F_{st_g}} \tag{3.60}$$

3.4 Comparison of θ and G_{st}

3.4.1 The functions $\hat{F}_{is}(F_{is}), \theta(F_{st})$ and $G_{st}(F_{st})$

A first step in understanding the differences between θ and G_{st} and their relation to F_{st} consists in studying them as a function of F_{st}. Plots as a function of F_{st} for different combinations of sample size N and number of samples D are found in Figure 3.5–3.8. For completeness, a plot of $\hat{F}_{is}$ as a function of Wright's F_{is} is given in Figure 3.4. Table 3.6 summarises the functional analysis of $\hat{F}_{is}, \theta$ and G_{st}. The

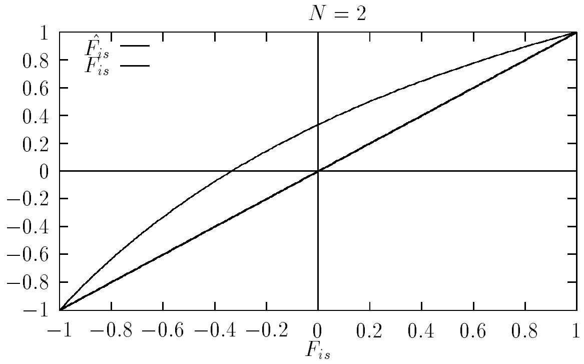

Figure 3.4: $\hat{F}_{is}$ as a function of Wright's F_{is}

three functions are continuously increasing over their domain of definition, $\mathcal{D}$ (positive derivative). They cross the x-axis when x is equal to its expectation in a finite sample, under random-mating and no subdivisions. The three functions converge to Wright's F-statistics when N and D tend to ∞. More interesting is the behaviour of θ and G_{st} when *either* D or N tends to ∞. As N tends to ∞, G_{st} tends to Wright's $F_{st} \forall D$, whereas the expression of θ still depends on D (Table 3.6). As D tends to ∞, both the expression of θ and G_{st} depend on N, the only difference between the two being a -1 in the denominator for θ. Other observations can be made:

- Both estimators differ from F_{st} for low sample sizes and number of demes sampled.
- Both can be negative (this needs stressing, Weir & Cockerham's estimator is not the only one leading to negative estimates of F_{st}).
- They are equal to F_{st} when $F_{st} = 1$.
- Increasing the sample size will lower the intersection with the x-axis, because the allelic frequencies per sample is more accurate with an increased number of individuals per sample. On the other hand, increasing the number of samples without increasing their size has no effect on the individual $p'_k s$. Therefore, if negative values of either estimator are to be avoided, increasing the sample size is needed. *What is meant by a negative estimate of either θ or G_{st} is that the population under investigation display less variation in allelic frequencies between samples than that which is expected just by chance.*

- An increase in the number of samples lessens the difference between G_{st} and θ (Table 3.6 & Figure 3.7).
- Increasing the sample size is sufficient to reduce the discrepancy between G_{st} and F_{st}, (Table 3.6 & Figure 3.6), whereas it has little effect on θ.
- $G_{st} \leq F_{st}, \forall D, N$. If $(D-1) > (2N-1)$, $\theta \leq F_{st}$. The latter may be found when expansive molecular techniques (sequences...) are used.
- θ is defined in the interval $[-1, 1]$ when $N = 1$, the interval for F_{is}. As $\hat{F}_{is}$ is undefined for $N = 1$, we see that the only fixation index that can be estimated in this case is θ.
- The absolute value of θ is always larger than the absolute value of G_{st}.

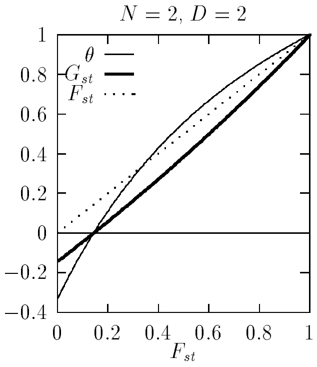

Figure 3.5: Estimators of F_{st} as a function of F_{st}

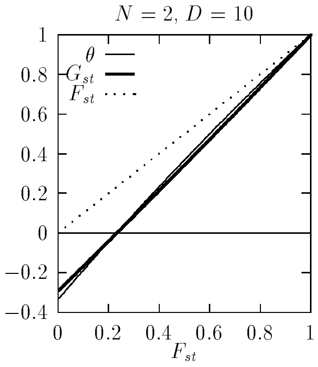

Figure 3.7: Estimators of F_{st} as a function of F_{st}

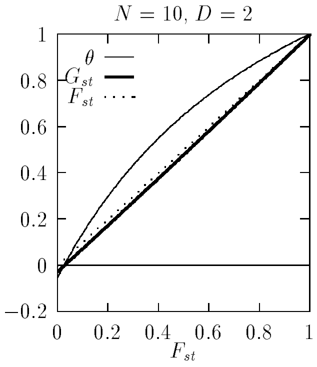

Figure 3.6: Estimators of F_{st} as a function of F_{st}

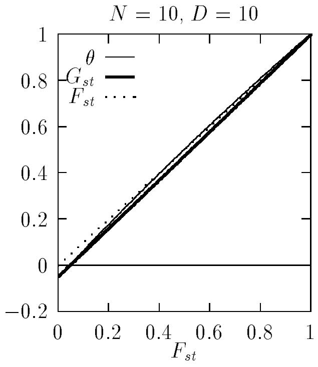

Figure 3.8: Estimators of F_{st} as a function of F_{st}

3.4.2 Experimental design

In order to assess the quality of the two estimators in respect of their behaviour under different sampling strategies, I will consider an adaptation of the experimental design described in Slatkin & Barton (1989). The different population structures were simulated using the model described in Chapter 2. The difference between this model and Slatkin's is that at generation 0, rather than having a completely monomorphic population, the population is as polymorphic as possible ($2DN$ unique alleles if there is DN individuals). There is therefore no need for mutation as long as the number of generations is not too large (≤ 10000 depending on the amount of migration and the variance effective size, D_eN_e).

Samples of varying sizes and of varying number were taken at random from an island model of population (gametic cloud) and a 2-dimensional stepping stone model. In Slatkin & Barton (1989), sampling was at random with respect to individuals and demes for the island model, but was a function of a parameter k representing the spacing between demes for the stepping stone model. In this design, sampling is at random over the total population for the stepping-stone model to allow an investigation of the dependance of θ and G_{st} on sampling strategy. That is, do the estimates of θ and G_{st} differ if $5, 10, 20$ or 50 demes are sampled, at random, from the total population. As there is geographical structuring in a stepping-stone model, sampling for different k values will lead to different estimates (cf. section on the stepping-stone model in Chapter 2).

For each model of population structure, deme size and number of demes were fixed at 64, for a grand total of 4096 individuals. Three levels of migration were considered, $0.005, 0.05$ and 0.1, leading to an Nm product of $0.32, 3.2$ and 6.4 respectively. These values for migration were chosen so that the product Nm lies on each side of the threshold 1 (If $Nm \gg 1$, the population behaves as effectively panmictic). Two levels of selfing were considered, 0% and 90%, corresponding to a typical outcrosser and a typical selfer. Fifty replicates were run until they reached equilibrium and the following sampling strategy was adopted:

	Number of samples			
Numb. of Ind. per sample	5	10	20	50
5	50	50	50	50
10	50	50	50	50
20	50	50	50	50
50	50	50	50	50

Figures 3.9—3.12 show the results of the effect of the number of samples. Each point corresponds to the average of 200 data points. The error bars are the 95% confidence interval (CI) of the mean. In each case we can see that θ seems to be the same for all sampling schemes, whereas G_{st} increases asymptotically toward θ as the number of samples increases. In fact, leaving out the case of 5 samples — in the third graph of figure 3.10 and the first graph of figure 3.12, the 95% CI of θ for 5 samples do not overlap with the other CI — there is no difference in all estimates for θ (overlapping CI), whereas 95% CI of the mean for 10 and 20 samples never overlaps for G_{st}. Therefore G_{st} is not an unbiased estimator of F_{st} with respect to the number of samples. We can also see that 95% CI of G_{st} and θ do not overlap.

To test if θ is an unbiased estimator of F_{st}, we need to compute the expected value of F_{st}. For the island model, it can be calculated using (2.29):

Model	F_{is}	F_{st}
$m = 0.005, s = 0.0$	0.0	0.437
$m = 0.005, s = 0.9$	0.82	0.585
$m = 0.05, s = 0.0$	0.0	0.067
$m = 0.05, s = 0.9$	0.82	0.116
$m = 0.1, s = 0.0$	0.0	0.032
$m = 0.1, s = 0.9$	0.82	0.053

The graphed results for θ fit very well with the values in the table and, as the number of sample increases, G_{st} converges toward the values in the above table.

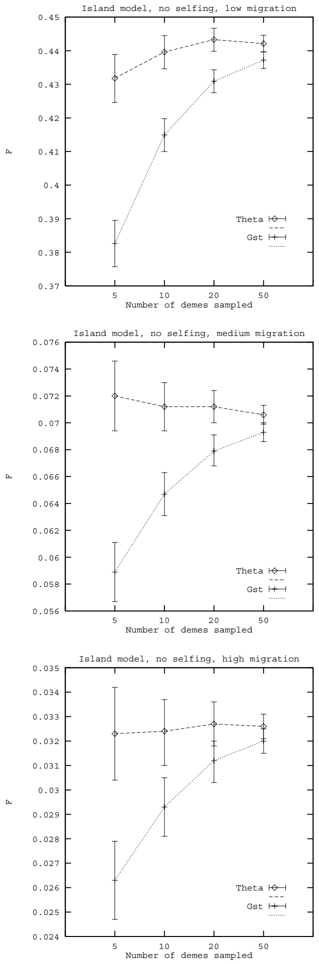

Figure 3.9: Estimators of F_{st} as a function of number of samples

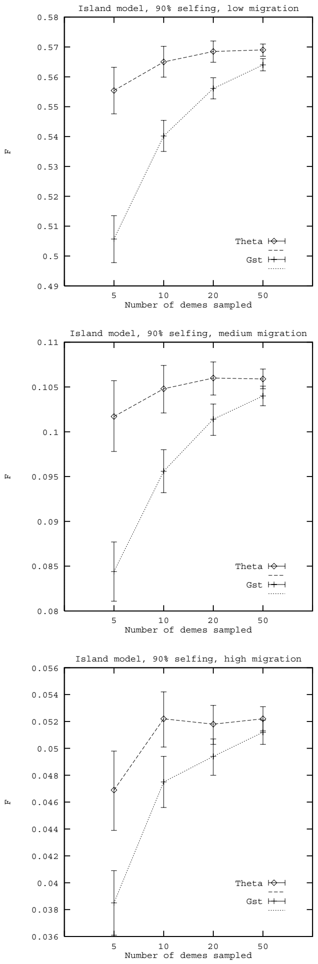

Figure 3.10: Estimators of F_{st} as a function of number of samples

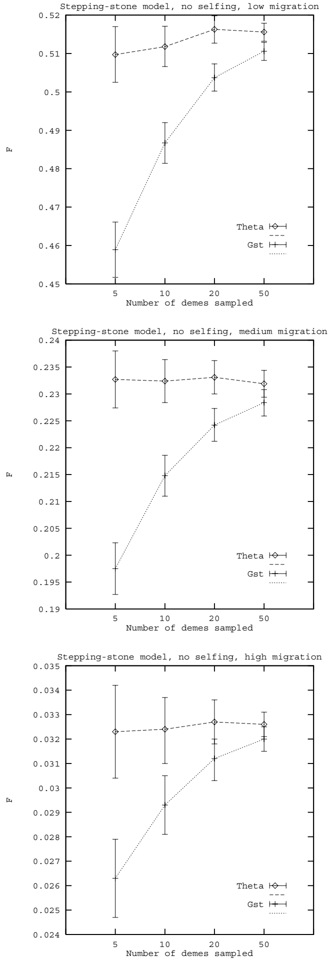

Figure 3.11: Estimators of F_{st} as a function of number of samples

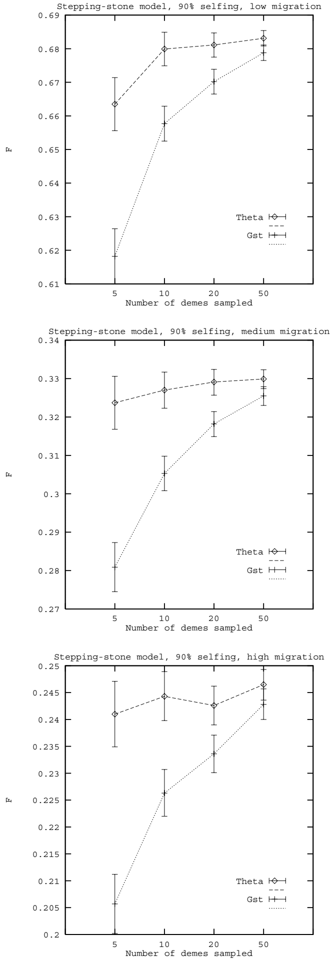

Figure 3.12: Estimators of F_{st} as a function of number of samples

Figures 3.13—3.16 focus on the effect of the number of individuals per sample. There is no sign of convergence of θ and G_{st} with an increasing number of individuals per sample, but both estimators remain fairly constant. The 95% CI of the means of θ and G_{st} do not overlap. Once again, the values for θ are in close agreement with the data in the table of expected values of F_{st} in an island model, but G_{st} is consistently lower. This could have been guessed from Table 3.6, where we saw that G_{st} converges toward F_{st} when N tends to ∞, for all D. The 95% CI of the mean does not seem to decrease significantly as the number of individuals per sample increases. For a given migration and selfing level, the values in a stepping-stone model of θ and G_{st} are higher than in an island model. This is to be expected, the input migration in a stepping-stone model being larger than the effective migration. This trend is stronger for large migration and is enhanced by selfing (which reduces the local effective size).

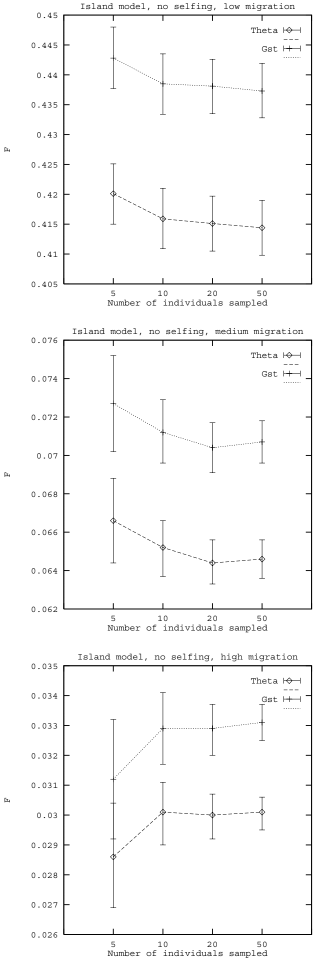

Figure 3.13: Estimators of F_{st} as a function of number of individuals per sample

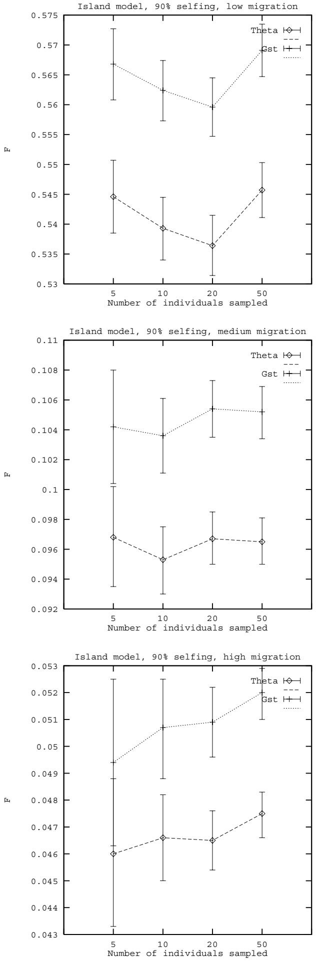

Figure 3.14: Estimators of F_{st} as a function of number of individuals per sample

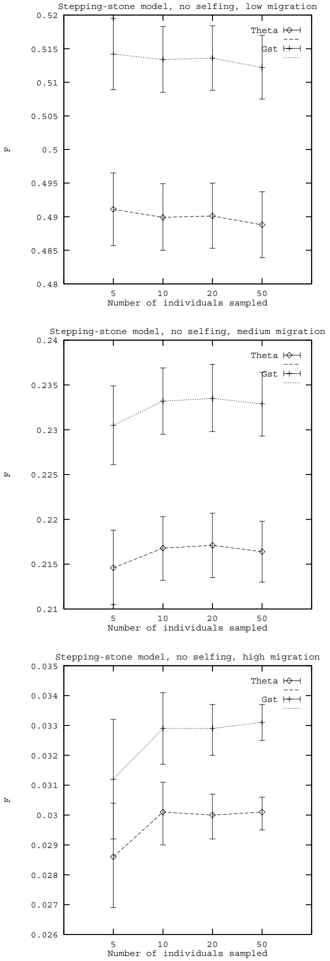

Figure 3.15: Estimators of F_{st} as a function of number of individuals per sample

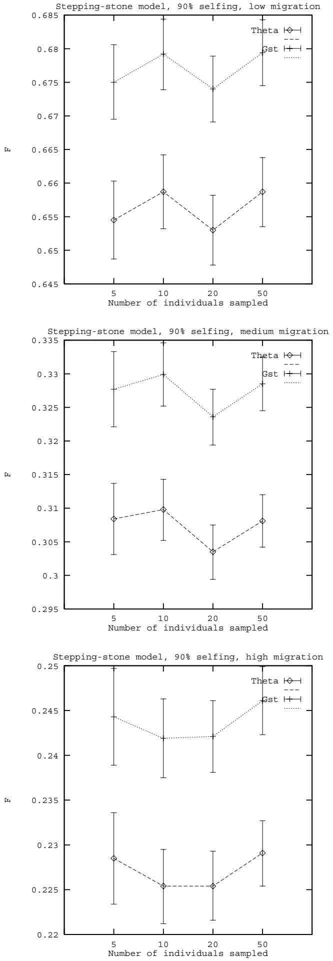

Figure 3.16: Estimators of F_{st} as a function of number of individuals per sample

Up to now, the focus has been on each effect separately. However, the number of samples and the number of individuals per sample may strongly interact. It is therefore necessary to carry out a statistical analysis that takes the interaction into account, to check that the preceding results are not flawed.

A first analysis is aimed at examining whether taking different numbers of samples and numbers of individuals per sample from the *same population* affect estimates of $\hat{F_{is}}$, θ and G_{st}. This experiment is designed to eliminate the variance due to genetic sampling, which adds an unnecessary level of noise.

Fifty replicates of each set of the parameters described below were run, and each of these replicates were then independently sampled using the 16 different sampling strategies. The number of individuals per sample and the number of samples are fixed effects, but the effect of replicates — which has to be taken into account because each replicate is used for all treatments — is random (Sokal & Rohlf, 1981). In summary, we have a 3-way mixed factorial design with no repetitions. We have to assume that the 3-way interaction is non-significant (additivity of the different effects) and will test the three 2-way interactions against the 3-way interaction, and the two fixed effects, number of samples and number of individuals per sample, against [number of samples*replicate] and [number of individuals*replicate] respectively. The effect of replicates is of no interest and could not be tested (no exact F-test can be calculated). The codes in the first column of table 3.7 have the following meanings:

- IS: Island model, SS: stepping-stone model
- LM: 1% migration, HM: 10% migration
- NS: no selfing, S, 90% selfing
- EG: generation 25, LG: generation 150

The early and late generation were chosen to mimic a non-equilibrium (half-way to equilibrium) and an equilibrium situation. Using (2.38) on an island model with 1% migration and no selfing, generation 25 correspond to the half-way, whereas generation 150 is at equilibrium. For the other island model patterns, equilibrium is reached faster (selfing and high migration speed up the process). Although it is impossible to predict analytically the time to equilibrium in a stepping-stone, it can be done graphically and it was checked that values of the estimators of F_{st} had reached a plateau.

Table 3.7: 3-way mixed factorial design for the effect of number of samples, number of individuals per sample and replicates (random).

Population type	$\hat{F_{is}}$			θ			G_{st}		
	NSP	NIND	INT	NSP	NIND	INT	NSP	NIND	INT
IS,LM,NS,EG	NS	NS	NS	NS	NS	NS	***	NS	NS
IS,LM,NS,LG	NS	NS	NS	NS	NS	NS	***	NS	NS
IS,LM, S,EG	NS	NS	NS	NS	NS	***	***	NS	***
IS,LM, S,LG	NS	NS	NS	NS	NS	NS	***	NS	NS
IS,HM,NS,EG	NS	NS	NS	**	NS	**	***	NS	**
IS,HM,NS,LG	NS	NS	NS	NS	NS	NS	***	NS	NS
IS,HM, S,EG	NS	NS	NS	NS	NS	NS	***	NS	NS
IS,HM, S,LG	NS	NS	NS	NS	NS	NS	***	NS	NS
SS,LM,NS,EG	NS	NS	NS	NS	NS	NS	***	NS	NS
SS,LM,NS,LG	NS	NS	NS	NS	NS	NS	***	NS	NS
SS,LM, S,EG	NS	NS	NS	NS	NS	NS	***	NS	NS
SS,LM, S,LG	NS	*	**	NS	NS	NS	***	NS	NS
SS,HM,NS,EG	NS	NS	NS	NS	NS	***	***	NS	***
SS,HM,NS,LG	NS	NS	NS	NS	NS	NS	***	NS	NS
SS,HM, S,EG	NS	NS	**	NS	NS	NS	***	NS	NS
SS,HM, S,LG	NS	NS	NS	NS	NS	*	***	NS	*

As we are not interested in the effect of replicates or the two-way interaction containing it (most of the 2-way interactions with replicates were non significant, appendix B), only the effects of number of individuals per sample, number of samples and their interactions are summarised in Table 3.7. The proportion of the variance explained by the model (R^2) ranges from 41% to 60% (appendix B), showing quite a good fit with the model. The first striking feature of this table is that there is always a highly significant effect of the number of samples on G_{st} ($P < 0.001$). G_{st} is therefore not an unbiased estimator of F_{st}, confirming the results of the graphical analysis. The second feature is that the interaction, while not significant in the majority of the cases, is significant in 25% of the cases for estimators of F_{st} and 12.5% of the cases for $\hat{F_{is}}$. One could suspect that the 3-way interaction and the 2 two-ways containing pep could be pooled together, as they are not significant (Sokal & Rohlf, 1981), but a quick look at the ANOVA in appendix B tells us that it is not necessary, the 3 MS [*nsp*rep, nind*rep, Error*] being of the same order of magnitude. We cannot, however, conclude that there is no additivity of the individual effects, for the interaction is not significant in the majority of the cases. The simplest explanation is that the data are heteroscedastic and not normally distributed, rendering the Type I error (the probability of rejecting the null hypothesis when it is true) larger than it should be.

The effect of the number of individuals per sample for both estimators of F_{st} is never significant, nor the effect of the number of samples for $\hat{F}_{is}$.

The effect of number of individuals per sample is significant in one case for $\hat{F}_{is}$ and the effect of the number of samples in one case for θ, leading to an acceptable Type I error of 6.25%.

Last, but not least, the 3 estimators are not affected by equilibrium or non equilibrium situations, levels of selfing or levels of migration.

We can now turn to a design that includes the effect of genetic sampling, for this effect will always be there when estimating F-statistics from natural populations. Figures 3.9 to 3.12 showed us that the 95% confidence interval for either estimator decreases as the number of samples increases. This will render the analysis of the data set using conventional parametric test such as the analysis of variance (ANOVA) very difficult, because one of the condition of application of ANOVA is homoscedasticity (Sokal & Rohlf, 1981). This condition is obviously not met here and we need to find a non parametric equivalent: for a one way ANOVA, this test is the Kruskall-Wallis test (Kruskall & Wallis,1952). In such a test, instead of working with the raw data, the data are ranked and the analysis of variance is carried out on the ranked data. In our case, we have two crossed factors (the number of individuals per sample and the number of samples), both being fixed effects. Friedman's test deals with two factors, but one of them is random. We therefore need a generalisation of the Kruskall-Wallis test for two fixed crossed factors and it can be found in Scheirer, Ray & Hare (1976) where the partitioning of the variance components is applied to rank rather than sum of squares. Contrasts could even be applied, but this is not the purpose of this experiment, as we are not really interested in comparing the effect of having 5 samples against the effect of having 50, but we want a general idea about the effect of increasing the number of samples.

640 replicated populations for each set of parameters were generated, and subsets of 40 were assigned at random to each sampling strategy. A MINITAB macro was written (Appendix C) and the results are presented in Table 3.8.

Table 3.8: 2-way Kruskall-Wallis with 40 repetitions per treatment.

Population type	$\hat{F}_{is}$			θ			G_{st}		
	NSP	NIND	INT	NSP	NIND	INT	NSP	NIND	INT
IS,LM,NS,EG	NS	NS	NS	NS	NS	NS	***	NS	NS
IS,LM,NS,LG	NS	NS	NS	**	NS	NS	***	NS	NS
IS,LM, S,EG	NS	NS	NS	NS	NS	NS	***	NS	NS
IS,LM, S,LG	NS	NS	NS	NS	NS	NS	***	NS	NS
IS,HM,NS,EG	*	NS	NS	*	NS	NS	***	NS	NS
IS,HM,NS,LG	NS	NS	NS	NS	NS	NS	***	NS	NS
IS,HM, S,EG	NS	NS	NS	NS	NS	NS	***	NS	NS
IS,HM, S,LG	NS	NS	NS	NS	NS	NS	***	NS	NS
SS,LM,NS,EG	NS	NS	NS	NS	NS	NS	***	NS	NS
SS,LM,NS,LG	NS	NS	NS	NS	NS	NS	***	NS	NS
SS,LM, S,EG	NS	NS	NS	NS	NS	NS	***	NS	NS
SS,LM, S,LG	NS	NS	NS	NS	NS	NS	***	NS	NS
SS,HM,NS,EG	NS	NS	NS	NS	NS	NS	***	NS	NS
SS,HM,NS,LG	NS	NS	NS	NS	NS	NS	***	NS	NS
SS,HM, S,EG	NS	NS	NS	NS	NS	NS	***	NS	NS
SS,HM, S,LG	NS	NS	NS	NS	NS	NS	***	NS	NS

The results are self-explanatory: in most instances, $\hat{F_{is}}$ and θ are unbiased (non significance of the effects of number of samples, number of individuals and interaction), whereas the effect of the number of samples on G_{st} is always highly significant ($P < 0.001$), confirming the two previous analyses. The interaction in all cases is non significant, as is the effect of the number of individuals per samples.

The Type I error seems however, to be slightly higher than 5%. This means that, when comparing two samples, we will find them significantly different one from another when they are not, with a higher probability than 5%.

The added effect of genetic sampling does not impair the results of the previous analysis, $\hat{F_{is}}$ and θ are unbiased estimators of F_{is} and F_{st} respectively, whereas G_{st} is biased by the number of samples.

3.5 Discussion and conclusions

A quick survey of the literature in population biology will show how widely the tools described in this chapter are used. It is astonishing to see how much has been written on the subject of F-statistics, without the reaching of a consensus on which sets of estimators are to be used to provide true, unbiased estimates. The ongoing polemic between Nei on the one hand (Nei, 1977, 1986, 1987; Nei & Chesser, 1983) and Cockerham and Weir on the other (Cockerham, 1969, 1973; Weir and Cockerham, 1984; Cockerham & Weir, 1986, 1987; Weir, 1990) does not seem to help the researcher in population biology to find the appropriate set of tools for his or her problem (θ and G_{st} are found in equal proportion in the literature, with no statements in general as to why one set of estimators has been chosen rather than the other). Even more surprising is the number of scientific publications using computer packages such as BIOSYS-1 (Swofford & Selander, 1981), which do not provide unbiased estimators of F-statistics. BIOSYS-1, in particular, uses the definitions of F-statistics of Nei (1977), which are not corrected for small sample sizes (estimation of H_o, H_s and H_t are not considered in this paper, but are presented in Nei & Chesser (1983)).

In their comparison of indirect estimators of gene-flow, Slatkin & Barton (1989) used the definition of G_{st} given in Nei's (1973) paper (Slatkin & Barton, 1989, p1356, equations 9a,b,c), while using Weir & Cockerham's (1984) definitions. This is slightly unfair, since Nei's paper does not address the question of estimation and a more appropriate analysis would have compared Weir & Cockerham (1984) with Nei & Chesser (1983). The relationship between G_{st} and θ given in Slatkin & Barton (1989) is for the estimators given in Weir & Cockerham (1984) and Nei & Chesser (1983) (in this relationship, if θ is negative, then G_{st} is negative, whereas Slatkin & Barton claim that G_{st} is always positive. This is true for the definition of G_{st} in Nei (1973, 1977), but not for Nei & Chesser (1983)).

Slatkin & Barton (1989) also find that θ gives an overestimate of Nm when Nm is large and suggest that this is so because θ is unbiased (G_{st}, on the other hand, always gives underestimates). This discrepancy between estimated and true Nm is more likely to come from the relation between F_{st} and Nm: it has been shown (Chapter 2) that the expression $F_{st} = 1/(4Nm+1)$ is only an approximation, that relies on m being small and N being large. These assumptions do not seem to hold true for $4Nm > 30$, as is shown on Figure 3.17, where both the exact and approximate formulae for the estimation of Nm from F_{st} are shown, for $N = 25$, as in Slatkin & Barton (1989). It is then possible to calculate what will be the approximate Nm value when the exact Nm is 51.2 and this gives 160 ($F_{st} = 0.0062125$). This is still lower that the estimate found by Slatkin & Barton (289 for the infinite allele model, Table 1, p 1360), but of the right order of magnitude (the coefficient of variation of θ is very large).

θ and G_{st} seem to behave similarly for low Nm values. This is not surprising, since the number of samples as well as the sample size are quite large. Indeed Figure 3.8 shows that there is little differences between θ, G_{st} and Wright's F_{st} with 10 samples of 10 individuals. For migration of 0.001 and 0.01, the inferred Nm is always slightly larger than its expectation, for θ as well as G_{st}. Figure 3.17 shows that the approximate estimation is always larger than the exact estimation, a finding that corroborates the results of Slatkin & Barton (1989). It should also be noted that, as there is an inverse relation between F_{st} and Nm, the closer the estimators of F_{st} are to 0, the larger will be the differences in Nm.

The differences between Nei's approach and Weir & Cockerham's revolves around models of fixed versus random effects. Nei considers that the species being surveyed is unique and that there is therefore no need to estimate the global allele frequencies (Nei, 1986; Cockerham & Weir, 1987; Weir, 1990). On the other hand, Weir & Cockerham (1984) point out that the observed allelic frequencies are the results of genetical sampling over generations as well as statistical sampling. That is, even if the total population is sampled, it is still necessary to estimate the global allele frequencies, because it is only one of the possible outcomes of the genetic sampling

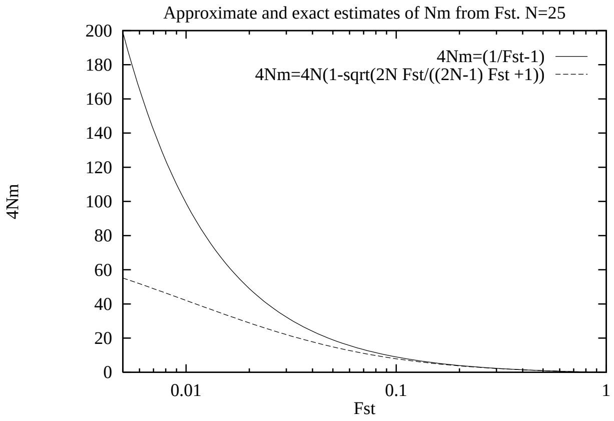

Figure 3.17: Discrepancies between the approximate and the exact estimation formula for Nm.

process that the species is undergoing.

An important finding in this work is that G_{st} depends upon the number of samples, whereas θ does not. This finding means that comparison of gene-flow estimates based on G_{st} from different samples is not reasonable. The number of samples used here is of the order of that which is found in the literature (5 to 50). Althought Nei (1986) suggested to use a correction on G_{st} to account for the effect of the number of samples, this estimator of F_{st} is very seldom found in the literature, probably because Nei (1986) does not give an explicit formula (the correction is given for D_{st}, one component of G_{st}).

Another point of interest concerns the value that should be assigned to the estimators of F_{st} when the samples are completely monomorphic. Nei & Chakravarti (1977) and Nei (1986) prefer to define the estimator as 0, while Weir & Cockerham (1984) suggest that it be left undefined. As the amount of polymorphism detected depends on the technique used (gel electrophoresis of isosymes, Random Fragment Length Polymorphism, Variable Number of Tandem Repeats such as minisatellites and microsatellites, Randomly Amplified Polymorphic DNA), one could find an F_{st} of 0 using a technique with a low power of resolution such as gel electrophoresis of isozymes, whereas another technique could give a completely different result (Cockerham, 1984; Cockerham & Weir, 1987). For this very reason, it seems logical to follow

Weir & Cockerham rather than Nei.

The approach used here for deriving unbiased estimates of Wright's F-statistics should help to clarify the circumstances in which it is preferable to use one or the other estimators. As Nei (1986) pointed out, he is interested in the degree of gene differentiation among populations rather than in the coefficient of inbreeding or coancestry within populations in which Weir & Cockerham (1984, p 1358) are interested. The two families of estimators can be derived from a unified approach that properly identifies the distribution generated by the sampling strategy as an hyper-geometric rather than a binomial. This finding allows the derivation of unbiased estimators of F_{is} and F_{st} providing that the null hypotheses are true (random mating and no subdivisions) and leads to the estimators F_{is_g} and F_{st_g}. In no circumstances, however, should these estimators be used, since there is no reason, *a priori*, to accept the null hypotheses. To obtain unbiased estimators if the null hypotheses are not true, it is necessary to formulate alternative hypotheses. The alternative hypothesis for random mating within sub-populations is that some selfing occurs, in which case N, the sample size, has to be replaced by the variance effective size due to selfing. This leads to equation 3.45, which is the same as both Nei & Chesser's G_{is} and Weir & Cockerham's f. For F_{st}, the alternative hypotheses formulated by Nei & Chesser (1983) and Weir & Cockerham (1984) are different: Nei & Chesser (1983) consider that the sampled population is unique, so that the global allelic frequency does not need estimating. This brings an alternative hypothesis which is cast in terms of allelic frequencies and the implicit assumption that there is homing: offspring will tend to stay where their parents lived. Using the variance effective size of a subdivided population, equation 3.52 is obtained. On the other hand, Weir & Cockerham (1984) consider that there is an extra level in the hierarchy: it could be other populations not sampled, or even non existing populations, but statistical outcomes of the genetic drift process. In this case, the global allelic frequency has to be estimated and the different correlations are relative to the highest level in the hierarchy, that is, the correlation between the least related genes (Cockerham, 1969, 1973). In this framework, the alternative hypothesis is cast in terms of rate of loss of heterozygosity, because rate of allele frequency drift could be affected by external inputs (migrants from populations not sampled, mutation etc...). The effective size to use to obtain an unbiased estimator of F_{st} is the variance effective size of a population where mating between relatives occurs and leads to equation 3.50. It should be stressed that the rate of loss of heterozygosity will be the same for both

Nei & Chesser's and Weir & Cockerham's alternative hypotheses, whereas the rate of allele frequency drift will be different. These differences stem from the implicit assumption of homing in Nei & Chesser, whereas there is an implicit assumption of no homing in Weir & Cockerham. These two different perspectives will help to answer different questions. Weir & Cockerham's estimators are appropriately used to compare estimates of gene-flow from different samples, either within a species or across species boundaries and will give unbiased answers to questions of the type: does species X at location A have the same breeding pattern as species X at location B. It has been shown that this type of questions cannot be answered with Nei's estimators, because of their dependance on the number of samples. On the other hand, Nei's estimators will be useful tools for the manager of a conservation reserve, who may be interested in measuring the extent of gene differentiation. It has been demonstrated, in particular, that G_{st} is an appropriate statistic to measure the variance effective size of a subdivided population. This measurement of the variance effective size not only does not require temporal data, which is always difficult to obtain, but seems also to be less subject to the large variations suffered by temporal estimates.

The framework described here to obtain unbiased estimators of F-statistics has other advantages: as long as an alternative hypothesis is clearly stated, it is possible to derive estimators for any type of evolutionary pressures. In particular, biased sex-ratio, unequal contribution of parents to the gametic pool and fluctuations in population sizes could be accounted for by including the effects in the formula for the effective size, as discussed in Chapter 2.

Chapter 4

Theoretico-realistic considerations?

4.1 Introduction

The aim of this chapter is to describe a series of techniques that have been developed to unravel levels of structuring in natural populations. This problem is very seldom addressed in the scientific literature and estimates of gene-flow are measured at a scale that is decided *a priori* by the researcher. No attempt is made in general to test if this scale is appropriate or not. This is unfortunate, since the behaviour of F-statistics, used to infer levels of gene-flow, are highly dependent on the underlying structure of the population (Chapter 3; Cockerham, 1969, 1973). One of the reasons why so little care is given to this problem is that natural populations live in general on a continuum (Wright, 1978), in which case discerning the boundaries of a sub-unit such as a deme will be a daunting task. To overcome this difficulty, researchers often use the concept of neighbourhood area, defined in Chapter 2. It was shown, however, that this concept is far from perfect, often misleading and, moreover, relies on the estimation of parent to offspring dispersal distance, which may well be highly variable in time and space. It is therefore necessary to outline a reasonably robust general strategy, tested on known models of population structure and, most importantly, transferable to the field.

Since F-statistics are designed to partition the heterozygote deficit into its different components and since unbiased estimators can be obtained, it is logical to start with them. Wright (1978, p. 115, Figure 4.2) used data of Dobzhansky & Epling (1944) on *Drosophila pseudoobsura* to plot the changes in F_{is} and F_{st} measured at different scales and showed that F_{is} increases with the size (area) of the sampling unit, whereas F_{st} decreases. He also applied this technique to the analysis of a data set from Epling & Dobzhansky (1942) on the desert snow *Linanthus parryae* (Wright, 1978, p. 203, Figure 6.3) and a data set collected by Lamotte (1951) on the land snail, *Cepaea nemoralis* (Wright, 1978, p. 231, Figure 6.16). These three surveys showed an increase in the value of F_{is} as the sampled area increases and a concurrent decrease in the value of F_{st}, an indication that gene-flow is somewhat restricted. However, the estimators of F_{is} and F_{st} used in these analyses are not unbiased and would show an increase in the value of F_{is} even if samples belonged to the same breeding unit, as is shown on Figure 3.2, Chapter 3. Since the objective is to identify correctly levels of structuring in natural populations, use will be made of the unbiased estimators derived in Chapter 3.

4.2 Raiders of the lost deme.

The smallest unit that one can possibly sample is the individual itself. We have seen in the Chapter 3 that at this level, F-statistics are not defined. One can start to look at a way of grouping these individuals in small units. From this level of grouping, F_{is} and F_{st} can be estimated. Grouping can then be made at a slightly larger scale and F-statistics recalculated. This procedure is repeated until all individuals are grouped into one single unit. Experiments such as these are widely used in ecology as a way to asses species richness at different scales (e.g. May, 1992).

Values of F_{is} and F_{st} can be plotted on a graph where the x-axis represents the different levels of grouping, and the y-axis the values of the F-statistics. As long as individuals belong to the same breeding unit, there is no changes in the values of either statistic, but F_{is} starts increasing (F_{st} decreasing) as soon as samples (groups) consist of more than one breeding unit. In Figure 3.2, unbiased F_{is} stays constant until groups are made of 64 individuals and then starts increasing, because the source of bias has changed from one due to selfing within a random-breeding unit to one due to partial isolation between breeding units. It is therefore possible to conclude that groups of 64 individuals constitute a random-breeding unit, or deme. Note in this case that $N_e m$ is $64/1.54 * 0.2 = 8.32$. This graphical method therefore seems to be able to detect structuring for Nm values larger than 4, in contradiction with what is generally reported in the literature (e.g. Slatkin, 1985, 1987; Crow & Kimura, 1970; Nunney & Campbell, 1993): if Nm is larger than 4, then the population behaves as if effectively panmictic.

The behaviour of F-statistics calculated at different levels of grouping have been investigated using data sets generated with **MODEL42** described in Chapter 2. As the expected distribution of F-statistics is unknown (Weir, 1990), use will be made of re-sampling techniques such as the jackknife and the bootstrap in significance testing.

4.2.1 Re-sampling techniques

The generalisation of personal computers in the office has allowed the development of new statistical techniques, known under the generic name of re-sampling techniques, or randomisation tests. These techniques are not subjected to the limitations suffered by parametric tests, such as normal distribution of the data, or homogeneity of the variances (Manly, 1991; Crowley, 1992).

The Jackknife

In ordinary usage, this word describes a large pocket knife, with a multitude of small pull-out tools, so that the owner is able to tackle many small tasks without having to look for something better. While this statistical technique was first described by Quenouille (1956), its name was given by Tukey (1958), who outlined that this method can be used where no better one can easily be.

Given a parameter, ϕ and a series of observations, $X_1, X_2, \ldots, X_n$, one can obtain an estimate, $\hat{\phi}$, of ϕ. The jackknife procedure consists of obtaining n new estimates of ϕ, ϕ_i by omitting each observation in turn. The mean of all these estimates is then just the average of all $\phi_i, \phi_{(.)}$ and a new estimator of ϕ, which should be less biased is:

$$\hat{\phi}_J = n\hat{\phi} - (n-1)\hat{\phi}_{(.)}$$

(Weir, 1990) and an estimate of the variance of $\hat{\phi}$ is:

$$Var(\hat{\phi})_J = \frac{n-1}{n} \sum_i \left(\hat{\phi}_i - \hat{\phi}_{(.)}\right)^2$$

Taking F_{is} for example, if n populations have been sampled, we can obtain an estimate of the variance of F_{is} by omitting each of the samples in turn. This will be the procedure to follow if one wants to find the confidence interval of F_{is} measured at a given locus. This would be a way to identify samples behaving oddly. On the other hand, if many loci are scored, each one can be omitted in turn, to give the confidence interval of F_{is} over loci. Comparisons of confidence intervals over populations and over loci would be a way to identify loci with peculiar behaviours (Goudet *et al*, In Press) and to eliminate them from subsequent analyses.

This technique, however, suffers some drawbacks. In particular, it is very sensitive to outliers, under which case, the jackknife variance is too large (Efron, 1982; Manly, 1991).

The Bootstrap

This method was first described by Efron (1979), who pointed out that the Jackknife can be regarded as an approximation to a more primitive method that he named the bootstrap, to reflect the fact that its use is analogous to someone pulling themselves up by their boot-laces. If there are n observational units, it consists of sampling *with replacement* a large number of times (1000) n observational units and to recalculate the statistics from this sample. For F-statistics, it would be sampling n loci from n with replacement and recalculating the F-statistics. This technique does not need to be applied with less than 5 observations, since it is possible to enumerate all combinations (there is $4(2^2)$ possible values with two loci, 27 with three loci, 256 with four). The different bootstrap estimates can then be sorted in ascending order and the inner 95% provide the bootstrap confidence interval. It should be noted that bootstrapping F_{st} or F_{it} values over populations would be meaningless, since the same population can be sampled many times.

Randomisation testing: the method of permutations.

The two techniques described above provide a confidence interval of the observed statistics. It is then possible to assess if the statistic is different from 0 by checking if 0 is included in the confidence interval. However, this relies on the assumption that the loci under scrutiny are neutral and also that they are a random sample of all possible loci.

A technique that proves useful in relaxing the above assumptions consists of permuting alleles within samples, alleles within the total and multi-locus genotypes among samples to test F_{is}, F_{it} and F_{st} respectively. This way, the distribution of the null hypothesis is obtained (e.g. alleles within samples are associated at random, therefore, there is random mating and F_{is} is not different from 0). Generating 4999

such permutations plus the observed value and sorting the data in ascending order will give the probability that the observed combination of alleles within individuals is due to random mating. If this probability is < 0.05, then the null hypothesis can be rejected at the 5% level.

Some problems arise as to how to generate the null distribution for F_{st}: if there is random mating within sub-populations, then permuting alleles or multi-locus genotypes will give essentially the same results, because each allele can be considered as being independent of the other allele present at the locus. However, if there is a certain amount of selfing, or mating with relatives, alleles within individuals are not independent one from another and testing F_{st} using the permutation of alleles within the total will lead to erroneous results, by increasing the probability of Type I error.

This last set of techniques are a special case of more general computer-intensive methods, known as Monte-Carlo tests (Manly, 1991). Permutation methods test the null-hypothesis that the observed distribution is random. In Monte-Carlo tests, the null hypothesis is more specific. In our case, it could be *'The observed samples behave in the same manner as an island model of populations, with 10% migration between samples and 70% selfing'.* Testing this hypothesis could be achieved using **MODEL42**, through the generation of many replicates with the parameters of the null hypothesis.

4.2.2 The island model

To assert if measuring F-statistics at different scales is able to unravel levels of structuring, the island model was used first. Twenty replicates of the gametic cloud island model described in Chapter 2 were simulated and run for one thousand generations. The genotypic array at the thousandth generation was saved. Each replicate was considered as an independent locus, from which f, θ and G_{st} were estimated using the program **FSTAT**, whose listing can be found in Appendix D. Confidence intervals on each point were obtained by jackknifing over loci. The confidence interval displayed on the graphs are the 95% confidence intervals, calculated as $\pm 1.96\sigma$. All the individuals in the population were sampled.

Figure 4.1 was obtained from an island model with 64 islands and 64 individuals on each island. Selfing occured at random (1/64) and migration between islands was set at 1%. Figure 4.1 displays the effect of grouping of samples on f (solid line), θ and G_{st} (dashed lines). Focusing on f, we see that it stays constant below the deme

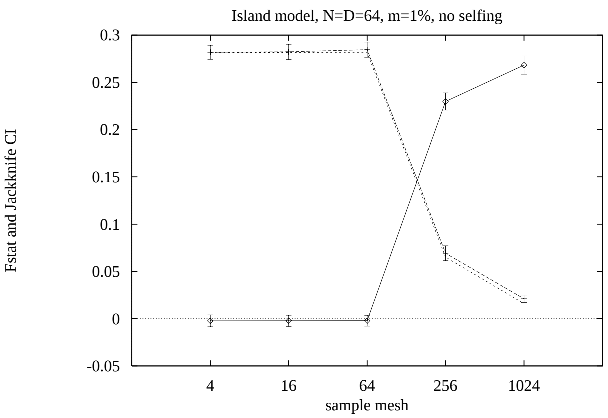

Figure 4.1: Changes in F-statistics with mesh size.

size, with a value of 0 (this is the expected value of f when there is random mating). As soon as more than one deme are pooled together, f increases suddenly to a value near its maximum (F_{it}). The differences between f as measured below deme size and above deme size is statistically significant (confidence intervals do not overlap). θ and G_{st} behave exactly in the opposite way: there is a decrease in their value after grouping of more than 1 deme.

Because we are using data generated by computer simulations, we expect each locus to behave in a similar manner to the others. As was mentioned in the previous section, this can be tested using jackknifing over population for each locus, while a jackknife (or bootstrap) over loci is also calculated. A confidence interval per locus is then obtained, as well as the overall confidence interval. If one or more loci have a confidence interval that does not overlap the over-loci confidence interval, it is discarded for the next analysis. Figures 4.2 to 4.4 display these confidence interval calculated at different mesh sizes: if natural populations are sampled, the sampling will operate either below or above the true deme size, if only because sampling is not exhaustive. We can see in these figures that as long as sampling is below the deme size (Figures 4.2 and 4.3), confidence intervals are very small, they all overlap and no locus displays an f significantly different from 0. One can however notice that there is variation among loci, although never enough to lead to the elimination of one of them. Figure 4.4 shows that when samples contain more than one random breeding

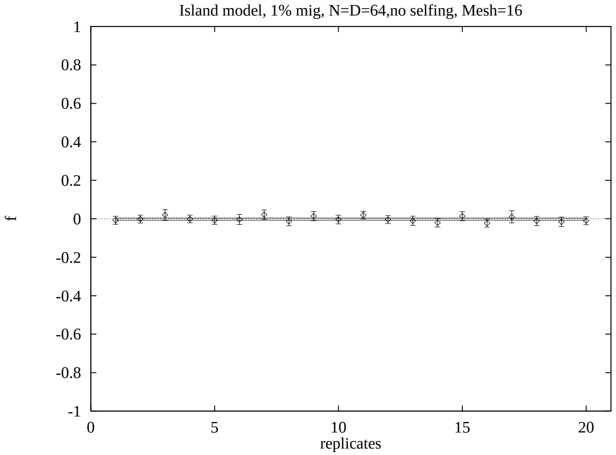

Figure 4.2: Detecting outlier loci at mesh=16

unit, confidence intervals widen, the values of f are shifted upwards and no locus has a confidence interval overlapping with the zero axis. Although all the over-population confidence intervals overlap with the over-loci confidence interval, some of the loci are at the limit of being excluded.

To answer the question of how much migration is necessary before the subdivided population behaves as effectively panmictic, a data set was generated with 30% migration, deme size of 64 and 16 demes, with no selfing. This leads to a Nm of 19.2. The results are displayed on Figure 4.5.

The first striking feature is how large the confidence interval on f is. This is because of the scale on the y-axis, which only covers the range [-0.005:0.03]. On the other hand, 0 is encompassed in the confidence interval for samples within deme, but excluded from it when samples are made of more than one deme. Furthermore, although not significant using the jackknife confidence interval of f, it is noticeable that f increases between sample mesh of 64 and sample mesh of 256. However, for this set of parameters, it seems that θ is a more appropriate statistics to use, since the decrease in its value between sample mesh 64 and 256 is statistically significant.

When the estimators are very close to 0, as in the present case, use can be made of the permutation procedure described above. The probabilities that the observed

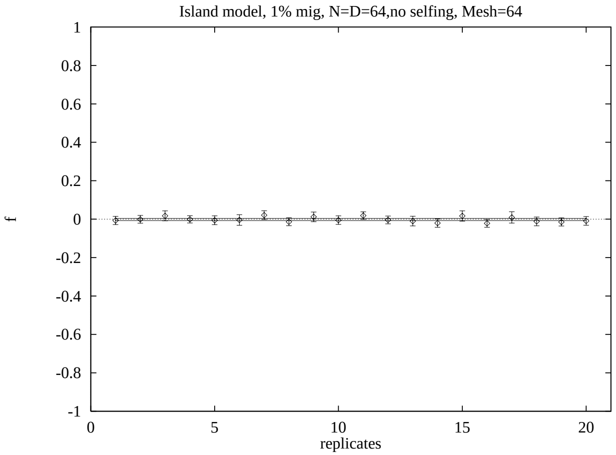

Figure 4.3: Detecting outlier loci at mesh=64

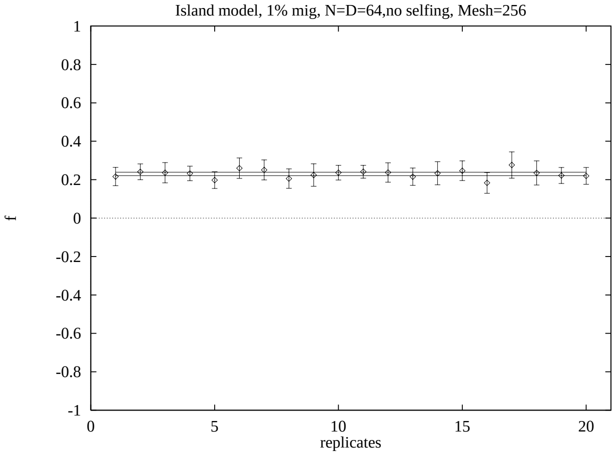

Figure 4.4: Detecting outlier loci at mesh=256

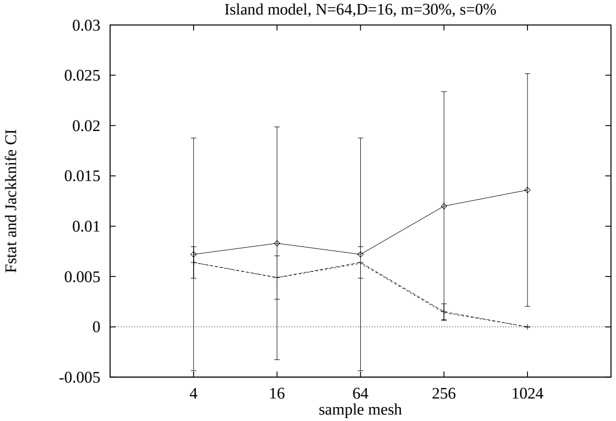

Figure 4.5: Changes in F-statistics with mesh size

f, θ and F come from a single large, random mating population are given below:

$Mesh$	f	θ	F
4	0.0672	< 0.0002	0.002
16	0.0426	< 0.0002	0.0024
64	0.0672	< 0.0002	0.002
256	0.00720	< 0.0002	0.0028
1024	0.00160	n/a	n/a

In all cases, θ is highly significant (the observed θ is the highest of 5000 estimates generated by permutations). With regard to f, although the probability does not allow the rejection of the null hypothesis for sample sizes 4 and 64, it is rejected for sample size 16 and is very close to the rejection level for 4 and 64. This remains unexplained. The non-availability of the probability levels of θ and F for a sample size of 1024 is because there is only one sample, in which case θ and F cannot be calculated. One could wonder if these tests would accept the null-hypothesis when it is true since all probability levels in the above table are very low. To test this, a single large, random breeding unit was modelled. The simulation was run until the thousandth generation and, as in the previous cases, 20 replicates were run to mimic 20 independent loci. This population was exhaustively sampled for mesh sizes of 64 and 256 and the probability that the observed f, θ and F come from a single large,

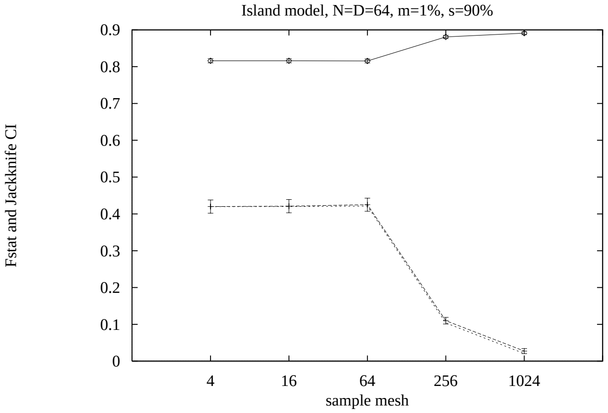

Figure 4.6: Changes in F-statistics with mesh size

random mating population are given below:

$Mesh$	f	θ	F
64	0.493	0.911	0.555
256	0.520	0.888	0.556

Obviously from the above table, the null hypothesis is accepted when it is true.

Next, it is of interest to see the effect of selfing on the behaviour of the different F-statistics. Figure 4.6 shows the changes in f (solid line) , θ and G_{st} (dashed lines) for an island model with 90% selfing and 1% migration. Although there is still a statistically significant increase in the value for f after pooling together more than 1 deme, it is much more difficult to discern, because f is near its maximum value of 1. On this graph, structuring is better inferred from θ and G_{st}. Figure 4.7 also shows that with selfing, even when loci are independent, it is possible to get outliers, since one of the loci CI does not overlap with the over-loci CI.

4.2.3 One-dimensional stepping-stone models

Since the two extremes of population structures with finite deme size are the island model and the one-dimensional stepping-stone model, with the two- and three-dimensional stepping-stone models being intermediate (cf. Chapter 2), only one-dimensional stepping-stone models will be treated here. For a given migration, population size, selfing proportion and number of demes, what differences are there between an island model and a 1-dimensional stepping stone model? Analysis of the behaviour of F_{st} over time was carried out in Chapter 2 and it was shown that models with geographical structuring take longer to reach equilibrium. One can therefore wonder if the technique presented above will work for stepping-stone models. A first step in understanding the differences between models with and without geographical structure is to keep all parameters (migration, deme size, selfing level, number of demes) constant and to follow the changes in F-statistics as a function of the mesh size.

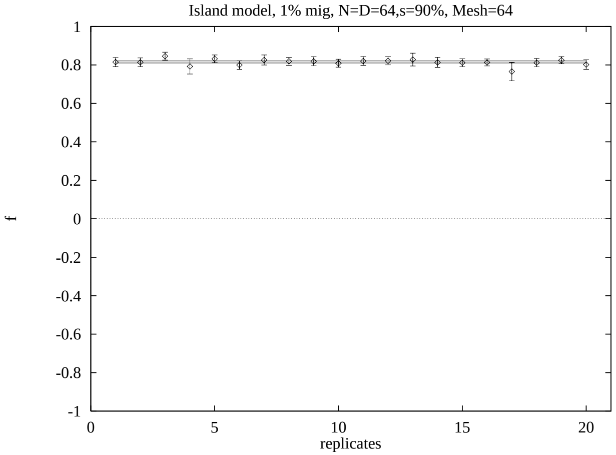

Figure 4.7: Detecting outlier loci at mesh size 64.

Figure 4.8 shows this comparison. The first striking feature is that θ is much larger in a stepping-stone than in an island. This is an indication that equilibrium has been reached for the island model, since F_{st} values in island and stepping-stones start diverging after equilibrium has been reached in the island model. There are no differences in f as long as it is measured below the deme size, a sign that f is not affected by geographical structuring as long as it is measured at an appropriate scale. However, as demes are pooled together, f in the stepping-stone model keeps increasing, whereas it stabilises very quickly in the island model. Most of the changes in the value of f occurs in a single step in island models (the curve is horizontal from deme size 2 to 64 and then from deme size 128 to 2048 — the same observation could be made from Figure 4.1). It is also remarkable that the amount of increase in f between deme size 64 and 128 is the same for both models. This is because

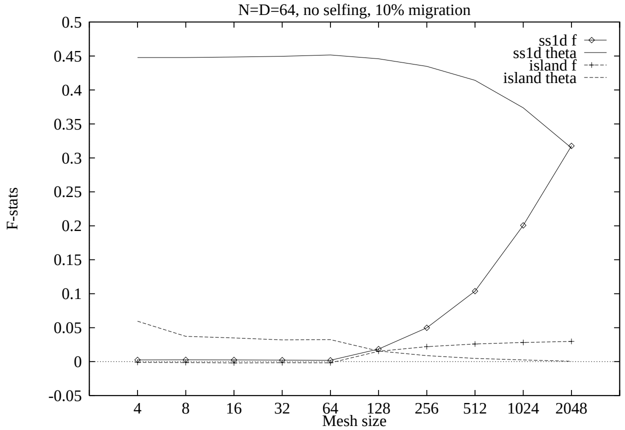

Figure 4.8: Comparison of a one dimensional stepping-stone model with an island model of population structure

neighbouring demes in a one-dimensional stepping stone model exchange the same number of migrants as any demes in the island model.

Figures 4.9, 4.10 and 4.11 display the changes in F-statistics for a 1-dimensional stepping-stone model. The percentage of migration in Figure 4.9 is 10%, which makes the product Nm 6.4. Structuring can still be detected, but two important changes can be seen. First, the increase in f when grouping more than 1 deme does not look as sharp as in the island model. This is because the range of the y-axis is much larger here since θ is larger. Second, the confidence interval on θ is much larger than in the island model. The first point can be understood as follows: in the island model, the different demes share the same 'level' of relatedness, whereas in the stepping-stone model, individuals in demes close one to another are likely to be more related than individuals in demes further apart. The second point emphasizes a facet of stepping stone versus island structure: a wider range of F_{st} values are obtained with a given set of parameters in a stepping stone model than in an island model, because the genetic sampling process is restricted in space.

Figure 4.10 displays essentially the same information, but the level of selfing this time is 90% instead of 0%. The increase in f is much more difficult to perceive, for the same reason as in the island model, namely, f is always near its maximum. It is noteworthy that f below and above deme size are not statistically different anymore.

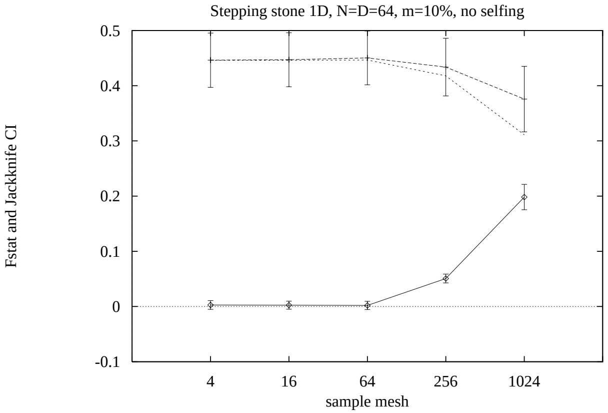

Figure 4.9: Changes in F-statistics as a function of mesh size

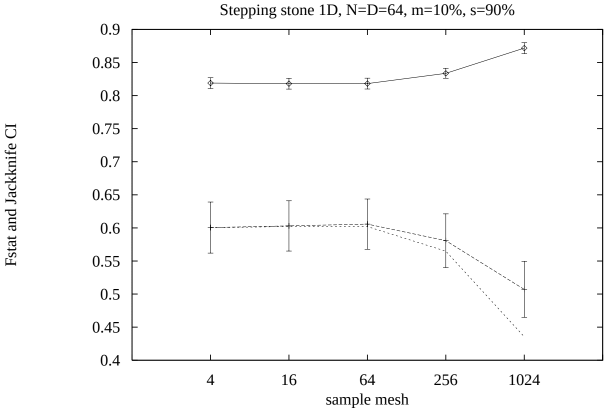

Figure 4.10: Changes in F-statistics as a function of mesh size

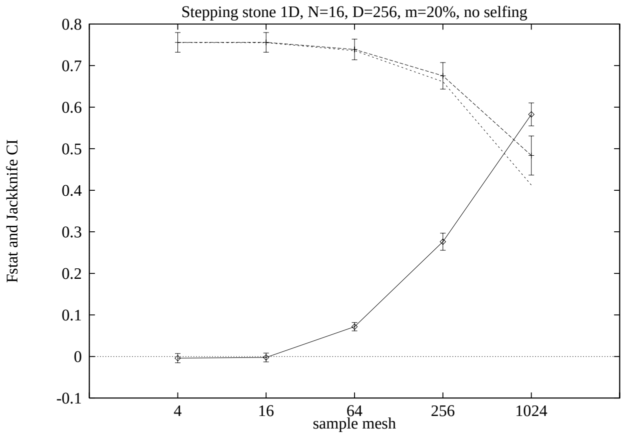

Figure 4.11: Changes in F-statistics as a function of mesh-size

This is also the case for θ. In this case, it is not possible to detect the deme size, whereas it was possible to do so in the island model. And, just as a classic film requires a good ending, one cannot fail to be disappointed by this negative result.

The values of θ when samples are taken within a deme is higher than in the case with no selfing: selfing accelerates the process of random genetic drift and therefore will increase the amount of differentiation between patches. The reverse is true when there is avoidance of mating between relatives, a negative f is obtained and θ will be lower than if mating was at random, because avoidance of mating with relatives will slow down the process of random genetic drift.

Figure 4.11 is another example with a deme size of 16 and migration of 20%, which give a Nm of 3.2. f below and above deme size are statistically different, whereas θ values are not.

The migration levels used in these investigations give Nm products larger than one, a level of gene-flow at which the population is supposed to behave as effectively panmictic. It has been possible, however, to detect structuring in most cases. Even when the deme size was not identifiable, as in Figure 4.10, there was a statistically significant difference between f calculated at the deme size and f calculated at the highest level of pooling.

A difference in behaviour of f as a function of pooling levels is also shown. While f in island models tends to level off quickly after pooling of more than one deme, it keeps increasing in one-dimensional stepping-stone models. It is therefore suggested that the technique presented here could also be used as a first appraisal of the presence of geographical structuring.

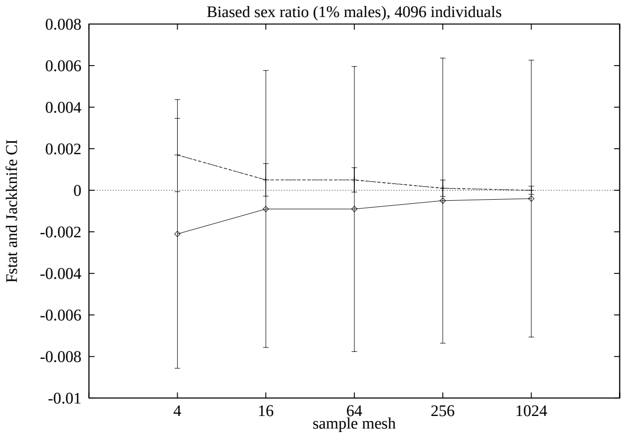

Figure 4.12: Behaviour of F-statistics when the sex-ratio is biased (1% of males in the total population)

4.2.4 Effect of a biased sex ratio

Before closing this section, some consideration needs to be made as to how the different estimators behave when some of the assumptions of the applications are relaxed. In particular, few species are monoecious, and a biased sex ratio is often found in social and domesticated animals, or in plants with peculiar reproductive systems, such as gynodioecy (presence of female and hermaphrodite plants in the same species). If the sex ratio is biased, the effective size of local population of the species is considerably lowered.

Figure 4.12 displays the changes in F-statistics for a single large population (4096 individuals), but with a very strong biased sex ratio (1% males in the population, which gives a N_e of 162). The expected value of both f and θ is zero and it is shown on Figure 4.12 that the observed values, at different levels of sampling, are not statistically different from 0.

Note also that a biased sex ratio implies a large variance in reproductive success

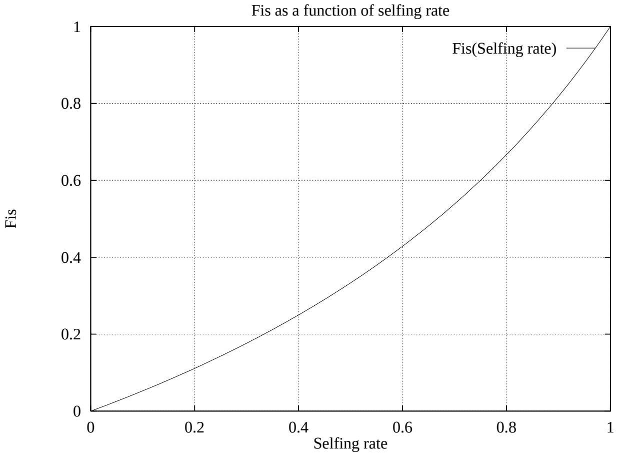

Figure 4.13: $F_{is}(s) = \frac{s}{2-s}$

(because males are producing more offspring than females). It is therefore likely that there will be no effect of differential reproductive success on the behaviour of unbiased F-statistics.

4.3 Estimation of Nm or N and m?

Since unbiased F-statistics have proved useful for identifying levels of structuring, it is possible to turn to the problem of biological inferences, namely inferring levels of selfing and migration from random breeding units. The case of selfing is straightforward: the relation between f and s was given in Chapter 2, equation 2.8 and does not depend on a combination of parameters. In all cases where selfing is 0, it has been shown that when f is measured below the deme size, it is not significantly different from 0, the expected value with no selfing. When selfing occurs, the expected value of f is given by equation 2.8 and is plotted in Figure 4.13. With 70% selfing, the expected value of f is 0.538 which is the observed value of f in Figure 3.2 and with 90% selfing, it is 0.82, the observed value in Figures 4.6 and 4.10.

Although this method of estimating s can be considered as 'rough and ready' compared to other methods, such as those of Ritland (1990), it gives accurate estimates when f is estimated at an appropriate scale.

The case for m is different. It has been shown in Chapter 2 that F_{st} relate both to N_e and m, and that, under the assumptions that m is small and N_e is large, it is a simple function of the product of these two parameters. On the other hand, when these assumptions are not met, an exact relationship between θ and N_e and m was given (equation 2.10). The discrepancy between these two relations was given in Chapter 3, Figure 3.17 for $N = 25$. Figure 4.14 displays the same relationship but for a wide range of values of N, between 2 and $10,000$. Here N refers to the effective sample size, since θ is estimated from samples. The first observation is that the approximation always leads to an overestimation of the number of migrants. This trend is stronger for small sample sizes than large ones, but still hold true for sample sizes of $10,000$ individuals when θ is small (0.0001). Noteworthy in Figure 4.14 is the independence of the approximate formula with regard to N (an inherent characteristic), which leads to some aberrant results, such as a number of effective migrants larger than the sample size (as an example, with samples of size 10 and an observed θ of 0.0001, $4Nm$ would be equal to 9999). On the other hand, the exact formula for $4Nm$ may look artificial, since it consists of multiplying by $4N$ both the right hand-side and the left hand-side of equation 2.10, which gives m. However, it gives results that look *a priori* sensible, since the inferred $4Nm$ is never larger than $4N$. A major inconvenience of this formula, for comparative purposes, is that it is not independent of sample size. $4Nm$ will increase as the sample size increases for a given θ. *The only appropriate measurement to compare different populations cannot be cast in term of biological parameters, such as Nm, but has to be achieved through an estimator independent of the sampling strategy, θ.* Unless θ is large, use of the approximation could give rise to highly erroneous results, an order of magnitude larger than the real parameters.

Bearing these considerations in mind, use can still be made of unbiased estimators of F-statistics to infer biological parameters, as long as the conditions of application are understood. It was shown that when the sample size is 25 individuals, the approximate formula holds when θ is larger than 0.1. 25 individuals is a sample size commonly found in the population biology literature. When molecular techniques are used (RFLP, VNTR, sequence), sample sizes tend to be smaller, because these techniques are more costly and time-consuming. In this case, even higher values of θ must be found before use can be made of the approximation. In cases where θ is less than 0.1, it is suggested that m be estimated from the exact relation rather than Nm from the approximate relation and to multiply the estimated m by the average effective

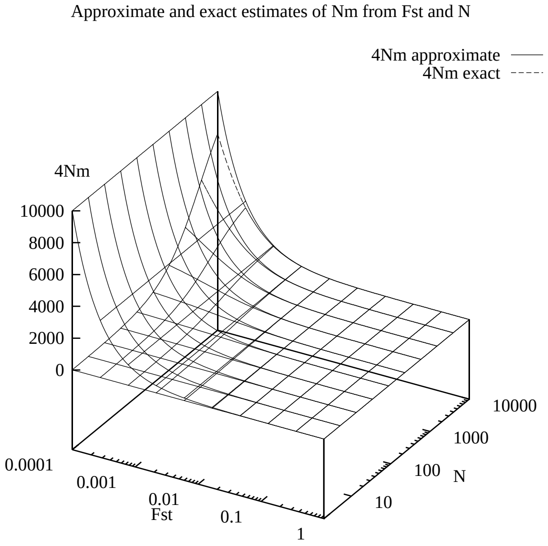

Figure 4.14: Approximate and exact relation between F_{st}, Nm and N.

