

User's Guide to the Program **NewHybrids** Version 1.1 beta

Eric C. Anderson*

April 7, 2003

Abstract

NewHybrids is a program for computing the posterior distribution that individuals in a sample fall into different hybrid categories. The details of the algorithm are published in ANDERSON and THOMPSON (2002). This document describes how to use the program. In order to make the program accessible to researchers, and also for my own use of the program, I have created a graphical interface that allows the observation of most of the variables involved in the Markov chain Monte Carlo (MCMC) simulation. This is valuable for assessing the reliability of the results from the MCMC run. In order to take full advantage of these features, you will want to read *A User's Guide to the GLUT for Markov Chain Monte Carlo Graphical Interface* which is included in the standard distribution of the **NewHybrids** software as the file `GFMCUserGuide.pdf`. If you are like me, you will find it nice to start using the software immediately. For this reason, I have included a test data file and describe how to get the program running on that as quickly as possible in the "Quick Start" section (page 3). After that I go into more detail on the required data file format, program options, and the graphical features unique to **NewHybrids**.

This version (1.1 beta) is a pre-distributed version. The added features in this version are:

- Support and documentation for dominant markers like AFLP's (Section 3.2) There is currently nothing published in the scientific white literature on this, but I hope to have something published soon.
- The ability to specify, via the **z** and the **s** options, individuals of known hybrid category that may or may not be considered part of the mixture of interest. (Section 3.4)
- A simple way to change between a Jeffreys-type prior and uniform prior for π or Θ . This can be done during run-time in the graphical version to gauge the sensitivity of the analysis to these two different types of reasonable "not-too-terribly informative" priors (Section 3.5).
- The program now reports the alleles read in at each locus in the output file "aa-LociAndAlleles.txt". This allows the user to know which allele index in the program corresponds to which allele name in the data file. In the program, alleles get numbered sequentially starting at 0. The order of the alleles is given in "aa-LociAndAlleles.txt".
- A method for placing priors on the allele frequencies in the separate species that does not require the genotypes of all previously sampled individuals (Section 3.5.1).
- A non-graphical version of the program. For very large data sets or for long runs, it may be preferable to not run **NewHybrids** with all the graphical interface. This distribution now includes a graphics free version that does not require any of the graphics libraries that the other version requires.

*Department of Integrative Biology, University of California, Berkeley, eriq@u.washington.edu

It still lacks some features that I hope to add to the software soon. A further shortcoming is that I am not yet retaining (in a user-accessible way) the codes given to different alleles in the data set. Within the program, alleles get numbered internally from 0 to the the number of alleles at the locus minus one, but that internal numbering system may correspond in a mysterious fashion to how the alleles are numbered in your dataset. I will address these issues in a later version.

There are bound to be some bugs in the program. I have developed it under the Mac Operating System, but have ported it to Windows. Since I don't have a real Windows machine at my disposal, I have not been able to test the program extensively in that environment. Please be patient! And don't hesitate to send me email, or call me at my office: (510) 643-6299.

Contents

1	Components of the Distribution	2
2	Quick Start	3
3	Inputs To NewHybrids	4
3.1	Data File	5
3.2	Entering AFLP data	6
3.3	Genotype Frequency Classes	7
3.4	Individual-specific options	7
3.5	Priors	9
3.5.1	Extra prior information	9
3.5.2	Priors for species-specific alleles	12
3.6	Program-Wide Options	12
4	Graphical Output of NewHybrids	12
4.1	"Info" Window	12
4.2	Observed Data	13
4.3	Category Probs	13
4.4	Allele Frequencies	14
4.5	Complete Data LogL Trace	14
4.6	Kullback Leibler Div By Locus	14
4.7	Allele Frequency Histograms	14
4.8	Category Prob Histograms	15
4.9	Pi Histograms	15
5	Text Output of NewHybrids	15
5.1	Echoed Data	15
5.2	Program Results Output	15
6	Strategies for Use of NewHybrids	15
7	Implementation Notes	16
7.1	Treatment of Missing Data	16
7.2	Implementation of the Individual-Specific Options	16
8	Software Agreement	16
A	Bug Fixes and Release History	16