Table 4.1: Results of biological inferences carried out on the data sets reviewed above

$Model$	s	m	N	N_e	$N_e m$	D	f	θ	$\hat{s}$	$N_e\hat{m}_{app}$	$\hat{N}_e$	$\hat{m}_{ex}$	$N_e\hat{m}_{ex}$
$IM1$	0	0.01	64	64	0.64	64	−0.002	0.285	−0.004	0.63	64	0.010	0.62
$IM2$	0.9	0.01	64	35	0.35	64	0.816	0.425	0.898	0.33	35	0.009	0.33
$IM3$	0	0.3	64	64	19.20	16	0.007	0.006	0.014	38.81	64	0.329	20.91
$SS1$	0	0.1	64	64	6.40	64	0.002	0.450	0.004	0.31	64	0.005	0.30
$SS2$	0.9	0.1	64	35	3.52	64	0.818	0.606	0.900	0.16	35	0.005	0.16
$SS3$	0	0.2	16	16	3.20	256	−0.002	0.756	−0.005	0.08	16	0.005	0.08

sample size, to get an estimator of Nm. One also needs to bear in mind that the relation between m, Nm and F_{st} is based on the assumption that in the populations under investigation, equilibrium between the opposing forces of random genetic drift and migration has been reached, an equilibrium that may well take a very long time to be achieved (cf. Chapter 2).

Table 4.1 summarises the results of biological inferences made on the data sets presented above. Focussing on $\hat{s}$ first, we see that the estimate is in very good agreement with the parameter entered in the model, both in the island model and the stepping-stone model. $N_e\hat{m}_{app}$ is a good estimator of $N_e m$ in island models and when the proportion of migration is small. When m is large, the discrepancy between $N_e\hat{m}_{apps}$ and the real value can be quite large (twice as large for $IM3$), whereas $N_e\hat{m}_{ex}$ is in very good agreement with the real value. Noteworthy also is the agreement between the inferred migration and the real value, even for large migration, as is the case for $IM3$. As was noted in Chapter 2, the effect of selfing is to diminish the local effective size and this is confirmed when comparison is made of $\hat{N}_e$ in $IM1$ and $IM2$.

Stepping-stone models show a different pattern: estimations of selfing rate and local effective size are in close agreement with the real values, but migration estimates, either on their own or in combination with the effective size, are very different from the input values. In this case, because θ values are quite high, there is a close agreement between the approximate and exact estimation of $N_e m$.

4.4 The isolation by distance model: a new scenario

While F-statistics are useful and accurate to measure the extent of isolation in structured populations with no overlapping demes, one needs to investigate their behaviour in isolation by distance models, to test:

- If the concept of neighbourhood size is meaningful
- If structuring can be detected, even though the biases on F-statistics for these models are not known.

To answer these questions, linear patterns of isolation by distance are modelled using an exponential decay for the parent-offspring dispersal distances. Selfing is not random, but fixed at 10% in all cases. The population consists of 1024 individuals, the simulations were run for one thousand generations. Twenty replicates, simulating twenty independent loci, were recorded at the thousandth generation. An exponential distribution of parent-offspring dispersal was used. Seven different standard deviation of dispersal, λ are discussed, ranging from very restricted dispersal ($\lambda = 1$) to very large dispersal ($\lambda = 99$). The distribution of dispersal distances generated is shown in Figure 4.15. A point of importance is the discontinuity between 0 and 1. This is due to the fixed proportion of selfing, or 'homing': only 10% of the gametes for all distributions stay at the location of their progenitor. This is the proportion that would be expected with a λ of 10, but would be higher for smaller λ and smaller for larger λ.

The changes in F-statistics as a function of mesh size are given in Figures 4.16 to 4.22. The behaviour of the changes in f and θ bears some resemblance to that of a linear stepping stone model: changes in values of f and θ are smooth, compared to an island model. However, a discontinuity is noticeable in linear stepping stone models, whereas it is not in isolation by distance.

A sharp increase in f occurs from the smallest mesh size (2) with very restricted dispersal (Figure 4.16), whereas f stays constant for all mesh sizes with $\lambda = 99$ (Figure 4.22). With an average dispersal of 2, no differences in the values of f could be detected for samples of sizes 2,4 and 8. As dispersal distance increases, changes in the values of f occur later. Note however, that even with an average dispersal of 99, θ is significantly different from 0. For a λ of 5 (Figure 4.18), structuring is only

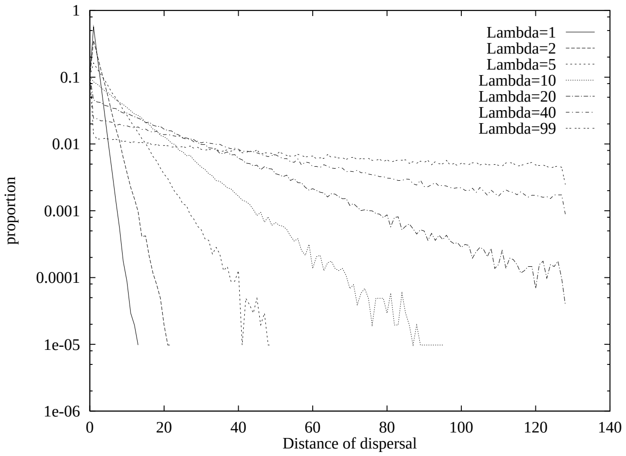

Figure 4.15: Distribution of dispersal distances in a linear isolation by distance model, with an exponential decay of dispersal distances, for seven different parameters of scale.

significantly detected for a mesh size of 64. It is 256 for a λ of 20 (Figure 4.20) and no differentiation between units could be detected for larger values, although a trend toward an increase in the value of f exists for $\lambda = 20$ and to a (much) lesser extant, 40 (Figure 4.21.

Further interpretation of these graphs seems difficult. Detection of units that are isolated will be dependent on the number of levels of grouping: should one consider the overlapping of confidence interval of 2 successive points on the graphs, or should one consider absolute differences? In Figure 4.16, the first statistical difference between adjacent points occurs between levels 8 and 16, while if absolute differences are considered, level 8 is different from level 2. The definition of the neighbourhood size is given in Chapter 2: the area from which the parent of the central individual could be considered as if drawn at random. Taking the inner 95% of the exponential distribution provides us with some measure of the neighbourhood size. On the other hand, Figures 4.16 to 4.22 could be used to infer neighbourhood sizes by considering non-overlapping of 2 neighbouring points (1) or non-overlapping with the first point on the graph (2). Table 4.2 gives the different values for the neighbourhood size.

Although these three measures increase with an increase in λ, little more can be said and the relations between these three sets of data do not seem straightforward.

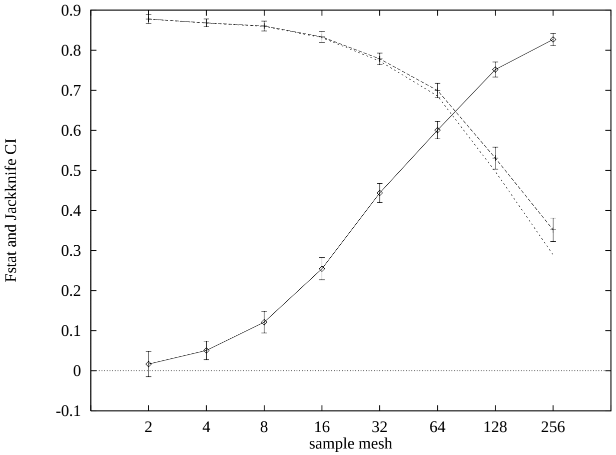

Figure 4.16: IBD model (1D) 10% selfing. $\lambda = 1$

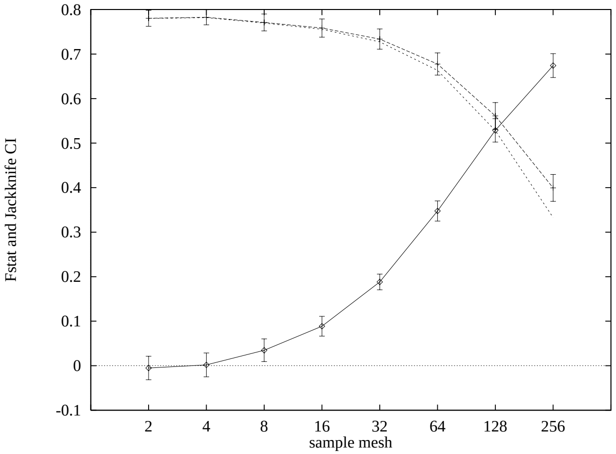

Figure 4.17: IBD model (1D) 10% selfing. $\lambda = 2$

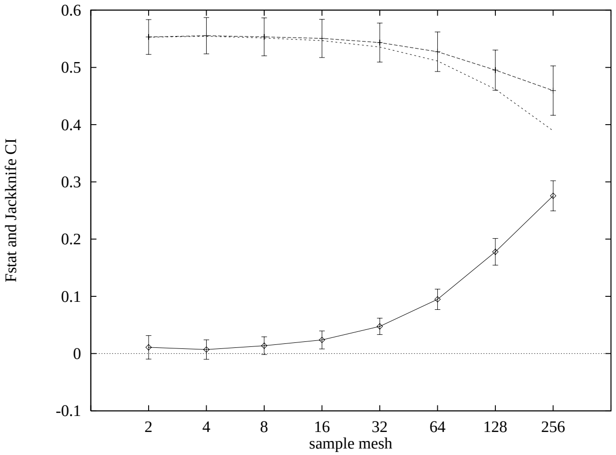

Figure 4.18: IBD model (1D) 10% selfing. $\lambda = 5$

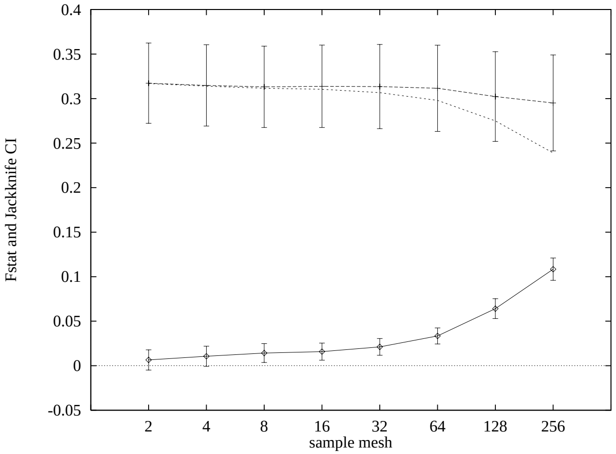

Figure 4.19: IBD model (1D) 10% selfing. $\lambda = 10$

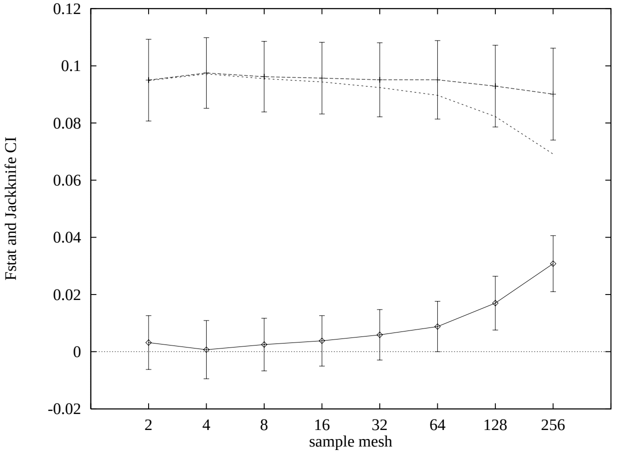

Figure 4.20: IBD model (1D) 10% selfing. $\lambda = 20$

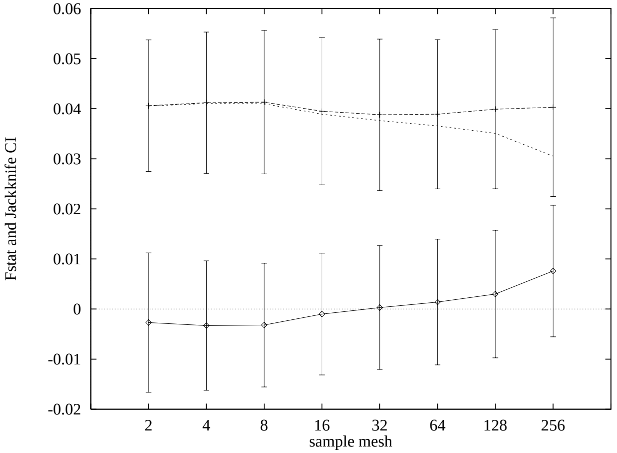

Figure 4.21: IBD model (1D) 10% selfing. $\lambda = 40$

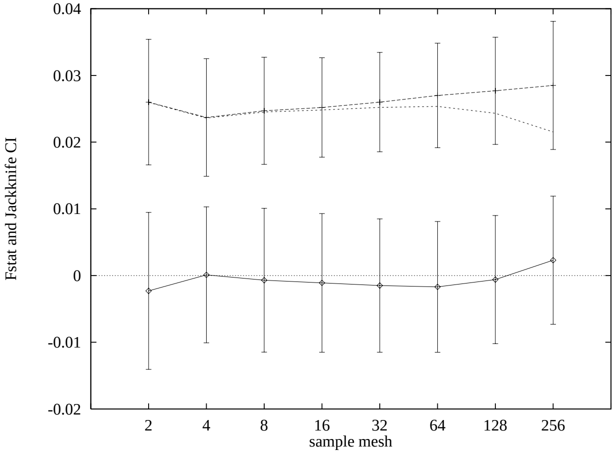

Figure 4.22: IBD model (1D) 10% selfing. $\lambda = 99$

Table 4.2: Possible estimates of neighbourhood size. (1) is for two consecutive points with non-overlapping CI. (2) is for non-overlapping CI with the first point. (3) is based on Wright's neighbourhood definition, adapted for a exponential parent to offspring dispersal

λ	$\lambda \log(\frac{1}{0.025})$(3)	(1)	(2)	(1)/(3)	(2)/(3)
1	4	8	8	2	2
2	7	16	16	2.29	2.29
5	18	64	32	3.56	1.78
10	37	128	64	3.46	1.73
20	74	?	256	?	3.46
40	148	?	?	?	?
99	365	?	?	?	?

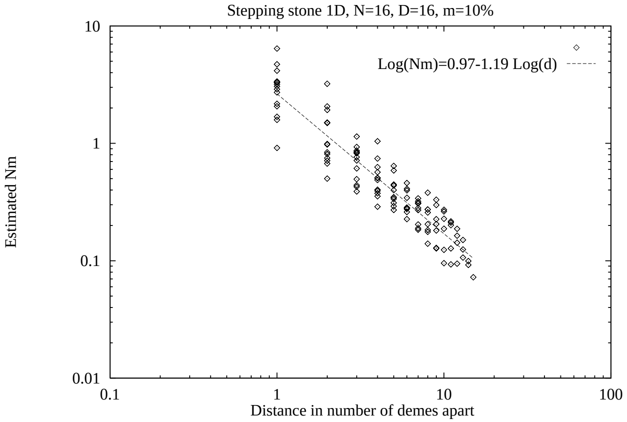

Figure 4.23: log—log regression of Nm on distance

It is therefore possible to detect isolation by distance using the changes in F-statistics with mesh size, but it seem difficult to find a structure that would bear resemblance to a neighbourhood size.

Other ways of detecting geographical structuring have been described in the scientific literature. In particular, Slatkin (1993), developed a method for detecting isolation by distance in equilibrium and non-equilibrium populations. It consists of calculating Nm per pair of samples using the relation $F_{st} = 1/(4Nm + 1)$, derived in Chapter 2, Equation 2.11. A linear regression of $\log(Nm)$ on $\log(Distance)$ is then applied. If the slope is significantly different from zero, then there is isolation by distance. This method has been applied to two levels of migration in a one-dimensional stepping-stone model. The results are displayed in Figure 4.23 and 4.24. Both regressions (for 10 and 50% migration, with 16 demes made of 16 individuals) are highly significant. As migration increases, both the constant and the regression coefficient increase.

This method was also applied to the one dimensional isolation by distance model, on the data sets used for the previous analyses. For the following, F-statistics were calculated for samples of size 16. There are therefore 64 samples and the largest distance between samples is 64. Table 4.3 gives the regression equations, together with their R^2.

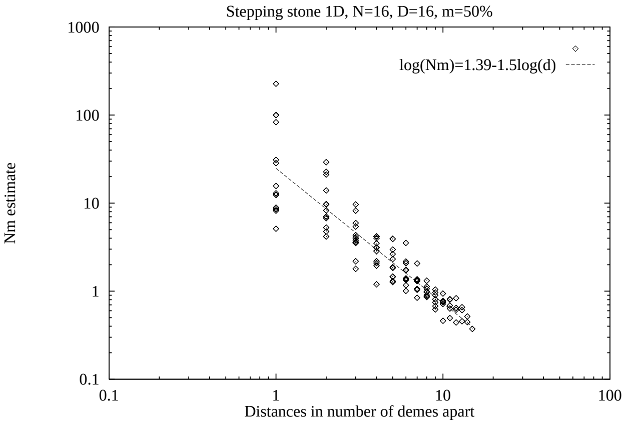

Figure 4.24: log—log regression of Nm on distance

Table 4.3: log—log regression of Nm on distances and R^2

λ	Regression equation	R^2	Neigbourhood size(4)
1	$\log(Nm) = -1.73 - 0.44\log(d)$	0.47	0.18
2	$\log(Nm) = -0.80 - 0.61\log(d)$	0.67	0.45
5	$\log(Nm) = 1.30 - 0.92\log(d)$	0.90	3.67
10	$\log(Nm) = 2.84 - 1.12\log(d)$	0.94	17.12
20	$\log(Nm) = 4.03 - 1.03\log(d)$	0.86	56.26
40	$\log(Nm) = 4.73 - 0.93\log(d)$	0.68	113.30
99	$\log(Nm) = 5.25 - 0.95\log(d)$	0.59	190.57

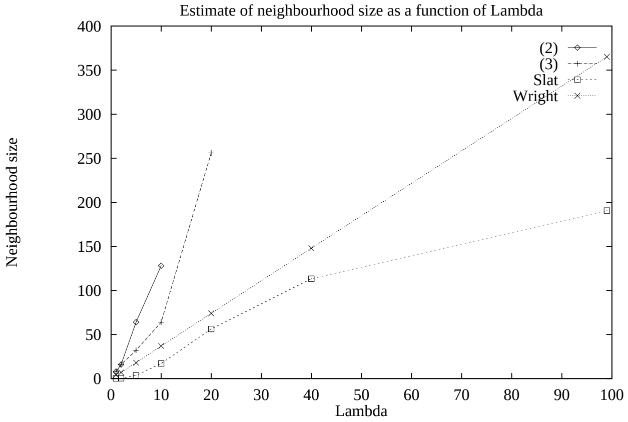

Figure 4.25: Comparison of the 4 estimators of neighbourhood size. Wright's is the expected.

The constants and the regression coefficients are highly significant in all cases ($P < 0.0001$). The percentage of the variance explained by the regression is also high, between 47 and 90%. This percentage is maximised for intermediate values of λ and decreases for very small or very large λs.

While no real trend is shown by the regression coefficients, the constant of the regression increases as λ increases. Slatkin suggested using the constant (the intercept of the regression line with the y-axis) as a measure of the neighbourhood size. The results are displayed in the right most column. The neighbourhood size increases with increasing λ, as expected. Comparison of the different estimates of neighbourhood sizes is given in Figure 4.25. While (2) and (3) are at best mythical, there is a good agreement between Slatkin's estimate and the expectation (Wright's estimate).

To explain the values of the regression coefficient and the percentage of the variance explained, it is useful to plot the data. Figures 4.26 to 4.32 show that the regression hides part of the story. Figures 4.26 and 4.27 show a decrease in the estimated Nm with distance for small distances, but no differences in the estimate of Nm for larger distances (> 10) (θ is the same between samples 10 units apart or 64 units apart). The relationship is truly linear in Figures 4.28 and 4.29. For Figures 4.30 to 4.32, no differences in the value of Nm can be detected for small distances (which means that there is no differences in θ for samples one unit apart or 10 units apart), whereas Nm diminishes for larger distances (> 10 for Figure 4.32). These behaviours emphasize the problem of scale: for very low dispersal distances, sampling locations far apart will not lead to any detection of isolation by distance and one would be tempted to conclude that the investigated population behaves as if it was an island model. The reverse is true for species with large dispersal distances.

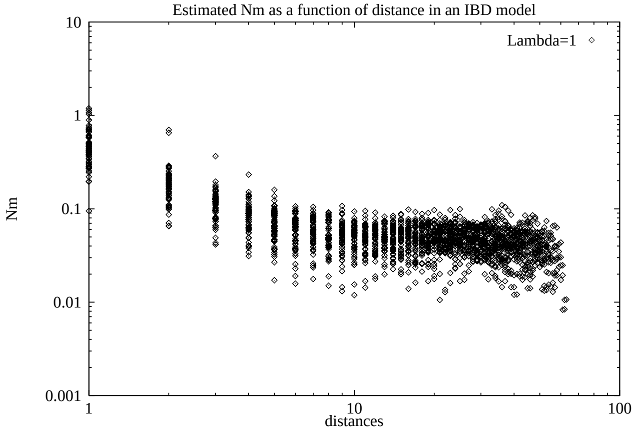

Figure 4.26: log—log plot of Nm on distance

One of the drawback of this technique is that a log—log linear relationship between migration and distance has to be assumed. This relationship, however, is not necessarily linear, even in a 1-dimensional habitat. One of the suggestions of Slatkin (1993) is that when the regression line is not statistically significant, it could be an indication that the population is not at equilibrium under the opposing forces of migration and random genetic drift. For large λ's, however, it has been checked that F_{st} reached a plateau. If only short distances are considered in these simulations, one would conclude that the population is not at equilibrium, whereas it is.

Since the log—log regressions of Nm values on distances seem to detect only some of the information present, other ways of presenting the data have been devised. Once the matrix of Nm values between samples has been calculated, it is possible to represent it on a three-dimensional graph, where the sample numbers are indexed along the x and the y-axes and the z-axis represents the migration. The picture generated will be symmetrical, by construction, with respect to the main diagonal. One expects on such a graph to see high values on the main diagonal for isolation

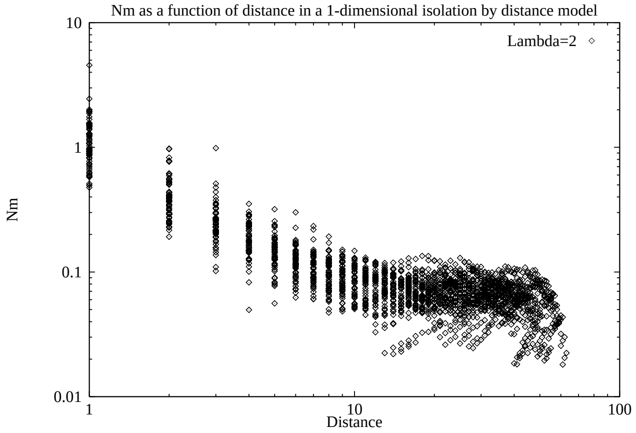

Figure 4.27: log—log plot of Nm on distance

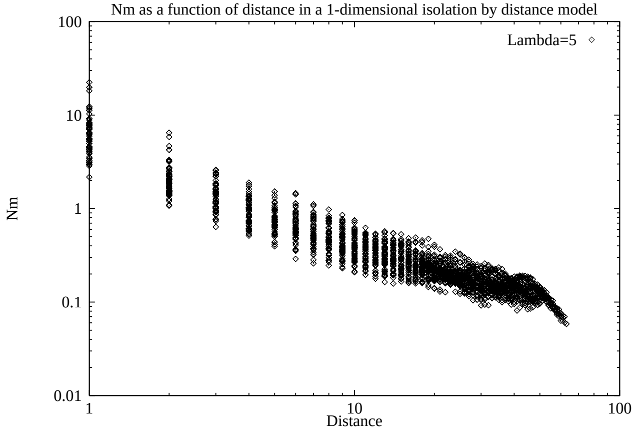

Figure 4.28: log—log plot of Nm on distance

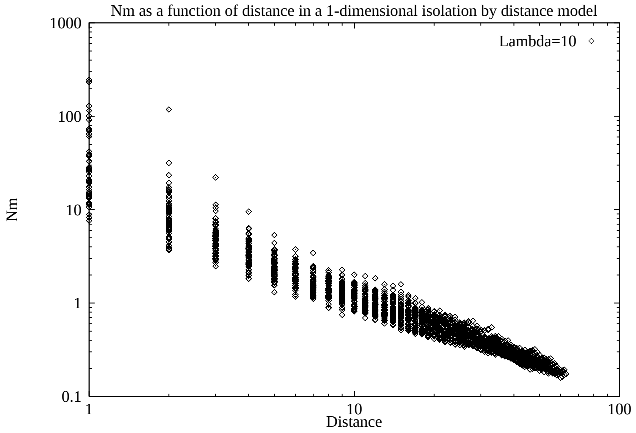

Figure 4.29: log—log plot of Nm on distance

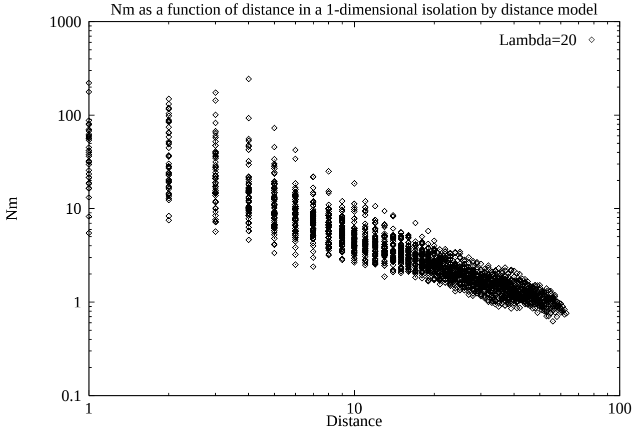

Figure 4.30: log—log plot of Nm on distance

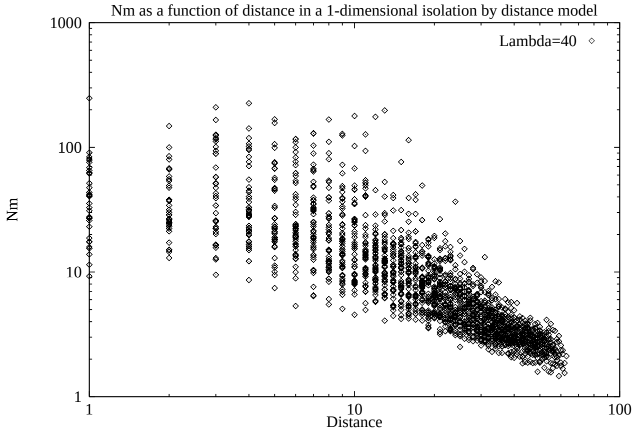

Figure 4.31: log—log plot of Nm on distance

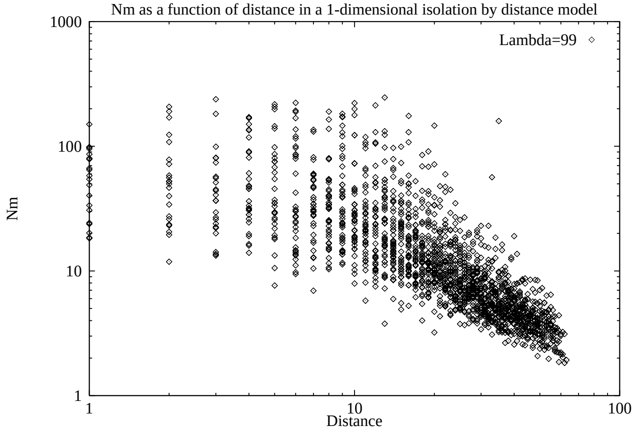

Figure 4.32: log—log plot of Nm on distance

by distance in a one dimensional habitat, since the main diagonal represents samples close one to each others and to see decreasing values as one moves away from this main diagonal. The picture created will therefore look like the crest or the ridge of a mountain landscape. On the other hand, if there is no isolation by distance, no distinctive patterns should emerge from the graph. Figures 4.33 to 4.35 show this graph for an island model of population structure, a two-dimensional, and one-dimensional stepping-stone. The graphs display the expected patterns: 'chaos' for the island model, whereas the main diagonal contains the highest migration values for the one-dimensional stepping-stone model. The pattern observed for the two-dimensional stepping-stone (Figure 4.34) is interesting: first of all, the effect of the ordering of the sample can be seen: the stepping stone was a 4*4 (16 demes) and it is noticeable that values of Nm follow the spatial arrangement (eg. sample 2 and 8). The other interesting feature displayed by this graph is the large variance of migration levels encountered: samples 12 to 16 display higher levels of migration than the others.

Enhancement of these figures is achieved through interpolation of the data using the computer package **UNIRAS**. The data are then transform logarithmically. The outcome is presented in Figures 4.36 and 4.37. The lower left picture represents the untransformed matrix of migration, the lower right the interpolated data and the upper figure the logarithmic transform of the interpolated data. For the one-dimensional stepping-stone model, the upper figure describes perfectly the gene-flow pattern: high migration occurs along the main diagonal and decreases as samples get further apart. Note however the edge effect, characterised by higher migration at the limit of the sampling range and the irregularity of the migration estimate between adjacent samples. This point is important and was already noted by Endler (1977): isolation by distance leads to the occurrence of clines of gene-frequencies that can persist for a long time, even in the absence of selection. This effect is also perceived in two-dimensional gene-flow patterns (Figure 4.37), where some adjacent samples seem to exchange more genetic material than others.

The same treatment has been applied to the isolation by distance model. Results are only given for $\lambda = 20$. F-statistics, from which the migration matrix was inferred, were calculated at a mesh size of 64. The results are presented in Figure 4.38. Note the similarity between the isolation by distance and the one-dimensional stepping-stone model. Once again, distinguishing between these two types of patterns of gene-flow will be quite difficult. If F-statistics are calculated at a smaller mesh size, the observed

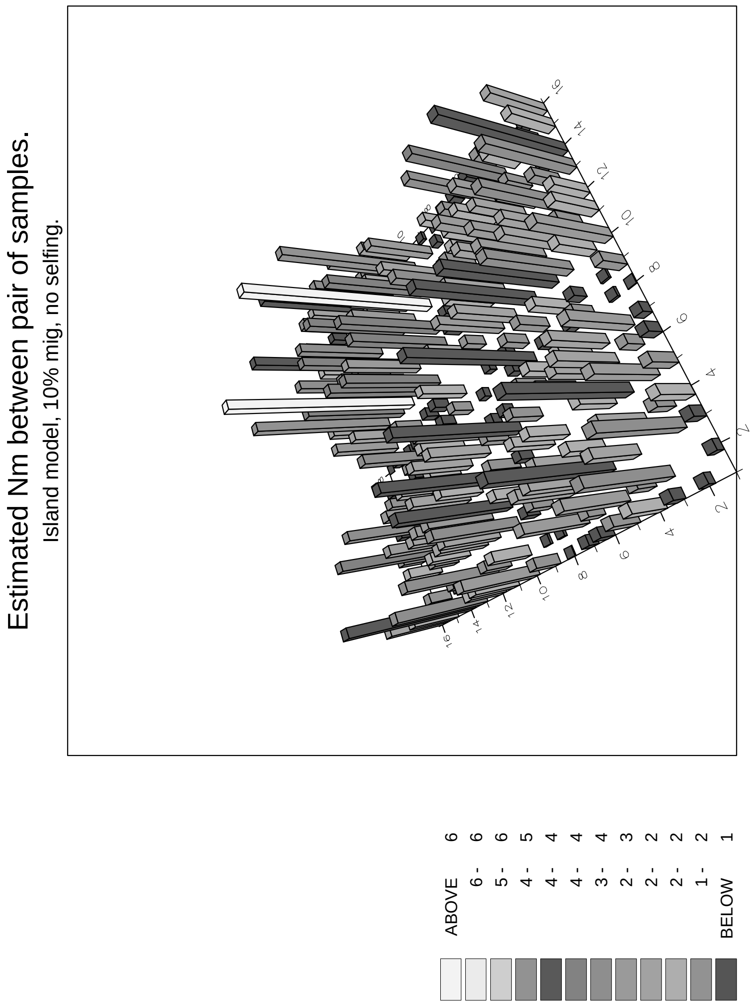

Figure 4.33: Estimated Nm between pairs of samples, island model

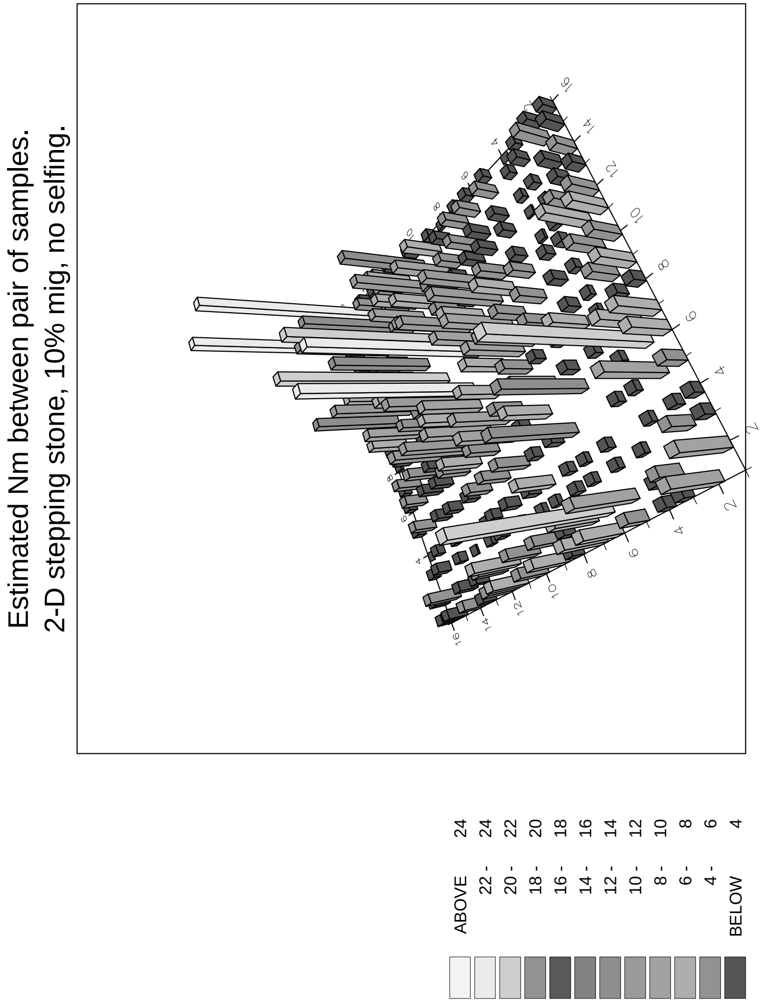

Figure 4.34: Estimated Nm between pairs of samples, two-dimensional stepping-stone model.

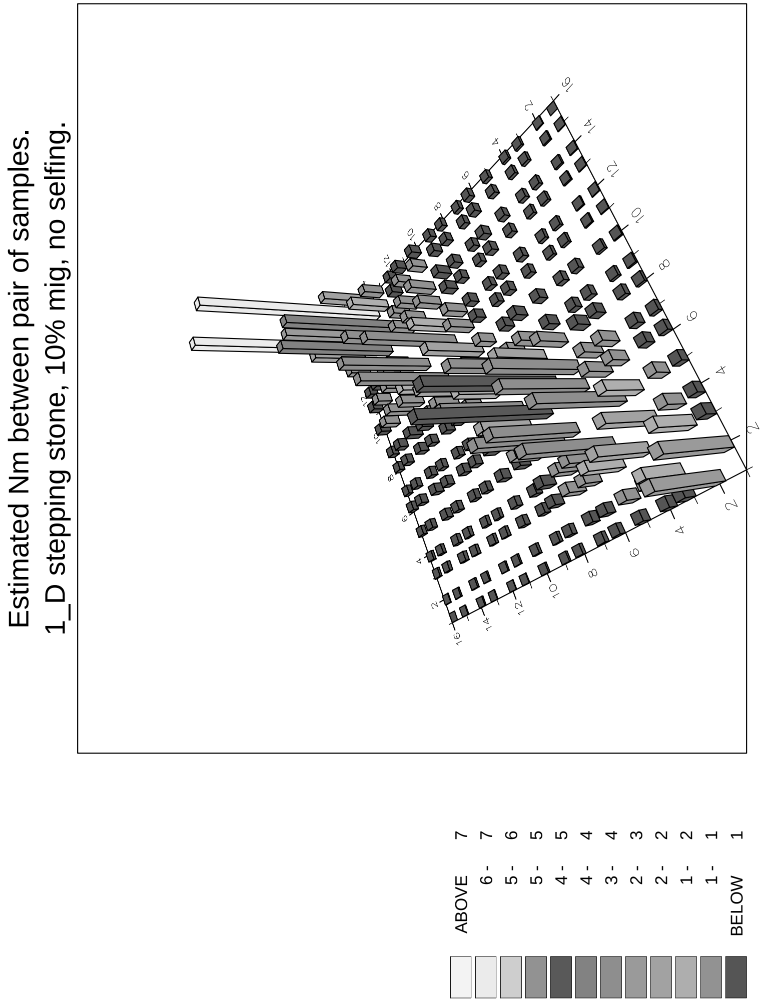

Figure 4.35: Estimated Nm between pairs of samples, one-dimensional stepping-stone model.

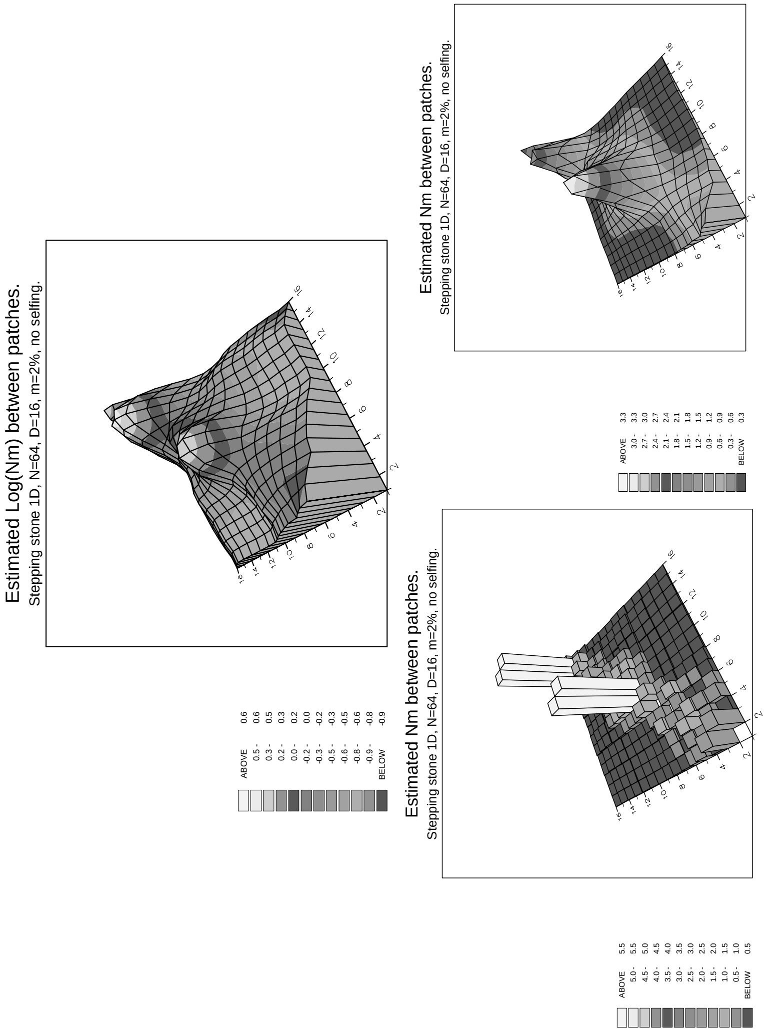

Figure 4.36: Nm between pairs of samples.

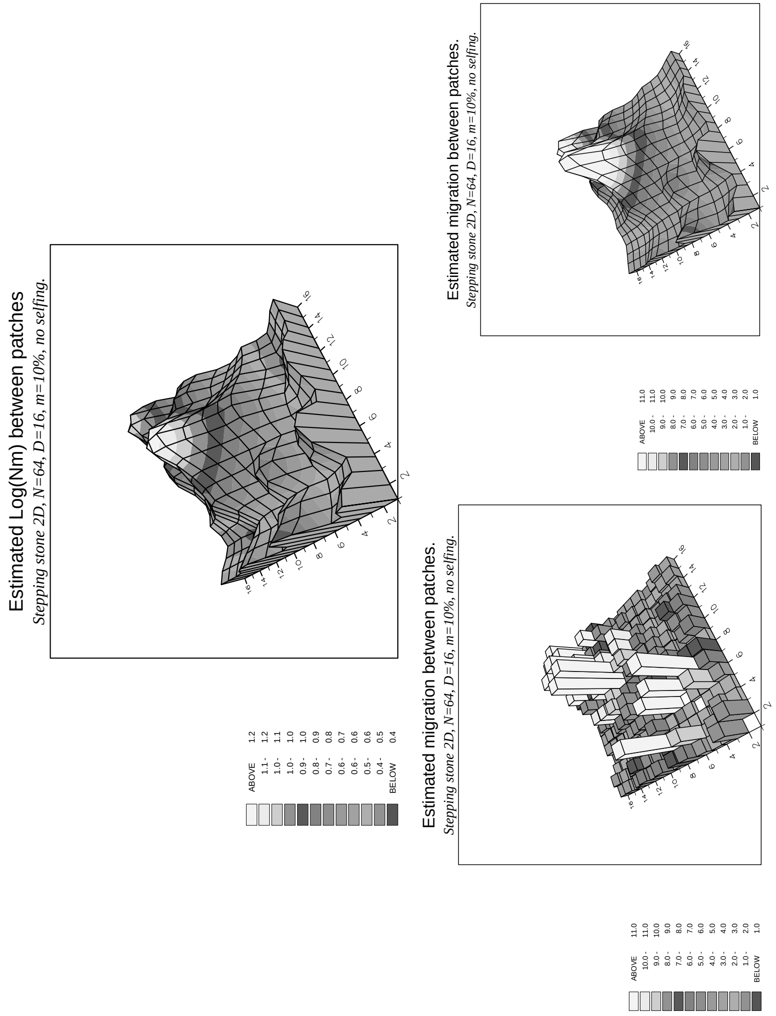

Figure 4.37: Nm between pairs of samples.

pattern is more rugged, but the main feature remains: high migration on the main diagonal, which decreases as demes get further apart.

4.5 The variance effective size, yet again!

It was shown in Chapters 2 and 3 that F-statistics could be useful tools to estimate the variance effective size of a population, since temporal data are not required for these estimations. It would therefore be of interest to compare estimates of the variance effective size measured with temporal data with those obtained from F-statistics. Criticisms could be made regarding the usefulness of estimates of N_e^v based on F-statistics, since they require an estimation of the census size, but Waples (1988) has shown that the estimate based on temporal data is also dependent on the census size. If no estimate of the census population size exists, at least the ratio of the census size to the effective size can be given.

To compare estimators of the variance effective size based on F-statistics (labelled N_eF on the graphs) and those based on temporal changes in allelic frequencies (labelled N_eVar), different patterns of gene-flow were modelled using **MODEL42** and calculation of the two estimators of variance effective sizes were carried out every 200 generations, for 10000 generations for 50 independant replicates.

Figure 4.39 gives the results for an island model of population structure, with 1% migration and no selfing, with a deme size of 16 and 256 demes. The two estimators give similar results, namely an effective size which is more than twice as large as the census size of 4096 individuals. As was discussed in Chapter 2, the variance effective size of a subdivided population is larger than the census size, a feature that needs stressing, since emphases on effective population sizes generally state that they are lower than the census size (Crow & Denniston, 1988; Gale, 1990; Gilpin, 1991; Ballou, 1992).

The other striking feature of this figure is the stability of N_eF, compared with N_eVar. The former is always between 10500 and 10800 whereas the latter ranges from less than 7000 to 13000. This is not surprising, since F-statistics measure the amount of differentiation that has been going on in the population from its foundation, whereas temporal data only take into account the variation that occurred between the two sampled generations.

Also noteworthy is the increase in fluctuation of the variance effective size as time

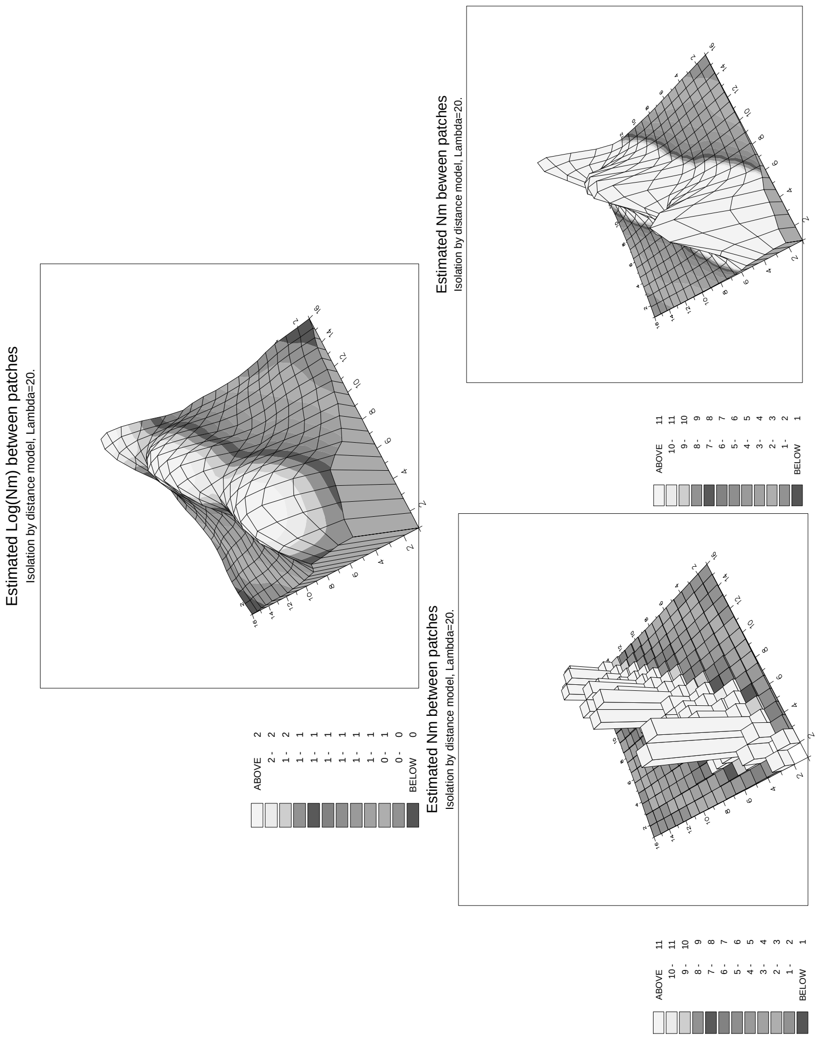

Figure 4.38: Nm between pairs of samples.

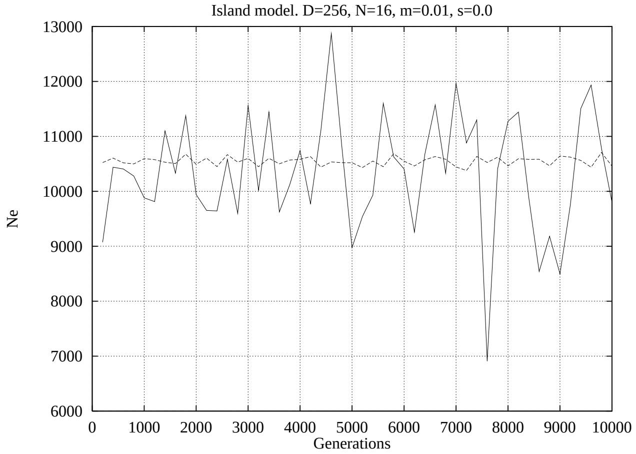

Figure 4.39: Estimates of variance effective sizes.

goes on. This is because there is an erosion of the genetic variability through time.

Next the effect of selfing is investigated (Figure 4.40). The trends are similar to the previous case. The variance effective size is still larger than the census number, even though selfing is present in the population. Fluctuations are larger for N_eVar than N_eF, but the two estimators stay in good agreement. The question as to how much selfing is necessary before its effect anhihilates that of structuring can be found in terms of F_{st} and F_{is}: if $F_{st} < \frac{F_{is}}{1+F_{is}}$ then the variance effective size is smaller than the census size.

This is the situation displayed in Figure 4.41. The average effective size in this case is less than 3000, compared to a census number of 4096. N_eF is still subject to less variation than N_eVar and again the two estimators stay in very good agreement.

The effect of geographical structuring on the variance effective size can be seen in Figure 4.42. The modeled population is a linear stepping-stone model, with 50% migration between adjacent demes. The first characteristic of this graph, compared to populations with no geographical structuring, is the time necessary for equilibrium of the variance effective size to be reached. Values of the effective size do not level off before the 8000th generation. Apart from this, the two estimators are in very good agreement and for the first 2000 generations, they are nearly identical. The amount of migration between adjacent demes (50%) is very large, but after 10000 generations, the variance effective size is 3.5 times the census size.

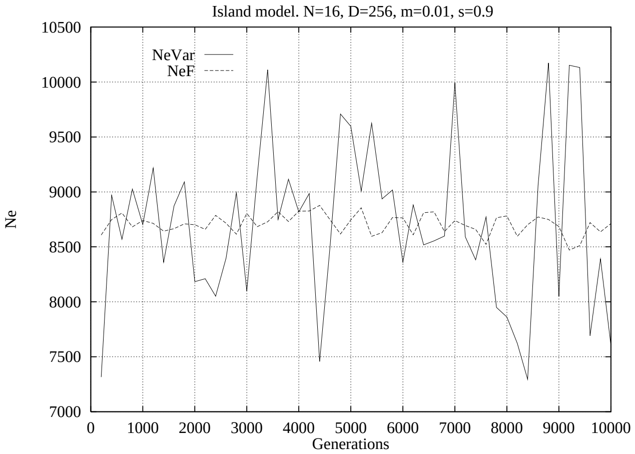

Figure 4.40: Estimates of variance effective sizes

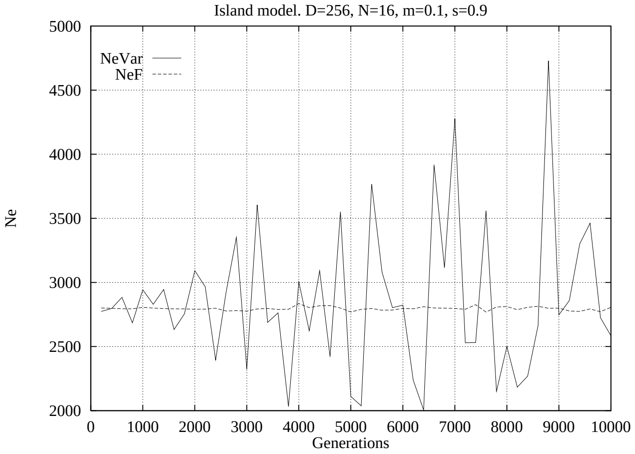

Figure 4.41: Estimates of variance effective sizes

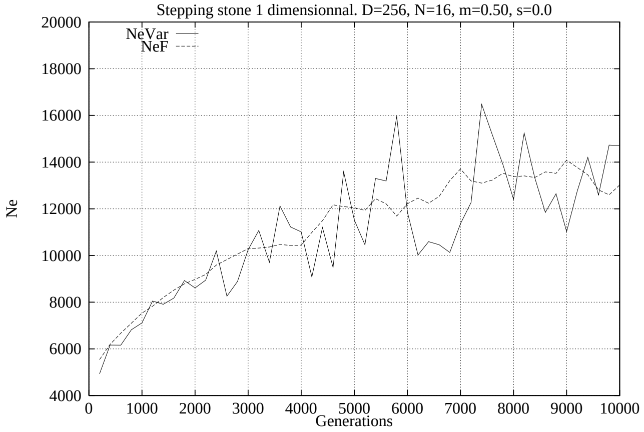

Figure 4.42: Estimates of variance effective sizes

N_eF stays a better estimator of N_e^v since its fluctuations are much smaller than those of N_eVar after the two thousandth generation.

N_eF and N_eVar are both good estimators of the variance effective size. Since the latter does not require spatial information such as the location and size of the demes but only temporal estimates of the changes in allele frequencies, it seems appropriate to use the concept of variance effective size to unmask what has been elusive so far: the neighbourhood size. Measurements of F-statistics were taken at mesh sizes 1, 4, 16 and 64 and N_eF calculated from it. The appropriate neighbourhood size corresponds to the best agreement between N_eVar and N_eF. Figure 4.43 gives the results for a 2-dimensional isolation by distance model, where the input parent-offspring dispersal distances should have led to a neighbourhood of size 4. It is obvious from Figure 4.43 that 4 is not the appropriate neighbourhood size and the best fit is for a sample size of 1. In other words, all the heterozygote deficit in the population is due to structuring and not to selfing or inbreeding. The equilibrium value of the variance effective size in this case is around 8000, twice the census number. This is surprisingly low since a neighbourhood of 4 corresponds to highly restricted gene-flow ($\sigma^2 = \pi^{-1}$, where σ^2 is the variance of the parent to offspring dispersal distance).

A similar pattern is observed in Figure 4.44. The input parent-offspring dispersal distance was such that the neighbourhood size should have been 16. It is clear from the figure that it is not the case and, once again, the best fit of N_eVar and N_eF is for

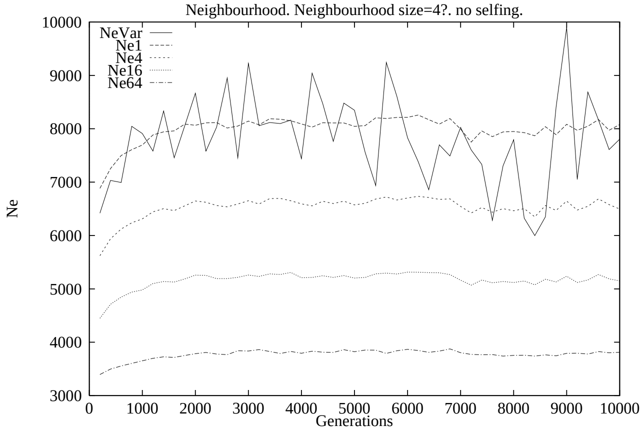

Figure 4.43: Estimates of variance effective sizes. $N_e 1, 4, 16$ and 64 are fore mesh sizes of 1, 4, 16 an 64 respectively.

samples of 1 individual. The average variance effective size this time is 5200, slightly larger than the census size. The scale of the x-axis of Figure 4.44 is different from the other graphs, since the number of generations looked at is only 1000.

4.6 Discussion and conclusions.

Unravelling the structure of natural populations remains one of the main preoccupations of population biologists. They have at their disposal a series of tools that are not necessarily designed to answer the questions they are asking, but which can nevertheless be adapted to meet their needs. It is of crucial importance that the capabilities of each of these tools is clarified and the conditions of application stated. The large number of new techniques to decipher the hidden variability render this task even more difficult and one only needs to read the type of questions being asked in internet news groups such as *bionet.general, bionet.molbio.rapds, bionet.population-bio* to appreciate the problems faced by researchers. The first problem is often one of scale, that is, to ensure that the samples taken are made of a single random breeding unit. For some species, the limit may be obvious, such as a barn in some species of mice, or the troupe in monkeys. In other cases, however, the limits are not easy to find and are often based on previous ecological studies, which measured dispersal of

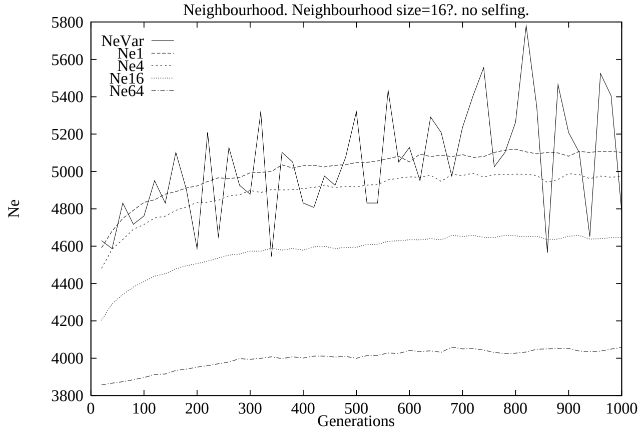

Figure 4.44: Estimates of variance effective sizes

individuals or of gametes (eg. Lamotte 1951, 1959 for the snail *Cepaea nemoralis*; Dice & Howard (1951) for the prairie deer mouse *Peromyscus maniculatus bairdi*; Blair (1960) for the rusty lizard *Sceloropus olivaceus*; Levin & Kerster (1968) for a perennial, insect-pollinated plant *Phlox pilosa*). However, it has been shown in this study that even when samples are taken in an area of the size of the neighbourhood, the deficit of heterozygotes measured within neighbourhood may well still be due to structuring (Figure 4.43 & 4.44). If samples are taken within a random mating area, then detection of its limit can be achieved by pooling recursively samples until an increase in the value of f is seen. This point is important, since, if samples are larger than a random breeding unit, then estimates of F_{st} will be lower than the correct value and, therefore, estimates of migration will be larger than the actuality. To demonstrate this property, Nm was inferred from θ using the approximate relation for the island model, with 1% migration and no selfing and a deme size of 64 ($Nm = 0.64$). The results are given in the following table:

$Mesh$	θ	Nm
4	0.2818	0.64
16	0.2823	0.64
64	0.2846	0.63
256	0.0692	3.36
1024	0.0211	11.60

As long as θ is measured below the deme size, the estimate of Nm is accurate, while it increases dramatically as soon as more than one deme is pooled together.

To test if the changes in the value of f are significant, statistical resampling methods were used. The usual test for significance of F-statistics are based on χ^2 (Li & Horvitz, 1953) and suffers from its limitations. In particular, the numbers of expected genotypes in each class have to be larger than five (Sokal & Rohlf, 1981). As the distribution of allele frequencies in natural populations tends to be U-shaped (Chakraborty et al., 1980; Latter, 1975; Ohta, 1976; Nei, 1987), it is likely that grouping of classes of genotypes will be necessary. On the other hand, resampling tests do not require these assumptions. It was shown that they prove useful for identifying levels of structuring, although jackknife and bootstrap methods seem to provide conservative estimates. Others resampling methods can be used instead, not to provide confidence intervals, but to test if the observed statistic is different from zero and it was shown that stucturing could be detected this way, even when the effective number of migrants per deme is as large as 20 in an island model.

Investigation of stepping stone models showed interesting patterns. It is still possible to detect structuring and to find the limit of the random breeding unit, but the changes in f after the pooling of more than one deme are not as apparent as in the island model. It was also noted that the confidence interval of θ is much larger than in island models, meaning that fluctuations in allelic frequencies from one replicate to the next are larger in the former than the latter.

Since estimates of F_{st} are often used to infer migration levels, it seemed necessary to review the suitabilty of application of the relationship between these two factors. It was shown that the usual approximation $F_{st} = 1/(4Nm + 1)$ is valid only for large values of F_{st} and was further contingent upon large sample sizes. Attention was drawn to this problem, with the increasing cost and time needed to unravel genetic variability when using new molecular techniques such as RFLPs and VNTRs, data sets are tending to become much smaller.

Since f changed as a function of the grouping of the samples and displayed a discontinuity at the level of the breeding unit, isolation by distance models in a one dimensional habitat were generated to assess if it was possible to measure indirectly the size of the neighbourhood area. These investigations showed that although f increased with the size of the sampled area, one could not perceive any discontinuities in these changes and that statistically significant differences between f's were not an appropriate measure of Wright's neighbourhood size whereas a method developed by Slatkin (1993), based on the log—log regression of estimates of Nm on distances, gave a good estimate of the neighboorhood size, apart for very large dispersal distances. On the other hand, this method assumes a linear relationship between $\log(Nm)$ and $\log(Distance)$. This linear relation was shown to exist in only certain cases, for intermediate values of the dispersal distances.

A graphical representation of the migration matrix looks to be a promising way of displaying the information and should allow the discrimation of species undergoing isolation by distances from species where there is no isolation by distance. The same graphical representation also allowed the discrimination between habitat structure of different dimensions.

Comparisons of estimates of the variance effective sizes, made using temporal data and spatial data, were carried out. Both estimates were in good agreement and displayed a trend seldom emphasized in the literature: *the variance effective size of a subdivided population can be larger than the census size.* It was, however, obvious that estimates based on temporal data are less accurate than those based on spatial data. On the other hand, to get an accurate estimate from spatial data, one needs to know the population structure of the species investigated. This knowledge is not required for temporal data. As deme size cannot be detected in isolation by distance models, then if only spatial data were available and if an estimate of effective population size were needed, this leads to a 'Catch22' situation. Comparisons of estimates of the variance effective size based on temporal data with those based on spatial data measured at different scales demonstrate that the concept of neighbourhood size is flawed. It may be a useful measure of parent to offspring dispersal distance but this should not be considered as a random breeding unit, that is, in any way, comparable to a deme in island or stepping-stone models.

Chapter 5

Applications to data from natural populations

5.1 Introduction

Investigation of the genetic structure of natural populations has monopolised the interest of population biologists for fifty years since the early work of Wright on the desert snow *Linanthus parryae*. The number of these studies has grown exponentially after the discovery of protein gel electrophoresis in 1966. These studies are usually intended to answer an evolutionary question but have been referred to as the *'Find'em and grind'em'* school of population genetics by some (Lewontin, 1991). The work presented here belongs to another category of population genetics studies that could be called the *'Find'em and scrounge'em'* school, as I did not myself collect any of the data presented here. I am indebted to Amanda Day for the dogwhelk data and to Alan Raybould and Alan Gray for the data on beet and cabbage.

Natural populations possess many very undesirable properties for the population geneticist, because they never seem to comply with the requirements of theoretical models. The population geneticist's task is therefore to find means of getting samples from natural populations to conform with the assumptions of one or the other of these models. Chapter 4 presented methods for assessing population structuring which were tested on computer generated data-sets corresponding to known structure of populations. The results of these investigations generally takes the form of an expected behaviour of some statistic when measured in a population with given parameters (increase in f when more than one breeding unit is pooled together, significant regression between log(Nm) and log(*distance*) when there is isolation by distance, chaotic pattern in the migration matrix for an island model of population structure). These processes were inductive. In the following, a deductive process will be presented: given the behaviour of a particular statistic, can biological parameters be deduced or inferred? (Chalmers, 1976).

5.2 *Brassica oleracea* ssp. *oleracea*

Known by the common name of wild cabbage, it is native to the coast of northwestern Europe as well as the Mediterranean (Thompson, 1976). Usually disliked by most children, probably because of a French legend which says that little boys are found under their leaves (as opposed to little girls, who are found under roses), the origin of its name stems from the old Norman French word, *caboche* (Collins, 1992), which meant head. This also explains the expression 'Cabbage head' describing somebody who is rather simple-minded.

Brassica oleracea is a polymorphic diploid species, containing many cultivars, such as *B. cauliflora*, the cauliflower, *B. oleracea* var. *gemmifera*, the Brussel sprout, *B. oleracea* var. *italica*, the broccoli and *B. oleracea* var.*capitata*, the cabbage (Thompson, 1976). However, it is doubtful whether the many cultivated species of *B. oleracea* evolved solely from the wild cabbage and several other wild diploid relatives such as *B. cretica*, *B. insularis* and *B. rupestris* may have contributed (Yarnell, 1956).

This species complex displays a strong self-incompatibility (Thompson & Taylor, 1966) which only tends to disappear in lines that have achieved greater uniformity through intense selection.

Since this polymorphic species i) is a typical outcrosser, ii) exists in cultivated as well as wild forms, iii) is likely to undergo genetic manipulation for crop improvement (Raybould & Gray, 1993), it would seem to be a judicious choice for use as a biological model of gene-flow between crops and their wild relatives.

5.2.1 Material and methods

A core population of 400 individuals divided into 20 patches of potentially interbreeding individuals was sampled from a more or less linear habitat on a stretch of the coastline of Dorset, Southern England. All patches were located on cliff-tops along a 30 km section of coastline between the Foreland (SZ 055824, east of Swanage) and

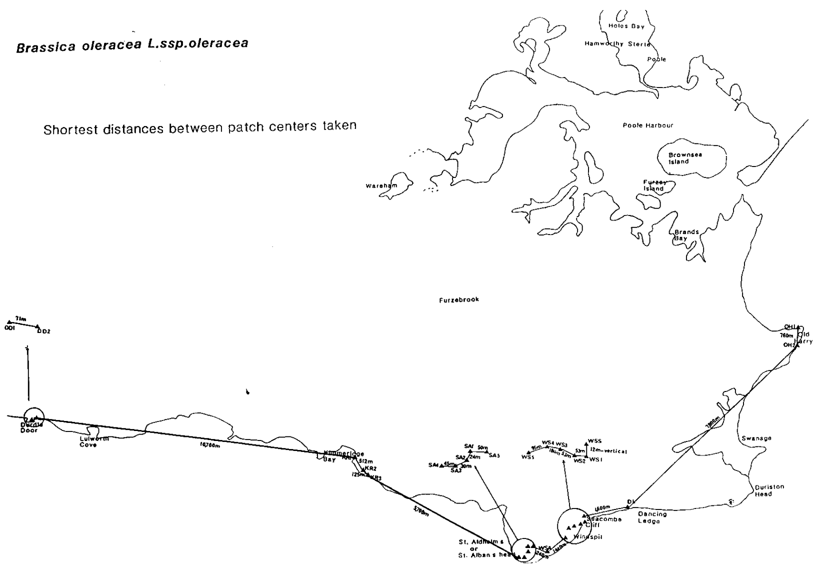

Figure 5.1: Samples location of *Brassica oleracea* ssp. *oleracea*

Durdle Door (SY 805803, west of Lulworth Cove) (Figure 5.1). At Windspit and St Aldhelm's Head, five patches were taken from more or less continuous populations. The remaining ten patches were taken at Durdle Door (2 samples), Old Harry (3 samples), Dancing Ledge, West Man and Kimmeridge (3 samples). For each patch, samples of leaf tissue were taken from 20 adult flowering plants (Gray *et al.*, 1992). In this species pollination is insect mediated, the main pollinators being, in this location, the bumble bee species *Bombus lapidarius* and *B. terrestris* and the bee *Anthophora plumipes* and *Apis mellifera*. It was noted (Gray *et al.*, 1992) that little competition for pollinators exists, since *Brassica* flowers before most species, but that there may well be a scarcity of pollinators. Also, the behaviour of pollinators appears to be strongly influenced by flower density, bees generally preferring high density patches.

Three out of 11 electrophoretic loci were found to be suitable for analysis (polymorphic), *SDH-2*, *PGI-1* and *APH-2* (Gray *et al*, 1992). The genotypic distribution in each population at these 3 loci can be found in Appendix E in a form suitable for input into the program FSTAT (Appendix D).

To assess whether samples corresponded to random breeding units, each was subdivided into 4 subsamples, with subsamples grouping together the closest individuals, according to a per-sample map provided by Alan Gray and Alan Raybould. The different fixation indices were calculated for this group of 80 samples. Fixation indices were then calculated for the 20 original samples. Samples were subsequently pooled as a function of distance, with the third pooling level for samples less than 25 metres apart, the fourth for samples less than 100 meters apart and so on. The different pooling stages are summarised in Table 5.1 (from right to left on the map). OH stands for Old Harry, DL for Dancing Ledge, WS for Wind Spit, SA for St Alban, KR for KimmeRidge and DD for Durdle Door. Populations at St Alban and WindSpit were more or less continuous, with the different samples at these locations being somewhat arbitrary. Pooling of these two groups occurs between stages 3 and 5 ($25m < x < 500m$).

Table 5.1: The eight levels of pooling of samples for *Brassica oleracea* ssp. *oleracea*.

Pooling	distance	Pooled samples				
1	<4m					
2	<10m					
3	<25m		WSS-1	SA1-2		
4	<100m		WSS-3, WS4-5	SA5-1		DD1-2
5	<500m		WSS-5	WSA-SA1	KR3-2	DD1-2
6	<2km	OH1-2	DL-WS5	WSA-SA1	KR3-1	DD1-2
7	<4km	OH1-2	DL-SA1		KR3-1	DD1-2
8	>4km	ALL SAMPLES TOGETHER				

5.2.2 Results

Appendix F gives the raw output of the program **FSTAT** (Appendix D). It can be detailed as follows:

- The allele frequencies per sample, as well as the size of each sample, for each allele at each locus. The observed and the expected heterozygosity per allele, for each sample and locus is then given (The expected heterozygosity is calculated using a hyper-geometric distribution, that is, instead of being $2Np(1-p)$, it is $4Np(1-p)/(2N-1)$).

- f per allele per sample. If the allele is not present in the sample, the table contains question marks, as f is undefined.
- Estimates of F, θ and f per allele and locus.
- The overall F, θ and f.
- The jackknife mean and standard deviation over samples, per locus.
- The jackknife mean and standard deviation over loci.
- The bootstrap over-loci confidence interval at 95% and 99%.
- The Pairwise θ estimates per locus.
- 95% and 99% confidence intervals for the null hypothesis that f is equal to zero (against the alternative hypothesis that f is larger than zero). The probability that the observed f is equal to zero.
- 95% and 99% confidence intervals for the null hypothesis that F is equal to zero (one sided test). The probability that the observed F is zero.
- 95% and 99% confidence intervals for the null hypothesis that θ is equal to zero (one sided test). The probability that the observed θ is zero.