1 Components of the Distribution

The distributions of `NewHybrids` Version 1.1 for the Mac and Windows come with two different executable programs. `NewHybrids_Mac_1.1` and `NewHybrids_PC_1.1` are the executable programs that include a graphical interface for observing the progress of the MCMC algorithm. This can be very valuable to assess how well the Markov chain is mixing, and is especially useful in the beginning stages of working with `NewHybrids`. The graphical interface is described in Section 4. However, for very long runs, or for very large data sets, the computational overhead associated with the graphical interface may make it less desirable. Additionally, older computer systems may lack some of the libraries needed to drive the graphical interface. For this reason, there is also included in the PC and Mac distributions, a version of the program without the graphical interface, these are `NewHybrids_PC_1.1_WOG` and `NewHybrids_Mac_1.1_WOG` respectively. The “WOG” ending stands for “without graphics.” This version gives a simple text console interface.

The Mac and PC distributions also include several more files:

1. `GFMCMC_UserGuide.pdf` The *GFMCMC Guide* which gives information on the features available in the graphical interface.
2. `new_hybs_doc.pdf` This user-documentation file.
3. `TestDat.dat` A file of simulated data to use for making sure the program is running, and to get a feel for how the program works.
4. `TestDatWithOptions.dat` The same file of simulated data, except this one demonstrates the use of the `z` and `s` options.
5. `TestAFLP.dat` A file of simulated AFLP data. (This is what a collection of very informative (*i.e.*, essentially diagnostic) AFLP markers would look like.)
6. `TestAFLPWithOptions.dat` A file of simulated AFLP data but this one demonstrates the use of the `z` and `s` options.
7. `TwoGensGtypFreq.txt` The file that holds the definitions of the genotype frequency classes possible after two generations of mating between two species. (Note that the ordering of the genotype frequency categories has been altered from the previously distributed version of this file.)
8. `NewHybrids_PreDefdViews.txt` A file used by the graphical program to know how many windows to open for a particular view of the variables involved in MCMC.

There is not currently a Linux or Unix (source-code) distribution. I’ll get around to that eventually.

2 Quick Start

To start using `NewHybrids` as quickly as possible:

1. Double click the self-extracting archive and put the resulting directory where you would like to keep it on your hard drive.
2. To use the version with the graphical interface, you may have to get some system components for your computer. You can try skipping this step and seeing if things work. If they do—super. If not, or if things are very slow, you may need to take care of adding some software to your system:

- If you have a Mac with OS 9 or greater you shouldn't have to do anything. Otherwise see the Macintosh section of the "System Requirements" section of the *GFMCMC Guide*, and follow the directions for downloading the required software.
 - On a Windows machine you will have to install the glut32.dll dynamic linked library and maybe select a graphics installer. See the Microsoft Windows section of the "System Requirements" section of the *GFMCMC Guide*, and follow the directions.
3. Double click on `NewHybrids_Mac_1_1` or `NewHybrids_PC_1_1` (or the corresponding "WOG" versions and follow the directions to use one of the four test data files (by entering a 0,1,2, or 3).
 4. Then, when prompted to, enter a 0 to read the genotype frequency classes in the file "TwoGens-GtypFreq.txt".
 5. When asked to "Enter the name of a file holding prior allele frequency information or just type 0 to not include any prior information" enter a 0.
 6. Enter two integers for random seeds (when prompted to)
 7. **In the non-graphical version:**
 - (a) Enter 0's to choose a Jeffrey's prior for π and for θ .
 - (b) You will be prompted for the number of sweeps to use for burn in with the line, "Please enter the number of sweeps for BurnIn (must be an integer)". If you are just getting warmed up with the program, enter 1000 (for a real run with your own data, you will want to use much more burn in).
 - (c) You will then be prompted for the number of sweeps to use in computing the actual Monte Carlo averages with the line, "Please enter the number of sweeps to be done after BurnIn (must be an integer)". For now, just enter 5,000. When analyzing a real data set, you will want to use far more than 5,000.
 - (d) After entering that information, the program will report back every 5 seconds with a report of its progress (several lines giving the expected time to completion and the current and average values of the mixing parameter π).
 - (e) When it completes, the program gives a brief message about the files in which the output from the program can be found.
 8. **In the graphical version**, once the "Info" window opens, hit "1" on your keyboard and many more windows should open. Then hit the space bar to begin the MCMC execution. You should see something like the screen shot of Figure 1.

That pretty much gets things going to give you a general idea of what this program does. To quit from the graphical version of the program, choose Quit from the main menu, accessed by doing right-mouseclick in a program window (on the Mac, that is ctrl-mouseclick).

At this point, you should read the *GFMCMC Guide* to learn about controlling the simulation and managing the windows on the system for the graphical version.

3 Inputs To NewHybrids

You will want to run `NewHybrids` on your own data. `NewHybrids` requires a certain simple data format. This is described in the next subsection.

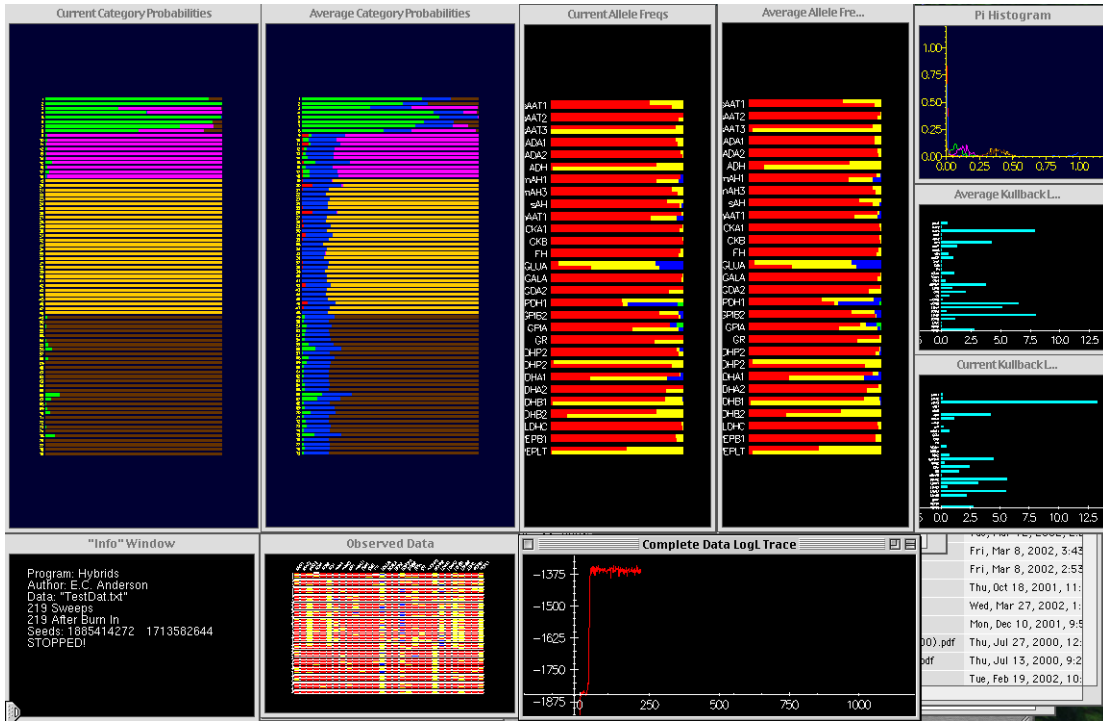


Figure 1: Screen grab of NewHybrids' first predefined view.

3.1 Data File

Open the file `TestDat.txt` to see what the structure is like. The first lines of any data file for NewHybrids must include some specific information. In the case of `TestDat.txt` that information is the prologue:

```

NumIndivs 79
NumLoci 49
Digits 1
Format Lumped

LocusNames  sAAT1 sAAT2 sAAT3 ADA1 ADA2 ADH. . .

```

The first line must have the word `NumIndivs` (spelled exactly) followed by whitespace (spaces and or tabs) and then the number of individuals in the data file. In this case 79. This is the number of individuals in the sample. The second line must have `NumLoci` followed by the number of loci at which the individuals have been typed. The third line has to have `Digits` followed by the number of digits used to denote a particular allele. In this case it is "1". This makes more sense when we consider the data format. On the fourth line the data format is given as `Lumped`. This is done by having `Format` followed by `Lumped`. The other option is `NonLumped` The `Lumped` data format means that the genotype at a single locus is given by a single number. We'll talk more about this in just a moment.

The following lines give the names of the loci. This is done by putting `LocusNames` in the file followed by the names of all the loci separated by white space. The number of names of loci must match the `NumLoci` given earlier in the file or strange/bad things may happen. If no locus names

are given (and `LocusNames` is omitted) then the loci will just be numbered by their order in the data set.