The estimate over loci of the pairwise θ is used to calculate the pairwise Nm, written into another file.

The overall estimates of $F = 0.328$, $\theta = 0.178$ and $f = 0.232$ show that there is a deficit of heterozygotes both within and among samples.

Within sample heterozygote deficit, as measured by f, is positive at the three loci (Figure 5.2). Jackknifing over samples lead to large confidence intervals (CI), an indication of the large variance of heterozygote deficit within samples. The largest is for *SDH-2*, with sample DL displaying an f of 0.8, while sample SA5 has an f of -0.15. Bearing in mind that there is a self-incompatibility system in *Brassica* (which should lead to an excess of heterozygotes and therefore, a negative f), this is a first indication that samples are larger than the random breeding unit.

Figure 5.3 displays the effect on f and θ of the pooling strategy. A sharp increase in f (decrease in θ) occurs up to 100m and then levels out. This is an indication of strong population structuring with gene-flow being restricted even at the level of samples (note the increase between level 1 and 2, i.e. sub-samples vs samples.). Comparison

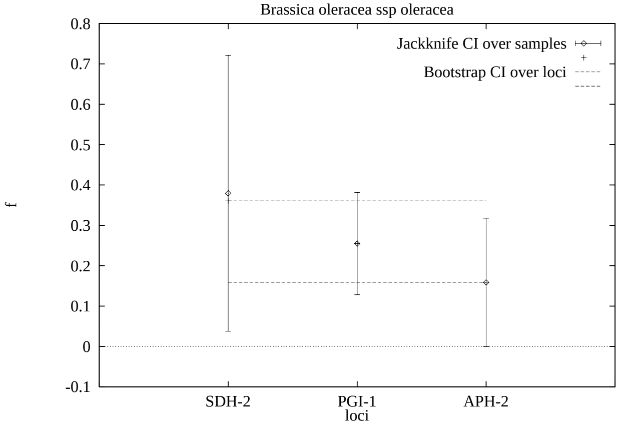

Figure 5.2: Jackknife CI over populations and over loci for *B. oleracea*

with the investigation of one-dimensional stepping stone and one-dimensional isolation by distance models (cf. Chapter 4) suggests that the average distance of dispersal must be very restricted (the rate of increase in f is similar to the shape for a λ of 1 in the isolation by distance model, Figure 4.16). Also, no real plateau is reached (slight increase of f up to the last point), an indication that isolation by distance occurs even for long distances. Focussing now on the Jackknife CI of f, there are no statistical differences among f's measured below 25m, but f measured for samples covering 100m is different from f estimated from the original samples.

Subsequently, Slatkin's method (1993), described and applied to isolation by distance models in Chapter 4, was used: θ was calculated for each pair of samples and the distance between samples was recorded. Nm was inferred from the pairwise θ values. A linear regression was carried out after a log—log transform of the data (Figure 5.4). The regression equation obtained for *Brassica* was:

$$\log(Nm) = 0.84 - 0.17 \log(distance)$$

where the distances are expressed in metres. Although the regression coefficient is significantly different from zero, the percentage of the variance explained by the regression is very low ($R^2 = 0.05$). We would expect to obtain a graph similar to Figure 4.26 but there are not enough points representing short distances to build up such a picture. Furthermore, the data set is based on only three loci and is much more

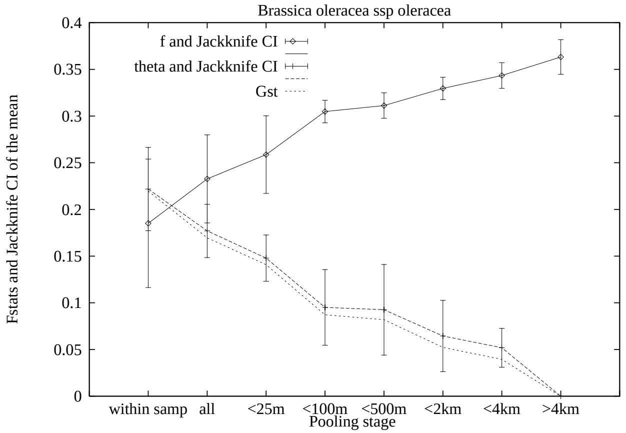

Figure 5.3: Changes in f and θ with the grouping of samples

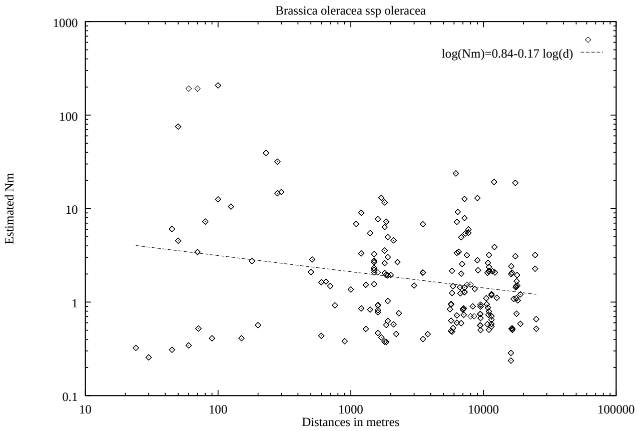

Figure 5.4: log—log regression of Nm on distance. *Brassica oleracea*

variable than a data set obtained from computer simulations. This analysis therefore falls short of a satisfactory explanation for the patterns of gene-flow occurring in *Brassica*.

Figure 5.5 displays a 3-dimensional plot of estimated migration per pair of samples. Ordering of samples along the x- and y-axes goes from right to left on the map, starting at OH1 and ending at DD2. The lower left graph gives the estimated migration, while the lower right graph displays a surface generated by interpolation of the data set with the computer package **UNIRAS**. The top graph represents a log transform of the bottom right. The emerging pattern looks quite dissimilar to the modelled one-dimensional gene-flow patterns. One main peak can be observed, which corresponds to one of the continuous populations sampled, Windspit. The amount of gene-flow between patches in this continuum is very high. Surprisingly, however, high levels of gene-flow are not observed in the other continuum, St Aldhem's Head. For the rest of the samples, distance between samples does not seem to be a very good predictor of the amount of gene-flow. The observed migration landscape appears to correspond to a species living in a habitat made of more than one dimension.

5.3 *Beta vulgaris* ssp. *maritima*

Known by the common name of sea beet, this subspecies is thought to be an ancestor of most, if not all, cultivars. Most of these cultivars belong to the sub-species *B. vulgaris* ssp. *vulgaris* and include sugar beets, beetroots, mangolds and fodder beets (Campbell, 1976). *Beta* is an old world genus virtually confined to Europe. Its use probably dates from prehistoric times. The Romans used *Beta vulgaris* ssp. *maritima* as feed for animals and man. It was taken from Italy to northern Europe by the barbarian invaders. Because the British blockaded the French ports, thereby creating a shortage of cane sugar from the West Indies, Napoleon published (in 1811) a series of decrees requiring beet to be grown and studied in schools. This led to a rapid improvement in sugar content through mass selection on multilines (twenty to thirty parental stocks, Campbell, 1976).

B. vulgaris ssp. *maritima* is diploid ($2n = 2x = 18$), largely anemophilous and outbred (Dark, 1971). It has, however, been classified as both anemophilous and insect pollinated by Raybould & Gray (1993). Its clifftop habitat is very similar to that of wild cabbage but it also occurs along driftlines in bays. The habitat could

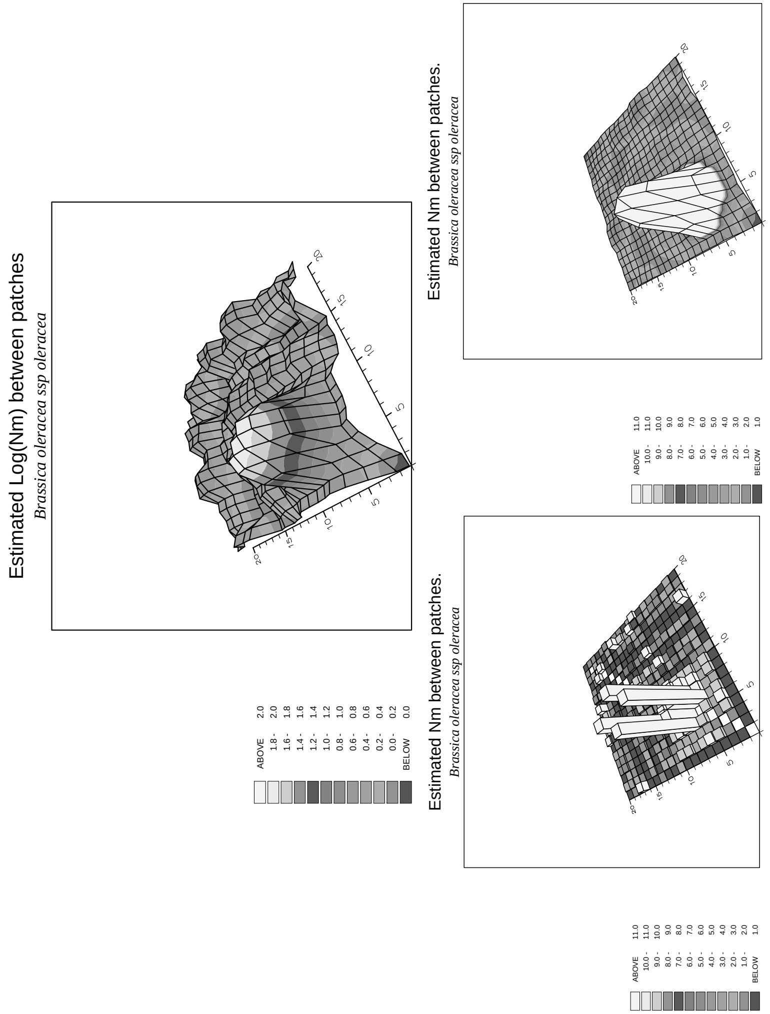

Figure 5.5: Estimates of migration between patches of *Brassica*

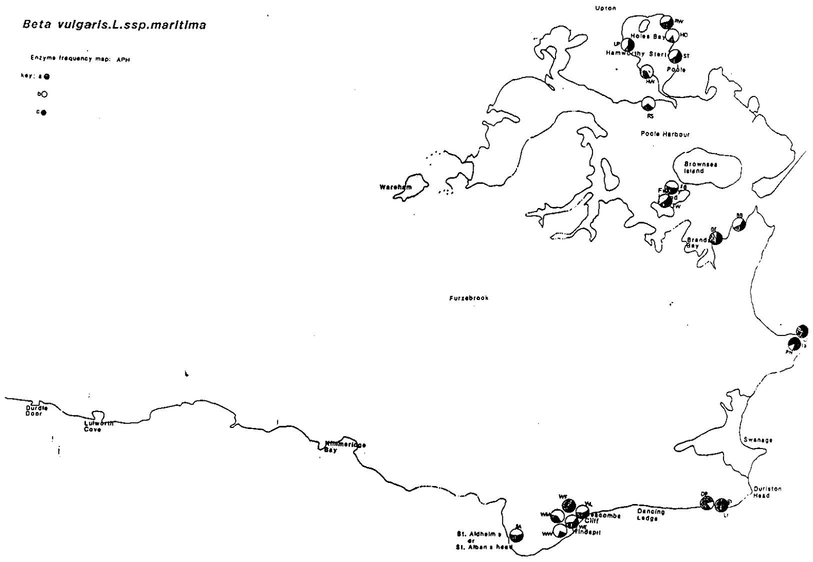

Figure 5.6: Samples location of *Beta vulgaris* ssp. *maritima*

be characterised as linear as far as cliff-tops are concerned but not for driftlines. Its usefulness as a biological model stems from the same considerations as for *Brassica*: *Beta vulgaris* is outcrossed, exists in cultivated as well as wild form and is likely to undergo genetic transformation (Raybould & Gray, 1993). Evidence for gene exchanges between wild and cultivated forms has already been published (Santoni, 1993; Santoni & Berville, 1992). A thorough appraisal of gene-flow patterns in this species complex seems, therefore, of prime interest.

5.3.1 Material and methods

Sampling took place along the Dorset coastline, Southern England. Its exact location overlaps that of *Brassica*. A core population of 400 individuals was divided into two major groups, ten patches from driftline populations and ten from cliff-top populations (Figure 5.6).

The first group comprised five patches from a stone embankment and the upper levels of saltmarshes around Holes Bay, a small bay on the northern edge of Poole Harbour and five patches at greater distance from one another around the shores of

Table 5.2: The eight levels of pooling of samples for *Beta vulgaris* ssp. *maritima*.

Pooling	distance	Pooled samples					
1	<4m						
2	<10m						
3	<25m	WE-WF	DP-LI				
4	<80m	WM-WL	DP-LI	PN-OH			
5	<500m	WW-WL	DP-LI	PN-OH		FB-FW	OH-RW
6	<2km	SA-WL	DP-LI	PN-OH	BE-BB	FB-FW	RS-HW, ST-RW
7	<4km	SA-WL	DP-LI	PN-OH	BE-FW		RS-RW
8	>4km	ALL SAMPLES TOGETHER					

Poole Harbour on shingle banks and tide lines (two on Furzey Island, one at Rockley Sands and two in Brand's Bay). The cliff-top patches, scattered from St Aldhem's Head to the Foreland included five populations at Windspit. Leaf samples from adult plants were taken for electrophoresis.

Six loci out of 13 showed polymorphism. The first 3, *Got-3, APH* and *SDH* were described in Gray *et al.* (1992). The last 3 *PGI, PER-1* and *MDH* were not because of difficulties in the interpretation of the gels. These problems now seem to have been resolved (Raybould, pers. comm.). Loci *PGI* and *PER-1* are included, although some sampled patches are missing (FB,HW,HO and RW for *PGI* and FB, FW and ST for *PER-1*). Results for these loci should be treated with caution because of the missing populations and also because of difficulties in interpreting the gels. The pooling procedure was similar to that adopted for *Brassica oleracea* ssp. *oleracea* and is summarised in the Table 5.2 (from left to right on the map). Pooling of the samples at WindSpit occurred between level 3 and 5, while pooling of the samples of Poole Harbour occurred between level 5 and 7.

5.3.2 Results

The raw results are presented in Appendix G, in the same format as for *Brassica*. The overall F was -0.08, θ was 0.167 and f was -0.295. Thus, there was a significant excess of heterozygotes within samples, whereas there was a significant deficit of heterozygotes between samples. f and θ cancel out for the global heterozygote deficit

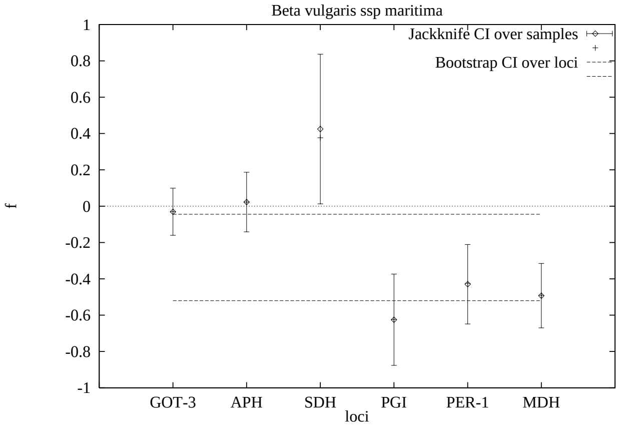

Figure 5.7: Jackknife CI over populations and over loci for *B. maritima*

F. This can be better understood by looking at Figure 5.7, which displays values of f per locus, together with over-samples Jackknife CI and over-loci bootstrapped CI.

GOT-3 and *APH-2* are not significantly different from zero, whereas *SDH* presents a significant heterozygote deficit and *PGI, PER-1* and *MDH* a significant excess. As already mentioned, the three loci displaying excess heterozygosity need to be treated cautiously because gel interpretation was difficult (A. Raybould, pers. comm.), and because some samples gave uninterpretable results. The three loci displaying an excess of heterozygotes may be linked to the self-incompatibility system found in *Beta*. *SDH* shows a strong deficit of heterozygotes and the two confidence intervals do not overlap (Jackknife over population and Bootstrap over loci). Furthermore, one sample (FB) at this locus was fixed for an allele at very low frequency in all other samples. This explains the structure of the matrix of pairwise θ for this locus (Appendix G) where most rows and columns are empty because the same allele is fixed in most populations. For this reason, the strategy for pooling samples together is divided into four steps: first of all, we analyse the six loci together. This leads to Figure 5.8. Notice first that f increases from the leftmost point, an indication that the population structure follows an isolation by distance model and that gene-flow is very restricted. f is negative for all the pooling levels, that is, even when samples more than 4km apart are pooled together there is still a deficit of heterozygotes. From the stand point of detecting random breeding units, confidence intervals for samples more than 4km apart with

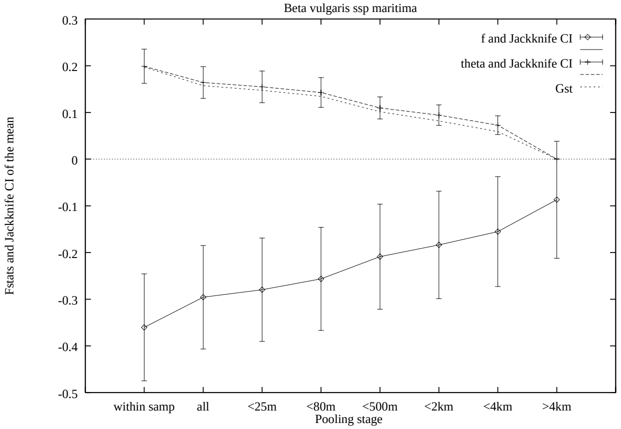

Figure 5.8: Changes in f with levels of grouping for all loci.

subsamples of the original are not statistically different. A negative f will also have some influence on θ by lowering it. Figure 5.9 displays the same analysis after removal of *SDH*. The behaviour of the changes in f is essentially similar to the previous graph, which is not surprising since *SDH* is nearly monomorphic (absent in most samples, apart from FB where its frequency reaches 0.89). It should also be noted that the Jackknife mean and standard deviation of θ and F are out of the domain of definition of these two statistics (cf. Appendix G). This is because sample FB behaves as an outlier and it has been pointed out in Chapter 4 that Jackknife statistics are very sensitive to outliers.

Figure 5.10 displays the results for the three loci displaying negative f.

Once again, the general trend is the same, although the values for θ are smaller.

Figure 5.11, at last, displays the results for the two 'well-behaved' loci. The changes in f follow the same trends as previously but are emphasised. The confidence interval displayed on this graph should not be taken too seriously because they are based on only two measures.

Even though a high level of heterogeneity exists among these loci, all graphs showed the same trends. f is increasing with the pooling level, a sign that there is some isolation by distance. This increase starts from the first point, which corresponds to f measured within samples (an area of four metres squared). It is tempting to conclude from this analysis that there is no such entity as a random breeding unit in

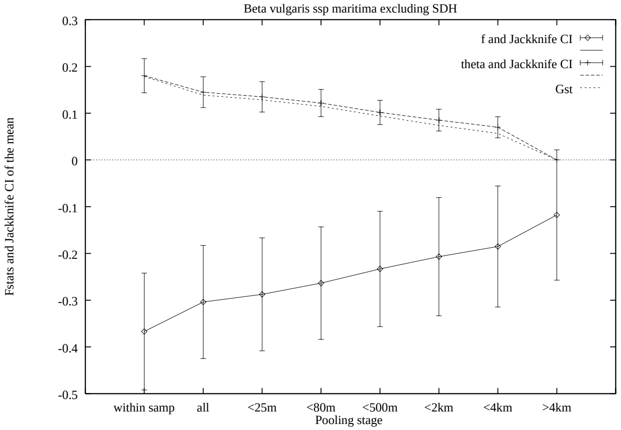

Figure 5.9: Changes in f with levels of pooling, *SDH* excluded.

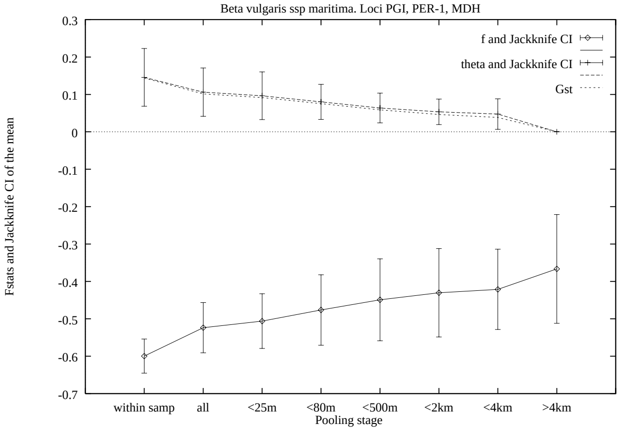

Figure 5.10: Changes in f with levels of pooling for *PGI, PER-1* and *MDH*

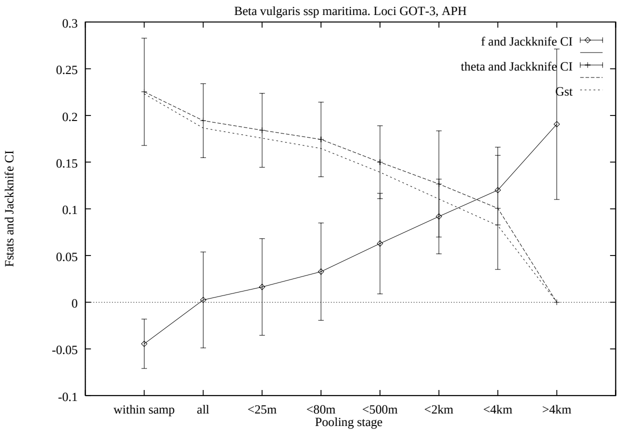

Figure 5.11: Changes in f with levels of pooling for *GOT-3* and *APH-2*

Beta vulgaris ssp. *maritima*.

Slatkin's method (1993) was applied and the results are displayed in Figure 5.12. The slope of the regression is not significant and the percentage of the variance explained by the model is very small ($R^2 < 0.01$). Also, data for very short distances were very scarce, as was the case for *Brassica*.

The equation for the regression is:

$$\log(Nm) = 0.51 - 0.09 \log(distance)$$

For this species too, it seems that detection of isolation by distance with Slatkin's method (1993) is compromised, whereas the technique of grouping samples and recalculating f for each level of grouping provides some evidence that isolation by distance is occurring.

The sampling strategy was designed to allow a comparison of samples from the bay (driftline populations) and from the cliff-tops. The two methods used above are not particularly well suited to this type of analysis, although it would have been possible to divide the samples into two groups and to carry out an analysis on each group. The graphical method presented first in Chapter 4 and used on *Brassica* seems a more appropriate way of distinguishing between these two groups. This is illustrated in (Figure 5.13). The cliff-top samples (1 to 10) on this figure do not seem to correspond to a one-dimensional habitat since there is no sign of increased migration along the

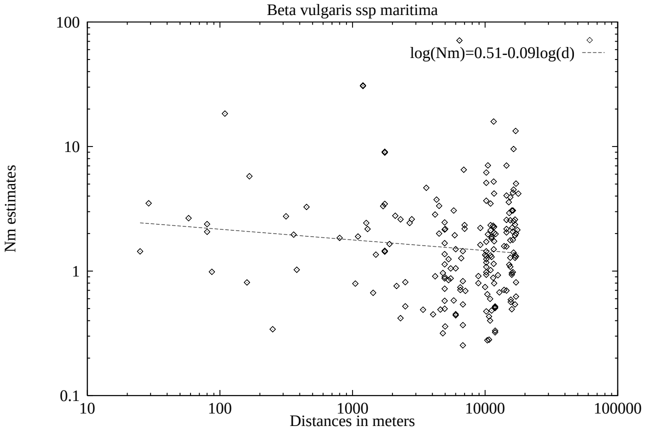

Figure 5.12: log—log regression of Nm on distance. *Beta vulgaris* ssp. *maritima*

main diagonal (neighbouring samples) for the first part of these graphs (Figure 5.13). However, these samples are easily distinguished from the driftlines populations which display much higher levels of gene-flow (samples 11 to 20). For this wind pollinated species, it therefore seems that gene-flow occurs mainly through a process of extinction and recolonisation which would be more frequent in bays (e.g. storms) than on the less disturbed habitat of a cliff-top. A consequence of this observation is that gene-flow occurs probably more through seed migration than through pollen dispersal.

5.4 The dogwhelk, *Nucella lapillus*

The dogwhelk *Nucella lapillus* is a widely distributed predatory intertidal gastropod, feeding on mussels and barnacles. It is ubiquitous on rocky substrates around the coasts of Great Britain and Ireland and extends from Iceland to Portugal (Berry and Crothers, 1974). The main reason for choosing this species rather than any others resides in its dispersal behaviour: restriction in adult movement to only a few metres only (Hughes, 1972), associated with the absence of a dispersed planktonic stage, are likely to result in pronounced local differentiation of subpopulations. Furthermore, since *Nucella lapillus* is restricted to the intertidal zone, it seems a very good biological model to test for the levels of connectidness in this habitat.

Shell shape is different in exposed and sheltered sites: whelks from exposed sites

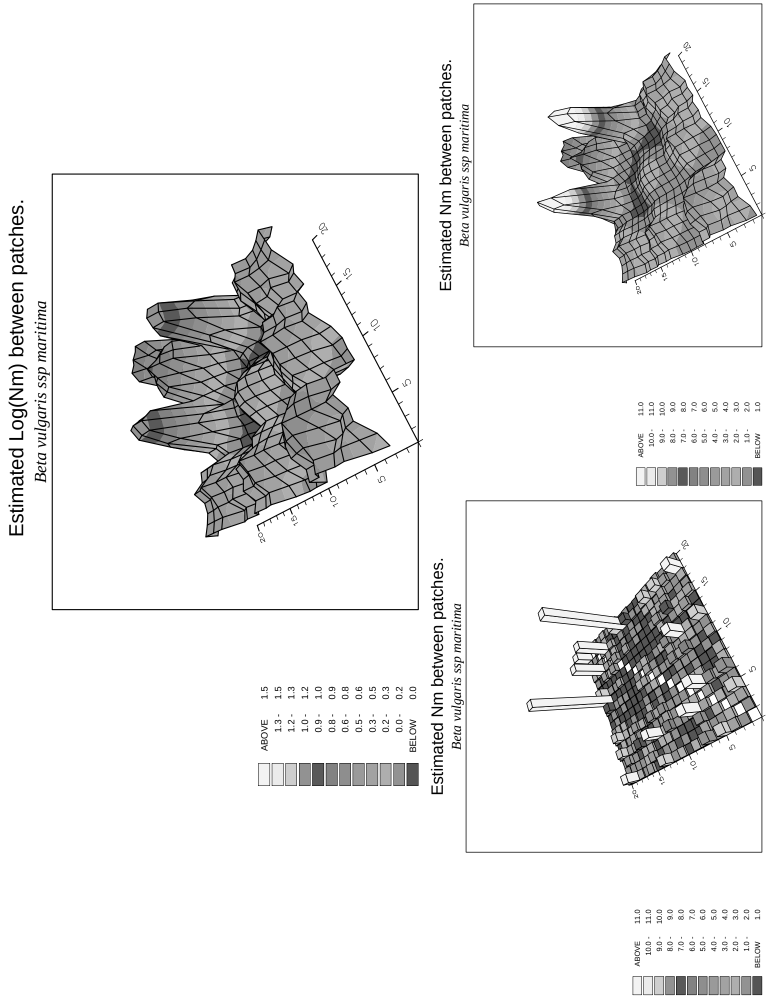

Figure 5.13: Estimated Number of migrants between patches of *Beta vulgaris* ssp. *maritima*

have a thinner, shorter shell than whelks from sheltered sites and a larger aperture that allows them to resist wave action. On the other hand, a thicker shell allows whelks from sheltered sites to resist the action of predatory crabs during their growth (Currey & Hughes, 1982).

Chromosome number has been found to vary between sheltered and exposed sites, with $2n = 26$ for exposed sites and $2n = 36$ for sheltered sites. This polymorphism is due to Robertsonian translocations (Bantock & Cockayne, 1976). However, when no chromosome number polymorphism can be found, the number of chromosomes is $2n = 26$ and this is the case in most of the British Isles apart from the English Channel (Bantock & Cockayne, 1976).

5.4.1 Material and methods

Allozyme data were obtained by A.J. Day (1990) on dogwhelks from 15 sites 50m to 21 km apart in S. Devon, S.W. England (Figure 5.14).

Sites 1 to 5 (between Blackpool and Start Point) are very exposed to wave action and are quite distant from each other (0.8-6.7 km). All sites with whelk populations on this stretch of coastline were sampled. These populations were usually dense, with easily identified breeding aggregations in crevices. Each sample consisted of all the whelks within a single aggregation (Day,1990). To the south of Start Point, the coastline is mainly sheltered. Ten sites were sampled along this strip of coast, one at Start Point itself (6 individuals), three around Lannacombe Bay, and six to the east of Prawle Point. At both Prawle and Lannacombe, the samples came from a 0.5km stretch of coastline and the distance between samples was 50–150m at the former and 150–300m at the latter (Day, 1990). Whelks were dispersed and no aggregations could be found so samples were taken within foraging areas of less than $4m^2$ (following estimates of maximum distance travelled by whelks (Hughes, 1972)) in an attempt to ensure that individuals would be part of the same breeding unit. The size of the sampled areas in the sheltered sites meant that no more than 21 whelks were found at a single sample location (Day, 1990).

Samples were analysed for allozyme variation at eight soluble enzyme loci *Est-3, Lap-1, Lap-2, Mdh-1, Pep-1, Pep-2, Pgm-1* and *Pgm-2*. Nomenclature, electrophoresis buffers and staining methods follow those of Day & Bayne (1988) modified by Day (1990).

Raw data were presented in Day (1990) and are given in Appendix H. All loci were

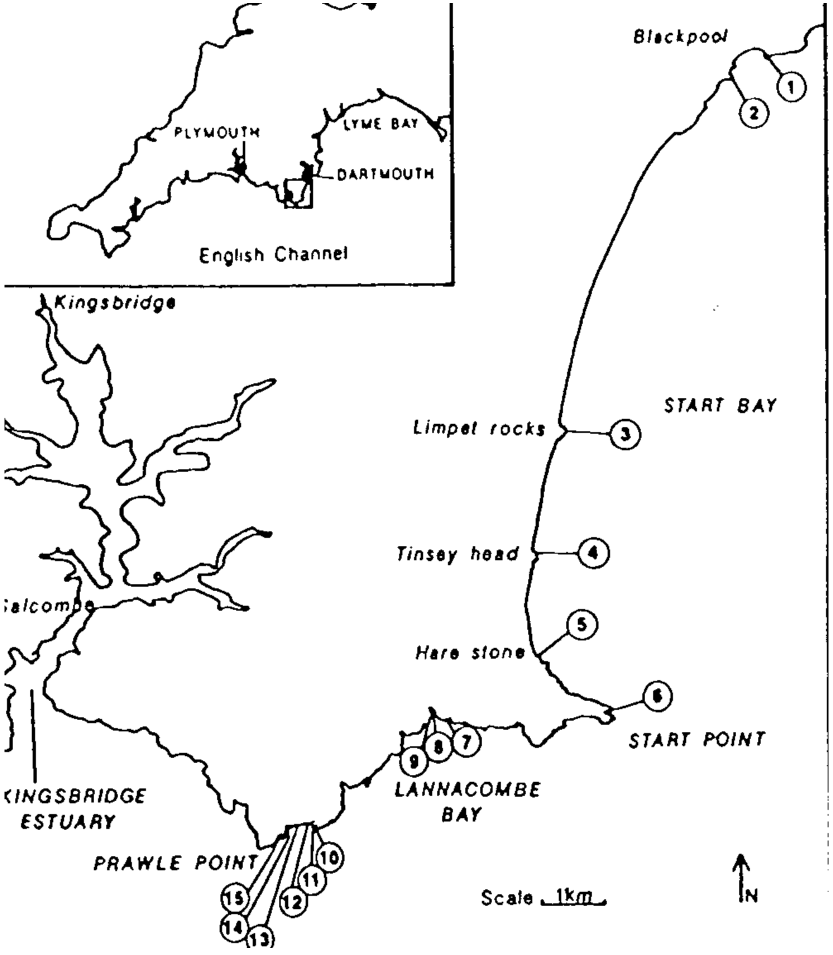

Figure 5.14: Sample locations of *Nucella lapillus*

polymorphic and the number of alleles per locus ranged from two for *Pep-1, Pgm-1, Est-3, Pep-2* to four for *Pgm-2, Lap-2*.

Since differences in morphology as well as genetic variability were noticed in the previous analysis, the strategy for grouping samples was divided into two parts. First, when all sites were considered, the following groupings were made:

1. all samples independently
2. 8-9, 10-11, 14-15 pooled together
3. 7-9, 10-12, 13-15 pooled together
4. 7-9, 10-15 pooled together
5. 1-2, 4-5, 7-9, 10-15 pooled together
6. 1-2, 3-5, 7-9, 10-15 pooled together
7. 1-5, 6-9, 10-15 pooled together
8. 1-5 and 6-15 pooled together
9. all sites together

The curves following changes in f will therefore be made of nine data points.

For the study of sheltered sites only, the pooling strategy was:

1. all sheltered samples independently
2. 8-9, 10-11, 14-15 pooled together
3. 8-9, 10-11, 12-13, 14-15 pooled together
4. 7-9, 10-12, 13-15 pooled together
5. 7-9, 10-15 pooled together
6. 6-9, 10-15 pooled together
7. all sheltered sites together

An extra level of pooling between level 2 and 4 (8-9 , 10-11, 12-13, 14-15) was added to check for the effect of the pooling strategy on the behaviour of the changes in f. There will therefore be 7 data points.

In a previous analysis, Day (1990) found that, in the whole sample, high F_{st} values resulted from a high degree of heterogeneity from exposed (quite monomorphic) to sheltered sites (rather polymorphic). Some evidence of a smaller scale of population structuring came from the analysis of some of the eight loci studied but without the opportunity of calculating confidence intervals or, consequently, the precise scale at which such structuring might take place.

5.4.2 Results

Overall F-statistics were 0.328 for F, 0.3327 for θ and -0.007 for f. There is a strong heterozygote deficit due to differences in allele frequencies among samples, whereas no heterozygote deficit or excess is detected at the scale of samples (P=0.573, according to the permutation test of alleles within samples, Appendix H). On the other hand, the probability that the observed θ is obtained by chance is less than 0.0002.

Analysis per site and per locus

The number of monomorphic loci per site was given in Goudet *et al.* (In Press). It is a good indicator of the amount of variability present in sheltered and exposed sites. For the exposed sites, an average of 60% of the loci within sites are monomorphic, whereas this number falls to 25% for the sheltered sites. These differences in levels of polymorphism among sampled areas led Goudet *et al.* (In Press) to reanalyse the data. Their results can be summarised as follows:

Exposed sites The f value (0.02) and confidence interval ([-0.11,0.09]) are similar to the analysis encompassing all three areas. *Pep-1* gives the highest deficit (0.11) and *Lap-1* and *Mdh-1* the greatest excess (-0.10 and -0.13 respectively). *Pep-1* is the only locus that shows more variability in the exposed than in the sheltered area.

Sheltered, Prawle Point The estimated f from Prawle Point is -0.08, with 95% CI [- 0.20,0.05], again similar to the three areas together, although slightly more negative. *Lap-1* shows a large deficit of heterozygotes (0.20) and *Lap-2* an excess (-0.34).

Sheltered, Lannacombe Bay Lannacombe Bay gives unexpected results in that the estimated f is 0.09, but the 95% CI is [0.02,0.18], implying that there is a heterozygote deficit in this area. This is also the area where allelic frequencies at 4 loci are most variable (allele 9 for *Lap-2*, *Est-3*, *Mdh-1* and allele 11 for *Pep-2*) as shown in Day (1990, Figure 2).

The variability between loci in terms of f is summarised in Figure 5.15.

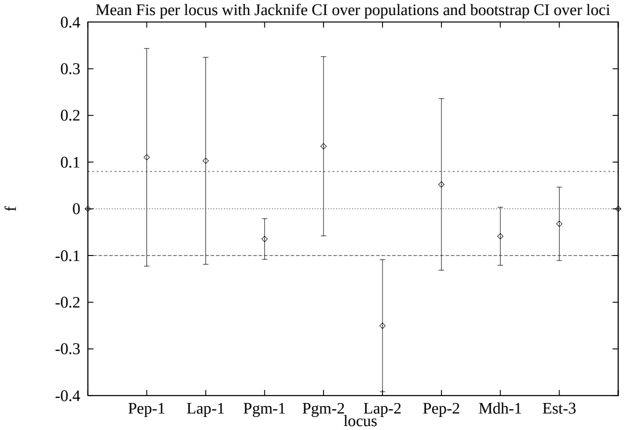

Figure 5.15: Jackknife CI per locus over samples and bootstrap CI over loci.

Recurrent pooling of sites

All sites Results for the effect of recurrent pooling of sites on f are displayed in Figure 5.16.

Pooling levels 2 and 3 lead to a slight increase in the value of f, but the first major discontinuity occurs at pooling level 4, when all sheltered Prawle sites are grouped together. This could be considered as a first level of structuring, although confidence intervals of the mean (Jackknifing over loci) between level 3 and 4 overlap. The next 3 pooling levels consist in grouping of exposed populations, and the f values obtained are constant. We have already mentioned that the exposed area was fairly monomorphic so it is not surprising to see that these pooling stages do not provide any new information. If there is structuring in this area, the data set is unable to detect it. Level 7 also adds the Start Point samples (only 6 individuals) to those of Lannacombe without changing the f value.

The next discontinuity on the graph occurs between level 7 and 8, when pooling together Lannacombe Bay samples with Prawle Point samples. Here, the difference in f value is large enough for the two CI not to overlap. This is the second level of structuring. Pooling all samples together reveals the third level of structuring, highlighting the difference in genetic make-up of exposed and sheltered sites.

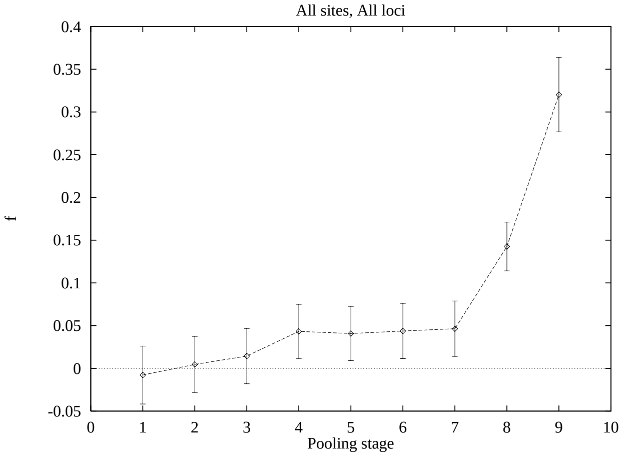

Figure 5.16: Changes in f with pooling stage in *Nucella lapillus*, all sites, all loci.

Sheltered sites only Figure 5.17 describes the change of f in sheltered sites only.

The graph shows a similar behaviour to that for all sites (Figure 5.16). The extra pooling level 3 (pooling of sites 12 and 13 together) leads to a slight increase in f, followed by a decrease when pooling 10-12 and 13-15. Although the difference is not significant, it suggests that sites 12 and 13, only 50 m apart, belong to different breeding units and confirms that the pooling strategy adopted is appropriate.

The observation that *Lap-2* is significantly different from all other loci (Figure 5.15) led to a reanalysis of sheltered sites excluding this locus. Results are given in Figure 5.18.

The graph shows essentially the same behaviour but confidence intervals of the mean are much narrower. This allows discrimination between pooling levels 4 and 5, the third level of structuring, which corresponds to the pooling together of all Prawle Point samples. The pooling of all sheltered sites (level 7) remains highly significant.

A matrix of pairwise estimates of Nm was calculated from over-loci pairwise θ. The results are shown on Figure 5.19.

The first striking feature of this figure is its similarity with the figure obtained for a one-dimensional stepping-stone model (Figure 4.35) and a one-dimensional isolation by distance model (Figure 4.38). Gene-flow is highest along the main diagonal, and decreases as samples get further apart. The lowest genetic exchange occurs between

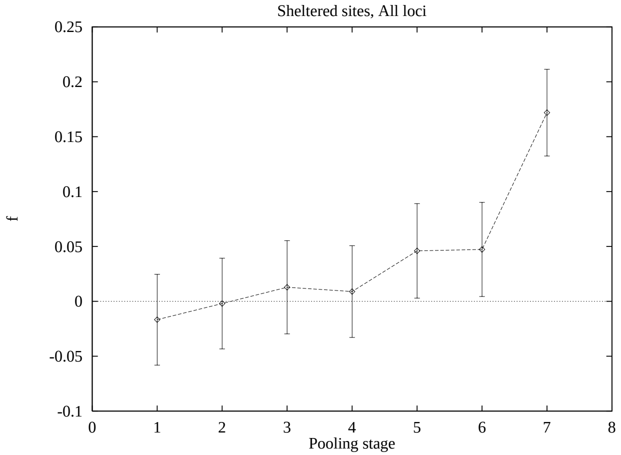

Figure 5.17: Changes in f with pooling stages in *Nucella lapillus*, sheltered sites, all loci.

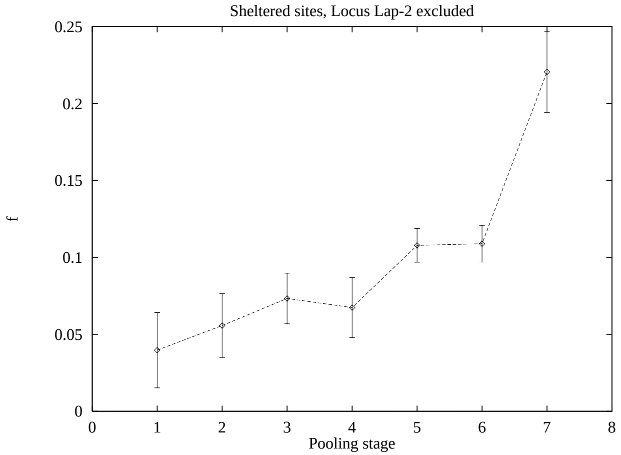

Figure 5.18: Changes in f with pooling stages in *Nucella lapillus*, sheltered sites, *Lap-2* excluded.

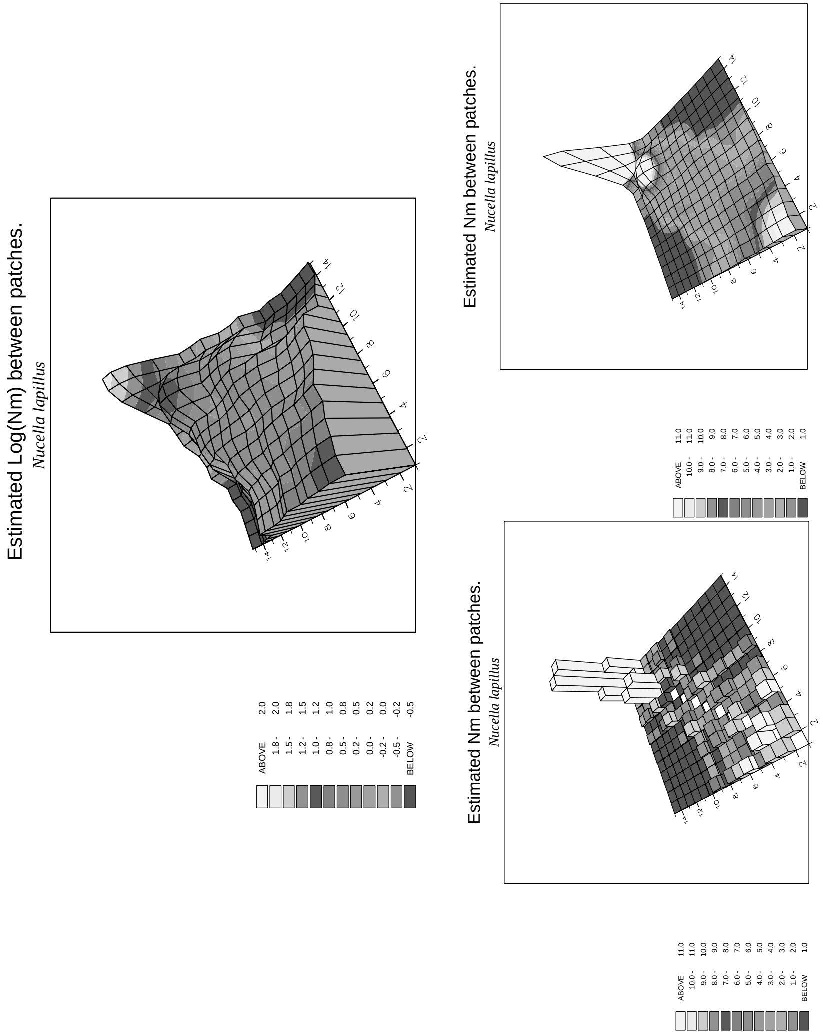

Figure 5.19: Estimated Nm between samples of *Nucella lapillus*.

samples at Prawle Point and those at exposed sites. The number of migrants is highest for Prawle Point and it is clear that this area is made up of two groups. This is in agreement with what had been detected using the changes in f with pooling stages. The number of migrants between exposed sites seems lower but more spread out. The 'bottom of the saddle' is at Lannacombe Bay where clines of gene frequencies are steepest (Day, 1990). Although it may seem naively inductivist (Chalmers, 1976), one may be tempted to conclude that *Nucella lapillus* lives in a one dimensional habitat. A falsificationist would say that gene-flow patterns in dogwhelks do not follow those of an island model or of a two-dimensional stepping-stone model.

The main points from this analysis are:

- There is a wide variation in polymorphism between sheltered and exposed sites. Exposed sites are more monomorphic and polymorphism is expressed at different loci from sheltered sites (Goudet *et al.*, In Press). Start Point seems to be a barrier to effective gene-flow.
- Lannacombe Bay and Prawle Point, separated by 3.3 km, correspond to 2 different populations with little genetic exchange.
- Prawle Point seems to be divided into two isolated breeding groups.
- Even though there seem to be two breeding groups at Prawle Point and one at Lannacombe Bay, these groups do not seem to be random breeding units.
- Gene-flow in *Nucella lapillus* seems to be restricted to one dimension.

5.5 Discussion and conclusions

Cabbage and beet are likely candidates for genetic improvement by means of genetic manipulation (Raybould & Gray, 1993). The possible effects of genes from genetically modified organisms (GMO's) escaping into the environment, either because the individual bearing the modification escapes or because there is hybridisation with a wild relative, remain largely unknown and will most certainly be dependent on particular genes and organisms. However, it seems more than likely that it will occur since crops could be thought of as an infinitely large pool of genetic material, constantly reimplanted into the environment. This is an effect very similar to that of recurrent mutation or migration (Gliddon, 1993) and well documented in the scientific literature (eg. Crow & Kimura, 1970). To delay the escape, minimum confinement distances of genetically modified crops are likely to be imposed and are documented in Raybould & Gray (1993), after Levin & Kerster (1974). Isolation requirements for *Brassica* is 900 m, whereas it is 3200 m in *Beta*. Although θ was higher in *Brassica* (0.328) than in *Beta* (0.167), it would be very difficult to compare the two species, since f values are so different. It was shown in Chapter 2 that the breeding system will affect measurements of θ, since selfing will lower the local effective size, whereas disassortive mating will enhance it (thereby speeding up or slowing down the process of random genetic drift). *Brassica* is insect pollinated and insect flight distances seem to be strongly influenced by flower density (Gray *et al.*, 1992). It was shown (Figure 5.5) that the amount of gene-flow is higher in the large continuous patches than in the rest of the samples, suggesting that pollen is the principal component of dispersal in this species. If this is the case, confinement distances have little meaning, since the length of the flight will be a function of the density of flowers encountered and a bee may well fly hundreds of metres to find a suitable plant. A better idea than confinement distances would be to surround the field of modified crop with a pollen donor that cannot hybridise with the crop.

Beta, on the other hand, is predominantly wind pollinated. Two different habitats were analysed and Figure 5.13 showed that levels of gene-flow are much higher in driftline than in cliff-top populations. If dispersal was to be mainly pollen mediated, one would expect to see either no differences or more gene-flow on cliff-tops where the wind is stronger. It seems, therefore, that seeds are the main element of dispersal in this species, at least for long distance migration. A study on the distribution and dispersal of *Beta vulgaris* spp. *maritima* germplasm in England, Wales and Ireland (Doney *et al.*, 1990) found that seed dispersal was mediated by tides, winds, animals and man. From the analysis carried out here, it seems that tides and/or storms are likely to be the main factor in the long distance dispersal of beet. One cannot fail to feel that lowering the probability of escape of modified *Beta* genes in the environment is difficult (Eijlander, 1989; Boudry *et al.* (In Press)) and that other solutions such as genes engineered to trigger death under environmental conditions alien to those of the crop (Ellstrand & Hoffman, 1990) should be looked into.

There are no proposals yet to genetically modify dogwhelks, (although a garlic butter flavouring gene would please the Mediterranean cooks) but this species is of interest in another evolutionary domain. Questions have been raised in order to explain the

genetic polymorphism observed in *Nucella lapillus*. The polymorphism observed at many electrophoretic loci seems to be linked to environmental conditions. In particular *Lap-2* was shown to be correlated with many ecological factors such as exposure to wave action, shell shape and chromosome polymorphism (Day, 1990). The same questions have been raised for other gastropods, in particular *Cepaea nemoralis*, the land snail, which displays a polymorphism of the colour and banding patterns of the shell. Many different selectionist arguments have been advanced to explain this trait ranging from predatory action (Cain & Sheppard, 1951) to effect the of temperature and albedo (Jones *et al.*, 1981). Nonetheless dispersal is extremely limited in this species (Lamotte, 1951), which would allow for differentiation to take place through the effect of random genetic drift. Figure 5.19 shows lower migration in the Lannacombe Bay area than elsewhere in the studied area. Since this is also where Robertsonian polymorphism is found (which may well be a partial fertility barrier and therefore, prevent or diminish gene-flow), one could be tempted to conclude that differences in environmental conditions (high exposure against low exposure) have favoured genetic differentiation of whelks each side of this bay. However, when comparing the picture obtained for whelks with those obtained via modelling of populations living in a one-dimensional habitat (Figure 4.38 & 4.36), the patterns are essentially similar. One is tempted to invoke the principle of parsimony (Occam's razor) to conclude that mere random genetic drift is sufficient to explain the observed pattern. Evidence for selection in this case would be better sought in laboratory experiments and it should always be remembered that it is rather too easy to commit suicide with Occam's razor (Gliddon & Gouyon, 1989).

Chapter 6

General discussion and conclusions

6.1 New developments in F-statistics.

A recent paper by Cockerham & Weir (1993) examines the estimation of gene-flow from F-statistics. The main thrust of this paper is

> 'to clarify the behaviour of F_{ST} and G_{ST} based estimators of gene-flow'.

Some remarks are necessary here to clarify what Cockerham & Weir (1993) mean by F_{ST} and G_{ST} statistics. F_{ST} based statistics, in their terms, are the correlation of genes within groups within populations. It is what has been called θ in this research, and what they call β. The difference between β and θ is that i) the model for β does not take the genotypes into consideration; ii) there is no mutation in the model used for θ whereas there is in the model for β. They state:

> 'the model under consideration is the standard island model, with a finite set of islands, each of size N. Individuals are monoecious, and *mating is at random including a random amount of selfing.*
>
> [...]
>
> Even though we generally assume the mutation rate to be much less than the migration rate, we cannot address questions about migration for a finite number of populations at equilibrium unless there is some mutation maintaining variation.'

Indeed, the quantities that they are estimating, X and Y, are the within-population and overall allelic frequencies respectively (as well as the expectation of f_0 and $\overline{f}$, the probabilities of identity by descent within and among populations respectively).

Another quantity would need to be estimated if they were to consider departure from random-mating within groups. As far as reaching equilibrium is concerned, it was shown in this research that, with the 2N-allele model with no mutation (cf. Chapter 2), simulations can be run for long enough to attain equilibrium without losing polymorphism (see below).

Cockerham & Weir (1993) also consider two G_{ST} based statistics, G_{CA} and G_{ST}. The first refers to a paper by Crow & Aoki (1984), and corresponds to what has been called G_{st} in this research. This quantity (equation 3 in Cockerham & Weir (1993)) is the same as the estimator of G_{st} derived in Nei & Chesser (1983). The second (equation 4 in Cockerham & Weir (1993)) is what has been called F_{st} in this research, namely, the weighted average of the different F_{st_u}, where u designates alleles and the weight is $p_u(1-p_u)$. The equivalence between the Cockerham & Weir (1993) estimators and mine is easily checked by comparing equations (3) and (4) in Cockerham & Weir (1993) with equations 3.59 and 3.57 here, respectively.

Mutation, μ, is considered in Cockerham & Weir (1993), as it was in the model of Crow & Aoki (1984) but has not been considered in this research. This added complication seems unnecessary since i) polymorphism is maintained for a long enough period in the 2N-allele model with no mutation; ii) it occurs in the equilibrium formula for β only as a product with the migration term (ρd in Cockerham & Weir (1993), equation 1). Figure 6.1 shows the changes of β as a function of migration and mutation rates. The striking feature of this figure is the independence between migration and mutation rates on β (additivity). If we replace migration in expression 2.10 by the sum of migration and mutation ($m \rightarrow (Dm/(D-1)+\mu)$), there is no difference between the two expressions apart from very high mutation and migration rates (Figure 6.2). Although mutation rates of the order of 10% have been found in some hyper variable and repetitive DNA, it is more often considered to be in the range of 10^{-7} to 10^{-5} per locus per generation (Maynard Smith, 1989). For these values, there is no differences between the two formulae.

Cockerham & Weir (1993) point out the differences between β and G_{CA} as being one of definition. First of all, they give the relation between identity by descent and identity in state, then derive unbiased estimators of the two identities in state $\hat{F}_0$ and $\hat{F}_1$, and point out that, G_{CA} is defined as

$$\frac{\hat{F}_0 - \hat{\overline{F}}}{1 - \hat{\overline{F}}},$$

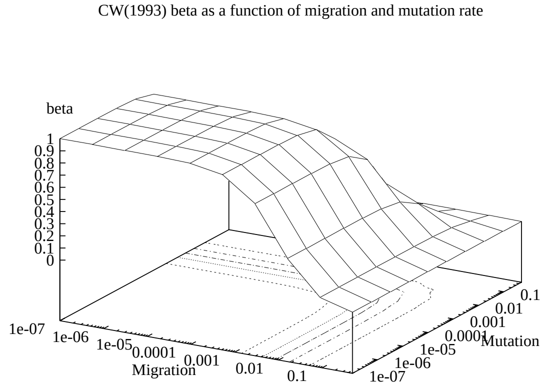

Figure 6.1: $\beta(m, \mu)$ (Cockerham & Weir, 1993, equation 1), with $r = 10$ and $M = 25$.

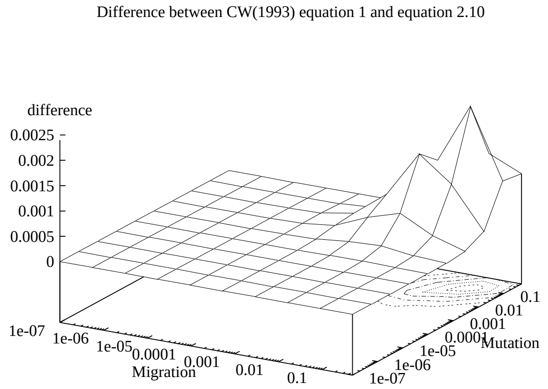

Figure 6.2: Differences between $\beta(m, \mu)$ and equation 2.10 when migration in the latter is replaced by the sum of migration and mutation rates

where $\overline{F}$ is the probability that two genes drawn at random from the entire population are identical. This is the definition of F that was given in Chapter 2. On the other hand, β is defined as

$$\frac{\hat{F}_0 - \hat{F}_1}{1 - \hat{F}_1}.$$

Cockerham & Weir (1993) say:

> 'β is preferable to G_{CA} for quantifying the relation between genes in this model. The argument is based on β not depending on the unknown quantity n [D here], on the use of each level of differentiation rather than the use of averages over levels, and on the use of intra-class correlations.'

While these remarks explain the discrepancies between the two estimators, in particular the dependence of G_{CA} on the number of samples, it does not lead to a clear statement about the underlying hypotheses needed to take estimators. It was pointed out in this work that the hypothesis behind the Weir & Cockerham estimators is one of rate of loss of heterozygosity, whereas that behind Crow & Aoki is one of rate of allele frequency drift. The two explanations are complementary but the latter provides a framework in which to include further complexity in the model.

The analysis carried out on estimation of gene-flow focussed on the inverse relationship $R(z) = 1/z - 1$, where z is one of the estimators of Wright's F_{st}, and $R(z)$ an estimator of $4Nm$. Note that there is a misprint and that the formula for $R(\overset{\circ}{G}_{ST})$ (p. 858) should read:

$$R(\overset{\circ}{G}_{ST}) = \frac{(2N-1)n}{2N(n-1)} \ldots$$

Cockerham & Weir (1993) state:

> 'Considerable simplification occurs for $R(G_{ST})$ when all individuals in all groups of a population at equilibrium are sampled.
>
> [...]
>
> [$R(\overset{\circ}{G}_{ST})$] provides a fairly close approximation to $4Nm$ for $\mu \ll m$, $m \leq 0.1$ and n large.
>
> [...]
>
> The percentage of discrepancy depends on m, and in the case of $R(\overset{\circ}{G}_{ST})$, on the sampling dimensions. When $m = 0.1$, censusing the population is far better than drawing a sample for $R(\overset{\circ}{G}_{ST})$.'

A complete census of the population does not alter the problem of estimation, as was pointed out in Chapter 3, since the genetical sampling is still present with a complete census. The simplifications for $R(G_{ST})$ are also valid for large sample size and large number of demes sampled. Indeed, if the total population comprises few populations and few individuals per population, the total census will still provide a highly biased estimator of F_{st}.

Table 1 in Cockerham & Weir (1993) points out the independence of the parameter β, with regard to M and r (the number of individuals per sample and the number of samples, respectively) and shows that the parameter G_{ST} (F_{st} here) is dependent on M and r. This is not surprising since G_{ST} is a statistic and not an estimator. As was pointed out in Chapter 3, a more appropriate comparison would be that of β and G_{CA}, a comparison that was carried out in Chapter 3 and which showed that, while independent of M, G_{CA} depends on r as expected from the relationship between β and G_{CA}.

Table 2 in Cockerham & Weir (1993) compares theoretical values of $R(z)$ with those obtained in Slatkin & Barton (1989). Note that superscripts ^{1} & ^{2} in the second and third column should read * and † respectively. They point out that the bias is always positive and sometimes very large, a finding that corroborates what has been found in this research (Figures 3.17 and 4.14).

Table 3 in Cockerham & Weir (1993) presents the results of their own simulation, where estimators were taken after the $101,000$th generation, since the calculations are based on X and Y which take a very long time to reach equilibrium. However, β reaches equilibrium much faster (Crow & Aoki, 1984; Chesser, 1991). This last feature is shown in Figure 6.3.

In this research, time to equilibrium was checked using expression 2.38 for the island model and by ensuring that estimators of F_{st} had reached a plateau for the stepping-stone models so that there was no need to run unnecessarily lengthy simulations.

Cockerham & Weir (1993) focus only on an island model at equilibrium with 10% migration. The results of Chapter 3 were for a much wider set of situations, since equilibrium as well as non equilibrium situations, random and partial selfing, and island as well as stepping-stone models were investigated. The results here generalised those of Cockerham & Weir (1993), showing that β is unbiased in all the situations examined. It was pointed out in this research that seeking estimators of Nm is valid

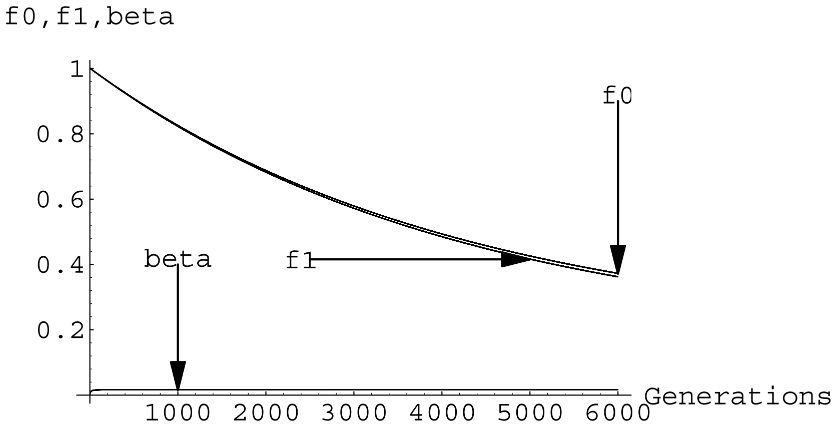

Figure 6.3: Changes over time of $\theta_{2,t}$,$\theta_{3,t}$ and β_t (f_0, f_1 and beta on the graph respectively), following Cockerham & Weir (1987). The parameters are $\mu = 0.0001$, $m = 0.1$, $N = 128$, $n = 100$. $\theta_{2,0} = \theta_{3,0} = 1$, as in Slatkin & Barton (1989). Note that equilibrium is reached faster for β than for the identity by descent coefficients.

only when θ is large, and that comparisons of different populations should be carried out using θ rather than Nm (cf. Chapter 4).

An interesting feature of Table 3 in Cockerham & Weir (1993) is that they consider two types of starting conditions: one which corresponds to the simulation of Slatkin & Barton (1989) where the entire population is monomorphic (labelled 'fixed' in Table 3), and one where they sampled 1000 unique alleles at random to create the genotypic array of the first generation.

> 'Instead of an infinite-allele model, we used one with 1000 alleles and equal mutation rate among all the alleles to make the simulations more manageable.' Cockerham & Weir (1993)

This last situation is similar to what has been implemented here but the infinite-allele model was made manageable by using the method presented in Chapter 2.

I have to disagree with the statement of Cockerham & Weir (1993) about source of the errors in Slatkin & Barton (1989).

> 'We do not know, of course, what contributed to the errors in the simulations of Slatkin & Barton (1989), but one possibility is that the populations were not at equilibrium.' Cockerham & Weir (1993)

Figure 6.3 pictures the exact changes in identity by descent coefficients and β, with the parameters and starting values used by Slatkin & Barton (1989). It is obvious from this figure that equilibrium for β is reached much faster than that for the identity by descent coefficients. Figure 6.3 also corroborates the finding that time to equilibrium depends on the larger of m and $1/N$. In the present case, m is much larger than $1/N$, and is the sole determinant of time to equilibrium. Although equation 2.38 could not be used here to assess time to equilibrium (since F_0 is undefined), the equation can be rewritten in terms of F_1:

$$F_t = B^{t-1}(F_1 - \frac{A}{1-B}) + \frac{A}{1-B}$$

from which

$$t - 1 = \frac{\ln\left(\frac{(x-1)A}{F_1(1-B)-A}\right)}{\ln(B)}.$$

In the present case, $\beta_1 = 0$ ($\theta_{2,1} = \theta_{3,1} = 0.9998$) and this expression reduces to

$$t = \frac{\ln(1-x)}{\ln B} + 1$$

where $B = (1 - (\mu + \frac{mn}{n-1}))^2(1 - 1/2N)$, to give 43.4 generations as the time necessary for β to reach 99.99% of its equilibrium value. Even if this result is an approximation, since the starting population is monomorphic, Slatkin & Barton (1989) must have allowed for mutation to create enough polymorphism which would have meant running the simulations for a large number of generations. The inaccuracy of Slatkin & Barton's results must be due to a different factor.

In their conclusions, Cockerham & Weir (1993) write:

> 'Finally, we note that there are conditions under which functions of F-statistics can provide gene-flow estimates with low bias. We must agree with a reviewer of this paper, however, and acknowledge a deficiency in the approaches discussed in this paper for providing such estimates. The problem is that these approaches are based on measures of population differentiation presumed to have been caused by gene-flow. No direct observations on gene-flow are used, and inferences are necessarily limited by the assumptions of the model, including neutral alleles and attainment of equilibrium. There is no basis for distinguishing between the events of the migration model assumed and any other evolutionary scenario that could lead to the same pattern of gene frequencies within and between groups. Unless the various assumptions of a model such as the island model are verified by direct observations, there must continue to be doubts about analyses based on assumption-laden theories—whether or not theses analyses rest on simulations.'

Unfortunately, as was pointed out in this research (cf. Chapter 3), direct measurements of gene-flow are likely to be even more inacurate than indirect measurement and could therefore mislead the researcher. Examples of such discrepancies have been found in many species and are likely to have arisen because of the large variance of dispersal over time. Methods presented in Chapter 4 and 5 should help to discriminate between the hypotheses of selective pressure and gene-flow. As Slatkin (1985a) stated:

> 'Estimates based on data from one or two loci should be suspect, but if estimates are based on data from numerous loci and there is consistency in the estimates using different methods, it is reasonable to have some confidence in the conclusions.'

One of the findings in Chapter 5 is that the behaviour of the changes in f, when pooling samples together, is very robust in situation which depart from random mating (eg. graphs for *Beta vulgaris* ssp. *maritima*). Even when loci present obvious signs of selection, such as *Lap-2* in *Nucella lapillus*, we have been able to identify them. Unless all loci are submitted to the same type of selection, it seems that the methods presented here are a step forward in the identification of selection acting at some loci.

Similarly, distinguishing between breeding patterns was made possible by plotting the matrix of pairwise Nm values. Some doubts may be cast as to the accuracy of such measurements, doubts with which I would agree. In particular, better estimates would be obtained by using the exact relationship between θ and Nm given in Chapter 3 and 4, rather than using the approximation. Another solution would be to portray the pairwise θ's themselves, but the outcome is likely to be difficult to interpret, since θ is constrained between $[-1/(2N-1):1]$, whereas exact Nm can vary between $[0:N]$ and approximate Nm between $[0:\infty]$. However, the outcome of this graphical representation should give a good indication of the underlying patterns of gene-flow. Even when the populations are not at equilibrium between the opposing forces of random genetic drift and migration, one would expect to see larger values of migration between adjacent populations than between populations further apart if isolation by distance (in a discrete or continuous form) is occurring. On the other hand a chaotic pattern would be a strong indication that there is no geographical structuring, whether at equilibrium or not.

The last remark concerns the quality of data. As pointed out by Slatkin (1985a), it is necessary to obtain many independent loci displaying a similar behaviour before estimates of gene-flow can be made. Of the three data sets presented here, only the dogwhelk data could be considered as potentially sufficient to quantify gene-flow but because there was no clear indication of the limit of the breeding unit, this quantification was avoided. This point is likely to be most constraining on the accurate inferences of levels of gene-flow since, without a notional breeding unit, one can never be sure that sampling was carried out at the appropriate scale.

In this research, results of simulations were always compared to those obtained from analytical theory, when available, and were in good agreement. When the analytical theory was unavailable or too complex to be solved even by numerical methods, at least the results followed expected patterns. Simulations were not considered on a par with analytical theory but were used to analyse otherwise intractable models.

Great care has been taken in verifying the results and ensuring that all the elements of the simulations were correct. In particular, great care was taken to choose truly random number generators, an aspect of stochastic simulations that, to my knowledge, is too often ignored.

6.2 Conservation genetics

The main scope of conservation biology is to identify the rules for maintaining the fitness of individuals and populations, and to understand the biological principles upon which these rules are based (Soulé, 1986). In a review called '*Conservation genetics and conservation biology: a troubled marriage*' Soulé & Mills (1992) write:

> 'Until the middle 1970s, most of the people in charge of conservation ignored genetics, and most of the people in charge of genetics ignored conservation. But beginning around 1970, plant geneticists started to become alarmed about the disappearance of primitive or traditional crop varieties and their replacement by modern, genetically uniform, cultivars. Geneticists suspected then as they do today that the seeds of the green revolution contained the agents of their own ultimate collapse, namely, genetic uniformity.'

Since those days, an international system of gene banks was endorsed by the United Nations Conference on the Human Environment in Stockholm (1972) and the International Board of Plant Genetic Resources (IBPGR) was established, in order to further the collection, conservation, documentation and use of germplasm crop species (Williams, 1988).

Genetics, however, remains a minor component of conservation biology for many reasons:

- It is usually accepted that the maintenance of genetic variability will affect the long-term survival of the species but does little for the short term (Goodman, 1987; Belovsky, 1987; Schwartz *et al.*, 1986; Dawson *et al.* 1987; Lande (1988)).

- Genetic variability on its own has little meaning and needs qualifiers such as electrophoretic, DNA, neutral, selected (Brown & Schoen, 1992)

- Depending on the type of variability being considered, the effects of bottlenecks are different: electrophoretic variability, and inbreeding effective size are reduced (increased homozygosity) (McComas & Bryant, 1990), whereas additive genetic variance seems to increase (Goodnight, 1987,1988; Carson, 1990; Lewin, 1990).

- Genetic variability may well be enhanced at the level of the total population through processes of local extinction and recolonisation (Ewens, 1990; Gliddon & Goudet, In Press).

- Moreover, genetics is an arcane field, in part because of its difficult jargon, and in part because it is quantitative (Soulé & Mills, 1992).

The first point seems to be dated now, and the latest reviews seem to emphasize the growing need for genetics to be taken into account in assessing chances of survival of a species (Soulé & Mills, 1992).