The next lines of the file are the actual genotype data. In `TestDat.txt` we have the first few lines of data: `TestDat.txt`:

```
1  11  11  11  0  11  11  11  11  11  11  11  11  11  32  11  11  21  11  11  11  11
   11  21  11  11  13  11  11  11  21  11  11  11  11  11  11  11  11  11  11  11
   11  11  11  11  11  11  11  11
2  21  11  21  11  11  12  11  11  12  11  11  11  0  11  11  11  12  11  12  11  11
   11  11  11  11  11  12  11  11  22  11  11  11  11  11  11  11  11  11  11  11  11
   11  11  11  11  11  11  11  11
```

The first character, the “1” is the number of the individual. You should index the individuals serially in your data file or the program will issue a number of warnings as it reads the data. Then, the next 49 numbers (they can be all on the same line or on different lines, it doesn’t matter) give the genotype of individual 1 at the 49 loci. These data are in the **Lumped** format. That means that 11 denotes that the individual has two copies of allele 1 at the locus in question. A 16 would mean that the individual carried a copy of allele 1 and a copy of allele 6 at the locus. The specifier `Digits` in the prologue of the data file refers to how many digits are used to represent each locus in lumped format. Thus, if `Digits` were followed by a “2” then the genotypes referred to above would be 0101 and 0106.

The numbers used to refer to the alleles need not be small numbers in series (*i.e.*, if you have only 4 alleles, they don’t have to be named 1, 2, 3, and 4. They could, for example, be lengths of microsatellite repeats, *etc.* They will automatically get converted into an index that is used internally by the program. I currently don’t have an easy way of retrieving which allele index used by the program corresponds to which allele number or length reported in the data file. I will rectify that eventually.

In the **Lumped** data format, missing data (*i.e.*, a locus that didn’t get scored at all) is denoted by a zero, “0”. This means that you can’t name any extant alleles “0”.

The other format is **NonLumped** in which the genotype at each locus is given by a consecutive pair of numbers that have white space between them. For example, with 9 loci, the data for the first four individuals might look like:

```
1 123 143 -1 -1 144 144 120 122 157 158 144 144 107 107 210 212 142 144
2 135 135 134 140 144 144 120 122 161 161 144 144 93 105 210 220 144 144
3 115 121 132 150 144 152 118 122 152 158 144 144 107 113 -1 -1 142 142
4 121 123 140 140 144 152 122 128 152 152 144 144 93 107 212 210 136 142
```

Once again we have the index of the individual followed by the genotypes. Individual 1 in the above example has a copy of allele 123 and a copy of allele 143 at the first locus. Missing data in this format is denoted by a -1, and each allele at the missing locus must have a -1.

Notice that the `Digits` parameter doesn’t really have any meaning here. Nonetheless, you have to include “`Digits X`”, where X is an integer, in the prologue of the data file for it to be read correctly.

3.2 Entering AFLP data

`NewHybrids` can deal with AFLP data (or other similar types of *dominant* autosomal markers), but you have to enter them in a certain way. The data format that must be used with AFLP markers (especially if they are used in combination with codominant markers in the same data set) is **Lumped**. AFLP loci either have a band present, denoted by a “+”, or the band is absent, denoted by a “-”, or the locus was not typed in an individual, in which case it is denoted by a “0”. Notice

that “0” and “-” mean very different things. As an example, the top lines of a dataset with four microsatellite loci and five AFLP markers should look something like:

```
NumIndivs 60
NumLoci 9
Digits 1
Format Lumped
```

```
LocusNames      m1 m2 m3 m4 A1 A2 A3 A4 A5
1      11  12  13  11  +  +  +  -  +
2      22  33  11  22  -  -  0  -  -
3      12  13  13  11  +  -  -  -  +
```

When `NewHybrids` reads the data, it will automatically classify the loci having -'s or +'s as AFLP loci and treat them appropriately.