The second point is of more interest, since the body of data obtained through studies of DNA polymorphism is growing and tends to replace electrophoretic work in the scientific reviews. The prime advantage of DNA techniques compared to protein electrophoresis resides in the amount of polymorphism detected. Species that appeared essentially monomorphic when screened for electrophoretic variants of proteins may reveal polymorphism with one or the other molecular tools now available to population geneticists. Questions arise as to how much polymorphism is necessary in order to estimate population structuring. It was shown in Chapter 3 that F-statistics are undefined when the locus is monomorphic. Similarly, when $2N$ alleles are obtained from a sample of N individuals, it is impossible to estimate Wright's Fixation indices. Total monomorphism is as bad as total polymorphism and one might wonder what the optimum amount of variability at a locus is in terms of the estimation of population structure. The information that one can extract from a locus should follow a parabola, crossing the zero-axis at 1 and $2N$ alleles, and with a maximum somewhere in between. For such systems, looking at a phylogeny of alleles has been proposed (e.g. Slatkin & Maddison, 1989, 1990, Excoffier *et al.*, 1992), but requires expensive sequencing techniques and may be difficult to apply since phylogeny can only safely be inferred in the absence of recombination (Slatkin & Maddison, 1989, 1990).

It was pointed out in Chapter 3 that the collection of genotypic data is crucial to obtain unbiased estimators of F-statistics. Randomly Amplified Polymorphic DNA (RAPD) are being used in population genetics and are often described as a cheap

and easy technique to detect polymorphism. However, the technique suffers many drawbacks since markers are dominant, lack repeatability, have been shown to follow non-Mendelian modes of inheritance (Riedy *et al.*, 1992), and give results highly dependent on the experimental conditions (McPherson *et al.*, 1993).

It is clear that, from a conservation point of view, an assessment of all types of variability is required, until some correlations can be found between the different sources. Neutral markers, however, can, under the conditions discussed in this research, provide information about gene-flow patterns in the past. This information could prove useful for the conservationist.

Studies of the effect of bottlenecks on populations have shown that while electrophoretic variability is reduced, additive variance seems to increase (Bryant *et al.*, 1986; McCommas & Bryant, 1990). This increase in additive genetic variance is due to the conversion of epistatic and dominant terms to additive terms (Templeton, 1991). This is a whole new facet of the field of population genetics and conservation and it is in line with the predictions of Wright's shifting balance theory (Wright, 1977). Most experimental studies of bottlenecks, however, have been carried out on laboratory organisms, such as *Drosophila melanogaster*, which tend to have a r-type life-history. Unfortunately, endangered species [that we want to preserve] tend to be of the K-type, and are more likely to suffer the effect of inbreeding-depression and/or outbreeding-depression (Templeton & Read, 1983; Templeton, *et al.* 1986). Indeed, 'r-species' such as the mosquito *Culex pipiens* goes through bottlenecks every year during the winter and therefore experience frequent local extinctions and recolonisations. Evolution must have favoured a genotypic make-up for these species insensitive or less-sensitive to these fluctuations than species with little fluctuation in population size. Furthermore, one can foresee an uproar if scientists were allowed to experiment with the endangered 'cute and cuddly' species, experiments that are necessary if better conservation strategies are to be defined and applied by conservationists.

The question of a Single Large Or Several Small (SLOSS) populations remains a classic conservation dilemma. From a pure demographic perspective, a single large population seems better since the probability of extinction of a patch is a convex increasing function of decline in patch size (this consideration needs moderating however, since a single catastrophe could wipe out the single large population). On the other hand several small populations would be the best way to maximise the maintenance of genetic variability at the level of the total population, if inbreeding depression were not a problem. This dilemma would have to be understood in the framework of metapopulations: a set of local populations which are established by colonists, survive for a while, send out migrants and eventually disappear (Levins, 1970). **MODEL42**, presented in this research, does not include this demographic feature of metapopulations but could still be considered as a reference for such studies. One of the criticisms that can be made of existing metapopulation models (see Olivieri *et al.* (1990) for notable specific exceptions) is that demography and genetics are usually uncoupled: some metapopulations models (eg. Wade & McCauley, 1988; Marayuma & Kimura, 1980) have no real dynamics locally since the local populations, after foundation by a specified numbers of individuals, grow instantly to their carrying capacity. Models in the same category have fixed r and K values, independent of the genotypes and, therefore, focus only on short term ecological effects rather than long term evolutionary consequences. A second category of models considers a constant fitness (independent of density and/or frequency). Possible ecological effects are therefore ignored to allow evolutionary consequences to be studied. As was pointed out by Gliddon & Goudet (In Press), given the relative lack of incorporation of genotypic effects on parameters of clear importance for colonisation such as colonising ability and extinction probability, it should come as no surprise that the majority of the models predict that demographic (genotype independent) effects are of major concern in designing conservation strategies. In a neutralist framework, a measure that may be more appropriate than the census size is the effective population size. We have seen throughout this work that many different, often contradictory, definitions of effective size exist. While inbreeding effective size is a measure of the rate of loss of heterozygosity, the variance effective size is a measure of the rate of loss of genetic variability from a population, or a measure of the rate of allele frequency drift. Crow & Denniston (1988) stated:

> 'If one is interested in conserving genetic variance,..., it [variance effective size] is the most appropriate effective number.'

A survey of the scientific literature will show at least three other types of effective sizes emerging:

- The extinction effective size of Haldane (1939), also called eigenvalue effective size, since it can be calculated by measuring the largest non-unit eigenvalue of the transition matrix of the Wright-Fisher model (Ewens, 1979, 1982).

- The mutation effective size, introduced by Marayuma & Kimura (1980), stems from the infinite allele model and is defined as a function of the probability that two individuals chosen at random from the entire population are of the same allelic type, the function being:

  $$P = \frac{(1-u)^2}{2N-(2N-1)(1-u)^2}$$

  (we recognise here equation 2.10, with the mutation rate u replacing the migration rate m). Providing that P and u can be estimated, and that the infinite allele model holds true, then solving for N would lead to the mutation effective size of the population (Ewens, 1989).

- The diversity effective size (Gregorius, 1991) , 'which accounts for the rate of loss of allelic variation, and not merely the rate of loss of heterozygosity' and is similar in concept to the variance effective size.

It has been shown in Chapter 4 that restricted gene-flow enhances the variance effective size of the population compared to the census size and therefore, maintains more genetic variability. Similar findings were described in metapopulation models. Wade & McCauley (1988, 1991) considered two different types of founders: a 'Propagule Pool' in which there is one large source population from which the migrants originate; and a 'Migrant Pool' in which migrants are drawn at random from the extant local populations. The 'Propagule Pool' confirms the results of a verbal model of Wright (1940), namely, that F_{st} was increased, relative to a no extinction control and, therefore, the variance effective size was increased providing that the number of founders was less than the carrying capacity. In the 'Migrant Pool' model, F_{st} was increased and, hence, variance effective size, providing that $4Nm+1$ was larger than twice the number of founders, where N is the local carrying capacity and m the rate at which local populations exchange migrants.

This view was expressed by Ewens in 1989:

> 'It is usually accepted that a subdivided population subject to extinction of subpopulations will lose genetic variation more rapidly than an equally large random mating population, or equivalently that it has a smaller eigenvalue effective population size. The above shows that it is not necessarily true. [...] We will see later than when mutation exists, the subdivided population can maintain more genetic variation, on the

> average, than a random mating population of the same size, again against accepted views.'

However in 1990 Ewens states:

> 'Except in extreme cases involving very many subpopulations each of small size, the rate of loss of genetic variation is greater than that for a single random mating population [...] and the implication for MVPS (Minimum Viable Population Size) is that a larger [global] population size is needed in the substructured case if the rate of loss of genetic variation is to be kept at the same value as that of an undivided population.'

In both quotations, reference is made to maintenance of genetic variation but no mention is made of the level at which this variation is maintained. Conclusions from the present research are that if variation is to be maintained within subpopulations, in order to avoid inbreeding depression, then maximisation of the inbreeding effective size is the goal, and a single large would be better than several small populations. On the other hand, if variation is to be maintained globally, then maximising the variance effective size is in order, and several small is better than a single large population. Note that the lenght of time over which this variation has to be maintained has not yet entered the argument but it is obvious from my results that, if genetic variation is to be maintained in the long term then several small is also better than a single large.

This discussion of effective sizes is a typical example of what Soulé & Mills (1992) meant when they mentioned the difficult jargon of population genetics. One might wonder whether the terminology of effective sizes, confusing even for geneticists, should not abandoned altogether and replaced by clear definitions such as rate of allele frequency drift (variance effective size), rate of loss of heterozygosity (inbreeding effective size), variance in the number of successful gametes (extinction effective size). Although these definitions have less impact than the catch-words 'effective sizes', they would avoid misuse since many conservationists have ignored the differences between census and effective size and used the latter as an estimate of MVPS (Harris & Allendorf, 1989; Ewens, 1990).

6.3 Risk assessment: releasing GMO's

It was pointed out in Chapter 5 that population genetics theory can be used to try and predict the effect of releasing Genetically Modified Organisms (GMO's) in the environment. Direct measurement of escapes of modified crops, or of their gametes, into the wild proves difficult since what needs to be measured are the long distance events. As was pointed out by Gliddon (pers. comm.), virtually all of the sampling methods and monitoring protocols described in the literature fail to describe the minimum levels of detection which could be achieved using their particular protocol. This problem is exacerbated by the design of the experiments– in the vast majority of cases using higher plants, the marked organisms are in a small minority of total organisms in the design. This results in the experimental design making it difficult to detect the spread of the marker in relation to the probability of recovering the non-marked gene. For example, if a marker is represented by 1% of the total organisms, even if it is distributed uniformly across the entire experimental area, it will only be recovered in 1% of the samples. This fault of experimental design could well account for the very small distances that have been reported for the spread of GMO's. Darmency & Renard (1992) referred to one experiment with transgenic oil-seed rape in which a small plot (10m x 10m) of recipient plants was situated at 800 m from a large plot (100m x 100m). The recipient plot consisted of 50% male-sterile and 50% hermaphrodite plants. On average, 1.5% of the seeds recovered from the male-steriles in the small plot had been pollinated by plants from the large plot, which was 800m distant. This should be compared with results (Darmency & Renard, 1992) in which no pollen was detected at 100m from a small source of transgenic rape located in the centre of a 1 hectare field.

A second point that needs emphasising is the crucial need to fit a distribution to the data collected. It was mentioned in Chapter 4 that the projection of bivariate distributions of the exponential family in one-dimension gives rise to a power function. Kareiva *et al.* (1993) and Manasse (1992) came to similar conclusions. Note that most reported results are expressed in terms of marker genes at a given distance as a percentage of total genes sampled. This is inappropriate as it is scale dependent, the correct form being of marker genes at a given distance as a percentage of the total number of marker genes recovered. This last method removes the dependence on size of the source of marker genes and correctly emphasizes the rate of decrease of marker genes recovered with distance (Kareiva *et al.*, 1993).

While the above considerations will improve direct measurements of gene-flow, the space-time variability intrinsic to direct measurements remains. Furthermore, only part of the gene-flow occurring is measured, since what is followed are marker genes and there seems to be no need for a transgenic crop in these experiments because the foreign DNA is used merely as a marker. The debate then returns to the usefulness of direct versus indirect methods. While data sets provided for the study of *Beta* and *Brassica* in Chapter 5 were not sufficient for quantitative predictions to be made, I have been able to show that levels of gene-flow are higly dependent on environmental conditions. One is therefore tempted to regard direct estimations as an attempt to characterise levels of gene-flow in a given, monitored environment. For predictions to be of any use, the experiment would need to be repeated in many different environments and indirect methods should be used concurrently in a close wild relative (if it exists) on a large scale to assess how variable gene-flow could be and how much long distance migration occurs because this last category will be extremely difficult to measure with direct methods.

Risk to conservation should not be neglected. One possible effect of the escape of a large number of genetically uniform organisms (with wild relatives), be it a GMO or not, will certainly be to diminish the diversity of the wild relatives. Examples, sadly, already exist: supportive breeding of salmonid populations (releasing captive-bred animals into the wild to support weak and endangered populations), in an attempt to enhance wild stocks, results a in dramatic increase in the rate of loss of genetic heterozygosity of the wild population, as well as an increase allele frequency drift, thereby reducing both the inbreeding and the variance effective size of the wild population (Ryman & Laikre, 1991).

6.4 General conclusions

Gene-flow has been studied in this research from an analytical, theoretical and practical angle. While simple models of restricted gene-flow are tractable analytically and can produce very accurate predictions when compared with the results of computer simulations, models of discrete populations with geographical structure models of continuous populations need further research. Basic requirements for models of discrete populations and analytical models are highlighted. However it should be kept in mind that models of isolation by distance in a continuum are very difficult to relate to concepts familiar to the population geneticist since the basic concept linking continuous populations to discrete ones, the neighbourhood size, has been shown to be flawed.

Inferring gene-flow from indirect methods implies obtaining unbiased estimates of quantities such as F-statistics. The framework for estimation presented, which uses the concept of variance effective size to derive unbiased estimates in different situations, does help to clarify the underlying assumptions. In particular, the conditions under which the estimates of Nei & Chesser (1983) and Weir & Cockerham (1984) are best suited have been highlighted.

While analytical treatment of geographically structured populations is difficult, F-statistics can be used to unravel levels of genetic structuring when the ideal conditions of an island model are not met. Methods presented here yield ways of discriminating between samples taken within and among breeding units, a necessary distinction if levels of gene-flow are to be inferred. Calculation of pairwise F_{st}'s provides a picture of the geography of gene-flow in the population investigated, even in continuous populations.

Emergent properties are inherent in biological systems since they are hierarchical. Gliddon & Gouyon (1989) pointed out that the outcome of selection at any level of a hierarchy (molecule, individual, group..) must be the result of a successful selection at all underlying levels. In this research, individual and molecular levels were amalgamated because individuals were represented by a collection of independent (diploid) loci. It was pointed out that effects such as those of bottlenecks have a different outcome on electrophoretic and quantitative variation. Small interactions in systems with few components can be ignored but as the the number of components in the system increases, interactions, even if very small, take precedence over separate effects (Cohen & Stewart, 1991). The question 'why all this polymorphism?' is as bad a question as 'why sex?' or 'what is the unit of selection?'. Indeed one might be tempted to answer '42' and apply for funding to build a new computer!

References

Amos, B., Barrett, J. and Dover G.A. (1991) Breeding behaviour of pilot whales revealed by DNA fingerprinting. *Heredity*. 67:49–55.

Ballou, J. (1992) Small population overview. In *Population and Habitat Viability Analysis* Captive Sreeding Specialist Group, IUCN.

Bantock, C.R. and Cockayne, W.C. (1975) Chromosomal polymorphism in *Nucella lapillus*. *Heredity*. 34:231–245.

Barbujani, G. (1987) Autocorrelation of gene-frequencies under isolation by distance. *Genetics*. 117:777–782.

Bartlett, M.S. (1955) *An introduction to stochastic processes*. Cambridge University Press, Cambridge.

Barton, N.H. and Slatkin, M. (1986) A quasi equilibrium theory of the distribution of rare alleles in a subdivided population. *Heredity*. 56:409–415.

Belovsky, G.E. (1987) Extinction models and mammalian persistence. In *Viable population for conservation*. Ed. M.E. Soulé, Cambridge University Press, Cambridge.

Berry, R.J. and Crothers, J.H. (1974) Stabilizing selection in the dogwhelk (*Nucella lapillus*). *J. Zool. London*. 174:123–148.

Blair, W.F. (1960) *The rusty lizard: A population study*. Publication 1851. University of Texas Press. Austin.

Boudry, P., Mörchem, M. , Saumitou-Laprade, P., Vernet, P.H., Van Dijk, H. (In Press) The origin and evolution of weed-beet: consequences for the breeding and release of herbicide resistant transgenic sugar-beet. *Theor. Appl. Genet.*.

Brown, A.H.D. and Schoen, D.J. (1992) Plant population genetic structure and biological conservation. In *Conservation of biodiversity for sustainable development*. Ed. O.T. Sandlund, K. Hindar and A.H.D. Brown, Oxford University Press, Oxford.

Bryant, E.H., McCommas, S.A. and Combs, L.M. (1986) The effect of an experimental bottleneck upon quantitative genetic variation in the housefly. *Genetics*. 114:1191–1211

Caballero, A. and Hill, G.H. (1992) Effective Size of Nonrandom Mating Populations. *Genetics*. 130:909–916.

Cain, A.J. and Sheppard, P.M. (1950) Selection in the polymorphic land snail *Cepaea nemoralis*. *Heredity*. 4:275–294

Cain, A.J. and Sheppard, P.M. (1954) Natural selection in *Cepaea*. *Genetics*. 39:85–116.

Campbell, G.K.G. (1976) Sugar beet. In *Evolution of crop plants*. Ed. N.W. Simmonds. Longman, London.

Carson, H.L. (1990) Increased genetic variance after a population bottleneck. *Trends Ecol. Evol.*. 5:228–230.

Cavalli-Sforza, L.L. and Bodmer, W.F. (1971) *The genetics of Human Populations*. San Fransico: Freeman.

Cavalli-Sforza, L.L. and Edwards A.W.F. (1967) Phylogenetic analyses: Models and estimation procedures. *Amer. J. Hum. genet.*. 19:233–257.

Chakraborty, R., Fuerst, P.A. and Nei, M. (1980) Statistical studies on protein polymorphism in natural populations. III. Distribution of allele frequencies and the number of alleles per locus. *Genetics*. 94:1039–1063.

Chalmers, A.F. (1976) *What is this thing called science?*. Open University Press, Milton Keynes.

Chesser, R.K. (1991) Influence of gene flow and breeding tactics on gene diversity within populations. *Genetics*. 129:573–583.

Cliff, A.D. and Ord, J.K. (1981) *Spatial autocorrelation*. 2nd Ed. Pion. London

Cockerham, C.C. (1969) Variance of gene frequencies. *Evolution*. 23:72–84

Cockerham, C.C. (1973) Analysis of gene frequencies. *Genetics*. 74:679–700

Cockerham, C.C. (1984) Drift and mutation with a finite number of allelic states. *Proc. Natl. Acad. Sci. USA*. 81:530–534.

Cockerham, C.C. and Weir, B.S (1986) Estimation of inbreeding parameters in stratified populations. *Ann. Human Genet.*. 50:271:281.

Cockerham, C.C. and Weir, B.S (1987) Correlations, descent measures: drift with migration and mutation. *Proc. Natl. Acad. Sci. USA*. 84:8512–8514

Cockerham, C.C. and Weir, B.S. (1993) Estimation of gene flow from F-statistics. *Evolution.* 47:855–863

Cohen, J. and Stewart, I. (1991) Chaos, contingency and convergence. *Non-linear Science Today.* 1:9–13.

Collins (1992) *Concise English Dictionary.* 3rd Ed. Harper Collins.

Coyne, J.A., Boussy, I.A., Prout, T., Bryant, S.H., Jones, J.S. and Moore, J.A. (1982) Long distance migration of *Drosophila. Amer. Natur.*. 119:589–595.

Crawford, T. (1984) What is a population. In *Evolutionary ecology.* Ed. B. Shorrocks. Blackwell scientific publications, Oxford.

Crow, J.F. and Aoki, K. (1984) Groups selection for a polygenic behavioural trait: estimating the degree of population subdivisions. *Proc. Natl. Acad. Sci. USA.* 81:6073–6077

Crow, J.F. and Denniston, C. (1988) Inbreeding and variance effective population numbers. *Evolution* 42:482–495

Crow, J.F. and Kimura, M. (1970) *An introduction to population genetics theory.* Harper & Row.

Crowley, P.H. (1992) resampling methods for computation intensive data analysis in ecology and evolution. *Ann. Rev. Ecol. Syst.*. 23:405–447.

Crumpacker, D.W. and Williams, J.S. (1973) Density, dispersion and population structure in *Drosophila pseudoobscura. Ecol. Monogr.*. 43:499–538.

Currey, J.D. and Hughes, R.N. (1982) Strenght of the dogwhelk *Nucella lapillus* and the winkle *Littorina littorea* from different habitats. *J. Anim. Ecol.*. 51:47–56.

Darmency, H. and Renard, M. (1992) Efficiency of safety procedures in experiments with transgenic oil-seed rape. In *Second International Symposium on the Biosafety Results of Field Tests of Genetically Modified Plants and Microorgarnisms.* Eds. R. Casper and J. Landsmann. BBLF Braunschweig, Germany.

Dark, S.O.S. (1971) Experiment on the cross-pollination of sugar beet in the field. *J. Natur. Inst. Agric. Bot.*. 12:246–66.

Dawson, W.R, Ligon, J.D, Murphy, J.R., Myers, J.P., Simberloff, D. and Verner, J. (1987) Report of the scientific advisory panel on the spotted owl. *Condor.* 89:205-229.

Day, A.J. (1990) Microgeographic variation in allozyme frequencies in relation to the degree of exposure to wave action in the dogwhelk, *Nucella lapillus* (L.) (Prosobranchia:Muricacea). *Biol. J. Linn. Soc.*. 40:245–261.

Day, A.J. and Bayne, B.L. (1988) Allozyme variation in population of the dogwhelk, *Nucella lapillus* from the South West peninsula of England. *Marine Biol.*. 99:93–100.

Dice, L.R. and Howard, W.B. (1951) Distance of dispersal by prairie deer mice from birthplaces to breeding sites. *Contr. Lab. Vert. Biol. Univ. Mich.* 50:1–15.

Dobzhansky, Th. and Epling, C. (1944) *Contribution to the genetics, taxonomy and ecology of* Drosophila pseudoobscura *and its relatives.* Publication 554. Carnegie Institution of Washington.

Dobzhansky, Th. and Wright, S. (1941) Genetics of natural populations. V. Relation between the mutation rate and the accumulation of lethals in population of *Drosophila pseudoobscura. Genetics.* 26:23–51.

Dobzhansky, Th. and Wright, S. (1943) Genetics of natural populations. X. Dispersion rates in *Drosophila pseudoobscura.* 28:304–340.

Doebley, J. (1990) Molecular evidence for gene-flow among *Zea* species. *Bioscience.* 40:443–448

Doney, D.L., Whitney, E.D., Terry, J., Freese, L.,Fitzgerald, P. (1990) The distribution and dispersal of *Beta vulgaris* ssp. *maritima* germplasm in England, Wales and Ireland. *J. Sugar Beet Res.*. 27:29–36

Efron, B. (1979) Bootstrap methods: another look at the jackknife. *Ann. Stat.*. 7:1–26.

Eijlander, R. (1989) Yes or no to transgenic beet? *Prophyta.* 43:232–233

Ellstrand, N.C. and Hoffman, C.A. (1990) Hybridisation as an avenue of escape for engineered genes. *Bioscience.* 40:438–442

Endler, J.A. (1977) *Geographic variation, speciation, and clines.* Princeton University Press.

Eppling, C. and Dobzhansky, Th. (1942) Genetics of natural populations: VI. Microgeographical races in *Linanthus parryae. Genetics.* 27:317–332.

Ewens, W.J. (1972) The sampling theory of selectively neutral alleles. *Theoret. Pop. Biol.*. 3:87–112.

Ewens, W.J. (1979) *Mathematical Population Genetics.*. Berlin: Springer-Verlag.

Ewens, W.J (1982) On the concept of the effective population size. *Theoret. Pop. Biol.*. 21:373–378.

Ewens, W.J. (1989) The effective population sizes in presence of catastrophes. In *Mathematical Evolutionary Theory*, Ed. M.W. Feldman, Princeton University Press, princeton, NJ.

Ewens, W.J. (1990) The minimum viable population size as a genetic and demographic concept. In *Convergent issues in genetics and demography*. Ed. J. Adams, D.A. Lam, A.I. Hermalin and P.E. Smouse. Oxford University Press.

Excoffier, L., Smouse, P.E., Quattro, J.M. (1992) Analysis of molecular variance inferred from metric distances among DNA-haplotypes. Applications to human mitochondrial DNA restriction data. *Genetics*. 131:479–491.

Feldman, M.W. and Christiansen, F.B. (1975) The effect of population subdivision on two loci without selection. *Genet. Res.*. 24:151–162.

Felsenstein, J. (1975) A pain in the torus: some difficulties with models of isolation by distance. *Amer. Natur.* 109:359–68.

Felsenstein, J.(1976) The theoretical population genetics of variable selection and migration. *Ann. Rev. Genet.*. 10:253–280.

Gale, J.S. (1990) *Theoretical population genetics.* London: Unwin Hyman (Ed.).

Gilpin, M.E. (1991) The genetic effective size of a metapopulation. *Biol. J. Linn. Soc.*. 42:165–175.

Gliddon, C.J (In Press) The impact of hybrids between genetically modified crops and their related species: biological models and theoretical perspectives. *Mol. Ecol.*

Gliddon, C.J. and Goudet, J. (In Press) The genetic structure of metapopulations and conservation biology. in *Conservation Genetic*, Ed. V. Loeschke, J. Toniuk, and S. Jain. Exs series, Birkhauser Verlag.

Gliddon, C.J. and Gouyon, P.H. (1989) The units of selection. *Trends Ecol. Evol.*. 4:204–208.

Gliddon, C.J. and Saleem, (1985) Gene flow in *Trifolium repens*. An expanding genetic neighbourhood. In *Genetic Differentiation and Dispersal in Plants*. Ed. P. Jacquard, G. Heim and J. Antonovics, Springer-Verlag.

Golding, G.B. and Strobeck, C. (1983) Variance and covariance of homozygosity in a structured population. *Genetics*. 104:513–529.

Goodman, D. (1987) The demography of chance extinction. In Viable Population for Conservation. Ed. M.E. Soulé, cambridge University Press, Cambridge.

Goodnight, C.J. (1987) On the effect of founder events on epistatic genetic variance. *Evolution* 41:80-91.

Goodnight, C.J. (1988) Population differentiation and the transmission of density effects between generations. *Evolution*. 42:399–403.

Goudet, J., De Meeüs, T., Day, A.J. and Gliddon, C. J. (In Press) The different levels of population structuring of the dogwhelk, *Nucella lapillus*, along the South Devon Coast. In *Genetics and Evolution of Aquatic Organisms*, Ed. A.R. Beaumont, Chapman & Hall, London.

Gouyon, P.H. (1986) Comment le paramètre F mesure _t_il la divergence? Séminaire du département de génétique et d'amélioration des plantes. INRA, Méribel 86.

Gray, A.J., Raybould, A.F., Warman, E.A., Jeal, K., Gliddon, C.J., Goudet, J., and Mogg, R.J. (1992) *Gene flow in natural populations of* Brassica *and* Beta. Institute of Terrestrial Ecology (Natural environment research council) Project T08068G1, DoE/NERC Contract PECD/7/8/190 First year report.

Gregorius, H.R. (1991) On the concept of effective number. *Theoret. Pop. Biol.*. 40:269–283.

Haldane, J.B.S. (1939) The equilibrium between mutation and random extinctions. *Ann. Eugen.*. 9:400–405

Haldane, J.B.S. (1954) An exact test for randomness of mating. *J. Genet.*. 52:631–635.

Hardy, G.H. (1908). Mendelian proportions in a mixed population. *Science*. 28:49–50.

Harris, H. (1966) Enzyme polymorphism in man. *Proc. Roy. Soc. Ser. B.* 164:298–310.

Harris, R.B. and Allendorf, F.W. (1989) Genetical effective population size of large mammals: an assessment of estimators. *Conserv. Biol.* 3:181–191.

Hartl, D.E. and Clark, A.G. (1989) *Principles of population genetics.* 2nd Ed. Sinauer Associates. Sunderland, Mass.

Heyland, K.U. and Kaul, H.P. (1993) A model of weed population-dynamics used for the determination of optimal weed-control strategies with regard to crop management of winter-wheat and sugar-beet. *Bodenkultur.* 44:29–50.

Heywood, J.S. (1991) Spatial analysis of genetic variation in plant populations. *Ann. Rev. Ecol. Syst.*. 22:335–355.

Hubby, J.L. and Lewontin, R.C. (1966) A molecular approach to the study of genic heterozygosity in natural populations. I. The number of alleles at different loci in *Drosophila pseudoobscura*. *Genetics*. 54:577–594.

Hughes, R.N. (1972) Annual production of 2 Nova Scotian of *Nucella lapillus*(L.). *Oecologia*. 8:356–370.

Johnston J.S. and Heed, W.B. (1975) Dispersal of *Drosophila*: the effect of baiting on the behavior and distribution of natural populations. *Amer. Natur.*. 108:207–216.

Jones, J.S, Bryant, S.H., Lewontin, R.C., Moore, J.A. and Prout, T. (1981) Gene flow and the geographic distribution of molecular polymorphism in *Drosophila pseudoobscura*. *Genetics*. 98:157–178.

Jones, J.S., Leith, B.H. and Rawlings, P. (1977) Polymorphism in *Cepaea*. *Ann. Rev. Ecol. Syst.*. 8:109–143.

Kareiva, P., Morris, W. and Jacobi, C. (In Press) Studying and managing the risk of cross-fertilization between transgenic crops and wild relatives. *Mol. Ecol.*

Kimura, M. (1955) Solution of a process of random genetic drift with a continuous model. *Proc Natl. Acad. Sci. USA*. 41:144–50.

Kimura, M. (1983) *The neutral theory of molecular evolution.* Cambridge University Press.

Kimura, M. and Weiss, G.H. (1964) The stepping-stone model of population structure and the decrease of genetic correlation with distance. *Genetics*. 49:561–576.

Kirby, G.C. (1975) Heterozygote frequencies in small populations. *Theoret. Pop. Biol.*. 8:31–48.

Knuth, D.E. (1981) *The art of computer programming.* Vol. 2: seminumerical algorithms. 2nd Ed. Addison-Wesley series in computer science and information processing. Addison-Wesley Publishers, reading, Massachusetts.

Kruskall, W.H. and Wallis W.A. (1952) Use of ranks on one-criterion variance analysis. *J. Amer. Statist. Assoc.*. 47:583–621.

L'Ecuyer, P. (1988) Efficient and portable random number generators. *Communications of the ACM* 31:147–157.

Lamotte, M. (1951) Recherches sur la structure génétique des populations naturelles de *Cepaea nemoralis* (L.). *Bull. Biol. Fr. Belg.*. suppl. 35:1–238

Lamotte, M. (1959) Polymorphism of natural populations of *Cepaea nemoralis*. *Cold Spring harbor Symp. Quant. Biol.*. 24:65–86.

Lande (1988) genetic and demography in biological conservation. *Science*. 241:1455–1460.

Latter, B.D.H. (1973) The island model of population differentiation: a general solution. *Genetics*. 73:147–157.

Latter, B.D.H. (1975) Influence of selection pressures on enzyme polymorphisms in *Drosophila*. *Nature*. 257:590–592.

Levene, H. (1949) On a matching problem arrising in genetics.*Ann. Math. Stat.* 20:91–94.

Levin, D.A. and Kerster, H.W. (1968) Local gene dispersal in *Phlox pilosa*. *Evolution*. 22:130–139.

Levin, D.A. and Kerster, H.W. (1974) Gene flow in seed plants. *Evol. Biol.*. 7:139–220.

Levins, R. (1970) Extinctions. *Lect. Math. Life Sci.*. 2:75–77.

Lewin, R. (1990) Population bottlenecks enhance a species genetic-resources. *New Scientist*. 125:32–33.

Lewontin, R.C. (1974) *The genetic basis of evolutionary change.* New York: Columbia University Press.

Lewontin, R.C. (1991) Twenty five years ago in genetics: Electrophoresis in the development of evolutionary genetics: Milestone or Milstone? *Genetics*. 128:657–662

Lewontin, R.C. and Hubby, J.L. (1966) A molecular approach to the study of genic heterozygosity in natural populations. II. Amount of variation and degree of heterozygosity in natural populations *Drosophila pseudoobscura*. *Genetics*. 54:595–609.

Li, C.C. (1955) *Population genetics.* University of Chicago Press.

Li, C.C. and Horvitz, D.G. (1953) Some methods of estimating the inbreeding coefficient. *Amer. J. Hum. Genet.*. 5:107–117.

Long, J.C. (1986) The allelic correlation structure of Gainj and Kalam speaking people. I. The estimation and interpretation of Wright's F-statistics. *Genetics*. 112:629–647.

McCauley, D.E. (1991) Genetic consequences of local population extinction and recolonization. *Trends Evol. Ecol.*. 6:5–8.

McComas, S.A. and Bryant,E.H (1990) Loss of electrophoretic variation in serially bottlenecked populations. *Heredity* 64:315–321.

McPherson, J.M., Eckstein, P.E., Scoles, G.J. and Gajadhar, A.A. (1993) Variability of the Random Amplified Polymorphic DNA assay among thermal cyclers, and effects of primer and DNA concentration. *Mol. Cell. Prob.* 7:293–299.

Mahalanobis, P.C. (1936) On the generalised distance in statistics. *Proc. Natl. Inst. Sci. India.* 2:49–55.

Malécot, G. (1948) *Les Mathématiques de l'hérédité.* Paris: Masson

Manasse, R.S. (1992) Ecological risks of transgenic plants: effect of spatial dispersion on gene flow. *Ecol. Appl.*. 2:431–438.

Manly, B.F.J. (1991) *Randomization and Monte-Carlo methods in biology.* Chapman and all, London.

Marayuma, T. (1970) On the rate of decrease of heterozygosity in circular stepping stone models. *Theret. Pop. Biol.*. 1:101–119

Marayuma, T. (1971a) Speed of gene substitution in in a geographically structured population. *Amer. Natur.*. 105:253–265.

Marayuma, T. (1971b) An invariant property of structured population. *Genet. Res.*. 18:81–84.

Marayuma, T. (1971c) The rate of decrease of heterozygosity in a population occupying a circular or a linear habitat. *Genetics.* 67:437–454.

Marayuma, T. (1972a) Some invariant properties of a geographically structured finite population: distribution of heterozygotes under irreversible mutation. *Genet. Res.*. 20:141–149.

Marayuma, T. (1972b) Rate of decrease of genetic variability in a two dimensional continuous population of finite size. *Genetics.* 70:639–651.

Marayuma, T. (1972c) The rate of decay of genetic variability in a geographically structured finite population. *Math. Biosi.*. 14:325–35.

Marayuma, T. (1974) A simple proof that certain quantities are independant of the geographic structure of a population. *Theoret. Pop. Biol.*. 5:148–154.

Marayuma, T. and Kimura, M. (1980) Genetic variability and effective population size when local extinction and recolonization of subpopulations are frequent. *Proc. Natl. Acad. Sci. USA.* 77:6710–6714.

Marsaglia, G., McLaren, M.D. and Bray, T.A. (1964) A fast procedure for generating normal random variables. *Communications of the ACM.* 7:4–19.

May, R.M. (1992) How many species inhabit the earth. *Sci. Amer.*. 267:42–48.

Maynard Smith, J. (1989) *Evolutionary genetics.* Oxford University Press. New York.

Mayr, E. (1959) Where are we? *Cold Spring Harbor Symposia Quant. Biol.*. 24:1–14.

Nagylaki, T. (1983) The robustness of neutral models of geographical variation. *Theoret. Pop. Biol.* 24:268–294.

Nei, M. (1972) Genetic distances between populations. *Amer. Natur.*. 105:385–398.

Nei, M. (1973) Analysis of gene diversity in subdivided populations. *Proc. Natl. Acad. Sci. USA.* 70:3321–3323.

Nei, M. (1975) *Molecular Population Genetics and Evolution.* Amsterdam: North-Holland.

Nei, M. (1977) F-statistics and analysis of gene diversity in subdivided populations. *Ann. Hum. Genet.*. 41:225–233.

Nei, M. (1978) Estimation of average heterozygosity and genetic distance from a small number of individuals. *Genetics.* 89:583–590.

Nei, M. (1986) Definition and estimation of fixation indices. *Evolution.* 40:643–645

Nei, M. (1987) *Molecular evolutionary genetics.* Columbia University Press

Nei, M., Chakravarti, A. and Tateno, Y. (1977) Mean and variance of F_{st} in a finite number of incompletely isolated populations. *Theoret. Pop. Biol.*. 11:291–306.

Nei, M. and Chesser, R.K. (1983) Estimation of fixation indices and gene diversities. *Ann. Hum. Genet.*. 47:253–259.

Nei, M. and Tajima, F. (1981) Genetic drift and estimation of effective population size. *Genetics* 98:625–640.

Nevo, E., Beiles, A. and Ben-Shlomo, R. (1984) The evolutionary significance of genetic diversity: ecological, demographic and life history correlates. *Lecture Notes in Biomathematics.*. 53:13–213

Nunney, L. and Campbell, K.A. (1993) Assessing Minimum Viable Population Size: Demography Meets Population genetics. *Trends. Ecol. Evol.*. 8:234–239

Ohta, T. (1976) Role of very slightly deleterious mutations in molecular evolution and polymorphism. *Theoret. Pop. Biol.*. 10:254–275.

Ohta, T. (1982) Linkage disequilibrium due to random genetic drift in finite subdivided populations. *Proc. Natl. Acad. Sci. USA.* 79:1940–1944.

Olivieri, I., Couvet, D. and Gouyon, P.H. (1990) The genetics of transient populations–research at the metapopulation level. *Trends Ecol. Evol.*. 5:207–210.

Park, S.K. and Miller, K.W. (1988) Random number generators: good ones are hard to find.*Communications of the ACM.* 31:1192–1201.

Parzen, E. (1960) *Modern probability theory and its applications.* New York Wiley.

Pearson, K. (1926) On the coefficient of racial likeness. *Biometrika*. 18:337–343.

Pollak, E. (1983) A new method for estimating the effective population size from allele frequency changes. *Genetics*. 104:531–548.

Pollak, E. (1987) On the theory of partially inbreeding finite populations. I. Partial selfing. *Genetics*. 117:353–360.

Pollak, E.(1988) On the theory of partially inbreeding finite populations. II. Partial sib mating. *Genetics*. 120:303–311

Quenouille, M.H. (1956) Note on bias in estimation. *Biometrika*. 43:253–260.

Raybould, A.F. and Gray, A.J. (1993) Genetically modified crops and hybridization with wild relatives: a UK perspective. *J. Appl. Ecol.*. 30:199–219.

Riedy, M.F., Hamilton, W.J. and Aquadro, C.F. (1992) Excess of non-parental bands in offspring from known primate pedigrees assayed using RAPD PCR. *Nuc. Ac. Res.*. 20:918.

Ripley, B.D. (1987) *Stochastic simulation*. Wiley series in probability and mathematical ststistics. Wiley & sons.

Ritland, K.(1990) A series of FORTRAN computer-programs for estimating plant mating systems. *J. Hered.*. 81:235–237.

Rogers, J.S. (1972) Measures of genetic similarity and genetic distance. In *Studies in genetics VII* pp. 145–153. University of Texas Publication 7213. Austin, Tex.: University of Texas.

Rohlf, J.F. and Schnell, G.D. (1971) An investigation of the isolation by distance model. *Amer. Natur.*. 105:295–324.

Roughgarden, J. (1979) *Theory of population genetics and evolutionary ecology: an introduction*. Mac Millan, New York.

Ryman,N. and Laikre, L. (1991) Effects of supportive breeding on the genetically effective population size. *Conserv. Biol.*. 5:325–329.

Santoni, S. (1993) Evidence for gene exchanges between sugar-beet *Beta vulgaris* (L.) and wild beets. Consequences for transgenic sugar-beet. *Plant Mol. Biol.*. 21:413.

Santoni, S. and Berville, A. (1992) Evidence for gene exchanges between sugar-beet *Beta vulgaris* (L.) and wild beets. Consequences for transgenic sugar-beet. *Plant Mol. Biol.*. 20:575–580.

Scheirer, C.J. Ray, W.S. and Hare N. (1976) The analysis of ranked data from completely randomized factorial designs. *Biometrics*. 32:429–434.

Schwartz, O.A., Bleich, V.C. and Holl, S.A. (1986) Genetics and the conservation of mountain sheep *Ovis canadensis nelsoni*. *Biol. Conserv.*. 37:179–190.

Slatkin, M. (1977) Gene flow and genetic drift in a species subject to frequent local extinctions. *Theoret. Pop. Biol.*. 12:253–262.

Slatkin, M. (1985a) Gene flow in natural populations. *Ann. Rev. Ecol. Syst.*. 16:393–430.

Slatkin, M. (1985b) Rare alleles as indicators of gene flow. *Evolution*. 39:53–65.

Slatkin, M. (1987) Gene flow and the geographic structure of natural populations. *Science*. 236:787–792.

Slatkin, M. (1993) Isolation by distance in equilibrium and non-equilibrium populations. *Evolution*. 47(1):264–279.

Slatkin, M. and Arter, H.E. (1991a) Spatial autocorrelation methods in population genetics. *Amer. Natur.*. 138:499–517.

Slatkin, M. and Arter, H.E. (1991b) Reply to Sokal and Oden. *Amer. Natur.* 138:522-523.

Slatkin, M. and Barton, N.H. (1989) A comparison of three indirect methods for estimating levels of gene flow. *Evolution*. 43:1349–1368.

Slatkin, M. and Maddison W.P. (1989) A cladistic measure of gene flow from the phylogenies of alleles. *Genetics*. 123:603–613.

Slatkin, M. and Maddison W.P. (1990) Detecting isolation by distance using phylogenies of genes. *Genetics*. 126:249–260.

Slatkin, M. and Takahata, N. (1986) Private alleles in a partially isolated population. 2. Distribution of persistence time and probability of emigration. *Theoret. Pop. Biol.*. 30:180–193.

Sokal, R.R. and Oden N.L. (1991) Spatial autocorrelation analysis as an inferential tool in population genetics. *Amer. Natur.* 138:518–521.

Sokal, R.R. and Rohlf, F.J. (1981) *Biometry*. Freeman & Co. New York.

Soulé, M.E. (1986) *Conservation Biology. The science of scarcity and Diversity*. Sinauer Associates, Sunderland, Mass.

Soulé, M.E and Mills, L.S. (1992) Conservation genetics and conservation biology: a troubled marriage. In *Conservation of biodiversity for sustainable development*. Ed. O.T. Sandlund, K. Hindar and A.H.D. Brown, Oxford University Press, Oxford.

Swofford, D.L. and Selander, R.B. (1981) BIOSYS-1: a FORTRAN program for the comprehensive analysis of electrophoretic data in population-genetics and systematics. *J. Hered.*. 72:281–283.

Takahata, N. (1983) Gene identity and genetic differentiation of populations in the finite island model. *Genetics*. 104:497–512.

Takahata, N. and Nei, M. (1984) F_{ST} and G_{ST} statistics in the finite island model. *Genetics*. 107:501–504

Templeton, A.R. (1986) Coadaptation and outbreeding depression. In *Conservation Biology: The science of scarcity and diversity*, ed M.E. Soulé. Sinauer, Sunderland, Mass.

Templeton, A.R. (1991) Off-sites breeding of animals and implication for plant conservation strategies. In *Genetics and conservation of rare plants*. Ed. D.A. Falk and K.E. Holsinger. Oxford University Press, Oxford.

Templeton, A.R., Hemmer, H., Mace, J., Seal, U.S., Shields, W.M. and Woodruff, D.S. (1986) Local adaptation, coadaptation and population boundaries. *Zoo Biol.*. 5:115–125

Templeton, A.R. and Read, B. (1983) The elimination of inbreeding depression in a captive herd of Speke's gazelle. In *Genetics and Conservation: A reference for managing wild animal and plant populations*, ed. C.M. Schonewald-Cox, S.M. Chambers, B. MacBryde and L. Thomas. Benjamin-Cummings, Menlo Park, Calif.

Thompson, K.F. (1976) Cabbages, kales etc.. In *Evolution of crop plants*. Ed. N.W. Simmonds. Longman, London.

Thompson, K.F. and Taylor, J.P. (1966) The breakdown of self-incompatibility in cultivars of *Brassica oleracea*. *Heredity*. 21:345–362.

Tukey, J.W. (1958) Bias and confidence in not quite large samples. *Ann. Math. stat.* 29:614.

Wade, M.J. and McCauley, D.E. (1988) Extinction and recolonization: their effects on the genetic differentiation of local populations. *Evolution*. 42:995–1005.

Walhund, S. (1928) Zusammensetzung von Populationen und Korrelationserscheinungen vom Standpunkt der Vererbungslehre aus betrachtet. *Hereditas*. 11:65–106.

Wallace, B. (1966) On the dispersal of *Drosophila*. *Amer. Natur.*. 100:551–563.

Waples, R.S. (1989) A generalized approach for estimating effective population size from temporal changes in allele frequency. *Genetics* 121:379–391.

Watterson, G.A. (1978) The homozygosity test of neutrality. *Genetics*. 88:405–417.

Weinberg, W. (1908) Über den Nachweis der Vererbung beim Menschen. *Jahresh. Verein f. vaterl. Naturk. Württem.*. 64:368–382.

Weir, B.S. (1990) *Genetic data analysis*. Sinauer associates. Sunderland, Mass.

Weir, B.S. and Basten, C.J. (1990) Sampling strategies for distances between DNA sequences. *Biometrics*. 46:551–572.

Weir, B.S. and Cockerham, C.C. (1984) Estimating F-statistics for the analysis of population structure. *Evolution*. 43:1358–1370

Weiss, G.H. and Kimura, M. (1965) A mathematical analysis of the stepping-stone model of genetic correlation. *J. Appl. Prob.*. 2:129–149

Williams, J.T. (1988) Identifying and protecting the origin of our food plants. In *Biodiversity*. Ed. E.O. wilson, Natinal Academy press, Washington, DC.

Wilson, H.D. (1990) Gene-flow in squash species. *Bioscience*. 40:449–455

Wright, S. (1921) Systems of mating. *Genetics*. 6:111–178.

Wright, S. (1922) Coefficients of inbreeding and relationship. *Amer. Natur.*. 56:330–338.

Wright, S. (1931) Evolution in mendelian populations. *Genetics*. 16:97–159

Wright, S. (1940) Breeding structure of populations in relation to speciation. *Amer. Natur.*. 74:23–248

Wright, S. (1943) Isolation by distance. *Genetics*. 28:114–138

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

Wright, S. (1969) *Evolution and the genetics of Populations*. Vol 2. The theory of gene frequencies. University of Chicago Press.

Wright, S. (1977) *Evolution and the genetics of Populations*. Vol 3. Experimental results and evolutionary deductions. University of Chicago Press.

Wright, S. (1978) *Evolution and the genetics of Populations*. Vol 4. Variability within and among natural populations. University of Chicago Press.

Yarnell, S.H. (1956) Cytogenetics of the vegetable crops. II. Crucifers. *Bot. Rev.*. 22:81–166.

Appendix A

Listing of MODEL42

A.1 STEPINF.PAS

```
Unit stepinf;
(*****************************************************************************)
(*                                                                           *)
(*       WARNING !!! All the declarations within INTERFACE are               *)
(*         global;---> you can use them in all the programs                  *)
(*               using The unit StepInf                                      *)
(*                                                                           *)
(*****************************************************************************)

Interface


uses
crt,Graph,dos,Drivers{,Fonts};

const

            MaxInd       = 4096;
            n            = 64;
            DMaxInd      = 8192;

GFP_Mes : Array[0..9] of string[40]=
           ('',
           'ISLAND MODEL WITH INF. CONTINENT',
           'ISLAND MODEL WITH GAMETIC CLOUD',
           'STEP. STO. MODEL 1 DIM. EXP. DISTR.',
           'STEP. STO. 2 DIM. CON. 4',
           'STEP. STO. 2 DIM. CON. 8 EXP. DISTR.',
           'STEP. STO. 2 DIM. CON. 8 NOR. DISTR.',
           'STEPPING STONED NEIGHBOURHOOD',
           'PLANT LATTICE MODEL',
           'TRUE WRIGHT LATTICE MODEL');
type

Names         = String[40];
Ext           = string[3];

Struct        = array[1..3] of word;

W_Coord       = array[1..4] of word;

Point         = record
                  x,y    :integer;
                end;

ExistsPtr     =^Exists;
Exists        = Array[1..DMaxInd] of Boolean;


AliveGenPtr   =^AliveGen;
AliveGen      = Array[1..DMaxInd] of Word;

IntFldPtr     =^IntFld;
IntFld        = Array[1..MaxInd,1..2] of Word;

Descr_GFP1     = record
                       kind       : char;
                       self       :single;
                       mig        :single;
                       popsize    :word;
                       unused     :array[1..5] of byte;
                 end;
Descr_GFP4     = record
                       kind       : char;
                       self       :single;
```

```
                       mig        :single;
                       popsize    :word;
                       tor        :boolean;
                       unused     :array[1..4] of byte;
                 end;
Descr_GFP3     = record
                       kind       : char;
                       self       :single;
                       mig        :single;
                       popsize    :word;
                       tor        :boolean;
                       dist       :byte;
                       Aver       :Byte;
                       unused     :array[1..2] of byte;
                 end;
Descr_GFP7     = record
                       kind       : char;
                       self       :single;
                       popsize    :word;
                       tor        :boolean;
                       aver       :SINGLE;
                       unused     :array[1..4] of byte;
                 end;
Descr_GFP8     = record
                       kind       : char;
                       self       :single;
                       dispm,dispf:single;
                       tor,plant  :boolean;
                       unused     :array[1..1] of byte;
                 end;
DescrRec       = record
                       FilNam     :names;
                       NewData    :array[1..16] of byte;
                 end;

IntFldFile    = File Of IntFld;
AliveGenFile  = File Of AliveGen;
ExistsFile    = File Of Exists;
DescrRecFile  = File of DescrRec;

const
     W_Size_Small : W_Coord=(5,15,79,24);
     W_Size_Big   : W_Coord=(5,3,79,24);

var

  FilNamDat,Name,FilNamPar                  : Names;
  FilNamRed,FilNamTxt,FilNamFre             : Names;
  GrDriver,GrMode                           : integer;
  Graphics                                  : boolean;
  ErrCode                                   : Byte;
  FileDat,FileRed                           : IntfldFile;
  ThisDescrRec                              : DescrRec;
  FileDescrRec                              : DescrRecFile;
  FileTxt,File6,Filepar,FScreen             : Text;
  FileFre                                   : AliveGenFile;
  MaxAll                                    : Word;
  Ext1                                      : ext;
  S1,S2                                     : LongInt;
  D                                         : DirStr;
  Na                                        : NameStr;
  E                                         : ExtStr;
(*****************************************************************************)
Function lintrange(low,high:longint) : longint;
Function range(low,high:real):real;
Procedure FileList(ext1:ext);
Function Uniform : Double;
Function Grandom( N :LongInt) : Longint;
Procedure Window2(X : W_Coord);
Procedure Ini(var Fld:IntFldPtr);
Procedure FillBoolArray(Param:Single;var Arbool:ExistsPtr);
Function FileExist(Name :Names):Boolean;
Procedure Erase_File(var Name:Names);
Function Affirmed(Default,Ingraph:boolean):Boolean;
Procedure Message_End;
Procedure Write_Descr_GFP(ss:char;var F:IntFldFile);
Procedure Read_Descr_GFP(var ff:text;var F:IntFldFile);
Procedure UpDate_FileRec(Var NewRec:DescrRec);
Procedure InitTxt(Name:Names);
Function GetNewCoord(Tor:boolean;a,N:integer;Ofsa:ShortInt):integer;
(****************************************************************************
 ****************************************************************************
 *****************************************************************************)
IMPLEMENTATION
(***********************CHECK REAL AND INTEGER VALUES*************************)
function licheck : longint;
var x : longint;
begin
     {$I-}
     readln(x);
     {$I+}
     if IoResult<>0 then
           repeat
              writeln;
              write('Input not in correct form- Please retype ');
              {$I-}
              readln(x);
              {$I+}
           until IoResult=0;
     licheck:=x;
end;    (*licheck*)

function lintrange(low,high:longint) : longint;
var x : longint;
begin
     x:=licheck;
     while (x<low) or (x>high) do
     begin
          write('Out of range, Please retype: ? ');
          x:=licheck;
          writeln;
     end;
 lintrange:=x;
end; (*lintrange*)

function rcheck : real;
var x : real;
begin
     {$I-}
     readln(x);
     {$I+}
     if IoResult<>0 then
           repeat
              writeln;
              write('Input not in correct form- Please retype ');
              {$I-}
              readln(x);
              {$I+}
           until IoResult=0;
     rcheck:=x;
end; (*rcheck*)

function range(low,high:real) : real;
var x : real;
begin
     x:=rcheck;
     while (x<low) or (x>high) do
     begin
          write('Out of range, Please retype: ? ');
          x:=rcheck;
          writeln;
     end;
     range:=x;
end; (*range*)
(**********************FILES IN DIRECTORY*************************************)
procedure FileList(Ext1:ext);
var
DirInfo       : Searchrec;
Xpos,YPos     :Byte;
P             :PathStr;
D             :DirStr;
N             :NameStr;
E             :ExtStr;
begin
 XPos:=1;
 TextBackGround(Black);
 Window(5,3,79,12);
 clrscr;
 TextBackGround(LightGray);TextColor(Yellow);
 Window(3,2,77,11);
 ClrScr;
 GotoXY(23,1);write('LIST OF FILES IN DIRECTORY:');
 TextColor(Blue);
 FindFirst('*.'+ext1,Anyfile, DirInfo);
 while DosError = 0  do
     begin
      FSplit(Dirinfo.Name,D,N,E);
      If ((E='')
      or  (E='.SAG')
      or  (E='.FIQ')
      or  (E='.DAT')
      or  (E='.RED'))
      Then Begin
                If WhereY+1<(Hi(WindMax))
                Then GoToXY(Xpos,WhereY+1)
                Else Begin
                          Xpos:=XPos+14;
                          YPos:=2;
                          GoToXY(XPos,YPos);
                     End;
                write(dirinfo.name);
          end;
```

```
      findnext(dirinfo);
     end;
GoToXY(50,Hi(WindMax));TextColor(YELLOW);
Write('PRESS ANY KEY TO PROCEED.');
Repeat Until ReadKey>#1;
GotoXY(50,Hi(WindMax));ClrEol;
end;
(***************GET ANSWER Y,N OR DEFAULT AND PASS IT BACK*******************)
function affirmed(default,ingraph:boolean):boolean;
(* Waits for yes or no or default for return only *)
var
   gotit,yesno :  boolean;
   ans,dans : char;
begin
repeat
gotit:=true;
if default then
 begin
      if ingraph then outtext(' [Yes] : ')
                 else write(' [Yes] : ');
      dans:='y'
 end
else
 begin
      if ingraph then outtext(' [No] : ')
                 else write(' [No] : ');
      dans:='n';
 end;
if ingraph then outtext(' Y or N? ')
           else write(' Y or N? ');
repeat
      ans:=ReadKey;
until ans>=#13;
if ans=#13 then ans:=dans;
if Upcase(ans) in ['Y','N'] then
 case ans of
            'y','Y': yesno:=true;
            'n','N': yesno:=false;
 end (* ans *)
else
 begin
      if ingraph then begin
                        SetFillStyle(SolidFill,GetBkColor);
                        Bar(0,GetY-10,640,GetY);
                        MoveTo(5,GetY);
                        outtext('Please answer ');
                      end else begin
                                 writeln;
                                 write('Please answer ');
                               end;
      gotit:=false;
 end;
until gotit;

affirmed:=yesno;
if ingraph then outText(Upcase(ans)) else writeln(Upcase(ans));
end (* affirmed *) ;
(******************************MESSAGE_END**********************************)
Procedure Message_End;
Begin
     SetTextStyle(defaultFont,HorizDir,1);
     SetTextJustify(LeftText,CenterText);
     SetColor(getcolor);
     OutTextXY(5,GetMaxY-5,'PRESS ANY KEY TO PROCEED.');
     Repeat Until ReadKey>#1;
     cleardevice;
     setBkColor(Blue);
     RestoreCrtMode;
     Graphics:=False;
End; {Of Proc Message_End}
```

A.1.1 UNIFORM.PAS

```
(****************** UNIFORM FUNC*********************************************)
Function Uniform : Double;
Var
Z,k     : LongInt;
Begin
  k:=S1 div 53668;
  S1:=40014*(S1-k*53668)-k*12211;
  If S1 < 0 Then S1:=S1+2147483563;
  k:=S2 div 52774;
  S2:=409692*(S2-k*52774)-k*3791;
  If S2<0 Then S2:= S2+2147483399;
  Z:= s1-s2;
  If Z<1 Then Z:= z+2147483562;
  Uniform:=Z*4.656613E-10;
  end; {Of Func Uniform}

Function Grandom( N: Longint) : LongInt;
```

```
Begin
Grandom:=Trunc(UniForm*N);
end; {Of Func Grandom}


(*******************************2nd WINDOW************************************)
Procedure Window2( X:W_Coord);
Begin
     TextBackGround(black);
     Window(x[1],x[2],x[3],x[4]);ClrScr;
     TextBackGround(Blue);TextColor(15);
     Window(x[1]-2,x[2]-1,x[3]-2,x[4]-1);
     ClrScr;
End;
(********************INITIALISATION PROCEDURE*********************************)
Procedure Ini(Var Fld:IntFldPtr);
var i: word;
Begin
     MaxAll:=DMaxInd;
     For i:=1 to MaxInd do
     Begin
          Fld^[i,1]:=i;
          Fld^[i,2]:=MaxInd+i;
     end;
end;
(********************FILL_BOOLEAN_ARRAY PROCEDURE*****************************)
Procedure FillBoolArray(Param : Single;var ArBool   :ExistsPtr);
var i  : integer;
Begin
     For i:=1 to DMaxInd Do arbool^[i]:=Uniform<Param;
End;{Of Proc FillBoolArray}
(***********FILEEXIST FUNC***************************************************)
Function FileExist(Name :Names):Boolean;
var f  : File;
Begin
  Assign(f,Name);
{$I-}
  reset(f);
  close(f);
{$I+}
  FileExist:=IOResult=0;
End; { Of Function FileExist}
(***********ERASE_FILE PROC***************************************************)
Procedure Erase_File(Var Name:Names);
Var f :File;
begin
     ClrScr;
     graphics:=false;
     Window2(W_Size_Big);
     Write('Input File Name for Output (Without ext.): ');
     Readln(Name);
     If FileExist(name)
     Then Begin
               Write('WARNING!!. File Already exists. Do you want to Erase it?');
               If Affirmed(False,Graphics)
               Then Begin
                         assign(f,Name);
                         Erase(f);
                         If FileExist(Name+'.DAT')
                         Then Begin
                                   assign(f,Name+'.DAT');
                                   erase(f);
                              End;
                         If FileExist(Name+'.RED')
                         Then Begin
                                   assign(f,Name+'.RED');
                                   erase(f);
                              End;
                    End
               Else Begin
                         repeat
                               write('Input File Name For OutPut : ');
                               readln(Name);
                               Until (Not FileExist(Name));
                    End;
          End;
end;
(**************************WRITE_DESCR_GFP PROC********************************)
Procedure Write_Descr_GFP(ss:char;var F:IntFldFile);
var
ans                  : boolean;
NewName              : names;
P                    : word;
Begin
     ClrScr;
     Window2(W_Size_Big);
     GotoXY(10,2);
     Writeln(' Name of File is :',Name);
     case ss of
 '1','2'    :begin
                  With FileRec(F) Do
```

```
                  Begin
                       Descr_GFP1(UserData).kind:=ss;
                       write(' Input Proportion of selfing [0.0..1.0] : ');
                       Descr_GFP1(UserData).self:=range(0.0,1.0);
                       write(' Input Migration proportion [0.0..1.0] : ');
                       Descr_GFP1(UserData).mig:=range(0.0,1.0);
                       repeat
                             write(' Input deme size [1,4,16,64,256,1024] : ');
                             P:=LintRange(1,1024);
                       until ((P=1) or (P=4) or
                              (P=16) or (P=64) or (P=256) or (P=1024));
                       Descr_GFP1(UserData).PopSize:=P;
                  End;
             end;
 '4'        :begin
                  With FileRec(F) Do
                  Begin
                       Descr_GFP4(UserData).kind:=ss;
                       write('Input Proportion of selfing [0.0..1.0] : ');
                       Descr_GFP4(UserData).self:=range(0.0,1.0);
                       write(' Input Migration proportion [0.0..1.0] : ');
                       Descr_GFP4(UserData).mig:=range(0.0,1.0);
                       repeat
                             write(' Input deme size [1,4,16,64,256,1024] : ');
                             P:=LintRange(1,1024);
                       until ((P=1) or (P=4) or
                              (P=16) or (P=64) or (P=256) or (P=1024));
                       Descr_GFP4(UserData).PopSize:=P;
                       write(' Do you want the pattern to be toroidal ? ');
                       Descr_GFP4(UserData).tor:=affirmed(False,graphics);
                  End;
             end;
 '3','5','6':begin
                  With FileRec(F) Do
                  Begin
                       Descr_GFP3(UserData).kind:=ss;
                       write('Input Proportion of selfing [0.0..1.0] : ');
                       Descr_GFP3(UserData).self:=range(0.0,1.0);
                       write(' Input Migration proportion [0.0..1.0] : ');
                       Descr_GFP3(UserData).mig:=range(0.0,1.0);
                       repeat
                             write(' Input deme size [1,4,16,64,256,1024] : ');
                             P:=LintRange(1,1024);
                       until ((P=1) or (P=4)
                             or (P=16) or (P=64)
                             or (P=256) or (P=1024));
                       Descr_GFP3(UserData).PopSize:=P;
                       write(' Do you want the pattern to be toroidal ? ');
                       Descr_GFP3(UserData).tor
                       :=affirmed(False,graphics);
                       If Descr_GFP3(userdata).mig=0.0
                       Then Descr_GFP3(userdata).dist:=0
                       Else Begin
                                 write(' How far do you allow dispersal [1..99] ? ');
                                 Descr_GFP3(UserData).dist:=lintrange(1,99);
                                 write(' Average distance of dispersal [1..99] ? ');
                                 Descr_GFP3(userdata).Aver:=LintRange(1,99);
                            end;
                  end;
             end;
'7'         :begin
                  With FileRec(F) Do
                  Begin
                       Descr_GFP7(UserData).kind:=ss;
                       write('Input Proportion of selfing [0.0..1.0] : ');
                       Descr_GFP7(UserData).self:=range(0.0,1.0);
                       repeat
                             write(' Input deme size [1,4,16,64,256,1024] : ');
                             P:=LintRange(1,1024);
                       until ((P=1) or (P=4) or
                              (P=16) or (P=64) or (P=256) or (P=1024));
                       Descr_GFP7(UserData).PopSize:=P;
                       write(' Do you want the pattern to be toroidal ? ');
                       Descr_GFP7(UserData).tor:=affirmed(False,graphics);
                       write(' Average distance of dispersal [0..99.0] ? ');
                       Descr_GFP7(userdata).Aver:=Range(0.0,99.0);
                  End;
             end;
{
'8','9'     :begin
                  With FileRec(F) Do Descr_GFP8(UserData).kind:=ss;
                  write('Input Proportion of selfing [0.0..1.0] : ');
                  with FileRec(F) Do Descr_GFP8(UserData).self:=range(0.0,1.0);
                  With FileRec(F) Do Descr_GFP8(UserData).dispm:=sdm;
                  With FileRec(F) Do Descr_GFP8(UserData).dispf:=sdf;
                  With FileRec(F) Do Descr_GFP8(UserData).tor:=neibtor;
                  With FileRec(F) Do Descr_GFP8(UserData).plant:=plant;
             end;
}
             end;
end; {Of Proc Write_Descr_GFP}
```

```
(**************************READ_DESCR_GFP PROC*******************************)
Procedure READ_DESCR_GFP(var ff:text;var F:IntFldFile);
var
x   :byte;
ss  :string[80];
Begin
     ClrScr;
     Window2(W_Size_Big);
  gotoxy(2,1); writeln('');
  gotoxy(2,2); writeln('                                                                     ');
  gotoxy(2,3); writeln('                                                                     ');
  gotoxy(2,4); writeln('                                                                     ');
  gotoxy(2,5); writeln('                                                                     ');
  gotoxy(2,6); writeln('                                                                     ');
  gotoxy(2,7); writeln('                                                                     ');
  gotoxy(2,8); writeln('');
     with filerec(F) do
     Begin
          gotoxy(15,2);
          write (ff,'NAME OF FILE: ');
          x:=0;
          repeat
                write(ff,Upcase(name[x]));
                x:=x+1;
          until name[x]=#0;
          writeln(ff);
          x:=ord(Descr_GFP1(UserData).kind)-48;
          ss:=GFP_Mes[x];
          ss:=concat('TYPE of GFP: ',ss);
          GotoXY(20,3);
          Writeln(ff,ss);
          gotoxy(5,4);
          writeln(ff,'SELFING= ',Descr_GFP1(UserData).Self:6:4)
     end;
     case chr(FileRec(F).userdata[1]) of
'1','2'     :begin
                  With FileRec(F) do
                  Begin
                       gotoXY(50,4);
                       Writeln(ff,'MIGRATION= ',Descr_GFP1(UserData).Mig:6:4);
                       gotoxy(5,5);
                       Writeln(ff,'DEME SIZE IS: ',Descr_GFP1(userdata).PopSize:6);
                  end;
             end;
'4'         :begin
                  With FileRec(F) do
                  Begin
                       gotoXY(50,4);
                       Writeln(ff,'MIGRATION= ',Descr_GFP4(UserData).Mig:6:4);
                       gotoxy(5,5);
                       Writeln(ff,'DEME SIZE IS: ',Descr_GFP4(userdata).PopSize:6);
                       gotoxy(5,6);
                       Writeln(ff,'TOROIDAL IS: ',Descr_GFP4(userdata).tor);
                  end;
             end;
'3','5','6' :begin
                  With FileRec(F) do
                  Begin
                       gotoXY(50,4);
                       Writeln(ff,'MIGRATION= ',Descr_GFP3(UserData).Mig:6:4);
                       gotoxy(5,5);
                       Writeln(ff,'DEME SIZE IS: ',Descr_GFP3(userdata).PopSize:6);
                       gotoxy(5,6);
                       Writeln(ff,'TOROIDAL IS: ',Descr_GFP3(userdata).tor);
                       gotoxy(50,6);
                       writeln(ff,'MAX DIST OF DISP= ',Descr_GFP3(userData).Dist:3);
                       gotoxy(50,5);
                       writeln(ff,'AVER DIST Of DISP= ',Descr_GFP3(userdata).Aver:3);
                  end;
             end;
'7'         :begin
                  With FileRec(F) do
                  Begin
                       gotoxy(5,5);
                       Writeln(ff,'DEME SIZE IS: ',Descr_GFP7(userdata).PopSize:6);
                       gotoxy(50,5);
                       writeln(ff,'AVER DIST Of DISP= ',Descr_GFP7(userdata).Aver:5:3);
                       gotoxy(5,6);
                       Writeln(ff,'TOROIDAL IS: ',Descr_GFP7(userdata).tor);
                  end;
             end;
'8','9'     :begin
             end;
     end;
End;
(******************INITTXT PROC************************************************)
Procedure InitTxt(Name:Names);
var
i:word;
Begin
     for i:=1 to Length(Name) do Name[i]:=Upcase(Name[i]);
```

```
     FilNamTxt:=Name+'.TXT';
     If FileExist(FilNamTxt)
     Then Begin
               Assign(FileTxt,FilNamTxt);
               reset(FileTxt);
               Append(FileTxt);
          End
     Else Begin
               Assign(FileTxt,FilNamTxt);
               rewrite(FileTxt);
          End;
     writeln(FileTxt,
     '*****************DESCRIPTION OF THE POPULATION************************');
     Read_Descr_GFP(FileTxt,FileDat);
     writeln(FileTxt,
     '**********************************************************************');
     close(FileTxt);
End; {Of InitTxt Proc}
(**********************UPDATE FILEREC PROCEDURE*******************************)

Procedure UpDate_FileRec(Var NewRec:DescrRec);
var
i          :word;
Pos        :word;
DummyRec   :DescrRec;
Found_1    :Boolean;
begin
     Found_1:=False;
     i:=0;
     Repeat
           i:=i+1;
           NewRec.FilNam[i]:=Upcase(NewRec.FilNam[i]);
     Until NewRec.FilNam[i]=#0;
     For i:=i to Length(NewRec.FilNam) do NewRec.FilNam[i]:=#0;
     Reset(FileDescrRec);
     If FileSize(FileDescrRec)<>0
     Then
     Repeat
          Pos:=FilePos(FileDescrRec);
          read(FileDescrRec,DummyRec);
          If DummyRec.FilNam=NewRec.FilNam
          Then Begin
                    Seek(FileDescrRec,Pos);
                    write(FileDescrRec,NewRec);
                    Found_1:=True;
               End;
     Until ((Found_1) or (Eof(FileDescrRec)));
     If Not Found_1
     Then Begin
               Seek(FileDescrRec,FileSize(FileDescrRec));
               Write(FileDescrRec,NewRec);
          End;
     Close(FileDescrRec);
End;{Of Update_FileRec Proc}
(******************GETNEWCOORD FUNCTION***************************************)

Function GetNewCoord(Tor:boolean;a,N:integer;Ofsa:ShortInt):integer;
var Res : integer;
Begin
     If Tor
     Then Begin
               Res:=(a+N*50-1+Ofsa) mod N;
          end
     Else Begin
               If ((a+Ofsa)<1)
               Then Res:=GRandom(a)
               Else If ((a+Ofsa)>N)
                    Then Res:=a+Grandom(N-a)-1
                    Else Res:=a+Ofsa-1;
          end;
     GetNewCoord:=Res;
End; {Of Function GetNewCoord}
(*****************************************************************************)

Begin (*Main body*)
      TextBackGround(Blue);TextColor(White);
      Window(1,1,80,25);ClrScr;
      If FileExist('MODEL42.INI')
      then begin
                Assign(File6,'MODEL42.INI');
                reset(File6);
                read(file6,s1,s2);
                Close(File6);
           end
      else Begin
                Assign(File6,'MODEL42.INI');
                rewrite(File6);
                Close(File6);
                write('Input first seed [1..2147483562] : ');
                S1:=lintrange(1,2147483562);
                write('Input second seed [1..2147483398] : ');
                S1:=lintrange(1,2147483398);
           end;
      If FileExist('MODEL42.REC')
      Then Begin
                Assign(FileDescrRec,'MODEL42.REC');
                Reset(FileDescrRec);
                Close(FileDescrRec);
           End
      Else Begin
                Assign(FileDescrRec,'MODEL42.REC');
                Rewrite(FileDescrRec);
                Close(FileDescrRec);
           End;
      AssignCrt(FScreen);Rewrite(FScreen);
End.{Of Unit StepInf}
```

A.2 MODEL42.PAS

```
Program Model42;
(*****************************************************************************)
(* THIS PROGRAM IS JUST THE MENU FOR ALL THE OTHER BITS. FROM IT CALLS ARE
   MADE TO :
            -PROGRAM BLDGFPAT         (REPLICATES OF EITHER:
                     -ISLAND MODEL           (TO BUILD) .
                     -STEPPING STONE MODEL -|_|-,--..--|_|--..-- ETC...
                     -NEIGHBOURHOOD MODEL    (EXPO OR NORMAL)
            -PROGRAM BUILDGEN
            -PROGRAM PLOTFREQ
            -PROGRAM SHFIELD
            -PROGRAM DISPERSAL        (CALCULATION OF THE AV. DISP. IN A NEIB)
            -PROGRAM REARR            (FOR USE OF SAMPLING WITH A NEIB MODEL)
            -PROGRAM SAMPLING         (FOR NON NEIB MODEL)
            -PROGRAM CALCFSST         (CALCULATION OF FSTAT FOR NON NEIB MODEL)
    PLENTY OF COMMENTS TO HAD, BUT I CAN'T BE BOTHER.
    OH YES, SIZE OF FIELD=MAXIND IS DEFINED IN UNIT STEPINF.
                                                                             *)
(*****************************************************************************)
USES
STEPINF,DOS,CRT,GRAPH,DRIVers,FONTs,GRAPH2D,STATS_1,STATS_2;

var
choix                     :char;

{$I BLDGFPAT.PAS}
{$I DISPLIN.PAS}
{$I BUILDGEN.PAS}
{$I PLOTFREQ.PAS}
{$I REDUC.PAS}
{$I SHFIELD.PAS}
(***************************************************************************)

(*****************************MENU******************************************)

FUNCTION menu : char;


begin
 TextBackGround(cyan);
 Window(1,1,80,25);ClrScr;
 TextBackGround(Black);
 Window(5,3,79,24);ClrScr;
 TextBackGround(Blue);textcolor(15);
 Window(3,2,77,23);ClrScr;
 gotoxy(27,1);writeln('');
 gotoxy(27,2);writeln('       M E N U       ');
 gotoxy(27,3);writeln('');
 textcolor(15);
  gotoxy(17,4); writeln('');
  gotoxy(17,5); writeln(' 0         List contents of files               ');
  gotoxy(17,6); writeln(' 1         Build Gene Flow Patterns             ');
  gotoxy(17,7); writeln(' 2         Estimate dispersal                   ');
  gotoxy(17,8); writeln(' 3         Build Generations                    ');
  gotoxy(17,9); writeln(' 4         Plot Distr. of All. Freq.            ');
 gotoxy(17,10); writeln(' 5         Reduction of allele number           ');
 gotoxy(17,11); writeln(' 6         Sampling                             ');
 gotoxy(17,12); writeln(' 7         Calculation of F_Stat                ');
 gotoxy(17,13); writeln(' 8         Picture Of the Field Of GenoTypes    ');
 gotoxy(17,14); writeln(' 9         End Session                          ');
 gotoxy(17,15); writeln('');
{gotoxy(17,19); writeln('');}

 textcolor(15);
 gotoxy(45,16); writeln('');
 gotoxy(45,17); write('');
 textcolor(128+15);write(' Your Choice');textcolor(15);
 write(' :    ');
 gotoxy(45,18); writeln('');

 textcolor(15);
```

```
 repeat
     gotoxy(64,17);
      choix:=readkey;
      if not (choix in ['0'..'9'])
      then begin
                gotoxy(64,17);
                textcolor(15);
                write(choix,' ');
                gotoxy(44,20);
                textcolor(blink+Yellow);
                writeln(' Incorrect Answer');
                Delay(800);
                gotoxy(64,17);
                textcolor(15);
                write('  ');
                gotoxy(44,20);clreol;
           end;
 until choix in ['0'..'9'];
 gotoxy(64,17);
 write(choix);
 delay(100);
 menu:=choix;
 TextBackGround(Blue);
 textcolor(15)
end;

(************MAIN BODY********************************************************)

begin
      grdriver:=EGA; grmode:=EGAHi; name:='';
                     (******Check and initialise graphics******)
      If RegisterBGIDriver(@EGAVGADriverProc)<0 then Halt(1);
      If RegisterBGIFont(@TriplexFontProc)<0 then Halt(1);
      If RegisterBGIFont(@SmallFontProc)<0 then Halt(1);
      InitGraph(grdriver,grmode,'');
      ErrCode:=GraphResult;
      if ErrCode<>grOk
      Then begin
               writeln('Graphics Init. Error',ErrCode);
               Writeln('GrMode= ',GrMode,'  GrDriver= ',GrDriver);
               Halt(1);
           end
      Else begin
               SetGraphmode(grmode);
               Graphics:=true;
           end;
      SetBkColor(Blue);
      SetTextStyle(TriplexFont,HorizDir,4);
      SetTextJustify(CenterText,CenterText);
      SetColor(Yellow);
      OutTextXY(GetMaxX div 2,GetMaxY-4*(GetMaxY div 5),
                        'MODEL 42 or ');
      OutTextXY(GetMaxX div 2,GetMaxY-3*(GetMaxY div 5),
                'GENE FLOW PATTERNS AND F-STATS');
      OutTextXY(GetMaxX div 2,GetMaxY-2*(GetMaxY div 5),
                           'by');
      OutTextXY(GetMaxX div 2,GetMaxY-(GetMaxY div 5),
                        'JEROME GOUDET');
      SetTextStyle(DefaultFont,HorizDir,1);
      SetTextJustify(LeftText,LeftText);
      OutTextXY(5,GetMaxY-5,
                            'PRESS ANY KEY TO PROCEED ');
      repeat until ReadKey>#1;
      Graphics:=False;
      RestoreCrtMode;
     repeat
     ClrScr;
     choix:=menu;
     case choix of
       '0' : begin
                  gotoXY(20,22);
                  write('Not available yet. Press a key');
                  repeat until readkey>#1;
             end;
       '1' : begin
                  BLDGFPAT;
             End;
       '2' : Begin
                  DISPERSAL;
             end;
       '3' : Begin
                  BUILDGEN;
             End;
       '4'  :Begin
                  PLOTFREQ;
             End;
       '5' : Begin
                  REDUC;
             End;
       '6' : Begin
                  gotoXY(20,22);
                  write('Not available yet. Press a key');
                  repeat until readkey>#1;
             end;
       '7' : Begin

                  gotoXY(20,22);
                  write('Not available yet. Press a key');
                  repeat until readkey>#1;
             End;
       '8' : Begin
                  SHFIELD;
              end;
     end;

 until (choix='9');
 SetGraphMode(GrMode);
 SetBkColor(Blue);
 SetTextStyle(TriplexFont,HorizDir,4);
 SetTextJustify(CenterText,CenterText);
 SetColor(Yellow);
 OutTextXY(GetMaxX Div 2,GetMaxY div 3,'BYE FOR NOW');
 OutTextXY(GetMaxX div 2,2*(GetMaxY div 3),'TARA Y RWAN');
 Delay(800);
 rewrite(File6);
 write(File6,s1,'  ',s2);
                         {Grandom(2147483562)+1 is one of s1}
                         {GRandom(2147483398)+1 is one of s2}
 close(File6);
 CloseGraph;
 TextBackGround(0);TextColor(7);
 Window(1,1,80,25);
 clrscr;
 Close(FScreen);
 end.
```

A.3 BLDGFPAT.PAS

```
Procedure bldgfpat;
var
choix        :char;

{$I GFISINCO.PAS}
{$I GFISCLOU.PAS}
{$I GFSS1DEX.PAS}
{$I GFSS2DC4.PAS}
{$I GFSS8EXP.PAS}
{$I GFSSNEIB.PAS}
{$I INTRECS.PAS}
Function MenuRep:char;


Begin
 TextBackGround(cyan);
 Window(1,1,80,25);ClrScr;
 TextBackGround(Black);
 Window(5,3,79,24);ClrScr;
 TextBackGround(Blue);textcolor(15);
 Window(3,2,77,23);ClrScr;
 gotoxy(27,1);writeln('');
 gotoxy(27,2);writeln('      M E N U       ');
 gotoxy(27,3);writeln('');
 textcolor(15);
  gotoxy(17,5); writeln('');
  gotoxy(17,6); writeln(' 1         Island model with inf. continent     ');
  gotoxy(17,7); writeln(' 2         Island model with gametic cloud     ');
  gotoxy(17,8); writeln(' 3         Step. sto. model 1 dim. exp. distr. ');
  gotoxy(17,9); writeln(' 4         Step. sto. 2 dim. con. 4            ');
 gotoxy(17,10); writeln(' 5         Step. sto. 2 dim. con. 8 exp. distr.');
 gotoxy(17,11); writeln(' 6         Step. sto. 2 dim. con. 8 nor. distr.');
 gotoxy(17,12); writeln(' 7         Stepping stoned Neighbourhood       ');
 gotoxy(17,13); writeln(' 8         Plant Lattice model                 ');
 gotoxy(17,14); writeln(' 9         True Wright Lattice model           ');
 gotoxy(17,15); writeln('');

{gotoxy(17,19); writeln('');}

 textcolor(15);
 gotoxy(45,16); writeln('');
 gotoxy(45,17); write('');
 textcolor(128+15);write(' Your Choice');textcolor(15);
 write(' :    ');
 gotoxy(45,18); writeln('');

 textcolor(15);
 repeat
     gotoxy(64,17);
      choix:=readkey;
      if not (choix in ['1'..'9'])
      then begin
```

```
                gotoxy(64,17);
                textcolor(15);
                write(choix,' ');
                gotoxy(44,20);
                textcolor(blink+Yellow);
                writeln(' Incorrect Answer');
                Delay(800);
                gotoxy(64,17);
                textcolor(15);
                write('  ');
                gotoxy(44,20);clreol;
           end;
 until choix in ['1'..'9'];
 gotoxy(64,17);
 write(choix);
 delay(100);
 menurep:=choix;
 TextBackGround(Blue);
 textcolor(15)

End; {Of Func MenuRep}


Begin
     clrscr;
     choix:=menurep;
     case choix of
     '1' : GFISINCO;
     '2' : GFISCLOU;
     '3' : GFSS1DEX;
     '4' : GFSS2DC4;
     '5' : GFSS8EXP;
     '6' : Begin
                GoToXY(30,22);
                write('Not Available yet.');
                write('Press a key.');
                repeat until readkey>#1;
                exit;
           End;
     '7' : GFSSNEIB;
     '8' : Begin
                GoToXY(30,22);
                write('Not Available yet.');
                write('Press a key.');
                repeat until readkey>#1;
                exit;
           end;
     '9' : INTRECS
     End;
End; {Of BLDGFPAT Proc}
```

A.4 GFISINCO.PAS

```
Procedure GFISINCO;

var
  Champ                                      : IntFldPtr;
  a,i,x                                      : Integer;
  Mig,Self                                   : ExistsPtr;
  t,Reps                                     : byte;
  PropMig,SelfProp                           : single;
  NumbSP,PopSize                             : Integer;


(********************MIG PROC*************************************************)
Procedure Migr      (PopSize,NumbSp            : integer;
                     Mig,Sel                   : ExistsPtr;
                     var ParChamp2             : IntFldPtr);
var
Champ2                              :IntFldPtr;
Temp1,Temp2                         :Integer;
i,j,k,l,x                           :integer;
where1,where2                       :integer;


begin
     GetMem(Champ2,SizeOf(Champ2^));
     x:=0;
     for i:=1 to NumbSp Do
     For k:=1 to PopSize Do
     Begin
          x:=x+1;
          Temp1:=Grandom(PopSize)+1;
          Temp2:=Grandom(PopSize)+1;
          If Not Sel^[x]
          Then begin
                    If Mig^[x]
                    Then Where1:=Grandom(MaxInd)+1
                    else Where1:=(i-1)*PopSize+Temp1;
                    If Mig^[x+MaxInd]
                    Then Where2:=Grandom(MaxInd)+1
                    else Where2:=(i-1)*PopSize+Temp2;
               End
          Else Begin
                    If Mig^[x]
                    Then Where1:=Grandom(MaxInd)+1
                    Else Where1:=(i-1)*PopSize+Temp1;
                    Where2:=Where1;
               End;
          Champ2^[x,1]:=ParChamp2^[Where1,Grandom(2)+1];
          Champ2^[x,2]:=ParChamp2^[Where2,Grandom(2)+1];
     end;
     FreeMem(ParChamp2,SizeOf(ParChamp2^));
     ParChamp2:=Champ2;
end;{Of Proc Migr}
(************************************MAIN PROGRAM****************************)

Begin
     clrscr;
     Window2(W_Size_Big);
     Erase_File(Name);
     FilNamDat:=Name;
     Assign(FileDat,FilNamDat);
     Write_Descr_GFP('1',FileDat);
     With FileRec(FileDat) Do
     Begin
          SelfProp:=Descr_GFP1(userdata).self;
          PropMig:=Descr_GFP1(userdata).mig;
          PopSize:=Descr_GFP1(userdata).Popsize;
          With ThisDescrRec Do
          Begin
               FilNam:=Name;
               For i:=1 to 16 do
               NewData[i]:=userdata[i];
          End;
     End;
     Update_Filerec(ThisDescrRec);
     t:=0;Reps:=10;
     InitTxt(Name);
     Read_Descr_GFP(FScreen,FileDat);
     GotoXy(15,10);
     Write('How many replicates do you want [10..50] : ');
     Reps:=LintRange(10,50);
     Rewrite(FileDat);close(FileDat);
     NumbSp:=MaxInd div PopSize;
     GoToXY(15,12);
     write('Replicate no:      completed');
     While t<Reps Do
     Begin
         t:=t+1;
         GetMem(Champ,SizeOf(Champ^));
         Ini(Champ);
         GetMem(Mig,SizeOf(Mig^));GetMem(Self,SizeOf(Self^));
         FillBoolArray(PropMig,Mig);
         FillBoolArray(SelfProp,Self);
         Migr      (PopSize,NumbSp,
                    Mig,Self,
                    Champ);
         reset(FileDat);seek(FileDat,FileSize(FileDat));
         Write(FileDat,Champ^);
         Close(FileDat);
         FreeMem(Champ,SizeOf(Champ^));
         FreeMem(Mig,SizeOf(Mig^));
         FreeMem(Self,SizeOf(Self^));
         GoToXY(29,12);Write(t:4);
     End; {Of while t<Reps}
End;{Of Proc GFISINCO}
```

A.5 GFISCLOU.PAS

```
 (************BUILDREP PRO*******************************************************)
 Procedure GFISCLOU;

 var
   Champ                                      : IntFldPtr;
   a,i,x                                      : Integer;
   Mig,Self                                   : ExistsPtr;
   t,Reps                                     : byte;
   PropMig,SelfProp                           : Double;
   NumbSP,PopSize                             : Integer;

 (*******************MIG PROC*******************************************************)
 Procedure Migr      (PopSize,NumbSp           : integer;
      Mig,Sel                : ExistsPtr;
      var ParChamp2          : IntFldPtr);
 var
```

```
 Champ2                                :IntFldPtr;
 Temp1,Temp2                           :Integer;
 i,j,k,l,x                             :integer;
 where1,where2                         :integer;


 begin
      GetMem(Champ2,SizeOf(Champ2^));
      x:=0;
      for i:=1 to NumbSp Do
      For k:=1 to PopSize Do
      Begin
   x:=x+1;
   Temp1:=Grandom(PopSize)+1;
   Temp2:=Grandom(PopSize)+1;
   If Not Sel^[x]
   Then begin
     If Mig^[x]
     Then Begin
       Repeat
     Where1:=Grandom(MaxInd)+1
       Until ((Where1<=(i-1)*PopSize)
   or (Where1>i*Popsize));
  End
     Else Where1:=(i-1)*PopSize+Temp1;
     If Mig^[x+MaxInd]
     Then Begin
       Repeat
     Where2:=Grandom(MaxInd)+1
       Until ((Where2<=(i-1)*PopSize)
   or (Where2>i*Popsize));
  End
     Else Where2:=(i-1)*PopSize+Temp2;
End
   Else Begin
     If Mig^[x]
     Then Begin
       Repeat
     Where1:=Grandom(MaxInd)+1
       Until ((Where1<=(i-1)*PopSize)
   or (Where1>i*Popsize));
  End
     Else Where1:=(i-1)*PopSize+Temp1;
     Where2:=Where1;
End;
   Champ2^[x,1]:=ParChamp2^[Where1,Grandom(2)+1];
   Champ2^[x,2]:=ParChamp2^[Where2,Grandom(2)+1];
      end;
      FreeMem(ParChamp2,SizeOf(ParChamp2^));
      ParChamp2:=Champ2;
 end;{Of Proc Migr}
 (************************************MAIN PROGRAM****************************)

 Begin
      clrscr;
      Graphics:=False;
      Window2(W_Size_Big);
      Erase_File(Name);
      FilNamDat:=Name;
      Assign(FileDat,FilNamDat);
      Write_Descr_GFP('2',FileDat);
      With FileRec(FileDat) Do
      Begin
   SelfProp:=Descr_GFP1(userdata).self;
   PropMig:=Descr_GFP1(userdata).mig;
   PopSize:=Descr_GFP1(userdata).Popsize;
   With ThisDescrRec Do
   Begin
FilNam:=Name;
For i:=1 to 16 do
NewData[i]:=userdata[i];
   End;
      End;
      Update_Filerec(ThisDescrRec);
      t:=0;Reps:=10;
      InitTxt(Name);
      Read_Descr_GFP(FScreen,FileDat);
      GotoXy(15,10);
      Write('How many replicates do you want [10..50] : ');
      Reps:=LintRange(10,50);
      Rewrite(FileDat);close(FileDat);
      NumbSp:=MaxInd div PopSize;
      GoToXY(15,12);
      write('Replicate no:      completed');
      While t<Reps Do
      Begin
  t:=t+1;
  GetMem(Champ,SizeOf(Champ^));
  Ini(Champ);
  GetMem(Mig,SizeOf(Mig^));GetMem(Self,SizeOf(Self^));
  FillBoolArray(PropMig,Mig);
  FillBoolArray(SelfProp,Self);
  Migr      (PopSize,NumbSp,
     Mig,Self,
     Champ);
  reset(FileDat);seek(FileDat,FileSize(FileDat));
  Write(FileDat,Champ^);
  Close(FileDat);
  FreeMem(Champ,SizeOf(Champ^));
  FreeMem(Mig,SizeOf(Mig^));
  FreeMem(Self,SizeOf(Self^));
  GoToXY(29,12);Write(t:4);
      End; {Of while t<Reps}

 End;{Of Proc GFISCLOU}
```

A.6 GFSS2DC4.PAS

```
Procedure GFSS2DC4;

var
  Champ                                     : IntFldPtr;
  a,i,x                                     : Integer;
  Mig,Self                                  : ExistsPtr;
  t,Reps                                    : byte;
  PropMig,SelfProp                          : Double;
  NumbSP,PopSize                            : Integer;
  NumbRow,NumbCol                           : Integer;
  Tor                                       : boolean;

(**************GET THE HOR AND VERT OFFSET OF MIGR*************************)

Procedure GetOfset(Var OfsVer,OfsHor:Shortint);
var
temp, dist:byte;

begin
     OfsHor:=0;
     OfsVer:=0;
     temp:=GRandom(4);
     case temp of
          0 : OfsHor:=-1;
          1 : OfsHor:=1;
          2 : OfsVer:=-1;
          3 : OfsVer:=1;
     end;
end; {Of Proc GetOfs}

(******************MIG PROC***********************************************)
Procedure Migr      (Tor                     :boolean;
                     NumbCol,NumbRow,PopSize : integer;
                     Mig,Sel                 : ExistsPtr;
                     var ParChamp2           : IntFldPtr);
var
Champ2                              :IntFldPtr;
Temp1,Temp2                         :Integer;
i,j,k,l,All1,All2,x                 :integer;
Hor1,Hor2,Vert1,Vert2               :integer;
where1,where2                       :integer;
OfsHor,OfsVer                       :ShortInt;


begin
     GetMem(Champ2,SizeOf(Champ2^));
     x:=0;
            for i:=1 to NumbRow Do
            For j:=1 to NumbCol Do
            For k:=1 to PopSize Do
             Begin
                x:=x+1;
                Temp1:=Grandom(PopSize)+1;
                Temp2:=Grandom(PopSize)+1;
                hor1:=j-1;
                vert1:=i-1;
                hor2:=j-1;
                vert2:=i-1;
                If Not Sel^[x]
                Then begin
                      If Mig^[x]
                      Then Begin
                             GetOfset(OfsVer,OfsHor);
                             hor1:=GetNewCoord(Tor,j,NumbCol,OfsHor);
                             vert1:=GetNewCoord(Tor,i,NumbRow,OfsVer);
                           End;
                      Where1:=PopSize*Hor1+PopSize*NumbCol*Vert1+Temp1;
                      If Mig^[x+MaxInd]
                      Then Begin
                             GetOfset(OfsVer,OfsHor);
```

```
                             hor2:=GetNewCoord(Tor,j,NumbCol,OfsHor);
                             vert2:=GetNewCoord(Tor,i,NumbRow,OfsVer);
                           End;
                      Where2:=PopSize*Hor2+PopSize*NumbCol*Vert2+Temp2;
                     End
                Else Begin
                      If Mig^[x]
                      Then Begin
                             GetOfset(OfsVer,OfsHor);
                             hor1:=GetNewCoord(Tor,j,NumbCol,OfsHor);
                             vert1:=GetNewCoord(Tor,i,NumbRow,OfsVer);
                           End;
                      Where1:=PopSize*Hor1+PopSize*NumbCol*Vert1+Temp1;
                      Where2:=Where1;
                     End;
                Champ2^[x,1]:=ParChamp2^[Where1,Grandom(2)+1];
                Champ2^[x,2]:=ParChamp2^[Where2,Grandom(2)+1];
             end;
     FreeMem(ParChamp2,SizeOf(ParChamp2^));
     ParChamp2:=Champ2;
end;{Of Proc Migr}
(*************************************MAIN PROGRAM*************************)

Begin
     clrscr;
     Graphics:=False;
     Window2(W_Size_Big);
     Erase_File(Name);
     FilNamDat:=Name;
     Assign(FileDat,FilNamDat);
     Write_Descr_GFP('4',FileDat);
     With FileRec(FileDat) Do
     Begin
          SelfProp:=Descr_GFP4(userdata).self;
          PropMig:=Descr_GFP4(userdata).mig;
          PopSize:=Descr_GFP4(userdata).Popsize;
          Tor:=Descr_GFP4(userdata).tor;
          With ThisDescrRec Do
          Begin
               FilNam:=Name;
               For i:=1 to 16 do
               NewData[i]:=userdata[i];
          End;
     End;
     Update_Filerec(ThisDescrRec);
     t:=0;Reps:=10;
     InitTxt(Name);
     Read_Descr_GFP(FScreen,FileDat);
     GotoXy(15,10);
     Write('How many replicates do you want [10..50] : ');
     Reps:=LintRange(10,50);
     Rewrite(FileDat);close(FileDat);
     NumbSp:=MaxInd div PopSize;
     Case NumbSp of
              4096 :begin numbrow:=64;numbcol:=64;end;
              1024 :begin numbrow:=32;numbcol:=32;end;
              256  :begin NumbRow:=16;NumbCol:=16;end;
              64   :begin NumbRow:=8;NumbCol:=8;end;
              16   :begin NumbRow:=4;NumbCol:=4;end;
              4    :begin NumbRow:=2;NumbCol:=2;end;
              1    :begin NumbRow:=1;NumbCol:=1;end;
     end;
     GoToXY(15,12);
     write('Replicate no:      completed');
     While t<Reps Do
     Begin
         t:=t+1;
         GetMem(Champ,SizeOf(Champ^));
         Ini(Champ);
         GetMem(Mig,SizeOf(Mig^));GetMem(Self,SizeOf(Self^));
         FillBoolArray(PropMig,Mig);
         FillBoolArray(SelfProp,Self);
         Migr      (Tor,NumbCol,NumbRow,PopSize,
                    Mig,Self,
                    Champ);
         reset(FileDat);seek(FileDat,FileSize(FileDat));
         Write(FileDat,Champ^);
         Close(FileDat);
         FreeMem(Champ,SizeOf(Champ^));
         FreeMem(Mig,SizeOf(Mig^));
         FreeMem(Self,SizeOf(Self^));
         GoToXY(29,12);Write(t:4);
     End; {Of while t<Reps}

End;{Of Proc BuildField}
```

A.7 GFSS1DEX.PAS

```
Procedure GFSS1DEX;

var
  Champ                                      : IntFldPtr;
  a,i,x                                      : Integer;
  Mig,Self                                   : ExistsPtr;
  t,Reps                                     : byte;
  PropMig,SelfProp                           : Double;
  NumbSP,PopSize                             : Integer;
  Distance,Average                           : Byte;
  Tor                                        : boolean;

(**RETURN AN INTEGER FROM A NEGATIVE EXPONENTIAL******************************)

Function IntExpo(Dist,Av:byte) :Byte;

Var Temp  :Single;

begin
     repeat
           Temp:=Uniform;
     Until Temp>Exp(-1.0*Dist);
     Temp:=-1.0*av*Ln(Temp);
     IntExpo:=Trunc(Temp)+1;
End; {Of Function IntExpo}

(***************GET THE HOR AND VERT OFFSET OF MIGR**************************)

Function GetOfset :shortint;
var
temp, dist:byte;
offs       :shortint;
begin
     Offs:=-99;
     Dist:=IntExpo(Distance,average);
     temp:=GRandom(2);
     case temp of
          0 : Offs:=-dist;
          1 : Offs:=dist;
     end;
GetOfSet:=Offs
end; {Of Proc GetOfs}

(********************MIG PROC*************************************************)
Procedure Migr      (Tor                     : boolean;
                     PopSize,NumbSp          : integer;
                     Mig,Sel                 : ExistsPtr;
                     var ParChamp2           : IntFldPtr);
var
Champ2                              :IntFldPtr;
Temp1,Temp2                         :Integer;
i,j,k,l,x                           :integer;
Hor1,Hor2                           :integer;
where1,where2                       :integer;
OfsHor                              :ShortInt;


begin
     GetMem(Champ2,SizeOf(Champ2^));
     x:=0;
            for i:=1 to NumbSp Do
            For k:=1 to PopSize Do
             Begin
                x:=x+1;
                Temp1:=Grandom(PopSize)+1;
                Temp2:=Grandom(PopSize)+1;
                hor1:=i-1;
                hor2:=i-1;
                If Not Sel^[x]
                Then begin
                      If Mig^[x]
                      Then Begin
                             OfsHor:=GetOfset;
                             hor1:=GetNewCoord(Tor,i,NumbSp,OfsHor);
                           End;
                      Where1:=PopSize*Hor1+Temp1;
                      If Mig^[x+MaxInd]
                      Then Begin
                             OfsHor:=GetOfset;
                             hor2:=GetNewCoord(Tor,i,NumbSp,OfsHor);
                           end;
                      Where2:=PopSize*Hor2+Temp2;
                     End
                Else Begin
                      If Mig^[x]
                      Then Begin
                             OfsHor:=GetOfset;
                             hor1:=GetNewCoord(Tor,i,NumbSp,OfsHor);
                           End;
                      Where1:=PopSize*Hor1+Temp1;
                      Where2:=Where1;
```

```
                     End;
                Champ2^[x,1]:=ParChamp2^[Where1,Grandom(2)+1];
                Champ2^[x,2]:=ParChamp2^[Where2,Grandom(2)+1];
             end;
     FreeMem(ParChamp2,SizeOf(ParChamp2^));
     ParChamp2:=Champ2;
end;{Of Proc Migr}
(*************************************MAIN PROGRAM***************************)

Begin
     clrscr;
     Graphics:=False;
     Window2(W_Size_Big);
     Erase_File(Name);
     FilNamDat:=Name;
     Assign(FileDat,FilNamDat);
     Write_Descr_GFP('3',FileDat);
     With FileRec(FileDat) Do
     Begin
          SelfProp:=Descr_GFP3(userdata).self;
          PropMig:=Descr_GFP3(userdata).mig;
          PopSize:=Descr_GFP3(userdata).Popsize;
          Tor:=Descr_GFP3(userdata).tor;
          distance:=Descr_GFP3(userdata).dist;
          Average:=Descr_GFP3(userdata).Aver;
          With ThisDescrRec Do
          Begin
               FilNam:=Name;
               For i:=1 to 16 do
               NewData[i]:=userdata[i];
          End;
     End;
     Update_Filerec(ThisDescrRec);
     t:=0;Reps:=10;
     InitTxt(Name);
     Read_Descr_GFP(FScreen,FileDat);
     GotoXy(15,10);
     Write('How many replicates do you want [10..50] : ');
     Reps:=LintRange(10,50);
     Rewrite(FileDat);close(FileDat);
     NumbSp:=MaxInd div PopSize;
     GoToXY(15,12);
     write('Replicate no:      completed');
     While t<Reps Do
     Begin
         t:=t+1;
         GetMem(Champ,SizeOf(Champ^));
         Ini(Champ);
         GetMem(Mig,SizeOf(Mig^));GetMem(Self,SizeOf(Self^));
         FillBoolArray(PropMig,Mig);
         FillBoolArray(SelfProp,Self);
         Migr      (tor,PopSize,NumbSp,
                    Mig,Self,
                    Champ);
         reset(FileDat);seek(FileDat,FileSize(FileDat));
         Write(FileDat,Champ^);
         Close(FileDat);
         FreeMem(Champ,SizeOf(Champ^));
         FreeMem(Mig,SizeOf(Mig^));
         FreeMem(Self,SizeOf(Self^));
         GoToXY(29,12);Write(t:4);
     End; {Of while t<Reps}
End;{Of Proc GFSS1DEX}
```

A.8 GFSS8NOR.PAS

```
(************GENE FLOW PATTERN : STEPPING STONE 8 CON EXP DISP.**************)
Procedure GFSS8NOR;

var
  Champ                                      : IntFldPtr;
  a,i,x                                      : Integer;
  Mig,Self                                   : ExistsPtr;
  t,Reps                                     : byte;
  PropMig,SelfProp                           : Double;
  NumbSP,PopSize                             : Integer;
  NumbRow,NumbCol                            : Integer;
  Distance,Average                           : Byte;
  Tor                                        : boolean;

(**RETURN AN INTEGER FROM A NEGATIVE EXPONENTIAL******************************)

function Norm(var b:byte) : single;
var
 x,y,s,l,d1 : single;
begin
if b=0 then norm:=0 else
begin
s:=0.0;
```

```
repeat
 x:=2.0*Uniform-1.0; y:=2.0*Uniform-1.0;
 s:=sqr(x)+sqr(y);
 until s<1.0;
l:=sqrt(-2.0*ln(s)/s);
if Uniform<0.5 then d1:=x*l  else d1:=y*l;
norm:=d1*b;
end; (*if b*)
end (* norm *) ;

(***************GET THE HOR AND VERT OFFSET OF MIGR**************************)

Procedure GetOfset(Var OfsVer,OfsHor:Shortint);
var
temp, dist:byte;

begin
     OfsHor:=-99;
     OfsVer:=-99;
     Dist:=trunc(norm(Distance));
     temp:=GRandom(4);
     case temp of
          0 : OfsHor:=-dist;
          1 : OfsHor:=dist;
          2 : OfsVer:=-dist;
          3 : OfsVer:=dist;
     end;
     If Temp<=1
     Then OfsVer:=Grandom(2*Dist+1)-Dist
     Else OfsHor:=GRandom(2*Dist+1)-Dist;
end; {Of Proc GetOfs}

(********************MIG PROC*************************************************)
Procedure Migr      (Tor                     :boolean;
                     NumbCol,NumbRow,PopSize  : integer;
                     Mig,Sel                  : ExistsPtr;
                     var ParChamp2            : IntFldPtr);
var
Champ2                              :IntFldPtr;
Temp1,Temp2                         :Integer;
i,j,k,l,All1,All2,x                 :integer;
Hor1,Hor2,Vert1,Vert2               :integer;
where1,where2                       :integer;
OfsHor,OfsVer                       :ShortInt;


begin
     GetMem(Champ2,SizeOf(Champ2^));
     x:=0;
            for i:=1 to NumbRow Do
            For j:=1 to NumbCol Do
            For k:=1 to PopSize Do
             Begin
                x:=x+1;
                Temp1:=Grandom(PopSize)+1;
                Temp2:=Grandom(PopSize)+1;
                hor1:=j-1;
                vert1:=i-1;
                hor2:=j-1;
                vert2:=i-1;
                If Not Sel^[x]
                Then begin
                      If Mig^[x]
                      Then Begin
                             GetOfset(OfsVer,OfsHor);
                             hor1:=GetNewCoord(Tor,j,NumbCol,OfsHor);
                             vert1:=GetNewCoord(Tor,i,NumbRow,OfsVer);
                           End;
                      Where1:=PopSize*Hor1+PopSize*NumbCol*Vert1+Temp1;
                      If Mig^[x+MaxInd]
                      Then Begin
                             GetOfset(OfsVer,OfsHor);
                             hor2:=GetNewCoord(Tor,j,NumbCol,OfsHor);
                             vert2:=GetNewCoord(Tor,i,NumbRow,OfsVer);
                           End;
                      Where2:=PopSize*Hor2+PopSize*NumbCol*Vert2+Temp2;
                     End
                Else Begin
                      If Mig^[x]
                      Then Begin
                             GetOfset(OfsVer,OfsHor);
                             hor1:=GetNewCoord(Tor,j,NumbCol,OfsHor);
                             vert1:=GetNewCoord(Tor,i,NumbRow,OfsVer);
                           End;
                      Where1:=PopSize*Hor1+PopSize*NumbCol*Vert1+Temp1;
                      Where2:=Where1;
                     End;
                Champ2^[x,1]:=ParChamp2^[Where1,Grandom(2)+1];
                Champ2^[x,2]:=ParChamp2^[Where2,Grandom(2)+1];
             end;
     FreeMem(ParChamp2,SizeOf(ParChamp2^));
```

```
     ParChamp2:=Champ2;
end;{Of Proc Migr}
(*************************************MAIN PROGRAM***************************)

Begin
     clrscr;
     Graphics:=False;
     Window2(W_Size_Big);
     Erase_File(Name);
     FilNamDat:=Name;
     Assign(FileDat,FilNamDat);
     Write_Descr_GFP('6',FileDat);
     With FileRec(FileDat) Do
     Begin
          SelfProp:=Descr_GFP3(userdata).self;
          PropMig:=Descr_GFP3(userdata).mig;
          PopSize:=Descr_GFP3(userdata).Popsize;
          Tor:=Descr_GFP3(userdata).tor;
          Distance:=Descr_GFP3(userdata).Dist;
          Average:=Descr_GFP3(Userdata).Aver;
          With ThisDescrRec Do
          Begin
               FilNam:=Name;
               For i:=1 to 16 do
               NewData[i]:=userdata[i];
          End;
     End;
     Update_Filerec(ThisDescrRec);
     t:=0;Reps:=10;
     InitTxt(Name);
     Read_Descr_GFP(FScreen,FileDat);
     GotoXy(15,10);
     Write('How many replicates do you want [10..50] : ');
     Reps:=LintRange(10,50);
     Rewrite(FileDat);close(FileDat);
     NumbSp:=MaxInd div PopSize;
     Case NumbSp of
              4096 :begin numbrow:=64;numbcol:=64;end;
              1024 :begin numbrow:=32;numbcol:=32;end;
              256  :begin NumbRow:=16;NumbCol:=16;end;
              64   :begin NumbRow:=8;NumbCol:=8;end;
              16   :begin NumbRow:=4;NumbCol:=4;end;
              4    :begin NumbRow:=2;NumbCol:=2;end;
              1    :begin NumbRow:=1;NumbCol:=1;end;
     end;
     GoToXY(15,12);
     write('Replicate no:      completed');
     While t<Reps Do
     Begin
         t:=t+1;
         GetMem(Champ,SizeOf(Champ^));
         Ini(Champ);
         GetMem(Mig,SizeOf(Mig^));GetMem(Self,SizeOf(Self^));
         FillBoolArray(PropMig,Mig);
         FillBoolArray(SelfProp,Self);
         Migr      (Tor,NumbCol,NumbRow,PopSize,
                    Mig,Self,
                    Champ);
         reset(FileDat);seek(FileDat,FileSize(FileDat));
         Write(FileDat,Champ^);
         Close(FileDat);
         FreeMem(Champ,SizeOf(Champ^));
         FreeMem(Mig,SizeOf(Mig^));
         FreeMem(Self,SizeOf(Self^));
         GoToXY(29,12);Write(t:4);
     End; {Of while t<Reps}
End;{Of Proc GFSS8NOR}
```

A.9 GFSS8EXP.PAS

```
(************GENE FLOW PATTERN : STEPPING STONE 8 CON EXP DISP.**************)
Procedure GFSS8NOR;

var
  Champ                                      : IntFldPtr;
  a,i,x                                      : Integer;
  Mig,Self                                   : ExistsPtr;
  t,Reps                                     : byte;
  PropMig,SelfProp                           : Double;
  NumbSP,PopSize                             : Integer;
  NumbRow,NumbCol                            : Integer;
  Distance,Average                           : Byte;
  Tor                                        : boolean;

(**RETURN AN INTEGER FROM A NEGATIVE EXPONENTIAL*****************************)

function Norm(var b:byte) : single;
var
 x,y,s,l,d1 : single;
```

```
begin
if b=0 then norm:=0 else
begin
s:=0.0;
repeat
 x:=2.0*Uniform-1.0; y:=2.0*Uniform-1.0;
 s:=sqr(x)+sqr(y);
 until s<1.0;
l:=sqrt(-2.0*ln(s)/s);
if Uniform<0.5 then d1:=x*l  else d1:=y*l;
norm:=d1*b;
end; (*if b*)
end (* norm *) ;

(**************GET THE HOR AND VERT OFFSET OF MIGR*************************)

Procedure GetOfset(Var OfsVer,OfsHor:Shortint);
var
temp, dist:byte;

begin
     OfsHor:=-99;
     OfsVer:=-99;
     Dist:=trunc(norm(Distance));
     temp:=GRandom(4);
     case temp of
          0 : OfsHor:=-dist;
          1 : OfsHor:=dist;
          2 : OfsVer:=-dist;
          3 : OfsVer:=dist;
     end;
     If Temp<=1
     Then OfsVer:=Grandom(2*Dist+1)-Dist
     Else OfsHor:=GRandom(2*Dist+1)-Dist;
end; {Of Proc GetOfs}

(*******************MIG PROC***********************************************)
Procedure Migr      (Tor                        :boolean;
                     NumbCol,NumbRow,PopSize  : integer;
                     Mig,Sel                  : ExistsPtr;
                     var ParChamp2            : IntFldPtr);
var
Champ2                              :IntFldPtr;
Temp1,Temp2                         :Integer;
i,j,k,l,All1,All2,x                 :integer;
Hor1,Hor2,Vert1,Vert2               :integer;
where1,where2                       :integer;
OfsHor,OfsVer                       :ShortInt;


begin
     GetMem(Champ2,SizeOf(Champ2^));
     x:=0;
            for i:=1 to NumbRow Do
            For j:=1 to NumbCol Do
            For k:=1 to PopSize Do
             Begin
                x:=x+1;
                Temp1:=Grandom(PopSize)+1;
                Temp2:=Grandom(PopSize)+1;
                hor1:=j-1;
                vert1:=i-1;
                hor2:=j-1;
                vert2:=i-1;
                If Not Sel^[x]
                Then begin
                      If Mig^[x]
                      Then Begin
                             GetOfset(OfsVer,OfsHor);
                             hor1:=GetNewCoord(Tor,j,NumbCol,OfsHor);
                             vert1:=GetNewCoord(Tor,i,NumbRow,OfsVer);
                           End;
                      Where1:=PopSize*Hor1+PopSize*NumbCol*Vert1+Temp1;
                      If Mig^[x+MaxInd]
                      Then Begin
                             GetOfset(OfsVer,OfsHor);
                             hor2:=GetNewCoord(Tor,j,NumbCol,OfsHor);
                             vert2:=GetNewCoord(Tor,i,NumbRow,OfsVer);
                           End;
                      Where2:=PopSize*Hor2+PopSize*NumbCol*Vert2+Temp2;
                     End
                Else Begin
                      If Mig^[x]
                      Then Begin
                             GetOfset(OfsVer,OfsHor);
                             hor1:=GetNewCoord(Tor,j,NumbCol,OfsHor);
                             vert1:=GetNewCoord(Tor,i,NumbRow,OfsVer);
                           End;
                      Where1:=PopSize*Hor1+PopSize*NumbCol*Vert1+Temp1;
                      Where2:=Where1;
                     End;
```

```
                Champ2^[x,1]:=ParChamp2^[Where1,Grandom(2)+1];
                Champ2^[x,2]:=ParChamp2^[Where2,Grandom(2)+1];
             end;
     FreeMem(ParChamp2,SizeOf(ParChamp2^));
     ParChamp2:=Champ2;
end;{Of Proc Migr}
(************************************MAIN PROGRAM***************************)

Begin
     clrscr;
     Graphics:=False;
     Window2(W_Size_Big);
     Erase_File(Name);
     FilNamDat:=Name;
     Assign(FileDat,FilNamDat);
     Write_Descr_GFP('6',FileDat);
     With FileRec(FileDat) Do
     Begin
          SelfProp:=Descr_GFP3(userdata).self;
          PropMig:=Descr_GFP3(userdata).mig;
          PopSize:=Descr_GFP3(userdata).Popsize;
          Tor:=Descr_GFP3(userdata).tor;
          Distance:=Descr_GFP3(userdata).Dist;
          Average:=Descr_GFP3(Userdata).Aver;
          With ThisDescrRec Do
          Begin
               FilNam:=Name;
               For i:=1 to 16 do
               NewData[i]:=userdata[i];
          End;
     End;
     Update_Filerec(ThisDescrRec);
     t:=0;Reps:=10;
     InitTxt(Name);
     Read_Descr_GFP(FScreen,FileDat);
     GotoXy(15,10);
     Write('How many replicates do you want [10..50] : ');
     Reps:=LintRange(10,50);
     Rewrite(FileDat);close(FileDat);
     NumbSp:=MaxInd div PopSize;
     Case NumbSp of
              4096 :begin numbrow:=64;numbcol:=64;end;
              1024 :begin numbrow:=32;numbcol:=32;end;
              256  :begin NumbRow:=16;NumbCol:=16;end;
              64   :begin NumbRow:=8;NumbCol:=8;end;
              16   :begin NumbRow:=4;NumbCol:=4;end;
              4    :begin NumbRow:=2;NumbCol:=2;end;
              1    :begin NumbRow:=1;NumbCol:=1;end;
     end;
     GoToXY(15,12);
     write('Replicate no:      completed');
     While t<Reps Do
     Begin
         t:=t+1;
         GetMem(Champ,SizeOf(Champ^));
         Ini(Champ);
         GetMem(Mig,SizeOf(Mig^));GetMem(Self,SizeOf(Self^));
         FillBoolArray(PropMig,Mig);
         FillBoolArray(SelfProp,Self);
         Migr      (Tor,NumbCol,NumbRow,PopSize,
                    Mig,Self,
                    Champ);
         reset(FileDat);seek(FileDat,FileSize(FileDat));
         Write(FileDat,Champ^);
         Close(FileDat);
         FreeMem(Champ,SizeOf(Champ^));
         FreeMem(Mig,SizeOf(Mig^));
         FreeMem(Self,SizeOf(Self^));
         GoToXY(29,12);Write(t:4);
     End; {Of while t<Reps}
End;{Of Proc GFSS8NOR}
```

A.10 INTRECS.PAS

```
procedure intrecs;
(***************     Model for plants on a torus      ********************)
const
 maxplants = (*14641*)4096;
 n         = (*121*)64;
type
    genratns  =    1..2;
    sexes     =    1..2;
    fields    =    record
                      genos: array[1..maxplants,genratns] of byte;
                   end;
    field2s   =    record
                      geno2s: array[1..n,1..n] of byte;
                   end;
    intflds   =    record
```

```
                      gamets: array[1..maxplants,sexes] of word;
                   end;


var
   intfld                  :    intflds;
   file1                   :    file of intflds;
   NAME1                   :    names;
   rep,reps                  :    integer;
   sdm,sdf,s               :    single;
   selfing,plant           :    boolean;
   NeibSize                :    Single;
   ch,chr,chc,answer       :    char;

(**************************************************************************)

procedure self;
begin
write('"Random" selfing?');
if Affirmed(True,Graphics) then begin selfing:=true;s:=0.0; end
                  else
                       begin
                            selfing:=false;
                            write('Input selfing rate [0..1]? ');
                            s:=Range(0.0,1.0);
                            writeln;
                       end;
end;(* Procedure Self*)

(**************************************************************************)
function Norm(var a,b:single) : single;
var
 x,y,s,l,d1 : single;
begin
if b<0.00000001 then norm:=a else
begin
s:=0.0;
repeat
 x:=2.0*Uniform-1.0; y:=2.0*Uniform-1.0;
 s:=sqr(x)+sqr(y);
 until s<1.0;
l:=sqrt(-2.0*ln(s)/s);
if Uniform<0.5 then d1:=x*l  else d1:=y*l;
norm:=a+d1*b;
end; (*if b*)
end (* norm *) ;

(**************************************************************************)
function sqdispersed(var a:integer; sd:single): integer;
 (*using mean=0 and std. dev., sd, generates new location from a*)
 (*on a toroidal surface of n * n                               *)
var
 m       : single;
 x,y     : integer;

begin
 m:=0.0;
 y:= (a-1) div n;            {ordinate}
 x:= (a-1) mod n;            {abcissa}
 x:=x+round(Norm(m,sd));
 y:=y+round(Norm(m,sd));
 While x < 0 do x:=n+x;      {0 <= x}
 While y < 0 do y:=n+y;      {0 <= y}
 x:=x mod n;                 {0 <= x <=  n-1}
 y:=y mod n;                 {0 <= y <=  n-1}
sqdispersed:= 1 + x + n * y; {1 <= SqDisPersed <= n*(n-1)+n-1+1 = n*n}
end (*function sqdispersed*);
(**************************************************************************)
Function dispersed(var a:integer; sd:single):integer;
var
m           :single;
x,y         :integer;

begin

m:=0.0;
repeat
x:=((a-1) mod n) +1;
y:=((a-1) div n) +1;
x:=x+round(norm(m,sd));
y:=y+round(norm(m,sd));
until ((x>0) and (y>0) and (x<=n) and (y<=n));
dispersed:=x + n * (y-1);
end;(*Function dispersed*)

(**************************************************************************)
procedure intseeds(sd1,sd2:single);

var
   i,j,id         :     integer;
   male,female,who:     integer;
```

```
begin

 with intfld do
     for i:=1 to maxplants do
     begin
          if Uniform<0.5 then id:=0 else id:=maxplants;
          female:=dispersed(i,sd2);
          gamets[i,2]:=female+id;
          if plant then who:=female else who:=i;
          if selfing then male:=dispersed(who,sd1)
          else if Uniform<s then male:=female
               else
               repeat
                     male:=dispersed(who,sd1);
               until male<>who;
          if Uniform<0.5 then id:=0 else id:=maxplants;
          gamets[i,1]:=male+id;
     end; (*i loop *)

end;   (* integer seedling for true & torus -plant model*)


Procedure BuildRep;
var
rep   : integer;
Begin
   Rewrite(file1);
   Graphics:=False;
   plant:=true;
   repeat
     writeln(' Do you want a plant model? :');
     Plant:=Affirmed(False,graphics);
     write(' Input Neighbourhood size : ');readln(NeibSize);
     If Plant
     then begin
           write('Input female dispersal variance- ');
           readln(sdf);
           self;
           sdm:=2*(NeibSize/4/PI-sdf)/(1-s);
           writeln (' The calculated male dispersal variance is : ',sdm:10:7);
            write (' Do you want to alter it ?');
           If Affirmed(True,graphics)
           then begin
                 write('Input male dispersal variance- ');
                 readln(sdm);
                 NeibSize:=4*Pi*(sdm*(1-s)/2+sdf);
                 writeln (' The New Neighbourhood size is : ',NeibSize:10:0);
                end;
          end
     else begin
            self;
            sdf:=Neibsize/pi/(1-s);
            sdm:=NeibSize/pi/(1-s);
            writeln (' Dispersal is : ',sdm:10:7);
             write (' Do you want to alter it ?');
            If Affirmed(True,Graphics)
            then begin
                   write('Input dispersal variance- ');
                   readln(sdm);
                   sdf:=sdm;
                   NeibSize:=Pi*(1-s)*(sdm+sdf)/2;
                 end;
          end;
     writeln;
     write('You are currently modeling ');
     if plant then  writeln('a plant neighbourhood model')
     else writeln('a true wright neighbourhood model');
     writeln('     male disp var : ',sdm:10:7,'      female disp var : ',sdf:10:7);
     If not selfing then writeln(' The proportion of selfing individuals is : ',s:10:7)
     else writeln(' Selfing is random ');
     writeln(' The expected neighbourhood size is : ',Neibsize:10:0);
     write( 'is everything to your satisfaction?: ');
   until affirmed(true,graphics);

sdm:=sqrt(sdm)/2;
sdf:=sqrt(sdf)/2;

write('Input no. of replicates- ');readln(reps);
for rep:=1 to reps do
    begin
      intseeds(sdm,sdf);
      write(file1,intfld);
      writeln('Completed rep ',rep:3);
    end;
close(File1);
End;{Of Procedure BuildReps}

(**************************************************************************)

begin  (*MAIN PROGRAM*)
```

```
     clrscr;
     NAME1:='resul.dat';
     write('Input filename for Output fields- ');
     readln(NAME1);
     Assign(file1,NAME1);
     Buildrep;
end;
```

A.11 GFSSNEIB.PAS

```
(************GENE FLOW PATTERN : STEPPING STONE 8 CON EXP DISP.**************)
Procedure GFSSNEIB;

var
  Champ                                     : IntFldPtr;
  a,i,x                                     : Integer;
  Self                                      : ExistsPtr;
  t,Reps                                    : byte;
  PropMig,SelfProp                          : Double;
  NumbSP,PopSize                            : Integer;
  NumbRow,NumbCol                           : Integer;
  Distance                                  : BYTE;
  average                                   : SINGLE;
  Tor                                       : boolean;


(**RETURN AN INTEGER FROM A NEGATIVE EXPONENTIAL******************************)

Function IntExpo(Dist:BYTE;av:SINGLE) :Byte;

Var Temp  :Single;
    Res   :Byte;

begin
     Repeat
           Temp:=-1.0*av*Ln(Uniform);
           Res:=Trunc(Temp);
     Until  Res<=127;
     IntExpo:=res;
End; {Of Function IntExpo}

(**************GET THE HOR AND VERT OFFSET OF MIGR*************************)

Function GetOfset :shortint;
var
temp, dist:byte;
offs       :shortint;
begin
     Offs:=-99;
     Dist:=IntExpo(Distance,average);
     temp:=GRandom(2);
     case temp of
          0 : Offs:=-dist;
          1 : Offs:=dist;
     end;
GetOfSet:=Offs
end; {Of Proc GetOfs}

(*******************MIG PROC************************************************)
Procedure Migr      (Tor                      :boolean;
                     NumbCol,NumbRow,PopSize  : integer;
                     Sel                      : ExistsPtr;
                     var ParChamp2            : IntFldPtr);
var
Champ2                              :IntFldPtr;
Temp1,Temp2                         :Integer;
i,j,k,l,All1,All2,x                 :integer;
Hor1,Hor2,Vert1,Vert2               :integer;
where1,where2                       :integer;
OfsHor,OfsVer                       :ShortInt;


begin
     GetMem(Champ2,SizeOf(Champ2^));
     x:=0;
            for i:=1 to NumbRow Do
            For j:=1 to NumbCol Do
            For k:=1 to PopSize Do
             Begin
                x:=x+1;
                Temp1:=Grandom(PopSize)+1;
                Temp2:=Grandom(PopSize)+1;
                hor1:=j-1;
                vert1:=i-1;
                hor2:=j-1;
                vert2:=i-1;
                If Not Sel^[x]
                Then begin
                             OfsHor:=GetOfset;
```

```
                             hor1:=GetNewCoord(Tor,j,NumbCol,OfsHor);
                             OfsVer:=GetOfset;
                             vert1:=GetNewCoord(Tor,i,NumbRow,OfsVer);
                             Where1:=PopSize*Hor1+PopSize*NumbCol*Vert1+Temp1;
                             OfsHor:=GetOfset;
                             hor2:=GetNewCoord(Tor,j,NumbCol,OfsHor);
                             OfsVer:=GetOfset;
                             vert2:=GetNewCoord(Tor,i,NumbRow,OfsVer);
                             Where2:=PopSize*Hor2+PopSize*NumbCol*Vert2+Temp2;
                     End
                Else Begin
                             OfShor:=GetOfset;
                             hor1:=GetNewCoord(Tor,j,NumbCol,OfsHor);
                             OfsVer:=GetOfset;
                             vert1:=GetNewCoord(Tor,i,NumbRow,OfsVer);
                             Where1:=PopSize*Hor1+PopSize*NumbCol*Vert1+Temp1;
                             Where2:=Where1;
                     End;
                Champ2^[x,1]:=ParChamp2^[Where1,Grandom(2)+1];
                Champ2^[x,2]:=ParChamp2^[Where2,Grandom(2)+1];
             end;
     FreeMem(ParChamp2,SizeOf(ParChamp2^));
     ParChamp2:=Champ2;
end;{Of Proc Migr}
(*************************************MAIN PROGRAM***************************)

Begin
     clrscr;
     Graphics:=False;
     Window2(W_Size_Big);
     Erase_File(Name);
     FilNamDat:=Name;
     Assign(FileDat,FilNamDat);
     Write_Descr_GFP('7',FileDat);
     With FileRec(FileDat) Do
     Begin
          SelfProp:=Descr_GFP7(userdata).self;
          PopSize:=Descr_GFP7(userdata).Popsize;
          Tor:=Descr_GFP7(userdata).tor;
          Average:=Descr_GFP7(userdata).Aver;
          With ThisDescrRec Do
          Begin
               FilNam:=Name;
               For i:=1 to 16 do
               NewData[i]:=userdata[i];
          End;
     End;
     Update_Filerec(ThisDescrRec);
     t:=0;Reps:=10;
     InitTxt(Name);
     Read_Descr_GFP(FScreen,FileDat);
     GotoXy(15,10);
     Write('How many replicates do you want [10..50] : ');
     Reps:=LintRange(10,50);
     Rewrite(FileDat);close(FileDat);
     NumbSp:=MaxInd div PopSize;
     Case NumbSp of
              4096 :begin numbrow:=64;numbcol:=64;end;
              1024 :begin numbrow:=32;numbcol:=32;end;
              256  :begin NumbRow:=16;NumbCol:=16;end;
              64   :begin NumbRow:=8;NumbCol:=8;end;
              16   :begin NumbRow:=4;NumbCol:=4;end;
              4    :begin NumbRow:=2;NumbCol:=2;end;
              1    :begin NumbRow:=1;NumbCol:=1;end;
     end;
     GoToXY(15,12);
     write('Replicate no:      completed');
     While t<Reps Do
     Begin
         t:=t+1;
         GetMem(Champ,SizeOf(Champ^));
         Ini(Champ);
         GetMem(Self,SizeOf(Self^));
         FillBoolArray(SelfProp,Self);
         Migr      (Tor,NumbCol,NumbRow,PopSize,
                    Self,
                    Champ);
         reset(FileDat);seek(FileDat,FileSize(FileDat));
         Write(FileDat,Champ^);
         Close(FileDat);
         FreeMem(Champ,SizeOf(Champ^));
         FreeMem(Self,SizeOf(Self^));
         GoToXY(29,12);Write(t:4);
     End; {Of while t<Reps}
End;{Of Proc GFSSNEIB}
```

A.12 DISPERSAL.PAS

```
Procedure dispersal;
```

```
type
Square_FieldPtr       =^Square_Field;
Square_Field          = array[0..(n-1),0..(n-1),1..2]of integer;

var
name,filnam1,filnam3                  :names;
file1                                 :IntFldFile;
File3                                 :text;
SqField                               :Square_FieldPtr;
Field                                 :IntFldPtr;
ThisOne,Dispx,Dispy,IDispxy                   :integer;
histo                                 :array [0..200] of longint;
AveDisp,VarDisp,Thisdisp              :Extended;
avedispx,avedispy,Dispxy,absdisp,axdisp                     :Extended;
x,y,k,x0,y0,i                         :integer;

Function Max(x,y:integer):integer;
Begin
     If x>y Then Max:=x Else Max:=y;
end; {Of Function Max}
(*********************MAIN PROGRAM************************************)
begin
     clrscr;
     ext1:='';
     FileList(Ext1);
     Window2(W_Size_Small);
     Repeat
           writeln (' Input filnam for output : ');
           write(' Return to Exit. ');
           readln(name);
           If name='' then Exit;
     Until FileExist(Name);
     filnam1:=name;
     filnam3:=name+'.TTT';
     for i:=0 to 200 do histo[i]:=0;
     assign(File3,FilNam3);rewrite(File3);
     append(file3);
     writeln(File3,
     '    rep      avg.abs.disp    var.disp      avg.Axdisp');
     Close(File3);
     assign(file1,filnam1);reset(file1);
     GetMem(Field,SizeOf(Field^));
     GetMem(SqField,SizeOf(SqField^));
     while not eof(file1) do
     begin
          read(file1,field^);
          i:=0;
          for y:=0 to (n-1) do
          for x:=0 to (n-1) do
          begin
               i:=i+1;
               sqfield^[y,x,1]:=field^[i,1];
               sqfield^[y,x,2]:=field^[i,2];
          end;
          absDisp:=0.0;AxDisp:=0.0;VarDisp:=0.0;AveDispx:=0.0;AveDispy:=0.0;Thisdisp:=0.0;
          for y:=0 to (n-1) do
          for x:=0 to (n-1) do
          for k:=1 to 2 do
          begin
               ThisOne:=sqfield^[y,x,k];
               If ThisOne>MaxInd then ThisOne:=ThisOne-MaxInd;
               x0:=((ThisOne-1) mod n);  {provides a figure between 0 and (n-1)}
               y0:=((ThisOne-1) div n);  {provides a figure between 0 and (n-1)}
               dispx:=abs(x0-x);
               dispy:=abs(y0-y);
               If dispx>(n div 2) then dispx:=n-dispx;
               If dispy>(n div 2) then dispy:=n-dispy;
               DispXy:=sqr(Dispx)+Sqr(Dispy);
               Dispxy:=1.0*sqrt(Dispxy);
               IDispXY:=trunc(DispXy);
               Histo[IDispxy]:=histo[IDispxy]+1;
               ThisDisp:=Thisdisp+Dispxy;
               absDisp:=absDisp+Dispx+Dispy;
               VarDisp:=VarDisp+Sqr(Dispxy);
          end;
          axDisp:=Thisdisp/DMaxInd;
          absDisp:=absDisp/DMaxInd/2;
          VarDisp:=VarDisp/DMaxInd-sqr(axDisp);
          writeln(FilePos(File1):6,absDisp:8:4,'  ',VarDisp:8:4,'  ',axDisp:8:4);
          append(File3);
          writeln(File3,
          FilePos(File1):6,'       ',AbsDisp:8:4,'       ',VarDisp:8:4,'  ',axDisp:8:4);
          close(File3);
     end;
     append(File3);
     writeln(file3);
     for i:=0 to 200 do
     if histo[i]<>0 then writeln(file3,i+0.5:6:2,'  ',histo[i]);
     close(file3);
     Freemem(Field,SizeOf(Field^));
```

```
     FreeMem(SqField,SizeOf(SqField^));
end;
```

A.13 BUILGEN.PAS

```
Procedure buildgen;


var
   File1,File2          :IntFldFile;
   name                 :array [1..5] of names;
   Name1                :names;
   CountAll             :Real;
   Pres                 :Exists;
   ThisGen,NumbOfGen    :word;
   MaxInFile            :word;
   name2,name3          :names;
   field,prev_field     :intfldptr;
   temp,i,j,count       :word;
   jj,rep,MaxRep        :byte;
   col                  :array[1..5]of word;
   Write_It,Print       :Boolean;
   TEMP1                :BYTE;

begin
     ClrScr;
     Graphics:=False;
     Ext1:='';
     FileList(Ext1);
     Window2(W_Size_Small);
     Write('Do you want to write results to a file : ?');
     Write_It:=Affirmed(True,Graphics);
     Write(' Input number of generations [2..10000]: ');
     NumbOfGen:=LintRange(2,10000);
     writeln ('You can give up to five names :');
     rep:=0;
repeat
     Repeat
           rep:=rep+1;
           Write('Input one of the above (without ext.) ');
           Write('(Return to Exit.) :');
           readln(Name[rep]);
           If ((rep=1) and (Name[rep]='')) then Exit;
     Until (FileExist(Name[rep]));
Until ((Name[rep]='') or (rep=5));
     If Name[rep]='' then MaxRep:=Rep-1 else MaxRep:=rep;
     write(' Print the graph? : ');
     Print:=Affirmed(False,Graphics);
rep:=0;
     SetGraphMode(GrMode);
     If Print
     Then Begin
               for i:=1 to 5 do col[i]:=white;
               SetBkColor(Black);
          End
     Else Begin
               col[1]:=yellow;
               col[2]:=red;
               col[3]:=white;
               col[4]:=green;
               col[5]:=lightcyan;
               SetBkColor(Blue);
          End;

     Graphics:=True;
     axex:='Generations.';
     axey:='Number of extant Alleles.';
     Axes;
     settextstyle(defaultfont,horizdir,1);
                                 (*graduation axe des y*)
     outtextxy(xoi-2,yfi+1,'-');
     outtextxy(xoi-30,yfi+1,'100');
     outtextxy(xoi-2,yfi+y4+1,'-');
     outtextxy(xoi-30,yfi+y4+1,'75');
     outtextxy(xoi-2,yfi+2*y4+1,'-');
     outtextxy(xoi-30,yfi+2*y4+1,'50');
     outtextxy(xoi-2,yfi+3*y4+1,'-');
     outtextxy(xoi-30,yfi+3*y4+1,'25');
     outtextxy(xoi-2,yoi+2,'-');
     outtextxy(xoi-30,yoi+2,'0');
                                (*graduation axe des x*)
     outtextxy(xfi-3*x4-3,yoi+2,'+');
     outtextxy(xfi-2*x4-3,yoi+2,'+');
     outtextxy(xfi-x4-3,yoi+2,'+');
     outtextxy(xfi-3,yoi+2,'+');
Repeat
     rep:=rep+1;
     Name1:=Name[rep];
     Name2:=name1+'.DAT';
```

```
assign(File1,Name1);Reset(File1);
If Write_it
Then Begin
          assign(File2,Name2);
          If FileExist(Name2)
          Then Reset(File2) else rewrite(File2);
          Assign(FileTxt,Name1+'.TXT');
          If FileExist(Name1+'.TXT')
          Then reset(FileTxt)
          Else Rewrite(FileTxt);
          Close(FileTxt);
          Append(FileTxt);
          Writeln(FileTxt,
          '**************************************************');
          Writeln(FileTxt,
          'NUMBER OF ALLELES EXTANT IN POPULATION AT GEN. X');
          Close(FileTxt);
     End;
GetMem(prev_Field,SizeOf(Prev_Field^));

TEMP1:=GRANDOM(FILESIZE(FILE1));
seek(File1,TEMP1);
Read(File1,Prev_field^);
Count:=1;
MaxInFile:=1;
If Write_It
Then If NumbOfGen<=MaxInFile
     Then Count:=1
     else Count:= NumbOfGen div MaxInFile;
MoveTo(xo,-dy+yo);
For ThisGen:=1 to NumbOfGen do
Begin
     If Print
     Then SetLineStyle(((rep-1) mod 4),0,((rep-1) div 4)*2+1)
     Else SetLineStyle(SolidLn,0,NormWidth);
     SetColor(col[rep]);
     GetMem(Field,SizeOf(Field^));
     TEMP1:=GRANDOM(FILESIZE(FILE1));
     seek(File1,TEMP1);
     Read(File1,Field^);
     If ThisGen=1
     Then Begin
               For i:=1 to MaxInd do
               For j:=1 to 2 do
               Field^[i,j]:=Prev_Field^[i,j];
          end
     Else Begin
              For i:=1 to MaxInd do
              For j:=1 to 2 do
              Begin
                   Temp:=Field^[i,j];
                   If ((Temp>DMaxInd) or (Temp<1))
                   Then Begin
                             OutTextXY(5,10,
                             ' I found an unexisting  Allele. Program Stopped!');
                             Halt(1);
                        End;
                        If Temp>MaxInd
                        Then Begin
                                  jj:=2;
                                  Temp:=Temp-MaxInd;
                             end
                        Else   jj:=1;
                        Field^[i,j]:=Prev_Field^[Temp,jj];
              End;
          End;
     If Write_it
     Then If (ThisGen MOD COUNT)=0
     Then Begin
               Seek(File2,FileSize(File2));
               Write(File2,Field^);
          End;
     CountAll:=0.0;
     For i:=1 to DMaxInd Do Pres[i]:=False;
     For i:=1 to MaxInd do
     For j:=1 to 2 do Pres[Field^[i,j]]:=True;
     For i:=1 To DMaxInd Do
     If Pres[i] Then CountAll:=CountAll+1.0;
     If Write_It
     Then If (ThisGen MOD COUNT)=0
          Then Begin
                    Append(FileTxt);
                    Writeln(FileTxt,thisgen:4,'  ',Countall:6:0);
                    Close(FileTxt);
               End;
     G1y:=CountAll/100{DMaxInd};
     g1x:=(ThisGen+1)/(NumbOfGen+1);
     Px1:=Round(dx*g1x)+xo;
     Py1:=-Round(dy*g1y)+yo;
     LineTo(px1,py1);
     FreeMem(Prev_Field,SizeOf(Prev_Field^));Prev_Field:=Nil;
```

```
          Prev_Field:=Field;
     End;
     MoveTo(GetMaxX-250,10+10*Rep);
     LineTo(GetMaxX-200,10+10*Rep);
     SetTextStyle(DefaultFont,HorizDir,1);
     SetTextJustify(LeftText,CenterText);
     OutTextXy(GetMaxX-180,10+10*rep,name1);
     close(File1);
     If Write_It Then Close(File2);
     FreeMem(Field,SizeOf(Field^));
Until Rep=MaxRep;
Message_End;
End;
```

A.14 PLOTFREQ.PAS

```
Procedure plotfreq;


type

histo= array [0..DMaxInd] of Word;

var

filnam1                    : Names;
file1                      : IntFldfile;
i,j,k,numbclass,AllNumb    : integer;
Max,Incr                   : integer;
Gener                      : array[1..5] of Byte;
yoRep,dyRep                : array[1..5] of integer;
Rep,MaxRep                 : Byte;
MaxHisto                   : Word;
print                      : boolean;
histoAll            : Histo;
Field                      : IntFldPtr;
FreqOfAll            : AliveGenPtr;
Col                        : array[1..5] of word;
s1,s2                      : string[18];
(*******************************CATEGORISE***********************************)
procedure categorise ( FreqOfAll    : AliveGenPtr;
                       Max,incr     : integer;
                       NumbClass    :Integer;
                   var histos       : histo);

var
i,j,Temp                 : integer;
NClass                 : integer;

begin
     for NClass:=0 to NumbClass do histos[NClass]:=0;
     for i:=1 to dmaxind do
     begin
          temp:=FreqOfAll^[i];
          If Temp=0
          Then Histos[0]:=Histos[0]+1
          Else Begin
                    j:=0;
                    For NClass:=1 to NumbClass do
                    begin
                         If (Temp>j) and (Temp<=(j+incr))
                         Then Histos[NClass]:=Histos[NClass]+1;
                         j:=j+incr;
                    end;
                    MaxHisto:=0;
                    For NClass:=1 to NumbClass do
                    If Histos[NClass]>MaxHisto
                    Then MaxHisto:=Histos[NClass];
               End;
     End;
end; {Of Categorise}

(*********************************************************************)
begin                   (* MAIN PROGRAM*)

     ClrScr;
     Graphics:=False;
     For i:=1 to 5 do
     Begin
          yorep[i]:=0;
          dyRep[i]:=0;
     End;
     Ext1:='DAT';
     FileList(Ext1);
     Window2(W_Size_Small);
     Repeat
           write ( ' Input FilName (Without Ext.): ');
           write ( ' Return to exit. ');
           readln(FilNam1);
```

```
      If FilNam1='' then Exit;
Until FileExist(FilNam1+'.DAT');
assign(File1,Filnam1+'.DAT');reset(File1);
GetMem(FreqOfAll,SizeOf(FreqOfAll^));
Writeln('You can give up to 5 different generations.');
Rep:=0;
Repeat
     Rep:=Rep+1;
     Write('Which generations do you want to look at ? :');
     Write('(0 to exit).');
     Gener[rep]:=LintRange(0,FileSize(File1));
Until ((Gener[Rep]=0) or (Rep=5));
If Gener[Rep]=0 Then MaxRep:=Rep-1 Else MaxRep:=Rep;
write('Print Output? ');
Print:=Affirmed(false,Graphics);
If not print
Then For i:=1 to 5 do col[i]:=yellow
Else For i:=1 to 5 do col[i]:=white;
Close(File1);

rep:=0;
SetGraphMode(GrMode);
If Print Then SetBkColor(Black) else SetBkColor(Blue);
Graphics:=True;
axex:='CLASS OF NUMBER OF COPIES';
axey:='NUMBER OF ALLELES IN CLASS X.';
axes;
SetColor(col[1]);
SetTextStyle(SmallFont,HorizDir,4);
SetTextJustify(LeftText,CenterText);
OutTextXY(xo-70,yo-dy+10,'Generation');
For i:=1 to MaxRep do
Begin
     yoRep[i]:=yo-trunc((i-1)*(dy div MaxRep));
     dyRep[i]:=dy div MaxRep;
End;
Repeat
      Rep:=Rep+1;
      Max:=0;
      AllNumb:=0;
      Reset(File1);
      Seek(File1,Gener[rep]-1);
      GetMem(Field,SizeOf(Field^));
      read(File1,Field^);
      close(File1);
      for i:=1 to dmaxind do
      FreqOfAll^[i]:=0;
      For i:=1 to MaxInd do
      For j:=1 to 2 do
      FreqOfAll^[Field^[i,j]]:=FreqOfAll^[Field^[i,j]]+1;
      FreeMem(Field,SizeOf(Field^));
      For i:=1 to DMaxInd do
      if FreqOfAll^[i]>Max
      then Max:=FreqOfAll^[i];
      NumbClass:=Max;
      incr:=1;
      Categorise(FreqOfAll,Max,Incr,NumbClass,HistoAll);
      G1y:=HistoAll[0]/MaxHisto{DMaxInd};
      G1y:=1.0;
      G1x:=0;
      Px1:=xo;
      py1:=-Round(dyRep[rep]*g1y)+yoRep[rep];
      If Rep<MaxRep
      Then Begin
                MoveTo(px1,py1);
                SetLineStyle(SolidLn,0,NormWidth);
                setColor(White);
                LineTo(px1+dx+10,py1);
           End;
      MoveTo(px1,py1);
      SetColor(Col[Rep]);
      SetTextStyle(SmallFont,HorizDir,4);
      SetTextJustify(LeftText,CenterText);
      Str(MaxHisto,s1);
      s2:=Concat('Highest is: ',s1);
      OutTextXY(xo+dx+10,yorep[rep]-(dyRep[rep] div 3),s2);
      Str(Max,s1);
      s2:=Concat('Numb. Class: ',s1);
      OutTextXY(xo+dx+10,yorep[rep]-(2*(dyRep[rep] div 3)),s2);
      Str(Gener[rep],s1);
      OutTextXY(xo-20,yoRep[Rep]-(dyRep[rep] div 2),s1);
      For i:=1 to NumbClass Do
      Begin
           G1y:=HistoAll[i]/MaxHisto{DMaxInd};
           If G1y>1.0 Then G1y:=1.0;
           G1x:=i/Max;
           Px1:=Round(dx*G1x)+xo;
           Py1:=-Round(dyRep[rep]*g1y)+yoRep[rep];
           MoveTo(px1,yoRep[rep]);
           LineTo(px1,Py1);
      End;
```

```
     Until rep=MaxRep;
     Message_End;
     FreeMem(FreqOfAll,SizeOf(FreqOfAll^));
end; {Of Procedure PlotFreq}
```