3.3 Genotype Frequency Classes

Another required input to the program is the genotype frequency class file. This file specifies the different categories that individuals may fall into. These categories are specified in terms of the expected proportions of the loci within an individual which have 0, 1, or both genes originating from species 0 (or “population” 0) as opposed to species 1. (While we referred to the separate species as “A” and “B” in ANDERSON and THOMPSON (2002), we will refer to them as “0” and “1” here. The format for the file follows that of the file `TwoGensGtypFreq.txt` that comes with the distribution and looks like:

```
6
Pure_0      1.00000      0.00000      0.00000      0.00000
Pure_1      0.00000      0.00000      0.00000      1.00000
F1          0.00000      0.5          0.5          0.00000
F2          0.25000      0.250000    0.250000    0.25000
0_Bx       0.50000      0.250000    0.250000    0.00000
1_Bx       0.00000      0.250000    0.250000    0.50000
```

The first character in the file has to be an integer giving the number of different categories. In this case it is 6. Each category then gets a separate line. The first string on the line is the name given to the category. For example, “1_Bx” is the name for the category that corresponds to a Population 1 backcross (that is, the product of a Pure 1 individual mating with an F1 individual). The rest of the line gives the expected proportions of 00, 01, 10, and 11 genotypes (a 10 genotype is the one in which the first gene copy comes from population 1 and the second from population 0.) These proportions correspond to the quantities $G_{g,0}$, $G_{g,1}/2$, $G_{g,1}/2$, and $G_{g,2}$ in ANDERSON and THOMPSON (2002). Of course, it is silly to draw a distinction between a 10 and 01 genotype, because the gene copies are not really ordered in the individual. But, this is how the frequencies are fed into the program. The user should verify that the expected proportions for the 01 and 10 genotypes are the same for any category specifications they make. And, of course, the entries on every line should sum to one.

3.4 Individual-specific options

It may be the case that you have prior information about the hybrid status of some of the individuals you have collected. For example, you may have sampled some individuals in a region in which

hybrids are known not to occur. In that case, you would like to be able to specify that these individuals are of known and pure origin so that the genes within them can be used to estimate the allele frequencies from their species. However, you may not want those individuals to influence the estimate of the proportion of individuals from different hybrid categories in the mixture (since the mixture may have been collected from a different region). There are currently two options that may be given for each individual that influence these sorts of things. They are the **z** and **s** options.

These options are all passed to the program by including them on the individual's line in the dataset between the individual's number and the first locus. For example, the small example above might look like:

```
NumIndivs 60
NumLoci 9
Digits 1
Format Lumped
```

Indiv	LocusNames	m1	m2	m3	m4	A1	A2	A3	A4	A5
1	z0s	11	12	13	11	+	+	+	-	+
2	z1	22	33	11	22	-	-	0	-	-
3	z1 s	12	13	13	11	+	-	-	-	+
4	z 0	11	12	13	11	+	+	+	-	+
5		21	13	11	12	-	+	-	-	-
6		12	22	33	11	-	-	-	+	+

The meanings and usage of the options are as follows:

z: An individual is given the **z** option if it is known in advance which genotype frequency category it belongs to. This category must be specified by an integer that follows after the **z** in the data set. (There may be an arbitrary amount of spaces and tabs between the **z** and the integer, or no white space at all.) This integer should correspond to the number of the genotype frequency class as specified in the genotype frequency class file *where we start counting from 0*. For example, in the listing of `TwoGensGtypFreq.txt` given above, 0 is `Pure_0`, 1 is `Pure_1`, 2 is `F1`, and so forth. Hence, individuals #1 and #4 in the above listing have been specified as `Pure_0`, while individuals #2 and #3 have been specified as `Pure_1`, and the genotype frequency categories of #5 and #6 have not been specified.

s: The **s** option indicates that the individual is not to be considered part of the mixture relevant to the parameter π in ANDERSON and THOMPSON (2002). Such a circumstance arises when the researcher is interested in estimating the proportion of individuals sampled from a particular locale that fall into the different genotype frequency classes. For example, you might be interested in the occurrence of hybrids between escaped net-pen-reared Atlantic salmon and brown trout in a sample of juveniles taken from a particular river. However, you may also have sampled some adults from the net pens. Clearly, you would like to include those individuals in the analysis because they will help you infer the allele frequencies among the net-pen-reared salmon, *but* you can't simply include them in the sample of juveniles, because that would inappropriately inflate the estimate of the proportion of individuals among the juveniles that are pure net-pen-reared salmon. The solution is to apply the **s** option to the net-pen salmon (the **s** indicates that they were sampled separately from the mixture of interest.) In the example listing above, individuals #1 and #3 are indicated as sampled separately from the rest of the mixture of interest. Note: it doesn't really make sense to apply the **s** option to an individual unless you also apply the **z** option.¹