A.15 REDUC.PAS

```
(************REDUC PRO********************************************************)


Procedure reduc;

var

Champ,ParChamp                              : IntFldPtr;
{FreqOfAll                                  : AliveGenPtr;}
i,j,L,K                                     : Word;
Temp,jj,REP,Compt,X                         : word;


begin
     ClrScr;
     Graphics:=False;
     ext1:='DAT';
     FileList(ext1);
     Window2(W_Size_Small);
     Repeat
           Write('Input One Of the above ');
           Write('(Return to Exit.) : ');
           Readln(Name);
           If name='' then Exit;
     Until FileExist(Name+'.DAT');
     FilNamDat:=Name+'.DAT';
     FilNamRed:=Name+'.RED';
     assign(FileRed,FilNamRed);rewrite(FileRed);close(FileRed);
     assign(FileDat,FilNamDat);reset(FileDat);
     MaxAll:=2;
     write ('Input the Number Of Alleles you want [2..50]: ');
     MaxAll:=LintRange(2,50);
     REP:=1;Compt:=0;
     While Compt<rep Do
     Begin
          reset(FileDat);
          getmem(ParChamp,SizeOf(ParChamp^));
          X:=0;
          for i:=1 to MaxInd do
          for j:=1 to 2 do Parchamp^[i,j]:=GRandom(MaxAll)+1;
          For i:=1 to MaxAll Do FreqOfAll^[i]:=0;
          for i:=1 to MaxInd do
          for j:=1 to 2 do
          FreqOfAll^[ParChamp^[i,j]]:=FreqOfAll^[ParChamp^[i,j]]+1;
          reset(FileFre);seek(FileFre,FileSize(FileFre));
          Write(FileFre,FreqOfAll^);
          Close(FileFre);
          Freemem(FreqOfAll,SizeOf(FreqOfAll^));FreqOfAll:=Nil;
          getmem(Champ,SizeOf(Champ^));
          While not eof(FileDat) Do
          Begin
               read(FileDat,Champ^);
               For i:=1 to MaxInd do
               For j:=1 to 2 do
               Begin
                    Temp:=Champ^[i,j];
                    If Temp>MaxInd
                    Then Begin
                              jj:=2;
                              Temp:=Temp-MaxInd;
                         end
                    Else   jj:=1;
                    Champ^[i,j]:=ParChamp^[Temp,jj];
               End;
               reset(FileRed);
               seek(FileRed,FileSize(FileRed));
               write(FileRed,Champ^);
               close(FileRed);
          End; { Of While Not Eof(FileDat)}
          freemem(champ,SizeOf(Champ));
          Freemem(ParChamp,SizeOf(ParChamp^));ParChamp:=Nil;
          compt:=compt+1;
     End;
     Close(FileDat);
End; { Of Proc Reduc}
```

A.16 SHFIELD.PAS

```
Procedure shfield;
```

```
const
FieldSide  :Word=n;
SpSide1    :Word=1;
SpSide2    :word=1;

type
PicFldPtr  =^PicFld;
picfld   = array[1..MaxInd,1..3] of byte;
Geno     = array[0..(n-1),0..(n-1)] of byte;
colour   = array[0..15] of byte;
Sentence = string[80];
var

XTitOutp                                : integer;
file1                                   : intfldfile;
File2                                   : alivegenfile;
fld,Fldb                                : intfldPtr;
Changes                                 : picfldPtr;
fall                                    : word;
FreqOfAll                               : AliveGenPtr;
genos                                   : geno;
freq                                    : array[0..n] of integer;
XScale,YScale,XOri                      : integer;
xwidth,ywidth                           : word;
YOri                                    : integer;
x,Gen,LastXPos,Xpos,Numb,i,j,k,ii,jj    : longint;
a1,a2,aa                                : word;
countchange,temp,countgen               : word;
Name,FilNam1,FilNam2,s,FilNam3          : names;
Titre                                   : sentence;
Colours                                 : colour;
InfEdge,SupEdge,Posmin,PosMax,Dif       : integer;
PCol,GraphCard,Hom1,Hom2,Het            : Byte;
chc,Ans,ansb,choix,bb                   : char;
DrawGrid,Reduc,Print                    : boolean;
file3                                   : text;

(*************************************************************)

Procedure initialize;


begin
     Colours[0] :=Black;             colours[8]  :=DarkGray;
     Colours[1] :=Blue;              colours[9]  :=LightBlue;
     Colours[2] :=Green;             colours[10] :=LightGreen;
     Colours[3] :=Cyan;              colours[11] :=LightCyan;
     Colours[4] :=Red;               colours[12] :=LightRed;
     Colours[5] :=Magenta;           colours[13] :=LightMagenta;
     Colours[6] :=Brown;             colours[14] :=Yellow;
     Colours[7] :=LightGray;         colours[15] :=White;
     SetGraphMode(GrMode);
     Graphics:=True;
     XScale:=2*(GetMaxX Div 6)-1;
     YScale:=2*(GetMaxY div 3)-1;
     Xori:=GetMaxX div 6; {160}
     Yori:=GetMaxY-5*(GetMaxY div 6);  {80}
     XScale:=340;
     YScale:=220;
     xwidth:=XScale div n;
     ywidth:=YScale div n;
     XScale:=XWidth*n;
     YScale:=YWidth*n;

End;{of proc initialize}

Procedure Title(titre:sentence);
var
YPos,ZPos                            : integer;
x1,y1,x2,y2                          : word;
i,j,Stap,N                           : longint;
xwidth,ywidth                        : word;
s,w,t                                : string[40];
x                                    : real;
begin


      SetFillStyle(SolidFill,PCol);
      SetLineStyle(SolidLn,0,NormWidth);
                                        (*ALL THE SCREEN*)
      Bar(0,0,GetMaxX,GetMaxY);
      SetTextStyle(TriplexFont,HorizDir,2);
      SetTextJustiFy(centertext,CenterText);
      SetColor(15);
      SetLineStyle(SolidLn,0,ThickWidth);
                                         (*DEFINE BOX*)
      Rectangle(80,30,GetMaxX-80,GetMaxY-30);
      OutTextXY(GetMaxX div 2,40,titre);
      SetFillStyle(SolidFill,0);
```

```
      SetLineStyle(SolidLn,0,NormWidth);
                                             (*LEGEND COMMAND*)
      Rectangle(GetMaxX-150,100,GetMaxX-90,140);
      SetTextStyle(SmallFont,HorizDir,4);
      SetTextJustiFy(lefttext,centertext);
      SetFillStyle(SolidFill,Hom1);
      bar(GetMaxX-140,110,GetMaxX-135,115);
      rectangle(GetMaxX-140,110,GetMaxX-135,115);
      OutTextXY(GetMaxX-130,112,'HOM1');
      SetFillStyle(SolidFill,Hom2);
      bar(GetMaxX-140,120,GetMaxX-135,125);
      rectangle(GetMaxX-140,120,GetMaxX-135,125);
      OutTextXY(GetMaxX-130,122,'HOM2');
      SetFillStyle(SolidFill,Het);
      bar(GetMaxX-140,130,GetMaxX-135,135);
      rectangle(GetMaxX-140,130,GetMaxX-135,135);
      OutTextXY(GetMaxX-130,132,'HET');
      SetTextStyle(SmallFont,HorizDir,4);
      SetTextJustify(CenterText,CenterText);
      OutTextXY(GetMaxX div 2,GetMaxY-15,
      'PRESS ANY KEY FOR NEXT GENERATION, Q TO QUIT.');
end;{Of proc title}

Function DrawField(Newgenos :picfldPtr; countind,countgen :word):char;

var
i,XL,YL      :word;
PopSize      :word;

xx,yy        :word;
s,s1,s2        :string[40];

begin
     str(countind,s1);str(Countgen,s2);
     s:=concat(' GENERATION : ',s2);
     SetFillStyle(SolidFill,PCol);
     SetLineStyle(SolidLn,0,NormWidth);
     SetTextStyle(SmallFont,HorizDir,5);
     SetTextJustiFy(centertext,CenterText);
     bar(100,GetMaxY-50,GetMaxX-100,GetMaxY-33);
     setcolor(15);
     OutTextXY(round(GetMaxX div 2),GetMaxY-40,s);
     i:=1;
     While i<=CountInd do
     begin
          SetFillStyle(SolidFill,colours[NewGenos^[i,3]]);
          YL:=YOri+(NewGenos^[i,1])*ywidth;
          XL:=XOri+(NewGenos^[i,2])*xwidth;
          bar(XL,YL,XL+xwidth-1,YL+ywidth-1);
          i:=i+1;
     end;
     if not print then setcolor(0);
     If DrawGrid
     then for xx:=0 to n do
     begin
          if (xx mod a2)=0 then
          line(XOri+xx*xwidth-1,YOri-1,XOri+xx*xwidth-1,YOri+yscale-1);
          if (xx mod a1)=0 then
          line(XOri-1,YOri+xx*ywidth-1,XOri+xscale-1,YOri+xx*ywidth-1);
     end
     else rectangle(XOri-1,YOri-1,XOri+xscale,YOri+yscale);
     DrawField:=readkey;
end;
(**************************************************************************)
{1}Begin
        clrscr;
        DrawGrid:=True;
        Graphics:=False;
        Ext1:='RED';
        FileList(Ext1);
        Window2(W_Size_Small);
        Writeln
(' WARNING!! This procedure will only map a reduced field with 2 alleles!');
        writeln
(' If you did not build it, just press return when asked for the file name.');
        writeln
(' Otherwise, allele n 1 will be mapped against all the others!!!!');
        Repeat
        Write('Input one of the above (without ext.) ');
        Write('(Return to Exit.) :');
        readln(Name);
        If name='' then Exit;
        Until (FileExist(Name+'.RED')) {or (FileExist(Name+'.DAT')))};
        REDUC:=true{not Affirmed(False,Graphics)};
        Write ('Input Title (<80 Char) : ');Readln(Titre);
        write('Do you want to print the graph? : ');
        print:=Affirmed(False,Graphics);
        If Not Print
        Then begin
                  Hom1:=3;
                  Hom2:=1;
                  Het:=11;
                  Pcol:=7;
             end
        else begin
                  Hom1:=0;
                  Hom2:=15;
                  Het:=7;
                  Pcol:=0;
             End;
        GetMem(Fld,SizeOf(Fld^));GetMem(Fldb,SizeOf(FldB^));
        GetMem(Changes,SizeOf(Changes^));
        if Reduc
        then begin
                  FilNam1:=Name+'.RED';
                  Fall:=1;
                  assign(FileDat,name);
                  write(' Input Deme size(1 for lattice model): ');
                  readln(SPSide1);
                  SpSide1:=Trunc(Sqrt(SPSide1));
                  SpSide2:=SpSide1;
                  FieldSide:=n div SpSide1;
                  write('Do you want do draw a grid ?(Y/N) : ');
                  DrawGrid:=Affirmed(True,False);
                  A1:=SpSide1;
                  A2:=SpSide2;
                  for i:=1 to maxInd do
                  begin
                       Changes^[i,1]:=0;
                       Changes^[i,2]:=0;
                       Changes^[i,3]:=0;
                  end;
                  assign(File1,Filnam1);Reset(File1);
                  read(File1,Fldb^);
                  CountGen:=FilePos(File1);
                  For i:=1 to maxind do
{2}               begin
                       if Fldb^[i,1]=Fall Then Fld^[i,1]:=1 else Fld^[i,1]:=2;
                       if Fldb^[i,2]=Fall Then Fld^[i,2]:=1 else Fld^[i,2]:=2;
{2}               end;
                  close(File1);
                  For i:=0 to FieldSide-1 do
                  For j:=0 to FieldSide-1 do
                  for ii:=0 to SpSide1-1 do
                  for jj:=0 to SpSide2-1 do
{2}        Begin
                temp:=1+jj
                +ii*SpSide2
                + j*SpSide2*SpSide1
                + i*SpSide2*SpSide1*FieldSide;
                if ((Fld^[Temp,1]=1) xor (Fld^[Temp,2]=1))
                then genos[i*SpSide1+ii,j*SpSide2+jj]:=Het
                else If (Fld^[temp,1]=1)
                     then Genos[i*SpSide1+ii,j*SpSide2+jj]:=Hom1
                     else Genos[i*SpSide1+ii,j*SpSide2+jj]:=Hom2;
                changes^[temp,1]:=i*SpSide1+ii;
                changes^[temp,2]:=j*SpSide2+jj;
                changes^[temp,3]:=genos[i*SpSide1+ii,j*SpSide2+jj];
{2}        end;
           for i:=0 to n-1 do
           begin
                freq[i]:=0;
                for j:=0 to n-1 do
                if genos[i,j]=hom2
                then freq[i]:=freq[i]+2
                else if genos[i,j]=het
                     then freq[i]:=freq[i]+1;
           end;
           Initialize;
           Title(titre);
           choix:=DrawField(changes,maxind,CountGen);
           If Upcase(choix)=#81
           Then Begin
                     RestoreCrtMode;
                     Graphics:=False;
                     exit;
                End;
           reset(File1);
           Seek(File1,1);
           While not Eof(File1) Do
{2}        Begin
                countchange:=0;
                read(File1,Fldb^);
                CountGen:=FilePos(File1);
                For i:=1 to maxind do
  {3}           begin
                     if Fldb^[i,1]=Fall Then Fld^[i,1]:=1 else Fld^[i,1]:=2;
                     if Fldb^[i,2]=Fall Then Fld^[i,2]:=1 else Fld^[i,2]:=2;
  {3}           end;
                For i:=0 to FieldSide-1 do
                For j:=0 to FieldSide-1 do
                For ii:=0 to SpSide1-1 do
```

```
                For jj:=0 to SpSide2-1 do
{3}             Begin
                     temp:=1+jj
                     +ii*SpSide2
                     + j*SpSide2*SpSide1
                     + i*SpSide2*SpSide1*FieldSide;
                     If (Fld^[temp,1]=Fld^[temp,2])
                     then Begin
                               If ((Fld^[temp,1]=1)
                                  and (genos[i*SpSide1+ii,j*SpSide2+jj]
                                  <>Hom1))
                               then Begin
                                         CountChange:=CountChange+1;
                                         Changes^[CountChange,1]:=i*SpSide1+ii;
                                         Changes^[CountChange,2]:=j*SpSide2+jj;
                                         Changes^[CountChange,3]:=Hom1;
                                    end;
                               If ((Fld^[temp,1]=2)
                                  and (genos[i*SpSide1+ii,j*SpSide2+jj]
                                  <>Hom2))
                               then begin
                                         CountChange:=CountChange+1;
                                         Changes^[CountChange,1]:=i*SpSide1+ii;
                                         Changes^[CountChange,2]:=j*SpSide2+jj;
                                         Changes^[CountChange,3]:=Hom2;
                                     end;
                          end
                     else If ((Fld^[Temp,1]<>Fld^[Temp,2])
                             and (Genos[i*SpSide1+ii,j*SpSide2+jj]
                             <>Het))
                          then begin
                                    CountChange:=CountChange+1;
                                    Changes^[CountChange,1]:=i*SpSide1+ii;
                                    Changes^[CountChange,2]:=j*SpSide2+jj;
                                    Changes^[CountChange,3]:=Het;
                                end;
{3}             end;{Of ij loop}
                for i:=1 to countchange do
                genos[changes^[i,1],changes^[i,2]]:=changes^[i,3];
                for i:=0 to n-1 do
                begin
                     freq[i]:=0;
                     for j:=0 to n-1 do
                     if genos[i,j]=hom2
                     then freq[i]:=freq[i]+2
                     else if genos[i,j]=het
                          then freq[i]:=freq[i]+1;
                end;
                choix:=DrawField(changes,countchange,CountGen);
                If Upcase(choix)=#81
                Then Begin
                          RestoreCrtMode;
                          Graphics:=False;
                          close(file1);
                          exit;
                     End;
{2}          End;{While not Eof(File1)}
         end;
         FreeMem(Changes,SizeOf(Changes^));
         FreeMem(Fld,SizeOF(Fld^));FreeMem(FldB,SizeOf(FldB^));
         close(file1);
         Message_End;
{1}End;
```

Appendix B

Effects of number of samples & number of individuals per sample

Island model, $m = 0.01$, $s = 0.0$, 25th generation.

Analysis of Variance for $\hat{F}_{is}$

Source	*DF*	*SS*	*MS*	*F*	*P*	R^2
nsp	3	0.003154	0.001051	1.01	0.391	0.467
nind	3	0.000634	0.000211	0.15	0.931	
$nsp * nind$	9	0.002648	0.000294	0.25	0.986	
rep	49	0.077907	0.001590	$*$		
$nsp * rep$	147	0.153260	0.001043	0.90	0.769	
$nind * rep$	147	0.209261	0.001424	1.23	0.055	
Error	441	0.509644	0.001156			
Total	799	0.956508				

Analysis of Variance for θ

Source	*DF*	*SS*	*MS*	*F*	*P*	R^2
nsp	3	0.0005019	0.0001673	0.59	0.622	0.497
nind	3	0.0007509	0.0002503	0.61	0.609	
$nsp * nind$	9	0.0019107	0.0002123	0.63	0.775	
rep	49	0.0425860	0.0008691	$*$		
$nsp * rep$	147	0.0416334	0.0002832	0.84	0.902	
$nind * rep$	147	0.0602207	0.0004097	1.21	0.074	
Error	441	0.1495081	0.0003390			
Total	799	0.2971117				

Analysis of Variance for G_{st}

Source	*DF*	*SS*	*MS*	*F*	*P*	R^2
nsp	3	0.0580782	0.0193594	85.63	0.000	0.599
nind	3	0.0006707	0.0002236	0.68	0.563	
$nsp * nind$	9	0.0015447	0.0001716	0.64	0.764	
rep	49	0.0356708	0.0007280	$*$		
$nsp * rep$	147	0.0332347	0.0002261	0.84	0.892	
$nind * rep$	147	0.0480452	0.0003268	1.22	0.067	
Error	441	0.1184957	0.0002687			
Total	799	0.2957400				

Island model, $m = 0.01$, $s = 0.9$, 25th generation.

Analysis of Variance for $\hat{F_{is}}$

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.0007322	0.0002441	0.79	0.502	0.493
$nind$	3	0.0002148	0.0000716	0.22	0.881	
$nsp * nind$	9	0.0041677	0.0004631	1.66	0.097	
rep	49	0.0218950	0.0004468	*		
$nsp * rep$	147	0.0454585	0.0003092	1.11	0.217	
$nind * rep$	147	0.0473871	0.0003224	1.15	0.136	
$Error$	441	0.1231933	0.0002793			
$Total$	799	0.2430487				

Analysis of Variance for θ

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.0001908	0.0000636	1.42	0.240	0.520
$nind$	3	0.0000338	0.0000113	0.30	0.827	
$nsp * nind$	9	0.0017721	0.0001969	5.28	0.000	
rep	49	0.0036807	0.0000751	*		
$nsp * rep$	147	0.0065878	0.0000448	1.20	0.079	
$nind * rep$	147	0.0055704	0.0000379	1.02	0.442	
$Error$	441	0.0164354	0.0000373			
$Total$	799	0.0342710				

Analysis of Variance for G_{st}

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.0041815	0.0013938	40.73	0.000	0.587
$nind$	3	0.0000201	0.0000067	0.23	0.876	
$nsp * nind$	9	0.0013746	0.0001527	5.38	0.000	
rep	49	0.0028792	0.0000588	*		
$nsp * rep$	147	0.0050307	0.0000342	1.21	0.075	
$nind * rep$	147	0.0042956	0.0000292	1.03	0.404	
$Error$	441	0.0125086	0.0000284			
$Total$	799	0.0302903				

Island model, $m = 0.1$, $s = 0.0$, 25th generation.

Analysis of Variance for $\hat{F_{is}}$

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.000173	0.000058	0.04	0.988	0.447
$nind$	3	0.008703	0.002901	2.39	0.071	
$nsp * nind$	9	0.013732	0.001526	1.08	0.377	
rep	49	0.112286	0.002292	*		
$nsp * rep$	147	0.191729	0.001304	0.92	0.718	
$nind * rep$	147	0.178288	0.001213	0.86	0.865	
$Error$	441	0.623820	0.001415			
$Total$	799	1.128731				

Analysis of Variance for θ

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.0025341	0.0008447	4.83	0.003	0.490
$nind$	3	0.0007483	0.0002494	1.00	0.393	
$nsp * nind$	9	0.0059394	0.0006599	3.29	0.001	
rep	49	0.0134302	0.0002741	*		
$nsp * rep$	147	0.0257211	0.0001750	0.87	0.836	
$nind * rep$	147	0.0365568	0.0002487	1.24	0.050	
$Error$	441	0.0884503	0.0002006			
$Total$	799	0.1733801				

Analysis of Variance for G_{st}

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.0199270	0.0066423	47.47	0.000	0.554
$nind$	3	0.0004122	0.0001374	0.70	0.555	
$nsp * nind$	9	0.0041490	0.0004610	2.96	0.002	
rep	49	0.0110689	0.0002259	*		
$nsp * rep$	147	0.0205701	0.0001399	0.90	0.775	
$nind * rep$	147	0.0289360	0.0001968	1.27	0.036	
$Error$	441	0.0685836	0.0001555			
$Total$	799	0.1536468				

Island model, $m = 0.1$, $s = 0.9$, 25**th generation.**

Analysis of Variance for $\hat{F_{is}}$

Source	*DF*	*SS*	*MS*	*F*	*P*	R^2
nsp	3	0.005418	0.001806	1.09	0.355	0.434
nind	3	0.001614	0.000538	0.26	0.852	
nsp ∗ nind	9	0.012895	0.001433	0.73	0.685	
rep	49	0.102261	0.002087	∗		
nsp ∗ rep	147	0.243295	0.001655	0.84	0.897	
nind ∗ rep	147	0.301282	0.002050	1.04	0.382	
Error	441	0.870647	0.001974			
Total	799	1.537413				

Analysis of Variance for θ

Source	*DF*	*SS*	*MS*	*F*	*P*	R^2
nsp	3	0.000474	0.000158	0.11	0.955	0.470
nind	3	0.000919	0.000306	0.20	0.894	
nsp ∗ nind	9	0.005129	0.000570	0.38	0.946	
rep	49	0.151878	0.003100	∗		
nsp ∗ rep	147	0.212513	0.001446	0.96	0.618	
nind ∗ rep	147	0.220807	0.001502	0.99	0.507	
Error	441	0.666050	0.001510			
Total	799	1.257770				

Analysis of Variance for G_{st}

Source	*DF*	*SS*	*MS*	*F*	*P*	R^2
nsp	3	0.131069	0.043690	36.22	0.000	0.528
nind	3	0.000572	0.000191	0.15	0.929	
nsp ∗ nind	9	0.004021	0.000447	0.35	0.958	
rep	49	0.133018	0.002715	∗		
nsp ∗ rep	147	0.177325	0.001206	0.94	0.662	
nind ∗ rep	147	0.186370	0.001268	0.99	0.520	
Error	441	0.564583	0.001280			
Total	799	1.196958				

Island model, $m = 0.01$, $s = 0.0$, 150**th generation.**

Analysis of Variance for $\hat{F_{is}}$

Source	*DF*	*SS*	*MS*	*F*	*P*	R^2
nsp	3	0.009739	0.003246	1.48	0.221	0.523
nind	3	0.000566	0.000189	0.08	0.973	
nsp ∗ nind	9	0.016045	0.001783	0.93	0.497	
rep	49	0.214204	0.004372	∗		
nsp ∗ rep	147	0.321601	0.002188	1.14	0.152	
nind ∗ rep	147	0.361902	0.002462	1.29	0.027	
Error	441	0.843927	0.001914			
Total	799	1.767984				

Analysis of Variance for θ

Source	*DF*	*SS*	*MS*	*F*	*P*	R^2
nsp	3	0.001801	0.000600	0.37	0.774	0.466
nind	3	0.001001	0.000334	0.21	0.889	
nsp ∗ nind	9	0.011431	0.001270	0.67	0.741	
rep	49	0.248972	0.005081	∗		
nsp ∗ rep	147	0.237420	0.001615	0.85	0.885	
nind ∗ rep	147	0.233683	0.001590	0.83	0.906	
Error	441	0.842168	0.001910			
Total	799	1.576476				

Analysis of Variance for G_{st}

Source	*DF*	*SS*	*MS*	*F*	*P*	R^2
nsp	3	0.130200	0.043400	31.44	0.000	0.515
nind	3	0.000777	0.000259	0.19	0.902	
nsp ∗ nind	9	0.009927	0.001103	0.67	0.733	
rep	49	0.222666	0.004544	∗		
nsp ∗ rep	147	0.202925	0.001380	0.84	0.889	
nind ∗ rep	147	0.199361	0.001356	0.83	0.912	
Error	441	0.721903	0.001637			
Total	799	1.487757				

Island model, $m = 0.01$, $s = 0.9$, 150th generation.

Analysis of Variance for $\hat{F_{is}}$

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.020299	0.006766	3.06	0.030	0.452
$nind$	3	0.003256	0.001085	0.45	0.720	
$nsp * nind$	9	0.029542	0.003282	1.25	0.262	
rep	49	0.220294	0.004496	*		
$nsp * rep$	147	0.325201	0.002212	0.84	0.891	
$nind * rep$	147	0.356786	0.002427	0.92	0.711	
$Error$	441	1.157906	0.002626			
$Total$	799	2.113284				

Analysis of Variance for θ

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.021257	0.007086	1.53	0.209	0.551
$nind$	3	0.007505	0.002502	0.55	0.647	
$nsp * nind$	9	0.019666	0.002185	0.55	0.834	
rep	49	0.740714	0.015117	*		
$nsp * rep$	147	0.680023	0.004626	1.17	0.110	
$nind * rep$	147	0.665752	0.004529	1.15	0.143	
$Error$	441	1.737598	0.003940			
$Total$	799	3.872514				

Analysis of Variance for G_{st}

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.134474	0.044825	10.27	0.000	0.567
$nind$	3	0.008420	0.002807	0.65	0.583	
$nsp * nind$	9	0.020528	0.002281	0.61	0.786	
rep	49	0.708612	0.014461	*		
$nsp * rep$	147	0.641636	0.004365	1.17	0.111	
$nind * rep$	147	0.632287	0.004301	1.16	0.133	
$Error$	441	1.640922	0.003721			
$Total$	799	3.786880				

Island model, $m = 0.1$, $s = 0.0$, 150th generation.

Analysis of Variance for $\hat{F_{is}}$

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.0007358	0.0002453	0.49	0.692	0.481
$nind$	3	0.0008862	0.0002954	0.65	0.583	
$nsp * nind$	9	0.0030408	0.0003379	0.74	0.676	
rep	49	0.0424588	0.0008665	*		
$nsp * rep$	147	0.0741282	0.0005043	1.10	0.234	
$nind * rep$	147	0.0666915	0.0004537	0.99	0.525	
$Error$	441	0.2024102	0.0004590			
$Total$	799	0.3903517				

Analysis of Variance for θ

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.0003442	0.0001147	1.87	0.137	0.490
$nind$	3	0.0001397	0.0000466	0.78	0.506	
$nsp * nind$	9	0.0005833	0.0000648	1.23	0.276	
rep	49	0.0035752	0.0000730	*		
$nsp * rep$	147	0.0090137	0.0000613	1.16	0.127	
$nind * rep$	147	0.0087517	0.0000595	1.13	0.180	
$Error$	441	0.0233000	0.0000528			
$Total$	799	0.0457080				

Analysis of Variance for G_{st}

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.0052936	0.0017645	38.08	0.000	0.557
$nind$	3	0.0000914	0.0000305	0.67	0.570	
$nsp * nind$	9	0.0004158	0.0000462	1.16	0.320	
rep	49	0.0028325	0.0000578	*		
$nsp * rep$	147	0.0068109	0.0000463	1.16	0.125	
$nind * rep$	147	0.0066571	0.0000453	1.14	0.165	
$Error$	441	0.0175880	0.0000399			
$Total$	799	0.0396892				

Island model, $m = 0.1$, $s = 0.9$, 150th generation.

Analysis of Variance for $\hat{F}_{is}$

Source	*DF*	*SS*	*MS*	*F*	*P*	R^2
nsp	3	0.003271	0.001090	0.65	0.581	0.484
nind	3	0.001802	0.000601	0.43	0.729	
$nsp * nind$	9	0.019724	0.002192	1.58	0.119	
rep	49	0.101976	0.002081	*		
$nsp * rep$	147	0.244760	0.001665	1.20	0.082	
$nind * rep$	147	0.203304	0.001383	1.00	0.503	
Error	441	0.612401	0.001389			
Total	799	1.187238				

Analysis of Variance for θ

Source	*DF*	*SS*	*MS*	*F*	*P*	R^2
nsp	3	0.0011029	0.0003676	1.02	0.386	0.431
nind	3	0.0015081	0.0005027	1.58	0.197	
$nsp * nind$	9	0.0032322	0.0003591	0.98	0.452	
rep	49	0.0164057	0.0003348	*		
$nsp * rep$	147	0.0529951	0.0003605	0.99	0.526	
$nind * rep$	147	0.0467987	0.0003184	0.87	0.836	
Error	441	0.1608853	0.0003648			
Total	799	0.2829281				

Analysis of Variance for G_{st}

Source	*DF*	*SS*	*MS*	*F*	*P*	R^2
nsp	3	0.0070783	0.0023594	8.42	0.000	0.449
nind	3	0.0011176	0.0003725	1.56	0.203	
$nsp * nind$	9	0.0024270	0.0002697	0.97	0.466	
rep	49	0.0130393	0.0002661	*		
$nsp * rep$	147	0.0411790	0.0002801	1.01	0.476	
$nind * rep$	147	0.0351854	0.0002394	0.86	0.862	
Error	441	0.1228902	0.0002787			
Total	799	0.2229167				

Stepping-stone model, $m = 0.01$, $s = 0.0$, 25th generation.

Analysis of Variance for $\hat{F}_{is}$

Source	*DF*	*SS*	*MS*	*F*	*P*	R^2
nsp	3	0.004569	0.001523	1.25	0.294	0.478
nind	3	0.002919	0.000973	0.74	0.529	
$nsp * nind$	9	0.018635	0.002071	1.73	0.080	
rep	49	0.084744	0.001729	*		
$nsp * rep$	147	0.179036	0.001218	1.02	0.441	
$nind * rep$	147	0.193033	0.001313	1.10	0.238	
Error	441	0.528000	0.001197			
Total	799	1.010938				

Analysis of Variance for θ

Source	*DF*	*SS*	*MS*	*F*	*P*	R^2
nsp	3	0.0010589	0.0003530	0.84	0.474	0.528
nind	3	0.0014896	0.0004965	1.30	0.277	
$nsp * nind$	9	0.0026931	0.0002992	0.86	0.562	
rep	49	0.0485340	0.0009905	*		
$nsp * rep$	147	0.0617214	0.0004199	1.21	0.076	
$nind * rep$	147	0.0561173	0.0003818	1.10	0.238	
Error	441	0.1535082	0.0003481			
Total	799	0.3251225				

Analysis of Variance for G_{st}

Source	*DF*	*SS*	*MS*	*F*	*P*	R^2
nsp	3	0.0461157	0.0153719	46.66	0.000	0.599
nind	3	0.0011562	0.0003854	1.26	0.291	
$nsp * nind$	9	0.0020648	0.0002294	0.83	0.593	
rep	49	0.0405598	0.0008278	*		
$nsp * rep$	147	0.0484330	0.0003295	1.19	0.096	
$nind * rep$	147	0.0450401	0.0003064	1.10	0.226	
Error	441	0.1225461	0.0002779			
Total	799	0.3059157				

Stepping-stone model, $m = 0.01$, $s = 0.9$, **25th generation.**

Analysis of Variance for $\hat{F_{is}}$

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.0002767	0.0000922	0.20	0.897	0.461
$nind$	3	0.0001936	0.0000645	0.14	0.933	
$nsp * nind$	9	0.0009796	0.0001088	0.26	0.986	
rep	49	0.0253033	0.0005164	$*$		
$nsp * rep$	147	0.0681866	0.0004639	1.09	0.254	
$nind * rep$	147	0.0656879	0.0004469	1.05	0.352	
$Error$	441	0.1878078	0.0004259			
$Total$	799	0.3484357				

Analysis of Variance for θ

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.0003651	0.0001217	1.63	0.184	0.480
$nind$	3	0.0001606	0.0000535	0.77	0.513	
$nsp * nind$	9	0.0004545	0.0000505	0.69	0.716	
rep	49	0.0075202	0.0001535	$*$		
$nsp * rep$	147	0.0109446	0.0000745	1.02	0.429	
$nind * rep$	147	0.0102387	0.0000697	0.96	0.623	
$Error$	441	0.0321466	0.0000729			
$Total$	799	0.0618304				

Analysis of Variance for G_{st}

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.0044860	0.0014953	28.33	0.000	0.521
$nind$	3	0.0001142	0.0000381	0.74	0.528	
$nsp * nind$	9	0.0003307	0.0000367	0.68	0.727	
rep	49	0.0056891	0.0001161	$*$		
$nsp * rep$	147	0.0077594	0.0000528	0.98	0.559	
$nind * rep$	147	0.0075287	0.0000512	0.95	0.645	
$Error$	441	0.0238214	0.0000540			
$Total$	799	0.0497294				

Stepping-stone model, $m = 0.1$, $s = 0.0$, **25th generation.**

Analysis of Variance for $\hat{F_{is}}$

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.001058	0.000353	0.30	0.828	0.411
$nind$	3	0.000303	0.000101	0.09	0.968	
$nsp * nind$	9	0.015491	0.001721	1.17	0.310	
rep	49	0.085513	0.001745	$*$		
$nsp * rep$	147	0.175255	0.001192	0.81	0.931	
$nind * rep$	147	0.173397	0.001180	0.80	0.941	
$Error$	441	0.646492	0.001466			
$Total$	799	1.097508				

Analysis of Variance for θ

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.0010919	0.0003640	1.38	0.252	0.493
$nind$	3	0.0003366	0.0001122	0.28	0.838	
$nsp * nind$	9	0.0097903	0.0010878	3.43	0.000	
rep	49	0.0278828	0.0005690	$*$		
$nsp * rep$	147	0.0388006	0.0002639	0.83	0.907	
$nind * rep$	147	0.0584307	0.0003975	1.25	0.043	
$Error$	441	0.1399656	0.0003174			
$Total$	799	0.2762983				

Analysis of Variance for G_{st}

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.0131314	0.0043771	21.78	0.000	0.525
$nind$	3	0.0002189	0.0000730	0.23	0.874	
$nsp * nind$	9	0.0076666	0.0008518	3.48	0.000	
rep	49	0.0224738	0.0004586	$*$		
$nsp * rep$	147	0.0295493	0.0002010	0.82	0.922	
$nind * rep$	147	0.0462809	0.0003148	1.29	0.027	
$Error$	441	0.1080160	0.0002449			
$Total$	799	0.2273369				

Stepping-stone model, $m = 0.1$, $s = 0.9$, **25th generation.**

Analysis of Variance for $\hat{F}_{is}$

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.001647	0.000549	0.36	0.782	0.453
$nind$	3	0.004737	0.001579	0.83	0.478	
$nsp * nind$	9	0.042001	0.004667	2.64	0.006	
rep	49	0.095401	0.001947	*		
$nsp * rep$	147	0.224263	0.001526	0.86	0.856	
$nind * rep$	147	0.278783	0.001896	1.07	0.294	
$Error$	441	0.780145	0.001769			
$Total$	799	1.426977				

Analysis of Variance for θ

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.004517	0.001506	1.19	0.314	0.525
$nind$	3	0.000824	0.000275	0.21	0.890	
$nsp * nind$	9	0.006153	0.000684	0.60	0.797	
rep	49	0.165001	0.003367	*		
$nsp * rep$	147	0.185224	0.001260	1.11	0.219	
$nind * rep$	147	0.193332	0.001315	1.15	0.136	
$Error$	441	0.502508	0.001139			
$Total$	799	1.057559				

Analysis of Variance for G_{st}

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.1565939	0.0521980	49.37	0.000	0.596
$nind$	3	0.0007079	0.0002360	0.21	0.888	
$nsp * nind$	9	0.0055556	0.0006173	0.64	0.762	
rep	49	0.1448862	0.0029569	*		
$nsp * rep$	147	0.1554191	0.0010573	1.10	0.234	
$nind * rep$	147	0.1633317	0.0011111	1.15	0.135	
$Error$	441	0.4243199	0.0009622			
$Total$	799	1.0508143				

Stepping-stone model, $m = 0.01$, $s = 0.0$, **150th generation.**

Analysis of Variance for $\hat{F}_{is}$

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.004784	0.001595	0.48	0.698	0.446
$nind$	3	0.005075	0.001692	0.47	0.706	
$nsp * nind$	9	0.010515	0.001168	0.34	0.961	
rep	49	0.168285	0.003434	*		
$nsp * rep$	147	0.490153	0.003334	0.98	0.564	
$nind * rep$	147	0.532894	0.003625	1.06	0.323	
$Error$	441	1.507597	0.003419			
$Total$	799	2.719304				

Analysis of Variance for θ

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.003742	0.001247	0.55	0.650	0.536
$nind$	3	0.003436	0.001145	0.66	0.575	
$nsp * nind$	9	0.011335	0.001259	0.67	0.739	
rep	49	0.356096	0.007267	*		
$nsp * rep$	147	0.334595	0.002276	1.20	0.077	
$nind * rep$	147	0.253299	0.001723	0.91	0.744	
$Error$	441	0.833113	0.001889			
$Total$	799	1.795617				

Analysis of Variance for G_{st}

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.143903	0.047968	23.98	0.000	0.576
$nind$	3	0.003209	0.001070	0.70	0.552	
$nsp * nind$	9	0.010584	0.001176	0.70	0.705	
rep	49	0.323607	0.006604	*		
$nsp * rep$	147	0.294004	0.002000	1.20	0.084	
$nind * rep$	147	0.223844	0.001523	0.91	0.744	
$Error$	441	0.736266	0.001670			
$Total$	799	1.735418				

Stepping-stone model, $m = 0.01$, $s = 0.9$, 150th generation.

Analysis of Variance for $\hat{F}_{is}$

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.011867	0.003956	1.54	0.206	0.461
$nind$	3	0.024544	0.008181	3.65	0.014	
$nsp * nind$	9	0.057424	0.006380	2.49	0.009	
rep	49	0.163633	0.003339	$*$		
$nsp * rep$	147	0.377441	0.002568	1.00	0.481	
$nind * rep$	147	0.329396	0.002241	0.88	0.829	
$Error$	441	1.128323	0.002559			
$Total$	799	2.092628				

Analysis of Variance for θ

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.003937	0.001312	0.33	0.806	0.477
$nind$	3	0.011150	0.003717	0.82	0.485	
$nsp * nind$	9	0.049419	0.005491	1.12	0.349	
rep	49	0.653025	0.013327	$*$		
$nsp * rep$	147	0.591028	0.004021	0.82	0.925	
$nind * rep$	147	0.666330	0.004533	0.92	0.717	
$Error$	441	2.167547	0.004915			
$Total$	799	4.142436				

Analysis of Variance for G_{st}

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.213695	0.071232	18.11	0.000	0.503
$nind$	3	0.011557	0.003852	0.87	0.458	
$nsp * nind$	9	0.048078	0.005342	1.11	0.354	
rep	49	0.640403	0.013069	$*$		
$nsp * rep$	147	0.578101	0.003933	0.82	0.926	
$nind * rep$	147	0.651070	0.004429	0.92	0.721	
$Error$	441	2.121218	0.004810			
$Total$	799	4.264122				

Stepping-stone model, $m = 0.1$, $s = 0.0$, 150th generation.

Analysis of Variance for $\hat{F}_{is}$

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.0021716	0.0007239	1.38	0.252	0.493
$nind$	3	0.0036734	0.0012245	2.05	0.109	
$nsp * nind$	9	0.0042370	0.0004708	0.89	0.537	
rep	49	0.0521841	0.0010650	$*$		
$nsp * rep$	147	0.0773349	0.0005261	0.99	0.518	
$nind * rep$	147	0.0876370	0.0005962	1.12	0.187	
$Error$	441	0.2341281	0.0005309			
$Total$	799	0.4613662				

Analysis of Variance for θ

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.0007740	0.0002580	2.05	0.109	0.457
$nind$	3	0.0008260	0.0002753	2.39	0.071	
$nsp * nind$	9	0.0003786	0.0000421	0.33	0.964	
rep	49	0.0096560	0.0001971	$*$		
$nsp * rep$	147	0.0184713	0.0001257	0.99	0.516	
$nind * rep$	147	0.0169521	0.0001153	0.91	0.749	
$Error$	441	0.0558845	0.0001267			
$Total$	799	0.1029425				

Analysis of Variance for G_{st}

$Source$	DF	SS	MS	F	P	R^2
nsp	3	0.0056152	0.0018717	19.87	0.000	0.493
$nind$	3	0.0006572	0.0002191	2.60	0.054	
$nsp * nind$	9	0.0002611	0.0000290	0.31	0.972	
rep	49	0.0075854	0.0001548	$*$		
$nsp * rep$	147	0.0138493	0.0000942	1.00	0.488	
$nind * rep$	147	0.0123810	0.0000842	0.89	0.787	
$Error$	441	0.0415104	0.0000941			
$Total$	799	0.0818596				

Stepping-stone model, $m = 0.1$, $s = 0.9$, **150th generation.**

Analysis of Variance for $\hat{F_{is}}$

Source	*DF*	*SS*	*MS*	*F*	*P*	R^2
nsp	3	0.001053	0.000351	0.31	0.822	0.431
nind	3	0.008610	0.002870	1.98	0.119	
$nsp * nind$	9	0.006357	0.000706	0.50	0.875	
rep	49	0.073888	0.001508	*		
$nsp * rep$	147	0.169054	0.001150	0.81	0.931	
$nind * rep$	147	0.212716	0.001447	1.02	0.422	
Error	441	0.623336	0.001413			
Total	799	1.095014				

Analysis of Variance for θ

Source	*DF*	*SS*	*MS*	*F*	*P*	R^2
nsp	3	0.0017932	0.0005977	1.59	0.194	0.455
nind	3	0.0011546	0.0003849	0.77	0.512	
$nsp * nind$	9	0.0097642	0.0010849	2.07	0.031	
rep	49	0.0519594	0.0010604	*		
$nsp * rep$	147	0.0551509	0.0003752	0.72	0.992	
$nind * rep$	147	0.0734498	0.0004997	0.95	0.631	
Error	441	0.2312758	0.0005244			
Total	799	0.4245477				

Analysis of Variance for G_{st}

Source	*DF*	*SS*	*MS*	*F*	*P*	R^2
nsp	3	0.0320387	0.0106796	35.68	0.000	0.509
nind	3	0.0008884	0.0002961	0.76	0.521	
$nsp * nind$	9	0.0076247	0.0008472	2.08	0.030	
rep	49	0.0436529	0.0008909	*		
$nsp * rep$	147	0.0440004	0.0002993	0.74	0.986	
$nind * rep$	147	0.0575860	0.0003917	0.96	0.602	
Error	441	0.1794744	0.0004070			
Total	799	0.3652654				

Appendix C

MINITAB macro for two way Kruskall-Wallis

Dependant variable in column `C15`. Independant variables in columns `C2` and `C3`.

```
noecho
rank c15 c16
let c60=c2*10+c3
unstack c16 into c20 c21 c22 c23 c24 c25 c26 c27 c28 c29 c30 c31 c32 c33 c34 c35;
subscripts c60.
let k20=sum(c20)
let k21=sum(c21)
let k22=sum(c22)
let k23=sum(c23)
let k24=sum(c24)
let k25=sum(c25)
let k26=sum(c26)
let k27=sum(c27)
let k28=sum(c28)
let k29=sum(c29)
let k30=sum(c30)
let k31=sum(c31)
let k32=sum(c32)
let k33=sum(c33)
let k34=sum(c34)
let k35=sum(c35)
let k120=k20**2
let k121=k21**2
let k122=k22**2
let k123=k23**2
let k124=k24**2
let k125=k25**2
let k126=k26**2
let k127=k27**2
let k128=k28**2
let k129=k29**2
let k130=k30**2
let k131=k31**2
let k132=k32**2
let k133=k33**2
let k134=k34**2
let k135=k35**2
let k36=4
let k37=4
let k38=count(c20)
let k40=count(c15)
let k41=k40*(k40+1)*(k40+1)/4
let k42=k40*(k40+1)/12
let k43=(k120+k121+k122+k123+k124+k125+k126+k127+k128+k129+k130+k131+k132+k133+k134+k135)/k38-k41
let k220=(k20+k21+k22+k23)**2+(k24+k25+k26+k27)**2+(k28+k29+k30+k31)**2+(k32+k33+k34+k35)**2
let k44=k220/k38/k36-k41
let k221=(k20+k24+k28+k32)**2+(k21+k25+k29+k33)**2+(k22+k26+k30+k34)**2+(k23+k27+k31+k35)**2
let k45=k221/k38/k37-k41
let k46=k43-k44-k45
let k50=k44/k42
let k51=k45/k42
let k52=k46/k42
let k53=k43/k42
let k60=k36-1
let k61=k37-1
let k62=(k36-1)*(k37-1)
let k63=k36*k37-1
note k50 k51 k52 k53 contains kw statistics for treat1 (nsp) treat2 (nind)
note treat1*treat2 and treat with df in k60 k61 k62 k63 respectivly
print k50 k51 k52 k53
print k60 k61 k62 k63
cdf k50 into k200;
chisquare with k60.
let k201=1.0-k200
cdf k51 into k202;
chisquare with k61.
let k203=1.0-k202
cdf k52 into k204;
chisquare with k62.
let k205=1.0-k204
cdf k53 into k206;
chisquare with k63.
let k207=1.0-k206
note k201 k203 k205 and k207 contain P-values for treat1 treat2
note interaction and treatment respectively
print k201 k203 k205 k207
erase c15-c100
erase k1-k230
echo
```

Appendix D

FSTAT.C

```
#include <p2c/p2c.h>


#define npmax           1100
#define nlmax           20
#define numax           9
#define maxboot         5000
#define maxind          5000
#define modulo          10

typedef double frq[nlmax][numax];
typedef double fst[nlmax];
typedef float peral[nlmax][numax];
typedef float (*ptr_to_peral[npmax])[numax];
typedef float sorted[maxboot];
typedef unsigned short numb_per_loc[npmax][nlmax];

Static unsigned short anl, anu, np, nl, nu, ia, iu, iv, ip, il;
Static unsigned short (*an)[nlmax];
Static unsigned short anp[nlmax];
Static unsigned short npinf, npsup;
Static ptr_to_peral h, p;
Static double *anbar, *annbar, *anc, *term1, *term2, *term3, *term4, *term5;
Static double (*pbar)[numax], (*ppbar)[numax], (*varp)[numax], (*hbar)[numax],
      (*capf)[numax], (*theta)[numax], (*smallf)[numax];
Static double *capfl, *thetal, *smallfl;
Static float *capfb, *thetb, *smlfb;
Static FILE *filein, *fileout1, *fileout2, *fileboot1, *fileboot2, *fileboot3;
Static Char locname[nlmax][9];
Static double tterm1, tterm2, tterm3, tterm4, tterm5, a, b, c, tcapf, ttheta,
      tsmallf;
Static double real_f[3];
Static long s1, s2, pass;
Static FILE *fileini, *filepar, *filemig;
Static unsigned short numbperm;
Static Char name[256], namedat[256], nameout[256];
Static Char nametmp[256], nameboot1[256], nameboot2[256], nameboot3[256];
Static unsigned short stepi;
Static Char ans;
Static Char filein_NAME[_FNSIZE];
Static Char fileout1_NAME[_FNSIZE];
Static Char fileout2_NAME[_FNSIZE];
Static Char fileini_NAME[_FNSIZE];
Static Char filepar_NAME[_FNSIZE];
Static Char filemig_NAME[_FNSIZE];
Static Char fileboot1_NAME[_FNSIZE];
Static Char fileboot2_NAME[_FNSIZE];
Static Char fileboot3_NAME[_FNSIZE];

/****************** uniform func*********************************************/
Static double uniform()
{
  long z, k;

  k = s1 / 53668L;
  s1 = (s1 - k * 53668L) * 40014L - k * 12211;
  if (s1 < 0)
    s1 += 2147483563L;
  k = s2 / 52774L;
  s2 = (s2 - k * 52774L) * 409692L - k * 3791;
  if (s2 < 0)
    s2 += 2147483399L;
  z = s1 - s2;
  if (z < 1)
    z += 2147483562L;
  return (z * 4.656613e-10);
}  /*of func uniform*/
```

```
Static long grandom(n)
long n;
{
  return ((long)(uniform() * n));
}  /*of func grandom*/
/**************************************************************************/
Static Void readdata(f)
FILE **f;
{
  unsigned short FORLIM;
  Char *TEMP;
  unsigned short FORLIM1, FORLIM2;

  rewind(*f);
  fscanf(*f, "%hd%hd%hd%*[^\n]", &np, &nl, &nu);
  getc(*f);
  if (np <= 1) {
    printf(" only one population.  can not calculate theta. exiting.\n");
    printf(" if you want smallf, create a dummy population fixed for\n");
    printf(" 1 allele. theta and capf will then be meaningless.\n");
    _Escape(1);
  }
  if (np > npmax || nl > nlmax) {
    printf("  too many populations or loci. npmax= %ld nlmax= %ld\n",
   (long)npmax, (long)nlmax);
    printf("  recompile with a higher value for npmax or nlmax.\n");
    printf("  exiting...\n");
    _Escape(1);
  }
  FORLIM = nl;
  for (il = 1; il <= FORLIM; il++) {
    fgets(locname[il - 1], 9, *f);
    TEMP = strchr(locname[il - 1], '\n');
    if (TEMP != NULL)
      *TEMP = 0;
  }
  FORLIM = nl;
  for (il = 1; il <= FORLIM; il++)
    anp[il - 1] = np;
  anl = nl;
  FORLIM = np;
  for (ip = 1; ip <= FORLIM; ip++) {
    FORLIM1 = nl;
    for (il = 1; il <= FORLIM1; il++) {
      an[ip - 1][il - 1] = 0;
      FORLIM2 = nu;
      for (iu = 1; iu <= FORLIM2; iu++) {
h[ip - 1][il - 1][iu - 1] = 0.0;
p[ip - 1][il - 1][iu - 1] = 0.0;
      }
    }
  }
  while (!P_eof(*f)) {
    fscanf(*f, "%hd", &ip);
    if (ip != 0) {
      FORLIM = nl;
      for (il = 1; il <= FORLIM; il++) {
fscanf(*f, "%hd", &ia);
iu = ia / modulo;
iv = ia % modulo;
if (ia != 0) {
  an[ip - 1][il - 1]++;
  p[ip - 1][il - 1][iu - 1] += 1.0;
  p[ip - 1][il - 1][iv - 1] += 1.0;
  if (iu != iv) {
    h[ip - 1][il - 1][iu - 1] += 1.0;
    h[ip - 1][il - 1][iv - 1] += 1.0;
  }
}
      }
    }
    fscanf(*f, "%*[^\n]");
    getc(*f);
  }
}  /*of procedure readdata*/
/**************************************************************************/
Static Void basic_stats()
{
  double temp;
  unsigned short FORLIM, FORLIM1, FORLIM2;

  fprintf(fileout1, "      ");
  FORLIM = np;
  for (ip = 1; ip <= FORLIM; ip++)
    fprintf(fileout1, "%6u", ip);
  putchar('\n');
  FORLIM = nl;
  for (il = 1; il <= FORLIM; il++) {
    fprintf(fileout1, "\n    locus: %s\n", locname[il - 1]);
    fprintf(fileout1, "     n");
    FORLIM1 = np;
```

```
    for (ip = 1; ip <= FORLIM1; ip++)
      fprintf(fileout1, "%6u", an[ip - 1][il - 1]);
    putc('\n', fileout1);
    FORLIM1 = nu;
    for (iu = 1; iu <= FORLIM1; iu++) {
      fprintf(fileout1, "p: %3u", iu);
      FORLIM2 = np;
      for (ip = 1; ip <= FORLIM2; ip++) {
if (an[ip - 1][il - 1] != 0)
  fprintf(fileout1, "%6.3f",
  p[ip - 1][il - 1][iu - 1] / 2.0 / an[ip - 1][il - 1]);
else
  fprintf(fileout1, " ?????");
      }
      putc('\n', fileout1);
    }
    putc('\n', fileout1);
    FORLIM1 = nu;
    for (iu = 1; iu <= FORLIM1; iu++) {
      fprintf(fileout1, "ho: %2u", iu);
      FORLIM2 = np;
      for (ip = 1; ip <= FORLIM2; ip++) {
if (an[ip - 1][il - 1] != 0)
  fprintf(fileout1, "%6.3f",
  h[ip - 1][il - 1][iu - 1] / an[ip - 1][il - 1]);
else
  fprintf(fileout1, " ?????");
      }
      putc('\n', fileout1);
    }
    putc('\n', fileout1);
    FORLIM1 = nu;
    for (iu = 1; iu <= FORLIM1; iu++) {
      fprintf(fileout1, "he: %2u", iu);
      FORLIM2 = np;
      for (ip = 1; ip <= FORLIM2; ip++) {
if (an[ip - 1][il - 1] != 0) {
  temp = p[ip - 1][il - 1][iu - 1] / 2.0 / an[ip - 1][il - 1];
  fprintf(fileout1, "%6.3f",
  2.0 * temp * (1.0 - temp) * 2.0 * an[ip - 1]
  [il - 1] / (2.0 * an[ip - 1][il - 1] - 1));
} else
  fprintf(fileout1, " ?????");
      }
      putc('\n', fileout1);
    }
  }
  fflush(fileout1);
  P_ioresult = 0;
}  /*of procedure basic_stats*/
/**************************************************************************/
Static Void calcfstat(inf, sup, ans)
unsigned short inf, sup;
Char ans;
{
  double temp;
  unsigned short FORLIM, FORLIM1, FORLIM2;
  long TEMP1;
  double TEMP2;

  pass++;
  FORLIM = nl;
  for (il = 1; il <= FORLIM; il++)
    anp[il - 1] = sup - inf + 1;
  for (ip = inf; ip <= sup; ip++) {
    FORLIM1 = nl;
    for (il = 1; il <= FORLIM1; il++) {
      FORLIM2 = nu;
      for (iu = 1; iu <= FORLIM2; iu++) {
if (an[ip - 1][il - 1] != 0.0) {
  p[ip - 1][il - 1][iu - 1] /= 2.0 * an[ip - 1][il - 1];
  h[ip - 1][il - 1][iu - 1] /= an[ip - 1][il - 1];
} else {
  p[ip - 1][il - 1][iu - 1] = 0.0;
  h[ip - 1][il - 1][iu - 1] = 0.0;
  if (iu == 1)
    anp[il - 1]--;
}
      }
    }
  }
  FORLIM = nl;
  for (il = 1; il <= FORLIM; il++) {
    anbar[il - 1] = 0.0;
    annbar[il - 1] = 0.0;
    for (ip = inf; ip <= sup; ip++) {
      anbar[il - 1] += an[ip - 1][il - 1];
      TEMP1 = an[ip - 1][il - 1];
      annbar[il - 1] += TEMP1 * TEMP1;
    }
    if (anp[il - 1] > 1 && anbar[il - 1] != 0) {
      anbar[il - 1] /= anp[il - 1];
      anc[il - 1] = anp[il - 1] * anbar[il - 1] -
    annbar[il - 1] / anp[il - 1] / anbar[il - 1];
      anc[il - 1] /= anp[il - 1] - 1.0;
      FORLIM1 = nu;
      for (iu = 1; iu <= FORLIM1; iu++) {
pbar[il - 1][iu - 1] = 0.0;
ppbar[il - 1][iu - 1] = 0.0;
hbar[il - 1][iu - 1] = 0.0;
for (ip = inf; ip <= sup; ip++) {
  pbar[il - 1]
    [iu - 1] += an[ip - 1][il - 1] * p[ip - 1][il - 1][iu - 1];
  TEMP2 = p[ip - 1][il - 1][iu - 1];
  ppbar[il - 1][iu - 1] += an[ip - 1][il - 1] * (TEMP2 * TEMP2);
  hbar[il - 1]
    [iu - 1] += an[ip - 1][il - 1] * h[ip - 1][il - 1][iu - 1];
}
pbar[il - 1]
  [iu - 1] = pbar[il - 1][iu - 1] / anp[il - 1] / anbar[il - 1];
TEMP2 = pbar[il - 1][iu - 1];
varp[il - 1]
  [iu - 1] = ppbar[il - 1]
     [iu - 1] - anp[il - 1] * anbar[il - 1] * (TEMP2 * TEMP2);
varp[il - 1][iu - 1] = varp[il - 1]
       [iu - 1] / (anp[il - 1] - 1.0) / anbar[il - 1];
hbar[il - 1]
  [iu - 1] = hbar[il - 1][iu - 1] / anp[il - 1] / anbar[il - 1];
      }
    }
  }
  if (pass == 1) {
    putc('\n', fileout1);
    FORLIM = nl;
    for (il = 1; il <= FORLIM; il++) {
      fprintf(fileout1, "\n for locus : %s\n", locname[il - 1]);
      FORLIM1 = nu;
      for (iu = 1; iu <= FORLIM1; iu++) {
fprintf(fileout1, "fis: %2u", iu);
FORLIM2 = np;
for (ip = 1; ip <= FORLIM2; ip++) {
  if (an[ip - 1][il - 1] != 0 && p[ip - 1][il - 1][iu - 1] > 0.0001 &&
      p[ip - 1][il - 1][iu - 1] < 0.9999) {
    temp = 4.0 * p[ip - 1][il - 1]
   [iu - 1] * (1 - p[ip - 1][il - 1][iu - 1]) * an[ip - 1]
   [il - 1] / (2.0 * an[ip - 1][il - 1] - 1.0);
    temp = 1 - h[ip - 1][il - 1][iu - 1] / temp;
    fprintf(fileout1, "%8.4f", temp);
  } else
    fprintf(fileout1, " ???????");
}
putc('\n', fileout1);
      }
    }
  }

  tterm1 = 0.0;
  tterm2 = 0.0;
  tterm3 = 0.0;
  tterm4 = 0.0;
  tterm5 = 0.0;
  FORLIM = nl;
  for (il = 1; il <= FORLIM; il++) {
    if (anp[il - 1] > 1 && anbar[il - 1] != 0) {
      term1[il - 1] = 0.0;
      term2[il - 1] = 0.0;
      term3[il - 1] = 0.0;
      term4[il - 1] = 0.0;
      term5[il - 1] = 0.0;
      a = 0.0;
      b = 0.0;
      c = 0.0;
      FORLIM1 = nu;
      for (iu = 1; iu <= FORLIM1; iu++) {
if (pbar[il - 1][iu - 1] > 0.00001 && pbar[il - 1][iu - 1] < 0.99999) {
  a = pbar[il - 1][iu - 1] * (1.0 - pbar[il - 1][iu - 1]);
  a += (1.0 - anp[il - 1]) * varp[il - 1][iu - 1] / anp[il - 1];
  b = a;
  a -= hbar[il - 1][iu - 1] / 4.0;
  a = varp[il - 1][iu - 1] - a / (anbar[il - 1] - 1);
  a = anbar[il - 1] * a / anc[il - 1];
  b += (1.0 - 2.0 * anbar[il - 1]) * hbar[il - 1]
       [iu - 1] / 4.0 / anbar[il - 1];
  b = anbar[il - 1] * b / (anbar[il - 1] - 1.0);
  c = hbar[il - 1][iu - 1] / 2.0;
  capf[il - 1][iu - 1] = (a + b) / (a + b + c);
  theta[il - 1][iu - 1] = a / (a + b + c);
  smallf[il - 1][iu - 1] = b / (b + c);
  term1[il - 1] += a + b;
  term2[il - 1] += a + b + c;
  term3[il - 1] += a;
  term4[il - 1] += b;
```

```
  term5[il - 1] += b + c;
} else {
  capf[il - 1][iu - 1] = 0.0;
  theta[il - 1][iu - 1] = 0.0;
  smallf[il - 1][iu - 1] = 0.0;
}
      }
      if (pass == 1 && term2[il - 1] != 0.0 && term5[il - 1] != 0.0) {
fprintf(fileout1, "\n for locus : %s\n", locname[il - 1]);
fprintf(fileout1, "allele   capf      theta    smallf  \n");
      }
      if (term2[il - 1] != 0.0 && term5[il - 1] != 0.0) {
capfl[il - 1] = term1[il - 1] / term2[il - 1];
thetal[il - 1] = term3[il - 1] / term2[il - 1];
smallfl[il - 1] = term4[il - 1] / term5[il - 1];
      } else {
capfl[il - 1] = 0.0;
thetal[il - 1] = 0.0;
smallfl[il - 1] = 0.0;
      }
      if (pass == 1 && term2[il - 1] != 0.0 && term5[il - 1] != 0.0) {
FORLIM1 = nu;
for (iu = 1; iu <= FORLIM1; iu++)
  fprintf(fileout1, "%5u%10.4f%10.4f%10.4f\n",
  iu, capf[il - 1][iu - 1], theta[il - 1]
  [iu - 1], smallf[il - 1][iu - 1]);
fprintf(fileout1, " all  %9.4f%10.4f%10.4f\n",
capfl[il - 1], thetal[il - 1], smallfl[il - 1]);
      }
      tterm1 += term1[il - 1];
      tterm2 += term2[il - 1];
      tterm3 += term3[il - 1];
      tterm4 += term4[il - 1];
      tterm5 += term5[il - 1];
    }
  }
  tcapf = tterm1 / tterm2;
  ttheta = tterm3 / tterm2;
  tsmallf = tterm4 / tterm5;
  if (pass != 1)
    return;
  fprintf(fileout1, "\n over all loci\n");
  fprintf(fileout1, "          capf      theta    smallf\n");
  fprintf(fileout1, "     %10.4f%10.4f%10.4f\n", tcapf, ttheta, tsmallf);
  fflush(fileout1);
  P_ioresult = 0;
  real_f[0] = tcapf;
  real_f[1] = ttheta;
  real_f[2] = tsmallf;
}  /*of procedure calcfstat*/
/***********************************************************************/
Static Void jack_ov_pop(inf, sup)
unsigned short inf, sup;
{
  double capflj, capfjj, smallflj, smallfjj, thetalj, thetajj, anbarj,
 annbarj, ancj, term1j, term2j, term3j, term4j, term5j, pbarj, ppbarj,
 hbarj, varpj, aj, bj, cj;
  unsigned short anp1, anp2, FORLIM, FORLIM1;
  long TEMP;
  unsigned short FORLIM2;
  double TEMP1;

  fprintf(fileout1, "\njackknifing over populations.\n");
  fflush(fileout1);
  P_ioresult = 0;
  FORLIM = nl;
  for (il = 1; il <= FORLIM; il++) {
    anbar[il - 1] *= anp[il - 1];
    anp1 = anp[il - 1] - 1;
    anp2 = anp1 - 1;
    capflj = 0.0;
    capfjj = 0.0;
    thetalj = 0.0;
    thetajj = 0.0;
    smallflj = 0.0;
    smallfjj = 0.0;
    FORLIM1 = nu;
    for (iu = 1; iu <= FORLIM1; iu++) {
      pbar[il - 1][iu - 1] *= anbar[il - 1];
      hbar[il - 1][iu - 1] *= anbar[il - 1];
    }
    for (ip = inf; ip <= sup; ip++) {
      if (an[ip - 1][il - 1] != 0) {
anbarj = (anbar[il - 1] - an[ip - 1][il - 1]) / anp1;
TEMP = an[ip - 1][il - 1];
annbarj = annbar[il - 1] - TEMP * TEMP;
ancj = (anp1 * anbarj - annbarj / anp1 / anbarj) / anp2;
term1j = 0.0;
term2j = 0.0;
term3j = 0.0;
term4j = 0.0;
```

```
term5j = 0.0;
FORLIM2 = nu;
for (iu = 1; iu <= FORLIM2; iu++) {
  pbarj = pbar[il - 1]
  [iu - 1] - an[ip - 1][il - 1] * p[ip - 1][il - 1][iu - 1];
  TEMP1 = p[ip - 1][il - 1][iu - 1];
  ppbarj = ppbar[il - 1]
   [iu - 1] - an[ip - 1][il - 1] * (TEMP1 * TEMP1);
  hbarj = hbar[il - 1]
  [iu - 1] - an[ip - 1][il - 1] * h[ip - 1][il - 1][iu - 1];
  pbarj = pbarj / anp1 / anbarj;
  varpj = (ppbarj - anp1 * anbarj * pbarj * pbarj) / anp2 / anbarj;
  hbarj = hbarj / anp1 / anbarj;
  if (pbarj > 0.00001 && pbarj < 0.99999) {
    aj = pbarj * (1.0 - pbarj) - anp2 * varpj / anp1;
    bj = aj;
    aj = varpj - (aj - hbarj / 4.0) / (anbarj - 1.0);
    aj = aj * anbarj / ancj;
    bj += (1.0 - 2.0 * anbarj) * hbarj / 4.0 / anbarj;
    bj = anbarj * bj / (anbarj - 1.0);
    cj = hbarj / 2.0;
    term1j += aj + bj;
    term2j += aj + bj + cj;
    term3j += aj;
    term4j += bj;
    term5j += bj + cj;
  }
}
if (term2j != 0.0 && term5j != 0.0) {
  capflj += term1j / term2j;
  TEMP1 = term1j / term2j;
  capfjj += TEMP1 * TEMP1;
  thetalj += term3j / term2j;
  TEMP1 = term3j / term2j;
  thetajj += TEMP1 * TEMP1;
  smallflj += term4j / term5j;
  TEMP1 = term4j / term5j;
  smallfjj += TEMP1 * TEMP1;
}
      }
    }
    if (term2j != 0.0 && term5j != 0.0) {
      fprintf(fileout1, "\n for locus : %s\n", locname[il - 1]);
      fprintf(fileout1, "          capf      theta    smallf\n");
      capfjj -= capflj * capflj / anp[il - 1];
      capfjj = sqrt(anp1 * capfjj / anp[il - 1]);
      capflj = anp[il - 1] * capfl[il - 1] - anp1 * capflj / anp[il - 1];
      thetajj -= thetalj * thetalj / anp[il - 1];
      thetajj = sqrt(anp1 * thetajj / anp[il - 1]);
      thetalj = anp[il - 1] * thetal[il - 1] - anp1 * thetalj / anp[il - 1];
      smallfjj -= smallflj * smallflj / anp[il - 1];
      smallfjj = sqrt(anp1 * smallfjj / anp[il - 1]);
      smallflj = anp[il - 1] * smallfl[il - 1] - anp1 * smallflj / anp[il - 1];
      fprintf(fileout1, " total%10.4f%10.4f%10.4f means\n",
      capflj, thetalj, smallflj);
      fprintf(fileout1, "      %10.4f%10.4f%10.4f std. devs.\n",
      capfjj, thetajj, smallfjj);
      fflush(fileout1);
      P_ioresult = 0;
    }
  }

}
/**************************************************************************/
Static Void jackknife(inf, sup, ans)
unsigned short inf, sup;
Char ans;
{
  double capflj, capfjj, smallflj, smallfjj, thetalj, thetajj, term1j, term2j,
 term3j, term4j, term5j;
  unsigned short anl1, FORLIM;
  double TEMP;

  if (sup > 2)
    jack_ov_pop(inf, sup);
  anl1 = anl - 1;
  capflj = 0.0;
  capfjj = 0.0;
  thetalj = 0.0;
  thetajj = 0.0;
  smallflj = 0.0;
  smallfjj = 0.0;
  FORLIM = nl;
  for (il = 1; il <= FORLIM; il++) {
    term1j = tterm1 - term1[il - 1];
    term2j = tterm2 - term2[il - 1];
    term3j = tterm3 - term3[il - 1];
    term4j = tterm4 - term4[il - 1];
    term5j = tterm5 - term5[il - 1];
    if (term2j != 0 && term5j != 0) {
      capflj += term1j / term2j;
```

```
      TEMP = term1j / term2j;
      capfjj += TEMP * TEMP;
      thetalj += term3j / term2j;
      TEMP = term3j / term2j;
      thetajj += TEMP * TEMP;
      smallflj += term4j / term5j;
      TEMP = term4j / term5j;
      smallfjj += TEMP * TEMP;
    }
  }
  capfjj = sqrt(anl1 * (capfjj - capflj * capflj / anl) / anl);
  capflj = anl * tcapf - anl1 * capflj / anl;
  thetajj = sqrt(anl1 * (thetajj - thetalj * thetalj / anl) / anl);
  thetalj = anl * ttheta - anl1 * thetalj / anl;
  smallfjj = sqrt(anl1 * (smallfjj - smallflj * smallflj / anl) / anl);
  smallflj = anl * tsmallf - anl1 * smallflj / anl;
  fprintf(fileout1, "\n jackknifing over loci.\n");
  if (fileout1 != NULL)
    fileout1 = freopen(fileout1_NAME, "a", fileout1);
  else
    fileout1 = fopen(fileout1_NAME, "a");
  if (fileout1 == NULL)
    _EscIO(FileNotFound);
  fprintf(fileout1, "\n          capf      theta    smallf\n");
  fprintf(fileout1, " total%10.4f%10.4f%10.4f means\n",
  capflj, thetalj, smallflj);
  fprintf(fileout1, "      %10.4f%10.4f%10.4f std. devs.\n",
  capfjj, thetajj, smallfjj);
  fflush(fileout1);
  P_ioresult = 0;
}  /*proc jackknife*/


typedef float perpop[npmax];
typedef float *perperpop[npmax];
/**************************************************************************/
Static Void perpair()
{
  double smallfp, tsmallfp, pqp, tpqp, ap, bp, cp, term1p, term2p, term3p,
 term4p, term5p;
  perperpop tterm2p, tterm3p;
  double thetapp, smallfpp, tthetapp, anbarp, annbarp, ancp, pbarp, ppbarp,
 varpp, hbarp;
  perperpop fpp, fppt, mppt;
  unsigned short ip1, ip2;
  float maxmig;
  unsigned short FORLIM, FORLIM1;
  Char STR2[256];
  unsigned short FORLIM2, FORLIM3;
  long TEMP, TEMP1;
  double TEMP2, TEMP3;

  FORLIM = np;
  for (ip = 1; ip <= FORLIM; ip++) {
    fpp[ip - 1] = (float *)Malloc(sizeof(perpop));
    fppt[ip - 1] = (float *)Malloc(sizeof(perpop));
    tterm2p[ip - 1] = (float *)Malloc(sizeof(perpop));
    tterm3p[ip - 1] = (float *)Malloc(sizeof(perpop));
    mppt[ip - 1] = (float *)Malloc(sizeof(perpop));
  }
  FORLIM = np;
  for (ip1 = 0; ip1 < FORLIM; ip1++) {
    FORLIM1 = np;
    for (ip2 = 0; ip2 < FORLIM1; ip2++) {
      fppt[ip1][ip2] = -9.99;
      tterm2p[ip2][ip1] = 0.0;
      tterm3p[ip2][ip1] = 0.0;
    }
  }
  fprintf(fileout1, "\n*************************************************\n");
  fprintf(fileout1, "  theta per locus over pair of populations.\n");

  FORLIM = nl;
  for (il = 1; il <= FORLIM; il++) {
    fprintf(fileout1, "\n for locus: %s\n\n", locname[il - 1]);
    fprintf(fileout1, "      ");
    FORLIM1 = np;
    for (ip = 1; ip <= FORLIM1; ip++)
      fprintf(fileout1, "%6u", ip);
    putc('\n', fileout1);

    FORLIM1 = np;
    for (ip1 = 0; ip1 < FORLIM1; ip1++) {
      FORLIM2 = np;
      for (ip2 = 0; ip2 < FORLIM2; ip2++)
fpp[ip1][ip2] = -9.99;
    }
    FORLIM1 = np;
    for (ip1 = 0; ip1 < FORLIM1; ip1++) {
      fprintf(fileout1, "%6u", ip1 + 1);
      FORLIM2 = np;
      for (ip2 = ip1; ip2 < FORLIM2; ip2++) {
if (ip1 + 1 == ip2 + 1) {   /*calculation of smallf per pop*/
  smallfp = 0.0;
  pqp = 0.0;
  tsmallfp = 0.0;
  tpqp = 0.0;
  FORLIM3 = nu;
  for (iu = 1; iu <= FORLIM3; iu++) {
    if (p[ip1][il - 1][iu - 1] > 0.00001 &&
p[ip1][il - 1][iu - 1] < 0.99999) {
      pqp = p[ip1][il - 1][iu - 1] * (1.0 - p[ip1][il - 1][iu - 1]);
      smallfp = 1 - h[ip1][il - 1][iu - 1] / 2.0 / pqp;
      tsmallfp += smallfp * pqp;
      tpqp += pqp;
    }
  }
  if (tpqp != 0.0) {
    tsmallfp /= tpqp;
    tpqp = 2.0 * an[ip1][il - 1] - 1.0;
    tsmallfp = (tpqp * tsmallfp + 1.0) / (tpqp + tsmallfp);
    fpp[ip1][ip2] = tsmallfp;
  }
} else {  /*calculation of theta and smallf per pair of pop*/
  if (an[ip1][il - 1] != 0 && an[ip2][il - 1] != 0) {
    anbarp = (an[ip1][il - 1] + an[ip2][il - 1]) / 2.0;
    TEMP = an[ip1][il - 1];
    TEMP1 = an[ip2][il - 1];
    annbarp = TEMP * TEMP + TEMP1 * TEMP1;
    ancp = 2.0 * anbarp - annbarp / 2.0 / anbarp;
    term1p = 0.0;
    term2p = 0.0;
    term3p = 0.0;
    term4p = 0.0;
    term5p = 0.0;
    FORLIM3 = nu;
    for (iu = 1; iu <= FORLIM3; iu++) {
      pbarp = an[ip1][il - 1] * p[ip1][il - 1][iu - 1] + an[ip2]
[il - 1] * p[ip2][il - 1][iu - 1];
      TEMP2 = p[ip1][il - 1][iu - 1];
      TEMP3 = p[ip2][il - 1][iu - 1];
      ppbarp = an[ip1][il - 1] * (TEMP2 * TEMP2) + an[ip2]
 [il - 1] * (TEMP3 * TEMP3);
      hbarp = an[ip1][il - 1] * h[ip1][il - 1][iu - 1] + an[ip2]
[il - 1] * h[ip2][il - 1][iu - 1];
      pbarp = pbarp / 2.0 / anbarp;
      varpp = ppbarp - 2.0 * anbarp * pbarp * pbarp;
      varpp /= anbarp;
      hbarp = hbarp / 2.0 / anbarp;
      ap = pbarp * (1.0 - pbarp) - varpp / 2.0;
      bp = ap;
      ap -= hbarp / 4.0;
      ap = varpp + ap / (1.0 - anbarp);
      ap = anbarp * ap / ancp;
      bp += (1.0 - 2.0 * anbarp) * hbarp / 4.0 / anbarp;
      bp = anbarp * bp / (anbarp - 1.0);
      cp = hbarp / 2.0;
      term1p += ap + bp;
      term2p += ap + bp + cp;
      term3p += ap;
      term4p += bp;
      term5p += bp + cp;
    }
    tterm2p[ip2][ip1] += term2p;
    tterm3p[ip2][ip1] += term3p;
    if (term2p != 0.0 && term5p != 0.0) {
      thetapp = term3p / term2p;
      smallfpp = term4p / term5p;
      fpp[ip1][ip2] = thetapp;
      fpp[ip2][ip1] = thetapp;
    }
    if (il == nl) {
      if (tterm2p[ip2][ip1] != 0.0) {
tthetapp = tterm3p[ip2][ip1] / tterm2p[ip2][ip1];
fppt[ip2][ip1] = tthetapp;
fppt[ip1][ip2] = tthetapp;
      }
    }
  }
}
      }

      FORLIM2 = np;
      /*
                      this is if matrices of theta per locus and perpair
                      need to be written to fileout1
      */
      for (ip2 = 0; ip2 < FORLIM2; ip2++) {
if (ip1 + 1 != ip2 + 1) {
  if (fpp[ip1][ip2] > -9.0)
    fprintf(fileout1, "%6.2f", fpp[ip1][ip2]);
  else
```

```
    fprintf(fileout1, "      ");
} else
  fprintf(fileout1, "      ");
      }
      putc('\n', fileout1);
    }
  }

  FORLIM = np;
  for (ip1 = 0; ip1 < FORLIM; ip1++) {
    FORLIM1 = np;
    for (ip2 = 0; ip2 < FORLIM1; ip2++) {
      if (fppt[ip1][ip2] > 0.0)
mppt[ip1][ip2] = (1.0 - fppt[ip1][ip2]) / 4.0 / fppt[ip1][ip2];
      else
mppt[ip1][ip2] = -999.999;
    }
  }
  maxmig = -1000.0;
  FORLIM = np;
  for (ip1 = 0; ip1 < FORLIM; ip1++) {
    FORLIM1 = np;
    for (ip2 = 0; ip2 < FORLIM1; ip2++) {
      if (mppt[ip1][ip2] > maxmig)
maxmig = mppt[ip1][ip2];
    }
  }
 /* printf("%10.3f\n", maxmig); */
  FORLIM = np;
  for (ip1 = 1; ip1 <= FORLIM; ip1++) {
    FORLIM1 = np;
    for (ip2 = 1; ip2 <= FORLIM1; ip2++) {
      sprintf(STR2, "%s.mig", name);
      strcpy(filemig_NAME, STR2);
    }
    /*     if (mppt[ip1]^[ip2]<0) then mppt[ip1]^[ip2]:=maxmig;*/
  }
  if (filemig != NULL)
    filemig = freopen(filemig_NAME, "w", filemig);
  else
    filemig = fopen(filemig_NAME, "w");
  if (filemig == NULL)
    _EscIO(FileNotFound);
  FORLIM = np;
  for (ip1 = 1; ip1 <= FORLIM; ip1++) {
    FORLIM1 = np;
    for (ip2 = 1; ip2 <= FORLIM1; ip2++)
      fprintf(filemig, "%10.3f%4u%4u\n", mppt[ip1 - 1][ip2 - 1], ip2, ip1);
    /* if ip1<>ip2 then*/
    /*if mppt[ip1]^[ip2]<>999.999 then*/
    putc('\n', filemig);
  }
  if (filemig != NULL)
    fclose(filemig);
  filemig = NULL;
  fflush(fileout1);
  P_ioresult = 0;
  FORLIM = np;
  for (ip = 1; ip <= FORLIM; ip++) {
    Free(fppt[ip - 1]);
    Free(fpp[ip - 1]);
    Free(tterm2p[ip - 1]);
    Free(tterm3p[ip - 1]);
    Free(mppt[ip - 1]);
  }
}  /*proc perpair*/


/* Local variables for quicksort: */
struct LOC_quicksort {
  float **a;
} ;

Local Void sort(l, r, LINK)
short l, r;
struct LOC_quicksort *LINK;
{
  short i, j;
  float x, y;

  i = l;
  j = r;
  x = (*LINK->a)[(l + r) / 2 - 1];
  do {
    while ((*LINK->a)[i - 1] < x)
      i++;
    while (x < (*LINK->a)[j - 1])
      j--;
    if (i <= j) {
      y = (*LINK->a)[i - 1];
      (*LINK->a)[i - 1] = (*LINK->a)[j - 1];
      (*LINK->a)[j - 1] = y;
      i++;
      j--;
    }
  } while (i <= j);
  if (l < j)
    sort(l, j, LINK);
  if (i < r)
    sort(i, r, LINK);
}
/**************************************************************************/
Static Void quicksort(a_, lo, hi)
float **a_;
short lo, hi;
{   /*quicksort*/
  struct LOC_quicksort V;

  V.a = a_;
  sort(lo, hi, &V);
}


/**************************************************************************/
Static Void bootstrap(ans)
Char ans;
{
  short mb, i, mbl, mbu, temp;
  float term1b, term2b, term3b, term4b, term5b, cafl, cafu, thetl, thetu,
smlfl, smlfu;
  short repbb, repuu;
  float cafll, cafuu, thetll, thetuu, smlfll, smlfuu;
  unsigned short FORLIM1;

  mb = 0;
  for (i = 1; i <= maxboot; i++) {
    term1b = 0.0;
    term2b = 0.0;
    term3b = 0.0;
    term4b = 0.0;
    term5b = 0.0;
    FORLIM1 = nl;
    for (il = 1; il <= FORLIM1; il++) {
      temp = grandom((long)nl) + 1;
      term1b += term1[temp - 1];
      term2b += term2[temp - 1];
      term3b += term3[temp - 1];
      term4b += term4[temp - 1];
      term5b += term5[temp - 1];
    }
    if (term2b != 0.0 && term5b != 0.0) {
      mb++;
      capfb[mb - 1] = term1b / term2b;
      thetb[mb - 1] = term3b / term2b;
      smlfb[mb - 1] = term4b / term5b;
    }
  }
  mbl = (long)floor(1.0 * maxboot / 40.0 + 0.5);
  mbu = (long)floor(39.0 * maxboot / 40.0 + 0.5);
  quicksort(&capfb, 1, maxboot);
  quicksort(&thetb, 1, maxboot);
  quicksort(&smlfb, 1, maxboot);
  cafl = capfb[mbl - 1];
  cafu = capfb[mbu - 1];
  thetl = thetb[mbl - 1];
  thetu = thetb[mbu - 1];
  smlfl = smlfb[mbl - 1];
  smlfu = smlfb[mbu - 1];
  repbb = (long)floor(1.0 * maxboot / 200.0 + 0.5);
  repuu = (long)floor(199.0 * maxboot / 200.0 + 0.5);
  cafll = capfb[repbb - 1];
  cafuu = capfb[repuu - 1];
  thetll = thetb[repbb - 1];
  thetuu = thetb[repuu - 1];
  smlfll = smlfb[repbb - 1];
  smlfuu = smlfb[repuu - 1];
  fprintf(fileout1, "\n*************************************************\n");
  fprintf(fileout1, "         bootstrapping over loci.\n\n");
  fprintf(fileout1, "         95%% confidence interval.\n\n");
  fprintf(fileout1, "      capf     theta    smallf\n");
  fprintf(fileout1, "%10.4f%10.4f%10.4f\n", cafl, thetl, smlfl);
  fprintf(fileout1, "%10.4f%10.4f%10.4f\n\n", cafu, thetu, smlfu);
  fprintf(fileout1, "         99%% confidence interval.\n\n");
  fprintf(fileout1, "      capf     theta    smallf\n");
  fprintf(fileout1, "%10.4f%10.4f%10.4f\n", cafll, thetll, smlfll);
  fprintf(fileout1, "%10.4f%10.4f%10.4f\n\n", cafuu, thetuu, smlfuu);
  fflush(fileout1);
  P_ioresult = 0;
}
/**************************************************************************/
Static Void permwithin()
{
```

```
  unsigned short rep, i, max, temp, temp1;
  uchar pop[maxind * 2];
  unsigned short n[numax];
  short repb, repu;
  float cafl, cafu, thetl, thetu, smlfl, smlfu;
  short repbb, repuu;
  float cafll, cafuu, thetll, thetuu, smlfll, smlfuu;
  unsigned short FORLIM, FORLIM1, FORLIM2, FORLIM3;

  numbperm = 5000;
  FORLIM = numbperm - 2;
  for (rep = 0; rep <= FORLIM; rep++) {
    FORLIM1 = np;
    for (ip = 1; ip <= FORLIM1; ip++) {
      FORLIM2 = nl;
      for (il = 1; il <= FORLIM2; il++) {
max = an[ip - 1][il - 1] * 2;
if (max != 0) {
  FORLIM3 = nu;
  for (iu = 1; iu <= FORLIM3; iu++)
    n[iu - 1] = 0;
  FORLIM3 = nu;
  for (iu = 1; iu <= FORLIM3; iu++) {
    p[ip - 1][il - 1][iu - 1] *= max;
    if (iu == 1)
      n[iu - 1] = (long)floor(p[ip - 1][il - 1][iu - 1] + 0.5);
    else
      n[iu - 1] = n[iu - 2] + (long)floor(p[ip - 1][il - 1]
  [iu - 1] + 0.5);
  }
  i = 0;
  FORLIM3 = nu;
  for (iu = 1; iu <= FORLIM3; iu++) {
    if (p[ip - 1][il - 1][iu - 1] != 0.0) {
      do {
i++;
pop[i - 1] = iu;
      } while (i != n[iu - 1]);
    }
  }
  i = max;
  do {
    temp = pop[i - 1];
    temp1 = grandom((long)i) + 1;
    pop[i - 1] = pop[temp1 - 1];
    pop[temp1 - 1] = temp;
    i--;
  } while (i != 1);
  FORLIM3 = nu;
  for (iu = 1; iu <= FORLIM3; iu++)
    h[ip - 1][il - 1][iu - 1] = 0.0;
  i = 1;
  while (i < max) {
    if (pop[i - 1] != pop[i]) {
      h[ip - 1][il - 1][pop[i - 1] - 1] += 1.0;
      h[ip - 1][il - 1][pop[i] - 1] += 1.0;
    }
    i += 2;
  }
}
      }
    }
    calcfstat(1, np, ans);
    capfb[rep] = tcapf;
    thetb[rep] = ttheta;
    smlfb[rep] = tsmallf;
  }
  capfb[numbperm - 1] = real_f[0];
  thetb[numbperm - 1] = real_f[1];
  smlfb[numbperm - 1] = real_f[2];
  quicksort(&capfb, 1, numbperm);
  quicksort(&thetb, 1, numbperm);
  quicksort(&smlfb, 1, numbperm);
  repb = (long)floor(1.0 * numbperm / 40.0 + 0.5);
  repu = (long)floor(39.0 * numbperm / 40.0 + 0.5);
  cafl = capfb[repb - 1];
  cafu = capfb[repu - 1];
  thetl = thetb[repb - 1];
  thetu = thetb[repu - 1];
  smlfl = smlfb[repb - 1];
  smlfu = smlfb[repu - 1];
  repbb = (long)floor(1.0 * numbperm / 200.0 + 0.5);
  repuu = (long)floor(199.0 * numbperm / 200.0 + 0.5);
  cafll = capfb[repbb - 1];
  cafuu = capfb[repuu - 1];
  thetll = thetb[repbb - 1];
  thetuu = thetb[repuu - 1];
  smlfll = smlfb[repbb - 1];
  smlfuu = smlfb[repuu - 1];
  FORLIM = numbperm -1;
  for (rep = 0 ; rep <= FORLIM ; rep++) {
    fprintf(fileboot1, "%6u%10.4f%10.4f%10.4f\n", (rep + 1), capfb[rep], thetb[rep], smlfb[rep]);
     }
  fflush(fileboot1);
  fprintf(fileout1, "\n**************************************************\n");
  fprintf(fileout1, "         permutting alleles within samples.\n\n");
  fprintf(fileout1, "         95%% confidence interval.\n\n");
  fprintf(fileout1, "      capf     theta    smallf\n");
  fprintf(fileout1, "%10.4f%10.4f%10.4f\n", cafl, thetl, smlfl);
  fprintf(fileout1, "%10.4f%10.4f%10.4f\n\n", cafu, thetu, smlfu);
  fprintf(fileout1, "         99%% confidence interval.\n\n");
  fprintf(fileout1, "      capf     theta    smallf\n");
  fprintf(fileout1, "%10.4f%10.4f%10.4f\n", cafll, thetll, smlfll);
  fprintf(fileout1, "%10.4f%10.4f%10.4f\n\n", cafuu, thetuu, smlfuu);
  rep = numbperm + 1;
  do {
    rep--;
  } while (smlfb[rep - 1] >= real_f[2]);
  rep++;
  if (rep < numbperm)
    fprintf(fileout1, "(prob fis=0)= %10.5f\n", 1.0 - (double)rep / numbperm);
  else
    fprintf(fileout1, "(prob fis=0)< %10.5f\n", 1.0 / numbperm);
  fflush(fileout1);
  P_ioresult = 0;
}  /*of proc permwithin*/
/************************************************************************/
Static Void permbetween()
{
  unsigned short i, rep, temp, temp1, max;
  uchar pop[maxind * 2];
  unsigned short n[npmax][numax];
  unsigned short sumn[npmax];
  short repb, repu;
  float cafl, cafu, thetl, thetu, smlfl, smlfu;
  short repbb, repuu;
  float cafll, cafuu, thetll, thetuu, smlfll, smlfuu;
  unsigned short FORLIM, FORLIM1, FORLIM2, FORLIM3;

  numbperm = 5000;
  FORLIM = numbperm - 2;
  for (rep = 0; rep <= FORLIM; rep++) {
    FORLIM1 = nl;
    for (il = 1; il <= FORLIM1; il++) {
      max = 0;
      sumn[0] = an[0][il - 1] * 2;
      FORLIM2 = np;
      for (ip = 1; ip <= FORLIM2; ip++)
max += an[ip - 1][il - 1] * 2;
      FORLIM2 = np;
      for (ip = 2; ip <= FORLIM2; ip++)
sumn[ip - 1] = sumn[ip - 2] + an[ip - 1][il - 1] * 2;
      if (max != 0) {
FORLIM2 = np;
for (ip = 1; ip <= FORLIM2; ip++) {
  FORLIM3 = nu;
  for (iu = 1; iu <= FORLIM3; iu++)
    n[ip - 1][iu - 1] = 0;
}
FORLIM2 = np;
for (ip = 1; ip <= FORLIM2; ip++) {
  FORLIM3 = nu;
  for (iu = 1; iu <= FORLIM3; iu++) {
    p[ip - 1][il - 1][iu - 1] *= an[ip - 1][il - 1] * 2.0;
    if (ip == 1 && iu == 1)
      n[ip - 1][iu - 1] = (long)floor(p[ip - 1][il - 1][iu - 1] + 0.5);
    else if (iu == 1)
      n[ip - 1]
[iu - 1] = n[ip - 2][nu - 1] + (long)floor(p[ip - 1][il - 1]
   [iu - 1] + 0.5);
    else
      n[ip - 1]
[iu - 1] = n[ip - 1][iu - 2] + (long)floor(p[ip - 1][il - 1]
   [iu - 1] + 0.5);
  }
}
i = 0;
FORLIM2 = np;
for (ip = 1; ip <= FORLIM2; ip++) {
  FORLIM3 = nu;
  for (iu = 1; iu <= FORLIM3; iu++) {
    if (p[ip - 1][il - 1][iu - 1] != 0.0) {
      do {
i++;
pop[i - 1] = iu;
      } while (i != n[ip - 1][iu - 1]);
    }
  }
}
i = max;
do {
  temp = pop[i - 1];
```

```
  temp1 = grandom((long)i) + 1;
  pop[i - 1] = pop[temp1 - 1];
  pop[temp1 - 1] = temp;
  i--;
} while (i != 1);
FORLIM2 = np;
for (ip = 1; ip <= FORLIM2; ip++) {
  FORLIM3 = nu;
  for (iu = 1; iu <= FORLIM3; iu++) {
    h[ip - 1][il - 1][iu - 1] = 0.0;
    p[ip - 1][il - 1][iu - 1] = 0.0;
  }
}
ip = 1;
i = 1;
do {
  while (i < sumn[ip - 1]) {
    p[ip - 1][il - 1][pop[i - 1] - 1] += 1.0;
    p[ip - 1][il - 1][pop[i] - 1] += 1.0;
    if (pop[i - 1] != pop[i]) {
      h[ip - 1][il - 1][pop[i - 1] - 1] += 1.0;
      h[ip - 1][il - 1][pop[i] - 1] += 1.0;
    }
    i += 2;
  }
  ip++;
} while (ip <= np);
      }
    }
    calcfstat(1, np, ans);
    capfb[rep] = tcapf;
    thetb[rep] = ttheta;
    smlfb[rep] = tsmallf;
  }
  capfb[numbperm - 1] = real_f[0];
  thetb[numbperm - 1] = real_f[1];
  smlfb[numbperm - 1] = real_f[2];
  quicksort(&capfb, 1, numbperm);
  quicksort(&thetb, 1, numbperm);
  quicksort(&smlfb, 1, numbperm);
  repb = (long)floor(1.0 * numbperm / 40.0 + 0.5);
  repu = (long)floor(39.0 * numbperm / 40.0 + 0.5);
  cafl = capfb[repb - 1];
  cafu = capfb[repu - 1];
  thetl = thetb[repb - 1];
  thetu = thetb[repu - 1];
  smlfl = smlfb[repb - 1];
  smlfu = smlfb[repu - 1];
  repbb = (long)floor(1.0 * numbperm / 200.0 + 0.5);
  repuu = (long)floor(199.0 * numbperm / 200.0 + 0.5);
  cafll = capfb[repbb - 1];
  cafuu = capfb[repuu - 1];
  thetll = thetb[repbb - 1];
  thetuu = thetb[repuu - 1];
  smlfll = smlfb[repbb - 1];
  smlfuu = smlfb[repuu - 1];
  FORLIM = numbperm -1;
  for (rep = 0 ; rep <= FORLIM ; rep++) {
    fprintf(fileboot2, "%6u%10.4f%10.4f%10.4f\n", (rep + 1), capfb[rep], thetb[rep], smlfb[rep]);
     }
  fflush(fileboot2);
  fprintf(fileout1, "\n**************************************************\n");
  fprintf(fileout1, "         permutting alleles within total.\n\n");
  fprintf(fileout1, "         95%% confidence interval.\n\n");
  fprintf(fileout1, "      capf     theta    smallf\n");
  fprintf(fileout1, "%10.4f%10.4f%10.4f\n", cafl, thetl, smlfl);
  fprintf(fileout1, "%10.4f%10.4f%10.4f\n\n", cafu, thetu, smlfu);
  fprintf(fileout1, "         99%% confidence interval.\n\n");
  fprintf(fileout1, "      capf     theta    smallf\n");
  fprintf(fileout1, "%10.4f%10.4f%10.4f\n", cafll, thetll, smlfll);
  fprintf(fileout1, "%10.4f%10.4f%10.4f\n\n", cafuu, thetuu, smlfuu);
  rep = numbperm + 1;
  do {
    rep--;
  } while (capfb[rep - 1] >= real_f[0]);
  rep++;
  if (rep < numbperm)
    fprintf(fileout1, "(prob fit=0)= %10.5f\n", 1.0 - (double)rep / numbperm);
  else
    fprintf(fileout1, "(prob fit=0)< %10.5f\n", 1.0 / numbperm);
  fflush(fileout1);
  P_ioresult = 0;
}
typedef uchar pops[maxind + 1][nlmax];
/************************************************************************/
Static Void permmultgen()
{
  unsigned short rep;
  uchar (*pop)[nlmax];
  unsigned short n[npmax];
  unsigned short temp, max, i;
  short repb, repu;
  float cafl, cafu, thetl, thetu, smlfl, smlfu;
  short repbb, repuu;
  float cafll, cafuu, thetll, thetuu, smlfll, smlfuu;
  Char STR1[256];
  unsigned short FORLIM, FORLIM1;
  int TEMP1;

  numbperm = 5000;
  pop = (uchar(*)[nlmax])Malloc(sizeof(pops));
/*
  sprintf(STR1, "%s.tmp", name);
  strcpy(fileout2_NAME, STR1);
*/
  fileout2 = tmpfile(fileout2);
  FORLIM = numbperm - 2;
  for (rep = 0; rep <= FORLIM; rep++) {
      rewind(filein);
      rewind(fileout2);
    fscanf(filein, "%hd%hd%hd%*[^\n]", &np, &nl, &nu);
    getc(filein);
    FORLIM1 = np;
    for (ip = 1; ip <= FORLIM1; ip++)
      n[ip - 1] = 0;
    i = 1;
    FORLIM1 = nl;
    for (il = 1; il <= FORLIM1; il++) {
      fscanf(filein, "%*[^\n]");
      getc(filein);
    }
    while (!P_eof(filein)) {
      fscanf(filein, "%hd", &ip);
      if (ip == 0)
continue;
      n[ip - 1]++;
      FORLIM1 = nl;
      for (il = 1; il <= FORLIM1; il++) {
fscanf(filein, "%d", &TEMP1);
pop[i][il - 1] = TEMP1;
      }
      fscanf(filein, "%*[^\n]");
      getc(filein);
      i++;
    }
    max = i - 1;
    FORLIM1 = np;
    for (ip = 2; ip <= FORLIM1; ip++)
      n[ip - 1] += n[ip - 2];
    i = max;
    do {
      FORLIM1 = nl;
      for (il = 1; il <= FORLIM1; il++)
pop[0][il - 1] = pop[i][il - 1];
      temp = grandom((long)i) + 1;
      FORLIM1 = nl;
      for (il = 1; il <= FORLIM1; il++) {
pop[i][il - 1] = pop[temp][il - 1];
pop[temp][il - 1] = pop[0][il - 1];
      }
      i--;
    } while (i != 1);
    fprintf(fileout2, "%3u%3u%3u\n", np, nl, nu);
    FORLIM1 = nl;
    for (il = 1; il <= FORLIM1;il++)
      putc('\n',fileout2);
    i = 0;
    ip = 1;
    do {
      do {
i++;
fprintf(fileout2, "%3u", ip);
FORLIM1 = nl;
for (il = 1; il <= FORLIM1; il++)
  fprintf(fileout2, "%3d", pop[i][il - 1]);
putc('\n', fileout2);
      } while (i != n[ip - 1]);
      ip++;
    } while (ip <= np);
    readdata(&fileout2);
    calcfstat(1, np, ans);
    capfb[rep] = tcapf;
    thetb[rep] = ttheta;
    smlfb[rep] = tsmallf;
  }
  capfb[numbperm - 1] = real_f[0];
  thetb[numbperm - 1] = real_f[1];
  smlfb[numbperm - 1] = real_f[2];
  quicksort(&capfb, 1, numbperm);
  quicksort(&thetb, 1, numbperm);
  quicksort(&smlfb, 1, numbperm);
  repb = (long)floor(1.0 * numbperm / 40.0 + 0.5);
```

```
  repu = (long)floor(39.0 * numbperm / 40.0 + 0.5);
  cafl = capfb[repb - 1];
  cafu = capfb[repu - 1];
  thetl = thetb[repb - 1];
  thetu = thetb[repu - 1];
  smlfl = smlfb[repb - 1];
  smlfu = smlfb[repu - 1];
  repbb = (long)floor(1.0 * numbperm / 200.0 + 0.5);
  repuu = (long)floor(199.0 * numbperm / 200.0 + 0.5);
  cafll = capfb[repbb - 1];
  cafuu = capfb[repuu - 1];
  thetll = thetb[repbb - 1];
  thetuu = thetb[repuu - 1];
  smlfll = smlfb[repbb - 1];
  smlfuu = smlfb[repuu - 1];
  FORLIM = numbperm -1;
  for (rep = 0 ; rep <= FORLIM ; rep++) {
    fprintf(fileboot3, "%6u%10.4f%10.4f%10.4f\n", (rep + 1), capfb[rep], thetb[rep], smlfb[rep]);
     }
  fflush(fileboot3);
  fprintf(fileout1, "\n************************************************\n");
  fprintf(fileout1, "         permutting genotypes within total.\n\n");
  fprintf(fileout1, "         95%% confidence interval.\n\n");
  fprintf(fileout1, "      capf     theta    smallf\n");
  fprintf(fileout1, "%10.4f%10.4f%10.4f\n", cafl, thetl, smlfl);
  fprintf(fileout1, "%10.4f%10.4f%10.4f\n\n", cafu, thetu, smlfu);
  fprintf(fileout1, "         99%% confidence interval.\n\n");
  fprintf(fileout1, "      capf     theta    smallf\n");
  fprintf(fileout1, "%10.4f%10.4f%10.4f\n", cafll, thetll, smlfll);
  fprintf(fileout1, "%10.4f%10.4f%10.4f\n\n", cafuu, thetuu, smlfuu);
  rep = numbperm + 1;
  do {
    rep--;
  } while (thetb[rep - 1] >= real_f[1]);
  rep++;
  putc('\n', fileout1);
  if (rep < numbperm)
    fprintf(fileout1, "(prob fst=0)= %10.5f\n", 1.0 - (double)rep / numbperm);
  else
    fprintf(fileout1, "(prob fst=0)< %10.5f\n", 1.0 / numbperm);
  fflush(fileout1);
  fclose(fileout2);
  P_ioresult = 0;
  Free(pop);
}
/**************************************************************************/
main(argc, argv)
int argc;
Char *argv[];
{
  unsigned short FORLIM;

  PASCAL_MAIN(argc, argv);
  filemig = NULL;
  filepar = NULL;
  fileini = NULL;
  fileout2 = NULL;
  fileout1 = NULL;
  filein = NULL;
  fileboot1 = NULL;
  fileboot2 = NULL;
  fileboot3 = NULL;
  ans = 'n';
  pass = 0;
  strcpy(fileini_NAME, "fstat.ini");
  if (fileini != NULL)
    fileini = freopen(fileini_NAME, "r", fileini);
  else
    fileini = fopen(fileini_NAME, "r");
  if (fileini == NULL)
    _EscIO(FileNotFound);
  fscanf(fileini, "%ld%ld", &s1, &s2);
  if (fileini != NULL)
    fclose(fileini);
  fileini = NULL;
  strcpy(name, P_argv[1]);
  sprintf(namedat, "%s.sdb", name);
  sprintf(nameout, "%s.out", name);
  sprintf(nameboot1, "%s.bo1", name);
  sprintf(nameboot2, "%s.bo2", name);
  sprintf(nameboot3, "%s.bo3", name);
  strcpy(filein_NAME, namedat);
  if (filein != NULL)
    filein = freopen(filein_NAME, "r", filein);
  else
    filein = fopen(filein_NAME, "r");
  if (filein == NULL)
    _EscIO(FileNotFound);
  fscanf(filein, "%hd%hd%hd%*[^\n]", &np, &nl, &nu);
  getc(filein);
  strcpy(fileout1_NAME, nameout);
  if (fileout1 != NULL)
    fileout1 = freopen(fileout1_NAME, "w", fileout1);
  else
    fileout1 = fopen(fileout1_NAME, "w");
  if (fileout1 == NULL)
    _EscIO(FileNotFound);
  strcpy(fileboot1_NAME, nameboot1);
  if (fileboot1 != NULL)
      fileboot1 =  freopen(fileboot1_NAME, "w", fileboot1);
  else
      fileboot1 = fopen(fileboot1_NAME, "w");
  if (fileboot1 == NULL)
    _EscIO(FileNotFound);
  strcpy(fileboot2_NAME, nameboot2);
  if (fileboot2 != NULL)
      fileboot2 =  freopen(fileboot2_NAME, "w", fileboot2);
  else
      fileboot2 = fopen(fileboot2_NAME, "w");
  if (fileboot2 == NULL)
    _EscIO(FileNotFound);
  strcpy(fileboot3_NAME, nameboot3);
  if (fileboot3 != NULL)
      fileboot3 =  freopen(fileboot3_NAME, "w", fileboot3);
  else
      fileboot3 = fopen(fileboot3_NAME, "w");
  if (fileboot3 == NULL)
    _EscIO(FileNotFound);
 FORLIM = np;
  for (ip = 1; ip <= FORLIM; ip++) {
    h[ip - 1] = (float(*)[numax])Malloc(sizeof(peral));
    p[ip - 1] = (float(*)[numax])Malloc(sizeof(peral));
  }
  anbar = (double *)Malloc(sizeof(fst));
  annbar = (double *)Malloc(sizeof(fst));
  anc = (double *)Malloc(sizeof(fst));
  term1 = (double *)Malloc(sizeof(fst));
  term2 = (double *)Malloc(sizeof(fst));
  term3 = (double *)Malloc(sizeof(fst));
  term4 = (double *)Malloc(sizeof(fst));
  term5 = (double *)Malloc(sizeof(fst));
  pbar = (double(*)[numax])Malloc(sizeof(frq));
  ppbar = (double(*)[numax])Malloc(sizeof(frq));
  varp = (double(*)[numax])Malloc(sizeof(frq));
  hbar = (double(*)[numax])Malloc(sizeof(frq));
  capf = (double(*)[numax])Malloc(sizeof(frq));
  theta = (double(*)[numax])Malloc(sizeof(frq));
  smallf = (double(*)[numax])Malloc(sizeof(frq));
  capfl = (double *)Malloc(sizeof(fst));
  thetal = (double *)Malloc(sizeof(fst));
  smallfl = (double *)Malloc(sizeof(fst));
  an = (unsigned short(*)[nlmax])Malloc(sizeof(numb_per_loc));
  capfb = (float *)Malloc(sizeof(sorted));
  thetb = (float *)Malloc(sizeof(sorted));
  smlfb = (float *)Malloc(sizeof(sorted));
  readdata(&filein);
  basic_stats();
  npinf = 1;
  npsup = np;
  calcfstat(npinf, npsup, ans);
  jackknife(npinf, npsup, ans);
  bootstrap(ans);
  perpair();
  permwithin();
  permbetween();
  permmultgen();
  FORLIM = np;
  for (ip = 1; ip <= FORLIM; ip++) {
    Free(h[ip - 1]);
    Free(p[ip - 1]);
  }
  Free(capfb);
  Free(thetb);
  Free(smlfb);
  Free(an);
  Free(annbar);
  Free(anc);
  Free(term1);
  Free(term2);
  Free(term3);
  Free(term4);
  Free(term5);
  Free(pbar);
  Free(ppbar);
  Free(varp);
  Free(hbar);
  Free(capf);
  Free(theta);
  Free(smallf);
  Free(capfl);
  Free(thetal);
  Free(smallfl);
```

```
  if (fileout1 != NULL)
    fclose(fileout1);
  fileout1 = NULL;
  if (fileini != NULL)
    fileini = freopen(fileini_NAME, "w", fileini);
  else
    fileini = fopen(fileini_NAME, "w");
  if (fileini == NULL)
    _EscIO(FileNotFound);
  fprintf(fileini, "%ld   %ld\n", s1, s2);
  if (fileini != NULL)
    fclose(fileini);
  fileini = NULL;
  if (filein != NULL)
    fclose(filein);
  if (fileout1 != NULL)
    fclose(fileout1);
  if (fileini != NULL)
    fclose(fileini);
  if (filepar != NULL)
    fclose(filepar);
  if (filemig != NULL)
    fclose(filemig);
  if (fileboot1 != NULL)
      fclose(fileboot1);
  if (fileboot2 != NULL)
      fclose(fileboot2);
  if (fileboot3 != NULL)
      fclose(fileboot3);
  exit(EXIT_SUCCESS);
}
/* End. */
```

Appendix E

Genotypic composition of Brassica

```
  20   3   3
SDH-2
PGI-1
APH-2
   1  22  13  22
   1  22  11  22
   1  22  13  22
   1  22  11  23
   1  22  11  23
   1  22  11  23
   1  22  11  23
   1  22  23  22
   1  22  11  33
   1  22  11  33
   1  22  11  23
   1  22  11  22
   1  22  11  22
   1  22  23  23
   1  22  11  23
   1  22  11  22
   1  22  11  22
   1  22  11  22
   1  22  11  22
   1  22  23  22
   2  22  11  22
   2  22  12  22
   2  22  12  22
   2  22  12  33
   2  22  22  33
   2  22  33  23
   2  22  33  23
   2  22  22  23
   2  22  33  23
   2  22  12  22
   2  22  12  22
   2  22  12  22
   2  22  33  22
   2  22  23  23
   2  22  23  22
   2  22  22  22
   2  22  23  23
   2  22  23  23
   2  22  23  23
   2  22  23  23
   3  12  11  22
   3  22  22  23
   3  12  11  23
   3  11  11  22
   3  11  22  22
   3  22  22  22
   3  22  22  22
   3  22  22  22
   3  11  22  22
   3  22  22  22
   3  11  12  22
   3  11  22  22
   3  11  11  22
   3  22  23  23
   3  22  12  22
   3  11  12  22
   3  22  22  22
   3  22  11  22
   3  22  33  23
   3  22  33  23
   4  22  23  22
   4  22  23  12
   4  22  23  23
   4  22  12  22
   4  22  12  12
   4  22  22  23
   4  22  22  22
   4  22  12  12
   4  22  12  12
   4  22  23  22
   4  22  23  22
   4  22  22  23
   4  22  12  23
   4  22  22  23
   4  22  22  23
   4  22  22  23
   4  22  33  23
   4  22  33  23
   4  22  23  23
   4  22  12  23
   5  22  23  22
   5  22  22  22
   5  22  22  22
   5  22  11  23
   5  22  22  22
   5  22  23  12
   5  22  22  13
   5  22  22  23
   5  12  12  22
   5  22  22  33
   5  22  33  22
   5  22  22  22
   5  12  23  23
   5  22  23  23
   5  12  22  22
   5  22  22  22
   5  22  23  23
   5  12  23  22
   5  22  23  11
   5  22  33  22
   6  22  22  22
   6  22  33  23
   6  11  23  23
   6  12  12  12
   6  22  11  22
   6  22  22  22
   6  22  22  33
   6  22  22  22
   6  22  33  23
   6  22  23  23
   6  22  12  22
   6  22  23  33
   6  22  33  12
   6  22  22  23
   6  22  22  23
   6  11  12  22
   6  22  23  33
   6  22  23  33
   6  22  33  33
   6  22  23  22
   7  22  22  12
   7  22  23  33
   7  22  11  22
   7  22  12  22
   7  12  22  12
   7  12  22  22
   7  22  22  23
   7  22  22  12
   7  22  23  22
   7  22  22  22
   7  22  22  22
   7  22  22  12
   7  22  12  23
   7  22  22  22
   7  22  22  33
   7  12  11  13
   7  22  22  22
   7  22  23  12
   7  12  22  12
   7  22  22  22
   8  22  22  11
   8  22  33  22
   8  22  33  22
   8  22  33  22
   8  22  22  22
   8  22  22  22
   8  12  11  11
   8  22  23  22
   8  22  23  22
   8  22  11  22
   8  22  22  23
   8  22  33  33
   8  22  22  23
   8  22  11  22
   8  22  00  23
   8  11  00  22
   8  22  33  23
   8  22  22  22
   8  22  33  23
   8  22  33  22
   9  22  22  22
   9  11  22  23
   9  22  23  23
   9  22  22  12
   9  22  22  22
   9  22  23  23
   9  22  23  33
   9  22  23  22
   9  22  22  23
   9  22  11  23
   9  22  11  22
   9  22  11  23
   9  22  22  12
   9  22  22  23
   9  22  23  23
   9  22  13  22
   9  22  22  22
   9  22  33  22
   9  22  13  22
   9  22  12  22
  10  22  11  22
  10  22  11  22
  10  22  12  22
  10  22  12  22
  10  22  11  22
  10  22  22  22
  10  22  12  22
  10  22  12  22
  10  22  12  22
  10  00  11  22
  10  11  12  22
  10  22  22  22
  10  22  11  22
  10  22  11  22
  10  22  11  12
  10  22  11  22
  10  22  11  22
  10  22  23  12
  10  22  13  22
  10  22  23  22
  11  22  12  22
  11  22  22  22
  11  22  22  22
  11  22  22  22
  11  22  22  12
  11  12  22  22
  11  22  22  12
  11  22  22  12
  11  12  33  22
  11  22  12  22
  11  22  12  22
  11  12  23  22
  11  22  22  22
  11  22  23  22
  11  12  12  22
  11  12  12  12
  11  22  12  22
  11  12  12  22
  11  22  11  22
  11  22  11  12
  12  22  11  12
  12  22  12  22
  12  22  11  22
  12  22  22  22
  12  22  11  22
  12  22  23  12
  12  12  22  22
  12  22  22  22
  12  22  22  22
  12  22  22  11
  12  22  22  12
  12  22  22  22
  12  22  22  12
  12  22  33  22
  12  22  22  22
  12  22  22  22
  12  22  12  22
  12  22  12  22
  12  22  23  22
  12  22  12  12
  13  22  22  33
  13  22  11  33
```

```
13 12 11 23
13 22 11 33
13 22 11 13
13 22 11 13
13 12 11 13
13 22 11 13
13 22 11 13
13 22 11 22
13 22 23 13
13 22 11 13
13 22 11 12
13 22 11 23
13 22 11 22
13 22 11 13
13 22 11 22
13 22 11 33
13 22 11 33
13 22 11 33
14 22 11 22
14 12 11 22
14 12 33 22
14 22 22 12
14 22 22 22
14 22 22 22
14 22 22 22
14 22 11 22
14 22 12 22
14 22 22 22
14 22 22 22
14 22 23 22
14 22 22 22
14 22 23 22
14 22 23 22
14 22 23 22
14 22 33 22
14 22 33 22
14 22 23 22
14 22 33 22
15 22 12 33
15 22 12 13
15 22 12 33
15 22 11 23
15 22 12 23
15 22 13 33
15 22 22 33
15 22 22 12
15 22 22 33
15 22 11 33
15 22 13 13
15 22 12 13
15 22 11 13
15 22 22 13
15 22 11 33
15 22 11 33
15 22 11 23
15 22 11 13
15 22 11 33
15 22 13 23
16 22 11 22
16 22 11 22
16 22 12 22
16 22 12 22
16 22 11 22
16 22 11 22
16 22 11 22
16 22 22 22
16 22 11 22
16 22 11 22
16 22 11 22
16 22 13 22
16 22 13 22
16 22 13 22
16 22 13 23
16 22 13 33
16 22 13 33
16 22 13 12
16 22 13 22
16 22 13 22
17 22 13 22
17 12 13 22
17 12 13 12
17 22 23 22
17 22 11 22
17 22 23 22
17 12 11 22
17 22 13 00
17 22 13 00
17 22 33 22
17 22 13 22
17 22 13 22
17 22 11 22
17 22 13 13
17 22 13 22
17 22 13 22
17 22 13 13
17 22 22 22
17 22 12 22
17 22 13 00
18 22 22 22
18 12 22 12
18 22 11 22
18 22 12 22
18 22 12 22
18 22 23 12
18 22 12 22
18 12 12 22
18 22 12 22
18 22 22 22
18 22 11 13
18 22 12 13
18 22 12 11
18 22 23 11
18 22 23 12
18 22 23 22
18 22 23 22
18 22 23 12
18 22 11 22
18 22 13 12
19 22 12 22
19 22 00 22
19 22 12 22
19 22 12 22
19 22 23 22
```

```
19 22 23 22
19 22 11 22
19 22 22 22
19 22 33 22
19 22 12 23
19 22 33 22
19 22 33 22
19 22 33 22
19 22 22 22
19 22 33 22
19 22 23 22
19 22 33 22
19 22 22 22
19 22 33 22
19 22 22 22
20 22 22 33
20 22 33 33
20 22 13 33
20 22 23 33
20 22 23 33
20 22 13 33
20 22 13 33
20 22 13 33
20 22 22 33
20 22 33 33
20 22 22 33
20 22 11 33
20 22 13 33
20 22 13 33
20 22 12 22
20 22 33 22
20 22 33 22
20 22 22 22
20 22 13 22
20 22 12 22
```

Appendix F

Raw output of the treatment of Brassica data

```
          1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20
   locus: SDH-2
     n   20   20   20   20   20   20   20   20   20   19   20   20   20   20   20   20   20   20   20   20
p:   1 0.00 0.00 0.40 0.00 0.10 0.12 0.10 0.07 0.05 0.05 0.15 0.03 0.05 0.05 0.00 0.00 0.07 0.05 0.00 0.00
p:   2 1.00 1.00 0.60 1.00 0.90 0.88 0.90 0.93 0.95 0.95 0.85 0.97 0.95 0.95 1.00 1.00 0.93 0.95 1.00 1.00
ho:  1 0.00 0.00 0.10 0.00 0.20 0.05 0.20 0.05 0.00 0.00 0.30 0.05 0.10 0.10 0.00 0.00 0.15 0.10 0.00 0.00
ho:  2 0.00 0.00 0.10 0.00 0.20 0.05 0.20 0.05 0.00 0.00 0.30 0.05 0.10 0.10 0.00 0.00 0.15 0.10 0.00 0.00
he:  1 0.00 0.00 0.49 0.00 0.18 0.22 0.18 0.14 0.10 0.10 0.26 0.05 0.10 0.10 0.00 0.00 0.14 0.10 0.00 0.00
he:  2 0.00 0.00 0.49 0.00 0.18 0.22 0.18 0.14 0.10 0.10 0.26 0.05 0.10 0.10 0.00 0.00 0.14 0.10 0.00 0.00
   locus: PGI-1
     n   20   20   20   20   20   20   20   18   20   20   20   20   20   20   20   20   20   20   19   20
p:   1 0.80 0.20 0.33 0.15 0.07 0.12 0.15 0.17 0.23 0.62 0.28 0.25 0.90 0.17 0.60 0.68 0.47 0.35 0.16 0.28
p:   2 0.07 0.45 0.55 0.60 0.65 0.53 0.78 0.39 0.55 0.30 0.62 0.65 0.07 0.50 0.33 0.10 0.12 0.47 0.39 0.30
p:   3 0.12 0.35 0.12 0.25 0.28 0.35 0.07 0.44 0.23 0.07 0.10 0.10 0.03 0.33 0.07 0.23 0.40 0.17 0.45 0.42
ho:  1 0.10 0.30 0.15 0.30 0.05 0.15 0.10 0.00 0.15 0.35 0.35 0.20 0.00 0.05 0.40 0.55 0.65 0.40 0.21 0.45
ho:  2 0.15 0.60 0.20 0.60 0.40 0.45 0.25 0.11 0.30 0.40 0.45 0.30 0.05 0.30 0.25 0.10 0.15 0.65 0.37 0.20
ho:  3 0.25 0.30 0.05 0.30 0.35 0.30 0.15 0.11 0.35 0.15 0.10 0.10 0.05 0.25 0.15 0.45 0.70 0.35 0.16 0.45
he:  1 0.33 0.33 0.45 0.26 0.14 0.22 0.26 0.29 0.36 0.48 0.41 0.38 0.18 0.30 0.49 0.45 0.51 0.47 0.27 0.41
he:  2 0.14 0.51 0.51 0.49 0.47 0.51 0.36 0.49 0.51 0.43 0.48 0.47 0.14 0.51 0.45 0.18 0.22 0.51 0.49 0.43
he:  3 0.22 0.47 0.22 0.38 0.41 0.47 0.14 0.51 0.36 0.14 0.18 0.18 0.05 0.45 0.14 0.36 0.49 0.30 0.51 0.50
   locus: APH-2
     n   20   20   20   20   20   20   20   20   20   20   20   20   20   20   20   20   17   20   20   20
p:   1 0.00 0.00 0.00 0.10 0.10 0.05 0.17 0.10 0.05 0.05 0.12 0.17 0.23 0.03 0.17 0.03 0.09 0.28 0.00 0.00
p:   2 0.72 0.68 0.88 0.62 0.70 0.55 0.65 0.72 0.70 0.95 0.88 0.82 0.23 0.97 0.12 0.85 0.85 0.68 0.97 0.30
p:   3 0.28 0.33 0.12 0.28 0.20 0.40 0.17 0.17 0.25 0.00 0.00 0.00 0.55 0.00 0.70 0.12 0.06 0.05 0.03 0.70
ho:  1 0.00 0.00 0.00 0.20 0.10 0.10 0.35 0.00 0.10 0.10 0.25 0.25 0.45 0.05 0.35 0.05 0.18 0.35 0.00 0.00
ho:  2 0.35 0.45 0.25 0.75 0.30 0.40 0.40 0.25 0.50 0.10 0.25 0.25 0.15 0.05 0.25 0.10 0.06 0.25 0.05 0.00
ho:  3 0.35 0.45 0.25 0.55 0.30 0.30 0.15 0.25 0.40 0.00 0.00 0.00 0.50 0.00 0.50 0.05 0.12 0.10 0.05 0.00
he:  1 0.00 0.00 0.00 0.18 0.18 0.10 0.30 0.18 0.10 0.10 0.22 0.30 0.36 0.05 0.30 0.05 0.17 0.41 0.00 0.00
he:  2 0.41 0.45 0.22 0.48 0.43 0.51 0.47 0.41 0.43 0.10 0.22 0.30 0.36 0.05 0.22 0.26 0.26 0.45 0.05 0.43
he:  3 0.41 0.45 0.22 0.41 0.33 0.49 0.30 0.30 0.38 0.00 0.00 0.00 0.51 0.00 0.43 0.22 0.11 0.10 0.05 0.43
 for locus : SDH-2
fis: 1  ???  ???  0.8  ??? -0.1  0.8 -0.1  0.6  1.0  1.0 -0.1  0.0  0.0  0.0  ???  ??? -0.1  0.0  ???  ???
fis: 2  ???  ???  0.8  ??? -0.1  0.8 -0.1  0.6  1.0  1.0 -0.1  0.0  0.0  0.0  ???  ??? -0.1  0.0  ???  ???
 for locus : PGI-1
fis: 1  0.7  0.1  0.7 -0.1  0.6  0.3  0.6  1.0  0.6  0.3  0.1  0.5  1.0  0.8  0.2 -0.2 -0.3  0.1  0.2 -0.1
fis: 2 -0.1 -0.2  0.6 -0.2  0.1  0.1  0.3  0.8  0.4  0.1  0.1  0.4  0.6  0.4  0.4  0.5  0.3 -0.3  0.2  0.5
fis: 3 -0.1  0.4  0.8  0.2  0.1  0.4 -0.1  0.8  0.0 -0.1  0.5  0.5  0.0  0.4 -0.1 -0.3 -0.4 -0.2  0.7  0.1
 for locus : APH-2
fis: 1  ???  ???  ??? -0.1  0.5  0.0 -0.2  1.0  0.0  0.0 -0.1  0.2 -0.3  0.0 -0.2  0.0 -0.1  0.1  ???  ???
fis: 2  0.1  0.0 -0.1 -0.6  0.3  0.2  0.1  0.4 -0.2  0.0 -0.1  0.2  0.6  0.0 -0.1  0.6  0.8  0.4  0.0  1.0
fis: 3  0.1  0.0 -0.1 -0.3  0.1  0.4  0.5  0.2  0.0  ???  ???  ???  0.0  ??? -0.2  0.8  0.0  0.0  0.0  1.0
```

```
************************************************
 for locus : SDH-2
allele   capf      theta     smallf
    1    0.4279    0.1053    0.3606
    2    0.4279    0.1053    0.3606
 all     0.4279    0.1053    0.3606
 for locus : PGI-1
allele   capf      theta     smallf
    1    0.4646    0.2341    0.3010
    2    0.3603    0.1518    0.2458
    3    0.2812    0.0816    0.2173
 all     0.3757    0.1618    0.2551
 for locus : APH-2
allele   capf      theta     smallf
    1    0.0992    0.0614    0.0403
    2    0.4000    0.2420    0.2085
    3    0.3810    0.2586    0.1651
 all     0.3421    0.2176    0.1590

 over all loci
          capf      theta     smallf
         0.3681    0.1773    0.2319
************************************************
jackknifing over populations.
 for locus : SDH-2
          capf      theta     smallf
 total    0.4686    0.1246    0.3794 means
          0.2103    0.0887    0.1743 std. devs.
 for locus : PGI-1
          capf      theta     smallf
 total    0.3756    0.1621    0.2548 means
          0.0625    0.0430    0.0646 std. devs.
 for locus : APH-2
          capf      theta     smallf
 total    0.3458    0.2217    0.1586 means
          0.0892    0.0658    0.0812 std. devs.

 jackknifing over loci.
          capf      theta     smallf
 total    0.3675    0.1770    0.2327 means
          0.0155    0.0252    0.0417 std. devs.
************************************************
         bootstrapping over loci.
         95% confidence interval.
      capf     theta    smallf
    0.3421    0.1053    0.1590
    0.4279    0.2176    0.3606

         99% confidence interval.
      capf     theta    smallf
    0.3421    0.1053    0.1590
    0.4279    0.2176    0.3606
```

```
  theta per locus over pair of populations.
 for locus: SDH-2
           1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17    18    19    20
     1              0.37        0.08  0.08  0.08  0.04  0.00  0.00  0.13  0.00  0.03  0.03              0.05  0.03
     2              0.37        0.08  0.08  0.08  0.04  0.00  0.00  0.13  0.00  0.03  0.03              0.05  0.03
     3  0.37  0.37        0.37  0.18  0.14  0.18  0.22  0.27  0.26  0.11  0.32  0.27  0.27  0.37  0.37  0.22  0.27  0.37  0.37
     4              0.37        0.08  0.08  0.08  0.04  0.00  0.00  0.13  0.00  0.03  0.03              0.05  0.03
     5  0.08  0.08  0.18  0.08       -0.03 -0.02 -0.03 -0.02 -0.02 -0.01  0.02 -0.01 -0.01  0.08  0.08 -0.02 -0.01  0.08  0.08
     6  0.08  0.08  0.14  0.08 -0.03       -0.03 -0.03 -0.01 -0.02 -0.03  0.03  0.00  0.00  0.08  0.08 -0.02  0.00  0.08  0.08
     7  0.08  0.08  0.18  0.08 -0.02 -0.03       -0.03 -0.02 -0.02 -0.01  0.02 -0.01 -0.01  0.08  0.08 -0.02 -0.01  0.08  0.08
     8  0.04  0.04  0.22  0.04 -0.03 -0.03 -0.03       -0.04 -0.04  0.00 -0.01 -0.03 -0.03  0.04  0.04 -0.03 -0.03  0.04  0.04
     9  0.00  0.00  0.27  0.00 -0.02 -0.01 -0.02 -0.04       -0.05  0.03 -0.03 -0.04 -0.04  0.00  0.00 -0.03 -0.04  0.00  0.00
    10  0.00  0.00  0.26  0.00 -0.02 -0.02 -0.02 -0.04 -0.05        0.02 -0.03 -0.04 -0.04  0.00  0.00 -0.03 -0.04  0.00  0.00
    11  0.13  0.13  0.11  0.13 -0.01 -0.03 -0.01  0.00  0.03  0.02        0.07  0.03  0.03  0.13  0.13  0.01  0.03  0.13  0.13
    12  0.00  0.00  0.32  0.00  0.02  0.03  0.02 -0.01 -0.03 -0.03  0.07       -0.02 -0.02  0.00  0.00  0.00 -0.02  0.00  0.00
    13  0.03  0.03  0.27  0.03 -0.01  0.00 -0.01 -0.03 -0.04 -0.04  0.03 -0.02       -0.02  0.03  0.03 -0.02 -0.02  0.03  0.03
    14  0.03  0.03  0.27  0.03 -0.01  0.00 -0.01 -0.03 -0.04 -0.04  0.03 -0.02 -0.02        0.03  0.03 -0.02 -0.02  0.03  0.03
    15              0.37        0.08  0.08  0.08  0.04  0.00  0.00  0.13  0.00  0.03  0.03              0.05  0.03
    16              0.37        0.08  0.08  0.08  0.04  0.00  0.00  0.13  0.00  0.03  0.03              0.05  0.03
    17  0.05  0.05  0.22  0.05 -0.02 -0.02 -0.02 -0.03 -0.03 -0.03  0.01  0.00 -0.02 -0.02  0.05  0.05       -0.02  0.05  0.05
    18  0.03  0.03  0.27  0.03 -0.01  0.00 -0.01 -0.03 -0.04 -0.04  0.03 -0.02 -0.02 -0.02  0.03  0.03 -0.02        0.03  0.03
    19              0.37        0.08  0.08  0.08  0.04  0.00  0.00  0.13  0.00  0.03  0.03              0.05  0.03
    20              0.37        0.08  0.08  0.08  0.04  0.00  0.00  0.13  0.00  0.03  0.03              0.05  0.03
 for locus: PGI-1
           1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17    18    19    20
     1        0.34  0.30  0.43  0.50  0.42  0.55  0.36  0.36  0.06  0.38  0.41  0.00  0.37  0.08  0.00  0.14  0.26  0.37  0.27
     2  0.34        0.03  0.00  0.02 -0.02  0.13 -0.03 -0.01  0.17  0.05  0.06  0.46 -0.03  0.15  0.23  0.11  0.02 -0.02  0.00
     3  0.30  0.03        0.01  0.05  0.04  0.04  0.07 -0.02  0.09 -0.03 -0.03  0.40  0.01  0.07  0.21  0.17 -0.03  0.08  0.08
     4  0.43  0.00  0.01       -0.02 -0.01  0.04  0.03 -0.02  0.22  0.01  0.00  0.53 -0.02  0.20  0.32  0.22  0.03  0.04  0.08
     5  0.50  0.02  0.05 -0.02       -0.01  0.04  0.05  0.00  0.29  0.04  0.03  0.60  0.00  0.27  0.39  0.27  0.07  0.05  0.11
     6  0.42 -0.02  0.04 -0.01 -0.01        0.10 -0.02 -0.01  0.23  0.05  0.05  0.53 -0.03  0.21  0.30  0.17  0.04 -0.01  0.03
     7  0.55  0.13  0.04  0.04  0.04  0.10        0.19  0.04  0.32  0.01 -0.01  0.64  0.09  0.29  0.45  0.38  0.10  0.19  0.24
     8  0.36 -0.03  0.07  0.03  0.05 -0.02  0.19        0.02  0.21  0.10  0.11  0.48 -0.02  0.19  0.23  0.09  0.05 -0.05 -0.02
     9  0.36 -0.01 -0.02 -0.02  0.00 -0.01  0.04  0.02        0.15 -0.01 -0.01  0.46 -0.02  0.13  0.25  0.16 -0.01  0.03  0.05
    10  0.06  0.17  0.09  0.22  0.29  0.23  0.32  0.21  0.15        0.16  0.18  0.13  0.19 -0.03  0.04  0.10  0.07  0.22  0.15
    11  0.38  0.05 -0.03  0.01  0.04  0.05  0.01  0.10 -0.01  0.16       -0.03  0.48  0.03  0.13  0.29  0.23  0.00  0.11  0.13
    12  0.41  0.06 -0.03  0.00  0.03  0.05 -0.01  0.11 -0.01  0.18 -0.03        0.51  0.03  0.15  0.32  0.25  0.01  0.12  0.14
    13  0.00  0.46  0.40  0.53  0.60  0.53  0.64  0.48  0.46  0.13  0.48  0.51        0.48  0.15  0.09  0.27  0.36  0.49  0.40
    14  0.37 -0.03  0.01 -0.02  0.00 -0.03  0.09 -0.02 -0.02  0.19  0.03  0.03  0.48        0.16  0.25  0.14  0.01 -0.02  0.01
    15  0.08  0.15  0.07  0.20  0.27  0.21  0.29  0.19  0.13 -0.03  0.13  0.15  0.15  0.16        0.05  0.10  0.05  0.20  0.14
    16  0.00  0.23  0.21  0.32  0.39  0.30  0.45  0.23  0.25  0.04  0.29  0.32  0.09  0.25  0.05        0.04  0.17  0.25  0.15
    17  0.14  0.11  0.17  0.22  0.27  0.17  0.38  0.09  0.16  0.10  0.23  0.25  0.27  0.14  0.10  0.04        0.12  0.10  0.03
    18  0.26  0.02 -0.03  0.03  0.07  0.04  0.10  0.05 -0.01  0.07  0.00  0.01  0.36  0.01  0.05  0.17  0.12        0.06  0.05
    19  0.37 -0.02  0.08  0.04  0.05 -0.01  0.19 -0.05  0.03  0.22  0.11  0.12  0.49 -0.02  0.20  0.25  0.10  0.06       -0.02
    20  0.27  0.00  0.08  0.08  0.11  0.03  0.24 -0.02  0.05  0.15  0.13  0.14  0.40  0.01  0.14  0.15  0.03  0.05 -0.02
 for locus: APH-2
           1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17    18    19    20
     1       -0.02  0.04  0.00 -0.01  0.02  0.02 -0.01 -0.02  0.18  0.13  0.12  0.25  0.21  0.38  0.02  0.06  0.10  0.20  0.28
     2 -0.02        0.09  0.00  0.00 -0.01  0.03  0.01 -0.01  0.24  0.18  0.16  0.21  0.27  0.33  0.06  0.11  0.12  0.25  0.22
     3  0.04  0.09        0.10  0.04  0.17  0.08  0.02  0.05  0.05  0.05  0.06  0.43  0.07  0.57 -0.03  0.00  0.13  0.05  0.49
     4  0.00  0.00  0.10       -0.01  0.00 -0.01  0.00 -0.01  0.22  0.14  0.11  0.17  0.25  0.29  0.07  0.09  0.06  0.24  0.21
     5 -0.01  0.00  0.04 -0.01        0.03 -0.02 -0.03 -0.02  0.13  0.07  0.05  0.23  0.17  0.37  0.01  0.02  0.02  0.16  0.29
     6  0.02 -0.01  0.17  0.00  0.03        0.04  0.05  0.02  0.32  0.24  0.21  0.10  0.35  0.20  0.14  0.17  0.13  0.34  0.10
     7  0.02  0.03  0.08 -0.01 -0.02  0.04       -0.02  0.00  0.16  0.08  0.04  0.20  0.20  0.34  0.05  0.04  0.00  0.20  0.29
     8 -0.01  0.01  0.02  0.00 -0.03  0.05 -0.02       -0.02  0.11  0.05  0.03  0.26  0.14  0.40 -0.01  0.00  0.02  0.13  0.32
     9 -0.02 -0.01  0.05 -0.01 -0.02  0.02  0.00 -0.02        0.17  0.11  0.09  0.23  0.20  0.36  0.02  0.05  0.07  0.19  0.27
    10  0.18  0.24  0.05  0.22  0.13  0.32  0.16  0.11  0.17        0.01  0.05  0.54 -0.02  0.67  0.03  0.01  0.16  0.00  0.62
    11  0.13  0.18  0.05  0.14  0.07  0.24  0.08  0.05  0.11  0.01       -0.02  0.46  0.05  0.60  0.02 -0.02  0.06  0.07  0.55
    12  0.12  0.16  0.06  0.11  0.05  0.21  0.04  0.03  0.09  0.05 -0.02        0.41  0.09  0.55  0.03 -0.01  0.01  0.11  0.51
    13  0.25  0.21  0.43  0.17  0.23  0.10  0.20  0.26  0.23  0.54  0.46  0.41        0.57  0.01  0.40  0.41  0.28  0.57  0.04
    14  0.21  0.27  0.07  0.25  0.17  0.35  0.20  0.14  0.20 -0.02  0.05  0.09  0.57        0.70  0.06  0.04  0.21 -0.01  0.65
    15  0.38  0.33  0.57  0.29  0.37  0.20  0.34  0.40  0.36  0.67  0.60  0.55  0.01  0.70        0.53  0.55  0.43  0.70  0.03
    16  0.02  0.06 -0.03  0.07  0.01  0.14  0.05 -0.01  0.02  0.03  0.02  0.03  0.40  0.06  0.53       -0.03  0.09  0.04  0.46
    17  0.06  0.11  0.00  0.09  0.02  0.17  0.04  0.00  0.05  0.01 -0.02 -0.01  0.41  0.04  0.55 -0.03        0.05  0.04  0.49
    18  0.10  0.12  0.13  0.06  0.02  0.13  0.00  0.02  0.07  0.16  0.06  0.01  0.28  0.21  0.43  0.09  0.05        0.22  0.39
    19  0.20  0.25  0.05  0.24  0.16  0.34  0.20  0.13  0.19  0.00  0.07  0.11  0.57 -0.01  0.70  0.04  0.04  0.22        0.64
    20  0.28  0.22  0.49  0.21  0.29  0.10  0.29  0.32  0.27  0.62  0.55  0.51  0.04  0.65  0.03  0.46  0.49  0.39  0.64
************************************************
         permutting alleles within samples.

         95% confidence interval.

      capf     theta    smallf
    0.1401    0.1811   -0.0530
    0.2269    0.1834    0.0559

         99% confidence interval.

      capf     theta    smallf
    0.1279    0.1807   -0.0684
    0.2430    0.1837    0.0761

(prob fis=0)<    0.00020
************************************************
         permutting alleles within total.

         95% confidence interval.

      capf     theta    smallf
   -0.0490   -0.0075   -0.0494
    0.0489    0.0087    0.0502

         99% confidence interval.

      capf     theta    smallf
   -0.0613   -0.0093   -0.0621
    0.0632    0.0122    0.0630

(prob fit=0)<    0.00020
************************************************
         permutting genotypes within total.

         95% confidence interval.

      capf     theta    smallf
    0.3624   -0.0101    0.3553
    0.3631    0.0121    0.3688

         99% confidence interval.

      capf     theta    smallf
    0.3623   -0.0128    0.3527
    0.3632    0.0161    0.3704

(prob fst=0)<    0.00020
```

Appendix G

Raw output of the treatment of Beta data

```
          1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20
    locus: GOT-3
     n   20   20   20   20   20   20   20   20   20   20   20   20   20   20   20   20   20   20   20   20
p:   1 0.30 0.82 0.53 0.57 0.85 0.53 0.40 0.17 0.90 0.60 0.55 0.33 0.38 0.82 0.62 0.95 0.65 0.50 0.45 0.38
p:   2 0.70 0.17 0.47 0.42 0.15 0.47 0.60 0.82 0.10 0.40 0.45 0.68 0.62 0.17 0.38 0.05 0.35 0.50 0.55 0.62
ho:  1 0.40 0.35 0.75 0.55 0.30 0.45 0.40 0.15 0.20 0.60 0.70 0.25 0.55 0.15 0.65 0.10 0.60 0.50 0.40 0.45
ho:  2 0.40 0.35 0.75 0.55 0.30 0.45 0.40 0.15 0.20 0.60 0.70 0.25 0.55 0.15 0.65 0.10 0.60 0.50 0.40 0.45
he:  1 0.43 0.30 0.51 0.50 0.26 0.51 0.49 0.30 0.18 0.49 0.51 0.45 0.48 0.30 0.48 0.10 0.47 0.51 0.51 0.48
he:  2 0.43 0.30 0.51 0.50 0.26 0.51 0.49 0.30 0.18 0.49 0.51 0.45 0.48 0.30 0.48 0.10 0.47 0.51 0.51 0.48
    locus: APH
     n   20   20   20   20   20   20   20   19   20   20   20   20   20   20   20   20   20   20   20   20
p:   1 0.30 0.23 0.10 0.40 0.38 0.35 0.10 0.37 0.80 0.68 0.53 0.42 0.17 0.12 0.12 0.45 0.07 0.50 0.05 0.40
p:   2 0.45 0.72 0.62 0.40 0.07 0.40 0.20 0.00 0.15 0.03 0.07 0.40 0.57 0.85 0.75 0.47 0.62 0.28 0.85 0.40
p:   3 0.25 0.05 0.28 0.20 0.55 0.25 0.70 0.63 0.05 0.30 0.40 0.17 0.25 0.03 0.12 0.07 0.30 0.23 0.10 0.20
ho:  1 0.60 0.45 0.20 0.50 0.65 0.40 0.10 0.32 0.40 0.45 0.65 0.15 0.25 0.05 0.25 0.40 0.15 0.20 0.10 0.60
ho:  2 0.70 0.55 0.65 0.40 0.15 0.40 0.40 0.00 0.30 0.05 0.15 0.30 0.15 0.10 0.30 0.35 0.55 0.15 0.20 0.80
ho:  3 0.50 0.10 0.55 0.40 0.60 0.30 0.50 0.32 0.10 0.50 0.80 0.15 0.40 0.05 0.25 0.15 0.60 0.05 0.20 0.40
he:  1 0.43 0.36 0.18 0.49 0.48 0.47 0.18 0.48 0.33 0.45 0.51 0.50 0.30 0.22 0.22 0.51 0.14 0.51 0.10 0.49
he:  2 0.51 0.41 0.48 0.49 0.14 0.49 0.33 0.00 0.26 0.05 0.14 0.49 0.50 0.26 0.38 0.51 0.48 0.41 0.26 0.49
he:  3 0.38 0.10 0.41 0.33 0.51 0.38 0.43 0.48 0.10 0.43 0.49 0.30 0.38 0.05 0.22 0.14 0.43 0.36 0.18 0.33
    locus: SDH
     n   19   19   19   19   19   19   19   19   19   19   19   19   19   19   19   19   19   19   19   19
p:   1 0.92 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.11 1.00 0.97 1.00 1.00 1.00 1.00 0.84
p:   2 0.08 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.89 0.00 0.03 0.00 0.00 0.00 0.00 0.16
ho:  1 0.16 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.11 0.00 0.05 0.00 0.00 0.00 0.00 0.11
ho:  2 0.16 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.11 0.00 0.05 0.00 0.00 0.00 0.00 0.11
he:  1 0.15 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.19 0.00 0.05 0.00 0.00 0.00 0.00 0.27
he:  2 0.15 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.19 0.00 0.05 0.00 0.00 0.00 0.00 0.27
    locus: PGI
     n   20   20   20   20   20   20   20   20   20   20   20   20    0   20   20   20    0   18    0    0
p:   1 0.60 0.50 0.50 0.50 0.50 0.38 0.50 0.70 0.50 0.38 0.50 0.65 ???? 0.23 0.50 0.50 ???? 0.61 ???? ????
p:   2 0.38 0.50 0.50 0.50 0.50 0.62 0.50 0.30 0.50 0.62 0.50 0.35 ???? 0.75 0.50 0.50 ???? 0.39 ???? ????
p:   3 0.03 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 ???? 0.03 0.00 0.00 ???? 0.00 ???? ????
ho:  1 0.70 1.00 1.00 1.00 1.00 0.45 1.00 0.60 1.00 0.75 1.00 0.40 ???? 0.35 1.00 1.00 ???? 0.22 ???? ????
ho:  2 0.65 1.00 1.00 1.00 1.00 0.45 1.00 0.60 1.00 0.75 1.00 0.40 ???? 0.40 1.00 1.00 ???? 0.22 ???? ????
ho:  3 0.05 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 ???? 0.05 0.00 0.00 ???? 0.00 ???? ????
he:  1 0.49 0.51 0.51 0.51 0.51 0.48 0.51 0.43 0.51 0.48 0.51 0.47 ???? 0.36 0.51 0.51 ???? 0.49 ???? ????
he:  2 0.48 0.51 0.51 0.51 0.51 0.48 0.51 0.43 0.51 0.48 0.51 0.47 ???? 0.38 0.51 0.51 ???? 0.49 ???? ????
he:  3 0.05 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 ???? 0.05 0.00 0.00 ???? 0.00 ???? ????
    locus: PER-1
     n   20   20   20   19   20   20   15   19   20   20   20   20    0    0   20   20   18    0   11   20
p:   1 0.60 0.82 0.93 0.50 0.42 0.70 0.23 0.58 0.90 0.57 0.97 0.53 ???? ???? 0.42 0.38 0.31 ???? 0.73 0.20
p:   2 0.40 0.17 0.07 0.50 0.57 0.30 0.77 0.42 0.10 0.42 0.03 0.47 ???? ???? 0.57 0.62 0.69 ???? 0.27 0.80
ho:  1 0.80 0.35 0.15 0.79 0.85 0.60 0.47 0.84 0.20 0.85 0.05 0.95 ???? ???? 0.85 0.65 0.39 ???? 0.55 0.00
ho:  2 0.80 0.35 0.15 0.79 0.85 0.60 0.47 0.84 0.20 0.85 0.05 0.95 ???? ???? 0.85 0.65 0.39 ???? 0.55 0.00
he:  1 0.49 0.30 0.14 0.51 0.50 0.43 0.37 0.50 0.18 0.50 0.05 0.51 ???? ???? 0.50 0.48 0.44 ???? 0.42 0.33
he:  2 0.49 0.30 0.14 0.51 0.50 0.43 0.37 0.50 0.18 0.50 0.05 0.51 ???? ???? 0.50 0.48 0.44 ???? 0.42 0.33
    locus: HDH
     n   20   20   20   20   20   20   20   20   20   20   20   20   20   20   20   20   20   20   20   20
p:   1 0.42 0.23 0.50 0.50 0.40 0.35 0.47 0.10 0.38 0.50 0.55 0.53 0.50 0.42 0.40 0.42 0.30 0.55 0.12 0.50
p:   2 0.57 0.78 0.50 0.50 0.60 0.65 0.53 0.90 0.62 0.50 0.45 0.47 0.50 0.57 0.60 0.57 0.70 0.45 0.88 0.50
ho:  1 0.35 0.15 0.80 1.00 0.80 0.60 0.75 0.20 0.75 0.90 0.70 0.95 0.90 0.85 0.80 0.85 0.40 0.90 0.15 0.80
ho:  2 0.35 0.15 0.80 1.00 0.80 0.60 0.75 0.20 0.75 0.90 0.70 0.95 0.90 0.85 0.80 0.85 0.40 0.90 0.15 0.80
he:  1 0.50 0.36 0.51 0.51 0.49 0.47 0.51 0.18 0.48 0.51 0.51 0.51 0.51 0.50 0.49 0.50 0.43 0.51 0.22 0.51
he:  2 0.50 0.36 0.51 0.51 0.49 0.47 0.51 0.18 0.48 0.51 0.51 0.51 0.51 0.50 0.49 0.50 0.43 0.51 0.22 0.51
 for locus : GOT-3
fis: 1  0.1 -0.2 -0.5 -0.1 -0.1  0.1  0.2  0.5 -0.1 -0.2 -0.4  0.4 -0.1  0.5 -0.4  0.0 -0.3  0.0  0.2  0.1
fis: 2  0.1 -0.2 -0.5 -0.1 -0.1  0.1  0.2  0.5 -0.1 -0.2 -0.4  0.4 -0.1  0.5 -0.4  0.0 -0.3  0.0  0.2  0.1
 for locus : APH
fis: 1 -0.4 -0.3 -0.1  0.0 -0.4  0.1  0.5  0.3 -0.2  0.0 -0.3  0.7  0.2  0.8 -0.1  0.2 -0.1  0.6  0.0 -0.2
fis: 2 -0.4 -0.3 -0.4  0.2 -0.1  0.2 -0.2  ??? -0.1  0.0 -0.1  0.4  0.7  0.6  0.2  0.3 -0.1  0.6  0.2 -0.6
fis: 3 -0.3  0.0 -0.3 -0.2 -0.2  0.2 -0.2  0.3  0.0 -0.2 -0.6  0.5  0.0  0.0 -0.1 -0.1 -0.4  0.9 -0.1 -0.2
 for locus : SDH
fis: 1 -0.1  ???  ???  ???  ???  ???  ???  ???  ???  ???  ???  ???  0.5  ???  0.0  ???  ???  ???  ???  0.6
fis: 2 -0.1  ???  ???  ???  ???  ???  ???  ???  ???  ???  ???  ???  0.5  ???  0.0  ???  ???  ???  ???  0.6
 for locus : PGI
fis: 1 -0.4 -1.0 -1.0 -1.0 -1.0  0.1 -1.0 -0.4 -1.0 -0.6 -1.0  0.1  ???  0.0 -1.0 -1.0  ???  0.5  ???  ???
fis: 2 -0.4 -1.0 -1.0 -1.0 -1.0  0.1 -1.0 -0.4 -1.0 -0.6 -1.0  0.1  ???  0.0 -1.0 -1.0  ???  0.5  ???  ???
fis: 3  0.0  ???  ???  ???  ???  ???  ???  ???  ???  ???  ???  ???  ???  0.0  ???  ???  ???  ???  ???  ???
 for locus : PER-1
fis: 1 -0.6 -0.2 -0.1 -0.5 -0.7 -0.4 -0.3 -0.7 -0.1 -0.7  0.0 -0.9  ???  ??? -0.7 -0.4  0.1  ??? -0.3  1.0
fis: 2 -0.6 -0.2 -0.1 -0.5 -0.7 -0.4 -0.3 -0.7 -0.1 -0.7  0.0 -0.9  ???  ??? -0.7 -0.4  0.1  ??? -0.3  1.0
 for locus : HDH
fis: 1  0.3  0.6 -0.6 -1.0 -0.6 -0.3 -0.5 -0.1 -0.6 -0.8 -0.4 -0.9 -0.8 -0.7 -0.6 -0.7  0.1 -0.8  0.3 -0.6
fis: 2  0.3  0.6 -0.6 -1.0 -0.6 -0.3 -0.5 -0.1 -0.6 -0.8 -0.4 -0.9 -0.8 -0.7 -0.6 -0.7  0.1 -0.8  0.3 -0.6
```

```
 for locus : GOT-3
allele   capf      theta    smallf
    1    0.1435    0.1685   -0.0300
    2    0.1435    0.1685   -0.0300
 all     0.1435    0.1685   -0.0300
 for locus : APH
allele   capf      theta    smallf
    1    0.2275    0.1708    0.0684
    2    0.3248    0.2777    0.0652
    3    0.0996    0.1645   -0.0777
 all     0.2266    0.2089    0.0225
 for locus : SDH
allele   capf      theta    smallf
    1    0.8139    0.7014    0.3766
    2    0.8139    0.7014    0.3766
 all     0.8139    0.7014    0.3766
 for locus : PGI
allele   capf      theta    smallf
    1   -0.5611    0.0420   -0.6295
    2   -0.5614    0.0398   -0.6262
    3   -0.0017   -0.0019    0.0002
 all    -0.5577    0.0407   -0.6238
 for locus : PER-1
allele   capf      theta    smallf
    1   -0.1121    0.2195   -0.4249
    2   -0.1121    0.2195   -0.4249
 all    -0.1121    0.2195   -0.4249
 for locus : HDH
allele   capf      theta    smallf
    1   -0.4032    0.0589   -0.4911
    2   -0.4032    0.0589   -0.4911
 all    -0.4032    0.0589   -0.4911

 over all loci
          capf      theta    smallf
        -0.0796    0.1665   -0.2953
************************************************
jackknifing over populations.
 for locus : GOT-3
          capf      theta    smallf
 total    0.1439    0.1683   -0.0307 means
          0.0874    0.0501    0.0662 std. devs.
 for locus : APH
          capf      theta    smallf
 total    0.2267    0.2091    0.0227 means
          0.0749    0.0445    0.0837 std. devs.
 for locus : SDH
          capf      theta    smallf
 total    1.1126    1.1394    0.4248 means
          0.4080    0.6055    0.2102 std. devs.
 for locus : PGI
          capf      theta    smallf
 total   -0.5576    0.0408   -0.6252 means
          0.1419    0.0204    0.1282 std. devs.
 for locus : PER-1
          capf      theta    smallf
 total   -0.1116    0.2192   -0.4296 means
          0.1607    0.0625    0.1117 std. devs.
 for locus : HDH
          capf      theta    smallf
 total   -0.4048    0.0585   -0.4928 means
          0.1029    0.0270    0.0905 std. devs.

 jackknifing over loci.
          capf      theta    smallf
 total   -0.0791    0.1641   -0.2958 means
          0.1570    0.0425    0.1385 std. devs.
************************************************
         bootstrapping over loci.
         95% confidence interval.
      capf     theta    smallf
   -0.3546    0.0944   -0.5177
    0.1989    0.2569   -0.0636

         99% confidence interval.
      capf     theta    smallf
   -0.4335    0.0700   -0.5592
    0.2853    0.3028    0.0272
 theta per locus over pair of populations.
for locus: GOT-3
         1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17    18    19    20
   1        0.42  0.08  0.12  0.46  0.07 -0.01  0.01  0.53  0.15  0.10 -0.03 -0.01  0.42  0.17  0.61  0.20  0.06  0.02 -0.01
   2  0.42        0.17  0.12 -0.02  0.16  0.30  0.58  0.00  0.10  0.15  0.39  0.33 -0.03  0.08  0.06  0.06  0.19  0.24  0.33
   3  0.08  0.17       -0.01  0.21 -0.02  0.01  0.22  0.28  0.00 -0.01  0.06  0.03  0.17  0.01  0.37  0.02 -0.02 -0.01  0.03
   4  0.12  0.12 -0.01        0.15 -0.02  0.03  0.27  0.22 -0.02 -0.02  0.09  0.06  0.11 -0.01  0.31 -0.01 -0.01  0.00  0.05
   5  0.46 -0.02  0.21  0.15        0.20  0.34  0.61 -0.01  0.13  0.18  0.43  0.37 -0.03  0.11  0.03  0.08  0.23  0.28  0.37
   6  0.07  0.16 -0.02 -0.02  0.20        0.00  0.21  0.27 -0.01 -0.02  0.05  0.02  0.16  0.00  0.36  0.01 -0.03 -0.02  0.02
   7 -0.01  0.30  0.01  0.03  0.34  0.00        0.09  0.41  0.05  0.02 -0.02 -0.02  0.30  0.08  0.50  0.10 -0.01 -0.03 -0.03
   8  0.01  0.58  0.22  0.27  0.61  0.21  0.09        0.68  0.30  0.25  0.02  0.07  0.58  0.33  0.75  0.36  0.19  0.13  0.07
   9  0.53  0.00  0.28  0.22 -0.01  0.27  0.41  0.68        0.20  0.25  0.50  0.45 -0.01  0.17 -0.01  0.15  0.30  0.36  0.45
  10  0.15  0.10  0.00 -0.02  0.13 -0.01  0.05  0.30  0.20       -0.01  0.12  0.08  0.09 -0.02  0.28 -0.01  0.00  0.02  0.08
  11  0.10  0.15 -0.01 -0.02  0.18 -0.02  0.02  0.25  0.25 -0.01        0.07  0.04  0.14  0.00  0.34  0.00 -0.02  0.00  0.04
  12 -0.03  0.39  0.06  0.09  0.43  0.05 -0.02  0.02  0.50  0.12  0.07       -0.02  0.38  0.14  0.58  0.17  0.03  0.00 -0.03
  13 -0.01  0.33  0.03  0.06  0.37  0.02 -0.02  0.07  0.45  0.08  0.04 -0.02        0.33  0.10  0.53  0.12  0.01 -0.01 -0.02
  14  0.42 -0.03  0.17  0.11 -0.03  0.16  0.30  0.58 -0.01  0.09  0.14  0.38  0.33        0.07  0.04  0.05  0.19  0.24  0.33
  15  0.17  0.08  0.01 -0.01  0.11  0.00  0.08  0.33  0.17 -0.02  0.00  0.14  0.10  0.07        0.26 -0.02  0.01  0.04  0.10
  16  0.61  0.06  0.37  0.31  0.03  0.36  0.50  0.75 -0.01  0.28  0.34  0.58  0.53  0.04  0.26        0.23  0.39  0.44  0.53
  17  0.20  0.06  0.02 -0.01  0.08  0.01  0.10  0.36  0.15 -0.01  0.00  0.17  0.12  0.05 -0.02  0.23        0.02  0.05  0.12
  18  0.06  0.19 -0.02 -0.01  0.23 -0.03 -0.01  0.19  0.30  0.00 -0.02  0.03  0.01  0.19  0.01  0.39  0.02       -0.02  0.01
  19  0.02  0.24 -0.01  0.00  0.28 -0.02 -0.03  0.13  0.36  0.02  0.00  0.00 -0.01  0.24  0.04  0.44  0.05 -0.02       -0.02
  20 -0.01  0.33  0.03  0.05  0.37  0.02 -0.03  0.07  0.45  0.08  0.04 -0.03 -0.02  0.33  0.10  0.53  0.12  0.01 -0.02
for locus: APH
         1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17    18    19    20
   1        0.09  0.04 -0.01  0.15 -0.02  0.20  0.22  0.27  0.21  0.14 -0.01  0.00  0.19  0.10  0.02  0.05  0.02  0.20  0.00
   2  0.09        0.06  0.11  0.41  0.11  0.44  0.49  0.46  0.45  0.38  0.11  0.04  0.01  0.00  0.08  0.08  0.20  0.05  0.12
   3  0.04  0.06        0.09  0.28  0.07  0.25  0.36  0.46  0.40  0.30  0.10 -0.02  0.10  0.02  0.12 -0.02  0.17  0.08  0.10
   4 -0.01  0.11  0.09        0.14 -0.02  0.24  0.22  0.18  0.15  0.10 -0.03  0.04  0.23  0.14 -0.01  0.11 -0.01  0.25 -0.02
   5  0.15  0.41  0.28  0.14        0.12  0.08 -0.01  0.32  0.12  0.02  0.15  0.23  0.52  0.41  0.24  0.29  0.09  0.52  0.15
   6 -0.02  0.11  0.07 -0.02  0.12        0.19  0.19  0.21  0.16  0.10 -0.03  0.02  0.22  0.13  0.01  0.08 -0.01  0.23 -0.02
   7  0.20  0.44  0.25  0.24  0.08  0.19        0.09  0.52  0.35  0.20  0.25  0.23  0.54  0.41  0.35  0.24  0.24  0.51  0.24
   8  0.22  0.49  0.36  0.22 -0.01  0.19  0.09        0.39  0.16  0.05  0.22  0.31  0.60  0.49  0.32  0.36  0.16  0.59  0.22
   9  0.27  0.46  0.46  0.18  0.32  0.21  0.52  0.39        0.09  0.17  0.16  0.38  0.60  0.51  0.18  0.48  0.09  0.63  0.19
  10  0.21  0.45  0.40  0.15  0.12  0.16  0.35  0.16  0.09        0.02  0.14  0.33  0.58  0.49  0.21  0.41  0.05  0.60  0.16
  11  0.14  0.38  0.30  0.10  0.02  0.10  0.20  0.05  0.17  0.02        0.10  0.24  0.51  0.41  0.18  0.31  0.03  0.52  0.11
  12 -0.01  0.11  0.10 -0.03  0.15 -0.03  0.25  0.22  0.16  0.14  0.10        0.04  0.23  0.14 -0.02  0.12 -0.02  0.25 -0.03
  13  0.00  0.04 -0.02  0.04  0.23  0.02  0.23  0.31  0.38  0.33  0.24  0.04        0.10  0.02  0.06 -0.01  0.11  0.09  0.04
  14  0.19  0.01  0.10  0.23  0.52  0.22  0.54  0.60  0.60  0.58  0.51  0.23  0.10        0.00  0.20  0.12  0.34 -0.01  0.24
  15  0.10  0.00  0.02  0.14  0.41  0.13  0.41  0.49  0.51  0.49  0.41  0.14  0.02  0.00        0.13  0.03  0.24  0.00  0.15
  16  0.02  0.08  0.12 -0.01  0.24  0.01  0.35  0.32  0.18  0.21  0.18 -0.02  0.06  0.20  0.13        0.14  0.02  0.24  0.00
  17  0.05  0.08 -0.02  0.11  0.29  0.08  0.24  0.36  0.48  0.41  0.31  0.12 -0.01  0.12  0.03  0.14        0.19  0.08  0.11
  18  0.02  0.20  0.17 -0.01  0.09 -0.01  0.24  0.16  0.09  0.05  0.03 -0.02  0.11  0.34  0.24  0.02  0.19        0.36 -0.01
  19  0.20  0.05  0.08  0.25  0.52  0.23  0.51  0.59  0.63  0.60  0.52  0.25  0.09 -0.01  0.00  0.24  0.08  0.36        0.26
  20  0.00  0.12  0.10 -0.02  0.15 -0.02  0.24  0.22  0.19  0.16  0.11 -0.03  0.04  0.24  0.15  0.00  0.11 -0.01  0.26
for locus: SDH
         1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17    18    19    20
   1        0.06  0.06  0.06  0.06  0.06  0.06  0.06  0.06  0.06  0.06  0.06  0.79  0.06  0.00  0.06  0.06  0.06  0.06 -0.01
   2  0.06                                                                    0.89        0.00                          0.12
   3  0.06                                                                    0.89        0.00                          0.12
   4  0.06                                                                    0.89        0.00                          0.12
   5  0.06                                                                    0.89        0.00                          0.12
   6  0.06                                                                    0.89        0.00                          0.12
   7  0.06                                                                    0.89        0.00                          0.12
   8  0.06                                                                    0.89        0.00                          0.12
   9  0.06                                                                    0.89        0.00                          0.12
  10  0.06                                                                    0.89        0.00                          0.12
  11  0.06                                                                    0.89        0.00                          0.12
  12  0.06                                                                    0.89        0.00                          0.12
  13  0.79  0.89  0.89  0.89  0.89  0.89  0.89  0.89  0.89  0.89  0.89  0.89        0.89  0.86  0.89  0.89  0.89  0.89  0.69
  14  0.06                                                                    0.89        0.00                          0.12
  15  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.86  0.00        0.00  0.00  0.00  0.00  0.06
  16  0.06                                                                    0.89        0.00                          0.12
  17  0.06                                                                    0.89        0.00                          0.12
  18  0.06                                                                    0.89        0.00                          0.12
  19  0.06                                                                    0.89        0.00                          0.12
  20 -0.01  0.12  0.12  0.12  0.12  0.12  0.12  0.12  0.12  0.12  0.12  0.12  0.69  0.12  0.06  0.12  0.12  0.12  0.12
```

```
for locus: PGI
         1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17    18    19    20
   1        0.02  0.02  0.02  0.02  0.09  0.02  0.00  0.02  0.09  0.02 -0.02        0.23  0.02  0.02       -0.03
   2  0.02        0.00  0.00  0.00  0.02  0.00  0.07  0.00  0.03  0.00  0.03        0.13  0.00  0.00        0.01
   3  0.02  0.00        0.00  0.00  0.02  0.00  0.07  0.00  0.03  0.00  0.03        0.13  0.00  0.00        0.01
   4  0.02  0.00  0.00        0.00  0.02  0.00  0.07  0.00  0.03  0.00  0.03        0.13  0.00  0.00        0.01
   5  0.02  0.00  0.00  0.00        0.02  0.00  0.07  0.00  0.03  0.00  0.03        0.13  0.00  0.00        0.01
   6  0.09  0.02  0.02  0.02  0.02        0.02  0.17  0.02 -0.02  0.02  0.12        0.02  0.02  0.02        0.07
   7  0.02  0.00  0.00  0.00  0.00  0.02        0.07  0.00  0.03  0.00  0.03        0.13  0.00  0.00        0.01
   8  0.00  0.07  0.07  0.07  0.07  0.17  0.07        0.07  0.18  0.07 -0.02        0.33  0.07  0.07       -0.01
   9  0.02  0.00  0.00  0.00  0.00  0.02  0.00  0.07        0.03  0.00  0.03        0.13  0.00  0.00        0.01
  10  0.09  0.03  0.03  0.03  0.03 -0.02  0.03  0.18  0.03        0.03  0.12        0.03  0.03  0.03        0.08
  11  0.02  0.00  0.00  0.00  0.00  0.02  0.00  0.07  0.00  0.03        0.03        0.13  0.00  0.00        0.01
  12 -0.02  0.03  0.03  0.03  0.03  0.12  0.03 -0.02  0.03  0.12  0.03              0.27  0.03  0.03       -0.03
  13
  14  0.23  0.13  0.13  0.13  0.13  0.02  0.13  0.33  0.13  0.03  0.13  0.27              0.13  0.13        0.22
  15  0.02  0.00  0.00  0.00  0.00  0.02  0.00  0.07  0.00  0.03  0.00  0.03        0.13        0.00        0.01
  16  0.02  0.00  0.00  0.00  0.00  0.02  0.00  0.07  0.00  0.03  0.00  0.03        0.13  0.00              0.01
  17
  18 -0.03  0.01  0.01  0.01  0.01  0.07  0.01 -0.01  0.01  0.08  0.01 -0.03        0.22  0.01  0.01
  19
  20
for locus: PER-1
         1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17    18    19    20
   1        0.10  0.25  0.01  0.05  0.01  0.23 -0.01  0.20 -0.01  0.34  0.01              0.05  0.09  0.14        0.02  0.27
   2  0.10        0.02  0.20  0.28  0.03  0.51  0.12  0.00  0.13  0.10  0.18              0.28  0.34  0.42        0.00  0.55
   3  0.25  0.02        0.36  0.44  0.14  0.67  0.27 -0.02  0.27  0.00  0.33              0.44  0.49  0.57        0.12  0.68
   4  0.01  0.20  0.36        0.00  0.07  0.13  0.00  0.31  0.00  0.45 -0.01              0.00  0.02  0.06        0.08  0.16
   5  0.05  0.28  0.44  0.00        0.13  0.07  0.04  0.40  0.04  0.53  0.02             -0.01 -0.01  0.01        0.16  0.09
   6  0.01  0.03  0.14  0.07  0.13        0.34  0.02  0.10  0.02  0.23  0.05              0.13  0.18  0.25       -0.02  0.38
   7  0.23  0.51  0.67  0.13  0.07  0.34        0.21  0.63  0.20  0.75  0.15              0.07  0.03 -0.02        0.38 -0.04
   8 -0.01  0.12  0.27  0.00  0.04  0.02  0.21        0.23 -0.01  0.37  0.00              0.04  0.07  0.12        0.03  0.24
   9  0.20  0.00 -0.02  0.31  0.40  0.10  0.63  0.23        0.23  0.02  0.29              0.40  0.45  0.53        0.08  0.65
  10 -0.01  0.13  0.27  0.00  0.04  0.02  0.20 -0.01  0.23        0.37  0.00              0.04  0.07  0.12        0.04  0.24
  11  0.34  0.10  0.00  0.45  0.53  0.23  0.75  0.37  0.02  0.37        0.42              0.53  0.57  0.65        0.24  0.75
  12  0.01  0.18  0.33 -0.01  0.02  0.05  0.15  0.00  0.29  0.00  0.42                    0.02  0.04  0.08        0.07  0.19
  13
  14
  15  0.05  0.28  0.44  0.00 -0.01  0.13  0.07  0.04  0.40  0.04  0.53  0.02                   -0.01  0.01        0.16  0.09
  16  0.09  0.34  0.49  0.02 -0.01  0.18  0.03  0.07  0.45  0.07  0.57  0.04             -0.01       -0.01        0.20  0.04
  17  0.14  0.42  0.57  0.06  0.01  0.25 -0.02  0.12  0.53  0.12  0.65  0.08              0.01 -0.01              0.28 -0.01
  18
  19  0.02  0.00  0.12  0.08  0.16 -0.02  0.38  0.03  0.08  0.04  0.24  0.07              0.16  0.20  0.28              0.42
  20  0.27  0.55  0.68  0.16  0.09  0.38 -0.04  0.24  0.65  0.24  0.75  0.19              0.09  0.04 -0.01        0.42
for locus: HDH
         1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17    18    19    20
   1        0.05 -0.01 -0.01 -0.02 -0.01 -0.02  0.22 -0.02 -0.01  0.01  0.00 -0.01 -0.02 -0.02 -0.02  0.00  0.01  0.18 -0.01
   2  0.05        0.13  0.14  0.05  0.01  0.11  0.02  0.03  0.13  0.18  0.16  0.13  0.07  0.05  0.07 -0.02  0.18  0.00  0.13
   3 -0.01  0.13       -0.01  0.01  0.03 -0.01  0.31  0.02 -0.01 -0.01 -0.01 -0.01  0.00  0.01  0.00  0.06  0.00  0.27 -0.01
   4 -0.01  0.14 -0.01        0.02  0.04 -0.01  0.32  0.03  0.00  0.00  0.00  0.00  0.01  0.02  0.01  0.07  0.00  0.27 -0.01
   5 -0.02  0.05  0.01  0.02       -0.01  0.00  0.20 -0.01  0.01  0.03  0.03  0.01 -0.01 -0.01 -0.01  0.00  0.04  0.16  0.01
   6 -0.01  0.01  0.03  0.04 -0.01        0.02  0.15 -0.01  0.03  0.06  0.05  0.03  0.00 -0.01  0.00 -0.02  0.07  0.11  0.03
   7 -0.02  0.11 -0.01 -0.01  0.00  0.02        0.28  0.01 -0.01  0.00  0.00 -0.01  0.00  0.00  0.00  0.04  0.00  0.24 -0.01
   8  0.22  0.02  0.31  0.32  0.20  0.15  0.28        0.18  0.31  0.36  0.34  0.31  0.23  0.20  0.23  0.09  0.37 -0.03  0.31
   9 -0.02  0.03  0.02  0.03 -0.01 -0.01  0.01  0.18        0.02  0.05  0.04  0.02  0.00 -0.01  0.00 -0.01  0.05  0.14  0.02
  10 -0.01  0.13 -0.01  0.00  0.01  0.03 -0.01  0.31  0.02       -0.01  0.00 -0.01  0.01  0.01  0.01  0.07  0.00  0.27 -0.01
  11  0.01  0.18 -0.01  0.00  0.03  0.06  0.00  0.36  0.05 -0.01       -0.01 -0.01  0.02  0.03  0.02  0.10 -0.01  0.32 -0.01
  12  0.00  0.16 -0.01  0.00  0.03  0.05  0.00  0.34  0.04  0.00 -0.01        0.00  0.02  0.03  0.02  0.09  0.00  0.30 -0.01
  13 -0.01  0.13 -0.01  0.00  0.01  0.03 -0.01  0.31  0.02 -0.01 -0.01  0.00        0.01  0.01  0.01  0.07  0.00  0.27 -0.01
  14 -0.02  0.07  0.00  0.01 -0.01  0.00  0.00  0.23  0.00  0.01  0.02  0.02  0.01       -0.01 -0.01  0.02  0.03  0.19  0.00
  15 -0.02  0.05  0.01  0.02 -0.01 -0.01  0.00  0.20 -0.01  0.01  0.03  0.03  0.01 -0.01       -0.01  0.00  0.04  0.16  0.01
  16 -0.02  0.07  0.00  0.01 -0.01  0.00  0.00  0.23  0.00  0.01  0.02  0.02  0.01 -0.01 -0.01        0.02  0.03  0.19  0.00
  17  0.00 -0.02  0.06  0.07  0.00 -0.02  0.04  0.09 -0.01  0.07  0.10  0.09  0.07  0.02  0.00  0.02        0.11  0.06  0.06
  18  0.01  0.18  0.00  0.00  0.04  0.07  0.00  0.37  0.05  0.00 -0.01  0.00  0.00  0.03  0.04  0.03  0.11        0.33  0.00
  19  0.18  0.00  0.27  0.27  0.16  0.11  0.24 -0.03  0.14  0.27  0.32  0.30  0.27  0.19  0.16  0.19  0.06  0.33        0.27
  20 -0.01  0.13 -0.01 -0.01  0.01  0.03 -0.01  0.31  0.02 -0.01 -0.01 -0.01 -0.01  0.00  0.01  0.00  0.06  0.00  0.27
        permutting alleles within samples.

        95% confidence interval.

   capf     theta    smallf
  0.1234    0.1591   -0.0449
  0.1975    0.1611    0.0457

        99% confidence interval.

   capf     theta    smallf
  0.1106    0.1588   -0.0606
  0.2083    0.1614    0.0588

(prob fis=0)<    0.00020

**********************************************
        permutting alleles within total.

        95% confidence interval.

   capf     theta    smallf
 -0.0419   -0.0065   -0.0421
  0.0420    0.0071    0.0422

        99% confidence interval.

   capf     theta    smallf
 -0.0544   -0.0083   -0.0548
  0.0529    0.0096    0.0533

(prob fit=0)=    0.99960

**********************************************
        permutting genotypes within total.

        95% confidence interval.

   capf     theta    smallf
 -0.0890   -0.0059   -0.0958
 -0.0883    0.0068   -0.0826

        99% confidence interval.

   capf     theta    smallf
 -0.0891   -0.0077   -0.0989
 -0.0882    0.0096   -0.0808

(prob fst=0)<    0.00020
```

Appendix H

Raw output of the treatment of Nucella data

```
          1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
    locus: EST-3
     n   13   30   39    7   24    5   13   19   17   16   21   17   18   21   21
p:   1 0.73 0.75 0.72 0.93 0.88 0.80 0.96 0.89 0.76 0.97 0.95 0.85 1.00 0.98 0.98
p:   2 0.27 0.25 0.28 0.07 0.12 0.20 0.04 0.11 0.24 0.03 0.05 0.15 0.00 0.02 0.02
ho:  1 0.54 0.37 0.26 0.14 0.25 0.40 0.08 0.21 0.24 0.06 0.10 0.29 0.00 0.05 0.05
ho:  2 0.54 0.37 0.26 0.14 0.25 0.40 0.08 0.21 0.24 0.06 0.10 0.29 0.00 0.05 0.05
he:  1 0.41 0.38 0.41 0.14 0.22 0.36 0.08 0.19 0.37 0.06 0.09 0.26 0.00 0.05 0.05
he:  2 0.41 0.38 0.41 0.14 0.22 0.36 0.08 0.19 0.37 0.06 0.09 0.26 0.00 0.05 0.05
    locus: LAP-1
     n   12   29   39    7   24    6   14   19   17   16   21   17   18   21   21
p:   1 0.88 0.88 1.00 1.00 0.98 0.75 0.96 0.87 0.79 0.88 0.88 0.97 1.00 0.86 0.86
p:   2 0.12 0.12 0.00 0.00 0.02 0.25 0.00 0.00 0.00 0.06 0.00 0.00 0.00 0.02 0.00
p:   3 0.00 0.00 0.00 0.00 0.00 0.00 0.04 0.13 0.21 0.06 0.12 0.03 0.00 0.12 0.14
ho:  1 0.25 0.24 0.00 0.00 0.04 0.17 0.07 0.16 0.41 0.25 0.14 0.06 0.00 0.10 0.29
ho:  2 0.25 0.24 0.00 0.00 0.04 0.17 0.00 0.00 0.00 0.12 0.00 0.00 0.00 0.05 0.00
ho:  3 0.00 0.00 0.00 0.00 0.00 0.00 0.07 0.16 0.41 0.12 0.14 0.06 0.00 0.05 0.29
he:  1 0.23 0.22 0.00 0.00 0.04 0.41 0.07 0.23 0.34 0.23 0.21 0.06 0.00 0.25 0.25
he:  2 0.23 0.22 0.00 0.00 0.04 0.41 0.00 0.00 0.00 0.12 0.00 0.00 0.00 0.05 0.00
he:  3 0.00 0.00 0.00 0.00 0.00 0.00 0.07 0.23 0.34 0.12 0.21 0.06 0.00 0.21 0.25
    locus: LAP-2
     n   14   30   39    7   24    6   14   19   17   16   21   17   18   21   21
p:   1 0.07 0.02 0.04 0.00 0.04 0.00 0.11 0.11 0.15 0.03 0.02 0.03 0.00 0.02 0.00
p:   2 0.93 0.98 0.96 1.00 0.96 1.00 0.89 0.89 0.85 0.97 0.98 0.97 1.00 0.98 1.00
ho:  1 0.14 0.03 0.08 0.00 0.08 0.00 0.21 0.21 0.29 0.06 0.05 0.06 0.00 0.05 0.00
ho:  2 0.14 0.03 0.08 0.00 0.08 0.00 0.21 0.21 0.29 0.06 0.05 0.06 0.00 0.05 0.00
he:  1 0.14 0.03 0.07 0.00 0.08 0.00 0.20 0.19 0.26 0.06 0.05 0.06 0.00 0.05 0.00
he:  2 0.14 0.03 0.07 0.00 0.08 0.00 0.20 0.19 0.26 0.06 0.05 0.06 0.00 0.05 0.00
    locus: MDH-1
     n   12   30   39    7   23    6   14   19   17   16   21   17   18   21   21
p:   1 0.00 0.00 0.00 0.00 0.00 0.00 0.04 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
p:   2 0.08 0.20 0.27 0.00 0.09 0.33 0.07 0.11 0.12 0.03 0.12 0.00 0.03 0.05 0.17
p:   3 0.92 0.80 0.73 1.00 0.91 0.67 0.89 0.89 0.88 0.94 0.88 0.97 0.97 0.95 0.83
p:   4 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.03 0.00 0.03 0.00 0.00 0.00
ho:  1 0.00 0.00 0.00 0.00 0.00 0.00 0.07 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
ho:  2 0.17 0.40 0.28 0.00 0.17 0.33 0.14 0.11 0.12 0.06 0.24 0.00 0.06 0.10 0.24
ho:  3 0.17 0.40 0.28 0.00 0.17 0.33 0.07 0.11 0.12 0.12 0.24 0.06 0.06 0.10 0.24
ho:  4 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.06 0.00 0.06 0.00 0.00 0.00
he:  1 0.00 0.00 0.00 0.00 0.00 0.00 0.07 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
he:  2 0.16 0.33 0.40 0.00 0.16 0.48 0.14 0.19 0.21 0.06 0.21 0.00 0.06 0.09 0.28
he:  3 0.16 0.33 0.40 0.00 0.16 0.48 0.20 0.19 0.21 0.12 0.21 0.06 0.06 0.09 0.28
he:  4 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.06 0.00 0.06 0.00 0.00 0.00
    locus: PEP-1
     n   14   30   39    7   21    6   14   19   17   16   21   17   18   21   21
p:   1 0.00 0.00 0.00 0.00 0.00 0.00 0.07 0.21 0.18 0.56 0.38 0.56 0.58 0.69 0.60
p:   2 0.96 1.00 1.00 1.00 1.00 1.00 0.93 0.79 0.82 0.44 0.62 0.41 0.42 0.31 0.40
p:   3 0.04 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
p:   4 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.03 0.00 0.00 0.00
ho:  1 0.00 0.00 0.00 0.00 0.00 0.00 0.14 0.32 0.24 0.62 0.67 0.65 0.72 0.52 0.71
ho:  2 0.07 0.00 0.00 0.00 0.00 0.00 0.14 0.32 0.24 0.62 0.67 0.59 0.72 0.52 0.71
ho:  3 0.07 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
ho:  4 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.06 0.00 0.00 0.00
he:  1 0.00 0.00 0.00 0.00 0.00 0.00 0.14 0.34 0.30 0.51 0.48 0.51 0.50 0.44 0.49
he:  2 0.07 0.00 0.00 0.00 0.00 0.00 0.14 0.34 0.30 0.51 0.48 0.50 0.50 0.44 0.49
he:  3 0.07 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
he:  4 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.06 0.00 0.00 0.00
```

```
    locus: PEP-2
     n   14   30   39    7   24    5   14   18   17   16   20   17   16   21   20
p:   1 0.96 1.00 1.00 1.00 1.00 0.80 0.89 0.58 0.76 0.69 0.47 0.56 0.28 0.48 0.33
p:   2 0.04 0.00 0.00 0.00 0.00 0.20 0.11 0.42 0.24 0.31 0.53 0.44 0.72 0.52 0.68
ho:  1 0.07 0.00 0.00 0.00 0.00 0.00 0.21 0.50 0.24 0.50 0.65 0.29 0.31 0.48 0.55
ho:  2 0.07 0.00 0.00 0.00 0.00 0.00 0.21 0.50 0.24 0.50 0.65 0.29 0.31 0.48 0.55
he:  1 0.07 0.00 0.00 0.00 0.00 0.36 0.20 0.50 0.37 0.44 0.51 0.51 0.42 0.51 0.45
he:  2 0.07 0.00 0.00 0.00 0.00 0.36 0.20 0.50 0.37 0.44 0.51 0.51 0.42 0.51 0.45
    locus: PGM-1
     n   14   30   39    7   24    6   14   19   17   16   21   17   18   21   21
p:   1 0.04 0.00 0.00 0.00 0.00 0.17 0.07 0.50 0.15 0.69 0.50 0.56 0.92 0.88 0.86
p:   2 0.93 0.97 0.86 0.79 0.85 0.83 0.93 0.47 0.71 0.31 0.50 0.44 0.08 0.12 0.14
p:   3 0.04 0.03 0.14 0.21 0.15 0.00 0.00 0.03 0.15 0.00 0.00 0.00 0.00 0.00 0.00
ho:  1 0.07 0.00 0.00 0.00 0.00 0.33 0.14 0.47 0.18 0.50 0.52 0.53 0.17 0.24 0.29
ho:  2 0.14 0.07 0.28 0.43 0.29 0.33 0.14 0.42 0.47 0.50 0.52 0.53 0.17 0.24 0.29
ho:  3 0.07 0.07 0.28 0.43 0.29 0.00 0.00 0.05 0.29 0.00 0.00 0.00 0.00 0.00 0.00
he:  1 0.07 0.00 0.00 0.00 0.00 0.30 0.14 0.51 0.26 0.44 0.51 0.51 0.16 0.21 0.25
he:  2 0.14 0.07 0.25 0.36 0.25 0.30 0.14 0.51 0.43 0.44 0.51 0.51 0.16 0.21 0.25
he:  3 0.07 0.07 0.25 0.36 0.25 0.00 0.00 0.05 0.26 0.00 0.00 0.00 0.00 0.00 0.00
    locus: PGM-2
     n   14   30   39    7   24    6   14   19   17   16   21   17   18   21   20
p:   1 0.04 0.00 0.00 0.00 0.00 0.17 0.07 0.45 0.15 0.66 0.50 0.56 0.92 0.88 0.88
p:   2 0.96 1.00 1.00 1.00 1.00 0.83 0.93 0.55 0.85 0.34 0.50 0.44 0.08 0.12 0.12
ho:  1 0.07 0.00 0.00 0.00 0.00 0.33 0.14 0.58 0.18 0.44 0.52 0.53 0.17 0.24 0.25
ho:  2 0.07 0.00 0.00 0.00 0.00 0.33 0.14 0.58 0.18 0.44 0.52 0.53 0.17 0.24 0.25
he:  1 0.07 0.00 0.00 0.00 0.00 0.30 0.14 0.51 0.26 0.47 0.51 0.51 0.16 0.21 0.22
he:  2 0.07 0.00 0.00 0.00 0.00 0.30 0.14 0.51 0.26 0.47 0.51 0.51 0.16 0.21 0.22

 for locus : EST-3
fis: 1 -0.3  0.0  0.4  0.0 -0.1 -0.1  0.0 -0.1  0.4  0.0  0.0 -0.1  ???  0.0  0.0
fis: 2 -0.3  0.0  0.4  0.0 -0.1 -0.1  0.0 -0.1  0.4  0.0  0.0 -0.1  ???  0.0  0.0
 for locus : LAP-1
fis: 1 -0.1 -0.1  ???  ???  0.0  0.6  0.0  0.3 -0.2 -0.1  0.3  0.0  ???  0.6 -0.1
fis: 2 -0.1 -0.1  ???  ???  0.0  0.6  ???  ???  ???  0.0  ???  ???  ???  0.0  ???
fis: 3  ???  ???  ???  ???  ???  ???  0.0  0.3 -0.2  0.0  0.3  0.0  ???  0.8 -0.1
 for locus : LAP-2
fis: 1  0.0  0.0  0.0  ???  0.0  ??? -0.1 -0.1 -0.1  0.0  0.0  0.0  ???  0.0  ???
fis: 2  0.0  0.0  0.0  ???  0.0  ??? -0.1 -0.1 -0.1  0.0  0.0  0.0  ???  0.0  ???
 for locus : MDH-1
fis: 1  ???  ???  ???  ???  ???  ???  0.0  ???  ???  ???  ???  ???  ???  ???  ???
fis: 2  0.0 -0.2  0.3  ??? -0.1  0.3  0.0  0.5  0.4  0.0 -0.1  ???  0.0  0.0  0.2
fis: 3  0.0 -0.2  0.3  ??? -0.1  0.3  0.6  0.5  0.4  0.0 -0.1  0.0  0.0  0.0  0.2
fis: 4  ???  ???  ???  ???  ???  ???  ???  ???  ???  0.0  ???  0.0  ???  ???  ???
 for locus : PEP-1
fis: 1  ???  ???  ???  ???  ???  ???  0.0  0.1  0.2 -0.2 -0.4 -0.3 -0.4 -0.2 -0.4
fis: 2  0.0  ???  ???  ???  ???  ???  0.0  0.1  0.2 -0.2 -0.4 -0.2 -0.4 -0.2 -0.4
fis: 3  0.0  ???  ???  ???  ???  ???  ???  ???  ???  ???  ???  ???  ???  ???  ???
fis: 4  ???  ???  ???  ???  ???  ???  ???  ???  ???  ???  ???  0.0  ???  ???  ???
 for locus : PEP-2
fis: 1  0.0  ???  ???  ???  ???  1.0 -0.1  0.0  0.4 -0.1 -0.3  0.4  0.3  0.1 -0.2
fis: 2  0.0  ???  ???  ???  ???  1.0 -0.1  0.0  0.4 -0.1 -0.3  0.4  0.3  0.1 -0.2
 for locus : PGM-1
fis: 1  0.0  ???  ???  ???  ??? -0.1  0.0  0.1  0.3 -0.1  0.0  0.0 -0.1 -0.1 -0.1
fis: 2  0.0  0.0 -0.1 -0.2 -0.1 -0.1  0.0  0.2 -0.1 -0.1  0.0  0.0 -0.1 -0.1 -0.1
fis: 3  0.0  0.0 -0.1 -0.2 -0.1  ???  ???  0.0 -0.1  ???  ???  ???  ???  ???  ???
 for locus : PGM-2
fis: 1  0.0  ???  ???  ???  ??? -0.1  0.0 -0.1  0.3  0.1  0.0  0.0 -0.1 -0.1 -0.1
fis: 2  0.0  ???  ???  ???  ??? -0.1  0.0 -0.1  0.3  0.1  0.0  0.0 -0.1 -0.1 -0.1
```

```
 for locus : EST-3
allele     capf     theta    smallf
    1    0.1543    0.0716    0.0891
    2    0.1543    0.0716    0.0891
 all     0.1543    0.0716    0.0891
 for locus : LAP-1
allele     capf     theta    smallf
    1    0.1322    0.0364    0.0994
    2    0.0974    0.0808    0.0180
    3    0.2100    0.0604    0.1593
 all     0.1516    0.0523    0.1048
 for locus : LAP-2
allele     capf     theta    smallf
    1   -0.0409    0.0191   -0.0612
    2   -0.0409    0.0191   -0.0612
 all    -0.0409    0.0191   -0.0612
 for locus : MDH-1
allele     capf     theta    smallf
    1    0.0008    0.0103   -0.0096
    2    0.1623    0.0493    0.1188
    3    0.1757    0.0379    0.1433
    4   -0.0016    0.0022   -0.0039
 all     0.1651    0.0426    0.1280
 for locus : PEP-1
allele     capf     theta    smallf
    1    0.2216    0.3802   -0.2558
    2    0.2284    0.3779   -0.2402
    3    0.0008    0.0103   -0.0096
    4    0.0002    0.0031   -0.0029
 all     0.2230    0.3757   -0.2445
 for locus : PEP-2
allele     capf     theta    smallf
    1    0.3800    0.3427    0.0567
    2    0.3800    0.3427    0.0567
 all     0.3800    0.3427    0.0567
 for locus : PGM-1
allele     capf     theta    smallf
    1    0.5623    0.5712   -0.0207
    2    0.3833    0.4208   -0.0648
    3   -0.0482    0.0730   -0.1308
 all     0.4220    0.4547   -0.0600
 for locus : PGM-2
allele     capf     theta    smallf
    1    0.5572    0.5703   -0.0306
    2    0.5572    0.5703   -0.0306
 all     0.5572    0.5703   -0.0306

 over all loci
           capf      theta    smallf
         0.3283    0.3327   -0.0066
************************************************
jackknifing over populations.
 for locus : EST-3
           capf      theta    smallf
 total     0.1756    0.0722    0.1104 means
           0.1188    0.0212    0.1189 std. devs.
 for locus : LAP-1
           capf      theta    smallf
 total     0.1475    0.0508    0.1029 means
           0.1012    0.0220    0.1131 std. devs.
 for locus : LAP-2
           capf      theta    smallf
 total    -0.0437    0.0200   -0.0646 means
           0.0108    0.0153    0.0223 std. devs.
 for locus : MDH-1
           capf      theta    smallf
 total     0.1752    0.0464    0.1339 means
           0.1056    0.0231    0.0979 std. devs.
 for locus : PEP-1
           capf      theta    smallf
 total     0.2176    0.3744   -0.2504 means
           0.1005    0.0737    0.0721 std. devs.
 for locus : PEP-2
           capf      theta    smallf
 total     0.3772    0.3437    0.0523 means
           0.0957    0.0860    0.0938 std. devs.
 for locus : PGM-1
           capf      theta    smallf
 total     0.4272    0.4602   -0.0587 means
           0.0830    0.0851    0.0317 std. devs.
 for locus : PGM-2
           capf      theta    smallf
 total     0.5623    0.5763   -0.0321 means
           0.1099    0.1072    0.0401 std. devs.

 jackknifing over loci.
           capf      theta    smallf
 total     0.3365    0.3434   -0.0078 means
           0.0648    0.0736    0.0489 std. devs.
************************************************
         bootstrapping over loci.
```

```
        95% confidence interval.
   capf      theta     smallf
 0.1926     0.1696    -0.1022
 0.4233     0.4396     0.0761

        99% confidence interval.
   capf      theta     smallf
 0.1468     0.1156    -0.1321
 0.4500     0.4649     0.0912
```

```
 theta per locus over pair of populations.
for locus: EST-3
          1     2     3     4     5     6     7     8     9    10    11    12    13    14    15
    1       -0.03 -0.03  0.08  0.05 -0.04  0.16  0.07 -0.03  0.19  0.17  0.02  0.29  0.23  0.23
    2 -0.03       -0.02  0.05  0.03 -0.06  0.11  0.04 -0.03  0.13  0.12  0.01  0.19  0.16  0.16
    3 -0.03 -0.02        0.06  0.05 -0.06  0.13  0.06 -0.02  0.15  0.14  0.02  0.20  0.17  0.17
    4  0.08  0.05  0.06       -0.03  0.00 -0.05 -0.04  0.02 -0.03 -0.04 -0.02  0.08 -0.02 -0.02
    5  0.05  0.03  0.05 -0.03       -0.03  0.02 -0.02  0.01  0.03  0.02 -0.02  0.09  0.05  0.05
    6 -0.04 -0.06 -0.06  0.00 -0.03        0.10 -0.02 -0.09  0.14  0.10 -0.05  0.34  0.20  0.20
    7  0.16  0.11  0.13 -0.05  0.02  0.10        0.00  0.10 -0.04 -0.03  0.03  0.01 -0.03 -0.03
    8  0.07  0.04  0.06 -0.04 -0.02 -0.02  0.00        0.03  0.01  0.00 -0.02  0.08  0.03  0.03
    9 -0.03 -0.03 -0.02  0.02  0.01 -0.09  0.10  0.03        0.13  0.11 -0.01  0.21  0.16  0.16
   10  0.19  0.13  0.15 -0.03  0.03  0.14 -0.04  0.01  0.13       -0.02  0.05  0.00 -0.03 -0.03
   11  0.17  0.12  0.14 -0.04  0.02  0.10 -0.03  0.00  0.11 -0.02        0.03  0.02 -0.02 -0.02
   12  0.02  0.01  0.02 -0.02 -0.02 -0.05  0.03 -0.02 -0.01  0.05  0.03        0.13  0.08  0.08
   13  0.29  0.19  0.20  0.08  0.09  0.34  0.01  0.08  0.21  0.00  0.02  0.13        0.00  0.00
   14  0.23  0.16  0.17 -0.02  0.05  0.20 -0.03  0.03  0.16 -0.03 -0.02  0.08  0.00       -0.02
   15  0.23  0.16  0.17 -0.02  0.05  0.20 -0.03  0.03  0.16 -0.03 -0.02  0.08  0.00 -0.02
for locus: LAP-1
          1     2     3     4     5     6     7     8     9    10    11    12    13    14    15
    1       -0.03  0.22  0.05  0.07 -0.02  0.05  0.03  0.08 -0.02  0.03  0.06  0.12  0.00  0.04
    2 -0.03        0.13  0.04  0.05  0.01  0.04  0.05  0.09 -0.01  0.04  0.05  0.08  0.03  0.06
    3  0.22  0.13              0.01  0.53  0.04  0.17  0.29  0.13  0.14  0.03        0.14  0.18
    4  0.05  0.04             -0.03  0.16 -0.03  0.03  0.12  0.02  0.02 -0.03        0.01  0.06
    5  0.07  0.05  0.01 -0.03        0.28 -0.01  0.08  0.18  0.04  0.07 -0.01 -0.01  0.06  0.10
    6 -0.02  0.01  0.53  0.16  0.28        0.19  0.08  0.08  0.03  0.09  0.23  0.35  0.04  0.10
    7  0.05  0.04  0.04 -0.03 -0.01  0.19        0.02  0.10  0.01  0.01 -0.03  0.01  0.01  0.04
    8  0.03  0.05  0.17  0.03  0.08  0.08  0.02       -0.01 -0.01 -0.03  0.03  0.10 -0.04 -0.03
    9  0.08  0.09  0.29  0.12  0.18  0.08  0.10 -0.01        0.03  0.00  0.12  0.19 -0.01 -0.01
   10 -0.02 -0.01  0.13  0.02  0.04  0.03  0.01 -0.01  0.03       -0.02  0.02  0.07 -0.03  0.00
   11  0.03  0.04  0.14  0.02  0.07  0.09  0.01 -0.03  0.00 -0.02        0.02  0.08 -0.03 -0.02
   12  0.06  0.05  0.03 -0.03 -0.01  0.23 -0.03  0.03  0.12  0.02  0.02        0.00  0.02  0.05
   13  0.12  0.08             -0.01  0.35  0.01  0.10  0.19  0.07  0.08  0.00        0.08  0.11
   14  0.00  0.03  0.14  0.01  0.06  0.04  0.01 -0.04 -0.01 -0.03 -0.03  0.02  0.08       -0.03
   15  0.04  0.06  0.18  0.06  0.10  0.10  0.04 -0.03 -0.01  0.00 -0.02  0.05  0.11 -0.03
for locus: LAP-2
          1     2     3     4     5     6     7     8     9    10    11    12    13    14    15
    1        0.02 -0.01  0.00 -0.02 -0.01 -0.03 -0.02  0.00 -0.02  0.00 -0.01  0.05  0.00  0.06
    2  0.02       -0.01 -0.04 -0.01 -0.04  0.07  0.06  0.11 -0.02 -0.02 -0.02 -0.01 -0.02 -0.01
    3 -0.01 -0.01       -0.02 -0.02 -0.03  0.02  0.02  0.07 -0.02 -0.01 -0.02  0.01 -0.01  0.01
    4  0.00 -0.04 -0.02       -0.02        0.03  0.03  0.07 -0.03 -0.03 -0.03       -0.03
    5 -0.02 -0.01 -0.02 -0.02       -0.03  0.01  0.01  0.05 -0.02 -0.02 -0.02  0.01 -0.02  0.02
    6 -0.01 -0.04 -0.03       -0.03        0.03  0.02  0.06 -0.04 -0.04 -0.04       -0.04
    7 -0.03  0.07  0.02  0.03  0.01  0.03       -0.03 -0.02  0.01  0.04  0.02  0.09  0.04  0.11
    8 -0.02  0.06  0.02  0.03  0.01  0.02 -0.03       -0.02  0.01  0.03  0.02  0.08  0.03  0.09
    9  0.00  0.11  0.07  0.07  0.05  0.06 -0.02 -0.02        0.05  0.08  0.06  0.13  0.08  0.14
   10 -0.02 -0.02 -0.02 -0.03 -0.02 -0.04  0.01  0.01  0.05       -0.03 -0.03  0.00 -0.03  0.01
   11  0.00 -0.02 -0.01 -0.03 -0.02 -0.04  0.04  0.03  0.08 -0.03       -0.03  0.00 -0.02  0.00
   12 -0.01 -0.02 -0.02 -0.03 -0.02 -0.04  0.02  0.02  0.06 -0.03 -0.03        0.00 -0.03  0.01
   13  0.05 -0.01  0.01        0.01        0.09  0.08  0.13  0.00  0.00  0.00        0.00
   14  0.00 -0.02 -0.01 -0.03 -0.02 -0.04  0.04  0.03  0.08 -0.03 -0.02 -0.03  0.00        0.00
   15  0.06 -0.01  0.01        0.02        0.11  0.09  0.14  0.01  0.00  0.01        0.00
for locus: MDH-1
          1     2     3     4     5     6     7     8     9    10    11    12    13    14    15
    1        0.03  0.06  0.01 -0.03  0.14 -0.04 -0.04 -0.04 -0.02 -0.02  0.02  0.00 -0.02 -0.01
    2  0.03        0.00  0.10  0.03  0.01  0.02  0.01  0.00  0.07  0.01  0.12  0.10  0.08 -0.02
    3  0.06  0.00        0.14  0.08 -0.06  0.06  0.05  0.04  0.11  0.04  0.16  0.15  0.13  0.01
    4  0.01  0.10  0.14        0.02  0.28 -0.01  0.00  0.01 -0.02  0.04 -0.03 -0.03 -0.02  0.07
    5 -0.03  0.03  0.08  0.02        0.17 -0.03 -0.03 -0.03 -0.01 -0.02  0.02  0.01 -0.01  0.00
    6  0.14  0.01 -0.06  0.28  0.17        0.12  0.11  0.08  0.24  0.10  0.35  0.34  0.29  0.02
    7 -0.04  0.02  0.06 -0.01 -0.03  0.12       -0.04 -0.04 -0.02 -0.02  0.01  0.00 -0.02 -0.01
    8 -0.04  0.01  0.05  0.00 -0.03  0.11 -0.04       -0.04 -0.01 -0.03  0.03  0.01 -0.01 -0.02
    9 -0.04  0.00  0.04  0.01 -0.03  0.08 -0.04 -0.04       -0.01 -0.03  0.04  0.02  0.00 -0.03
   10 -0.02  0.07  0.11 -0.02 -0.01  0.24 -0.02 -0.01 -0.01        0.01 -0.02 -0.02 -0.02  0.04
   11 -0.02  0.01  0.04  0.04 -0.02  0.10 -0.02 -0.03 -0.03  0.01        0.05  0.03  0.01 -0.02
   12  0.02  0.12  0.16 -0.03  0.02  0.35  0.01  0.03  0.04 -0.02  0.05       -0.01  0.00  0.09
   13  0.00  0.10  0.15 -0.03  0.01  0.34  0.00  0.01  0.02 -0.02  0.03 -0.01       -0.02  0.07
   14 -0.02  0.08  0.13 -0.02 -0.01  0.29 -0.02 -0.01  0.00 -0.02  0.01  0.00 -0.02        0.05
   15 -0.01 -0.02  0.01  0.07  0.00  0.02 -0.01 -0.02 -0.03  0.04 -0.02  0.09  0.07  0.05
```

```
for locus: PEP-1
         1     2     3     4     5     6     7     8     9    10    11    12    13    14    15
   1        0.03  0.04 -0.03  0.02 -0.04  0.00  0.12  0.09  0.49  0.29  0.48  0.50  0.61  0.51
   2  0.03                                0.09  0.24  0.21  0.64  0.42  0.63  0.65  0.73  0.64
   3  0.04                                0.11  0.27  0.24  0.68  0.46  0.67  0.69  0.76  0.67
   4 -0.03                                0.00  0.11  0.08  0.46  0.27  0.45  0.48  0.58  0.48
   5  0.02                                0.06  0.20  0.17  0.59  0.37  0.58  0.60  0.68  0.59
   6 -0.04                               -0.01  0.10  0.06  0.45  0.26  0.44  0.47  0.57  0.47
   7  0.00  0.09  0.11  0.00  0.06 -0.01        0.04  0.01  0.41  0.21  0.41  0.43  0.54  0.44
   8  0.12  0.24  0.27  0.11  0.20  0.10  0.04       -0.03  0.21  0.05  0.22  0.24  0.36  0.25
   9  0.09  0.21  0.24  0.08  0.17  0.06  0.01 -0.03        0.26  0.08  0.26  0.28  0.40  0.29
  10  0.49  0.64  0.68  0.46  0.59  0.45  0.41  0.21  0.26        0.05 -0.02 -0.02  0.01 -0.02
  11  0.29  0.42  0.46  0.27  0.37  0.26  0.21  0.05  0.08  0.05        0.05  0.07  0.16  0.08
  12  0.48  0.63  0.67  0.45  0.58  0.44  0.41  0.22  0.26 -0.02  0.05       -0.02  0.01 -0.02
  13  0.50  0.65  0.69  0.48  0.60  0.47  0.43  0.24  0.28 -0.02  0.07 -0.02        0.01 -0.01
  14  0.61  0.73  0.76  0.58  0.68  0.57  0.54  0.36  0.40  0.01  0.16  0.01  0.01        0.00
  15  0.51  0.64  0.67  0.48  0.59  0.47  0.44  0.25  0.29 -0.02  0.08 -0.02 -0.01  0.00
for locus: PEP-2
         1     2     3     4     5     6     7     8     9    10    11    12    13    14    15
   1        0.03  0.04 -0.03  0.02  0.07  0.00  0.30  0.11  0.20  0.41  0.32  0.64  0.40  0.58
   2  0.03                          0.41  0.14  0.48  0.28  0.38  0.57  0.51  0.78  0.56  0.72
   3  0.04                          0.48  0.17  0.52  0.32  0.43  0.61  0.55  0.81  0.60  0.75
   4 -0.03                          0.07  0.03  0.30  0.12  0.21  0.41  0.31  0.63  0.40  0.57
   5  0.02                          0.36  0.12  0.44  0.24  0.35  0.54  0.47  0.75  0.53  0.69
   6  0.07  0.41  0.48  0.07  0.36       -0.06  0.02 -0.10 -0.04  0.14  0.02  0.36  0.12  0.32
   7  0.00  0.14  0.17  0.03  0.12 -0.06        0.19  0.02  0.09  0.30  0.20  0.53  0.29  0.48
   8  0.30  0.48  0.52  0.30  0.44  0.02  0.19        0.04  0.00  0.00 -0.03  0.14  0.00  0.11
   9  0.11  0.28  0.32  0.12  0.24 -0.10  0.02  0.04       -0.02  0.14  0.05  0.35  0.13  0.31
  10  0.20  0.38  0.43  0.21  0.35 -0.04  0.09  0.00 -0.02        0.07  0.00  0.26  0.06  0.21
  11  0.41  0.57  0.61  0.41  0.54  0.14  0.30  0.00  0.14  0.07       -0.01  0.05 -0.02  0.03
  12  0.32  0.51  0.55  0.31  0.47  0.02  0.20 -0.03  0.05  0.00 -0.01        0.11 -0.02  0.08
  13  0.64  0.78  0.81  0.63  0.75  0.36  0.53  0.14  0.35  0.26  0.05  0.11        0.05 -0.02
  14  0.40  0.56  0.60  0.40  0.53  0.12  0.29  0.00  0.13  0.06 -0.02 -0.02  0.05        0.02
  15  0.58  0.72  0.75  0.57  0.69  0.32  0.48  0.11  0.31  0.21  0.03  0.08 -0.02  0.02
for locus: PGM-1
         1     2     3     4     5     6     7     8     9    10    11    12    13    14    15
   1       -0.01  0.02  0.07  0.02  0.02 -0.03  0.35  0.08  0.57  0.34  0.42  0.83  0.79  0.76
   2 -0.01        0.05  0.19  0.06  0.15  0.02  0.49  0.18  0.70  0.48  0.56  0.89  0.85  0.83
   3  0.02  0.05       -0.01 -0.01  0.05  0.05  0.37  0.05  0.56  0.36  0.43  0.77  0.74  0.72
   4  0.07  0.19 -0.01       -0.02  0.05  0.11  0.25 -0.01  0.46  0.26  0.32  0.76  0.71  0.68
   5  0.02  0.06 -0.01 -0.02        0.05  0.05  0.34  0.04  0.54  0.34  0.41  0.77  0.74  0.71
   6  0.02  0.15  0.05  0.05  0.05       -0.01  0.16 -0.01  0.38  0.16  0.22  0.74  0.68  0.64
   7 -0.03  0.02  0.05  0.11  0.05 -0.01        0.33  0.08  0.55  0.32  0.40  0.83  0.78  0.75
   8  0.35  0.49  0.37  0.25  0.34  0.16  0.33        0.14  0.03 -0.03 -0.02  0.30  0.25  0.22
   9  0.08  0.18  0.05 -0.01  0.04 -0.01  0.08  0.14        0.33  0.14  0.19  0.61  0.57  0.54
  10  0.57  0.70  0.56  0.46  0.54  0.38  0.55  0.03  0.33        0.05  0.01  0.13  0.09  0.06
  11  0.34  0.48  0.36  0.26  0.34  0.16  0.32 -0.03  0.14  0.05       -0.02  0.32  0.27  0.24
  12  0.42  0.56  0.43  0.32  0.41  0.22  0.40 -0.02  0.19  0.01 -0.02        0.27  0.22  0.18
  13  0.83  0.89  0.77  0.76  0.77  0.74  0.83  0.30  0.61  0.13  0.32  0.27       -0.02 -0.01
  14  0.79  0.85  0.74  0.71  0.74  0.68  0.78  0.25  0.57  0.09  0.27  0.22 -0.02       -0.02
  15  0.76  0.83  0.72  0.68  0.71  0.64  0.75  0.22  0.54  0.06  0.24  0.18 -0.01 -0.02
for locus: PGM-2
         1     2     3     4     5     6     7     8     9    10    11    12    13    14    15
   1        0.03  0.04 -0.03  0.02  0.07 -0.02  0.33  0.03  0.57  0.38  0.46  0.87  0.82  0.81
   2  0.03                          0.35  0.09  0.50  0.17  0.73  0.54  0.63  0.93  0.90  0.90
   3  0.04                          0.41  0.11  0.55  0.20  0.76  0.58  0.67  0.94  0.91  0.91
   4 -0.03                          0.12  0.00  0.33  0.05  0.56  0.37  0.45  0.88  0.83  0.82
   5  0.02                          0.31  0.07  0.47  0.14  0.70  0.51  0.59  0.93  0.89  0.88
   6  0.07  0.35  0.41  0.12  0.31       -0.01  0.11 -0.07  0.34  0.16  0.22  0.74  0.68  0.67
   7 -0.02  0.09  0.11  0.00  0.07 -0.01        0.27 -0.01  0.51  0.32  0.40  0.83  0.78  0.77
   8  0.33  0.50  0.55  0.33  0.47  0.11  0.27        0.17  0.06 -0.02  0.00  0.39  0.34  0.33
   9  0.03  0.17  0.20  0.05  0.14 -0.07 -0.01  0.17        0.41  0.22  0.29  0.74  0.69  0.68
  10  0.57  0.73  0.76  0.56  0.70  0.34  0.51  0.06  0.41        0.02 -0.01  0.16  0.11  0.10
  11  0.38  0.54  0.58  0.37  0.51  0.16  0.32 -0.02  0.22  0.02       -0.02  0.32  0.27  0.26
  12  0.46  0.63  0.67  0.45  0.59  0.22  0.40  0.00  0.29 -0.01 -0.02        0.27  0.22  0.20
  13  0.87  0.93  0.94  0.88  0.93  0.74  0.83  0.39  0.74  0.16  0.32  0.27       -0.02 -0.02
  14  0.82  0.90  0.91  0.83  0.89  0.68  0.78  0.34  0.69  0.11  0.27  0.22 -0.02       -0.02
  15  0.81  0.90  0.91  0.82  0.88  0.67  0.77  0.33  0.68  0.10  0.26  0.20 -0.02 -0.02
```

```
         permutting alleles within samples.

         95% confidence interval.

      capf     theta    smallf
    0.2985    0.3316   -0.0526
    0.3663    0.3336    0.0518

         99% confidence interval.

      capf     theta    smallf
    0.2898    0.3313   -0.0661
    0.3790    0.3338    0.0713

(prob fis=0)=    0.57280

************************************************
         permutting alleles within total.

         95% confidence interval.

      capf     theta    smallf
   -0.0433   -0.0071   -0.0440
    0.0442    0.0083    0.0449

         99% confidence interval.

      capf     theta    smallf
   -0.0585   -0.0089   -0.0581
    0.0587    0.0115    0.0577

(prob fit=0)<    0.00020

************************************************
         permutting genotypes within total.

         95% confidence interval.

      capf     theta    smallf
    0.3104   -0.0137    0.2992
    0.3120    0.0183    0.3197

         99% confidence interval.

      capf     theta    smallf
    0.3102   -0.0164    0.2936
    0.3124    0.0265    0.3213

(prob fst=0)<    0.00020
```