¹However, at some point, it might be nice to be able to specify several different samples that were taken along

The status of an individual with respect to the **z** and **s** options appears in the Category Probs Window described in Section 4.3. Basically, each individual's "color-bar" label is preceded by "z." if it has the **z** option only, and "z.s." if it has both the **z** and **s** options applied to it. The text label for individuals having the **z** option will appear in the color corresponding to the genotype frequency category to which it belongs (otherwise, the label will appear in the default text color for the window).

NOTICE!!! If you, for example, apply the **s** option to all the pure individuals that were sampled separately from the mixture of interest, *and* if there are no pure individuals in the mixture, then the estimate of the number of pure individuals in the mixture will be small, and on some iterations of the chain may be *very* small. This will cause the inferred current category probabilities for those individuals to jump around quite a bit. However, this shouldn't cause grave concern, so long as you seem to be getting good estimates of things for the individuals in the mixture.

A word to the wise—most typically individuals of known origin will be purely of one species or other. For this reason, it is most convenient to adopt the convention that the two pure categories always get the 0 and 1 labels (the first and second lines of genotype frequencies) in your file specifying the characteristics of the genotype frequency classes. That way, you can add other genotype frequency classes to the file (or use another file) without changing the numbers that follow the **z** option.

3.5 Priors

There are two types of priors—"Jeffreys-like" priors and the Uniform priors—available for both the mixing proportions and the allele frequencies. Both the Jeffreys-like prior (sometimes called the "unit-information" prior, in this case) and the Uniform prior should not influence the results of the analysis very much. However, one of the unfortunate features of this inference problem is that the prior on the nuisance parameter Θ (the allele frequencies) can be somewhat influential in the case where there are many alleles that are at very low frequencies *in both of the species*. In a perfect world, all species being compared would have large allele frequency differences, and our inference would not then rely to any great extent on alleles that appear at low frequencies in both of the species. However, sometimes that is not the case, and using the Jeffreys-like prior for the allele frequencies may provide more apparent sharpness to the inference about hybrid category of some individuals. However, these results may change when the Uniform prior is applied to that case. (The uniform prior asserts, in effect, that at least one copy of every allele has been found in both of the populations—thus downweighting the influence of an allele that is rare in one population, but absent in another). If the results seem to depend greatly on the choice of prior, then I would treat the results with considerable caution.

Instructions on changing between Jeffreys-like and Uniform priors is given in Section 4.1.

3.5.1 Extra prior information

If more prior information is available on the mixing proportions or on the allele frequencies, there are two ways to modify the priors that are used in the program. The first is to represent that prior information in terms of individuals that have already been sampled. For example, prior information probably will come from individuals of known origin that were previously sampled, and that information can be included in the analysis using the **z** option with those individuals. This prior information then gets added "on top of" the Jeffrey's-like or the Uniform prior².

a transect through a hybrid zone, say. Let me know at eriq@u.washington.edu if you have that sort of data situation—it might be just what I need to motivate me to implement a multiple-sample version of **NewHybrids**.

²In other words, the alleles counted in those individuals get added to the 1 (Uniform case) or the $1/(\# \text{of alleles})$ (Jeffrey's case) that already appears as the parameter of the Dirichlet prior)

The second way to incorporate prior information concerns only the allele frequencies, and involves including that information in a special file whose name you supply to the program at startup. It would be useful in the following type of scenario: Imagine you have sampled from a hybrid zone involving two species or populations that have been previously studied. In some of the previous studies, in which the species were sampled separately, allele frequency estimates for locus x were obtained from n_0 diploid individuals of species 0 and n_1 diploid individuals of species 1, and the estimated frequencies were $p_{0,1}, \dots, p_{0,k}$ and $p_{1,1}, \dots, p_{1,k}$ for the k alleles at the locus in species 0 and species 1 respectively. Clearly, you would like to use that prior information. Notice that this would be similar to you going out and collecting individuals and finding amongst the species 0 sample $2n_0p_{0,1}$ alleles of type 1, $2n_0p_{0,2}$ alleles of type 2, and so forth.

The procedure for including this sort of prior information is as follows:

1. First you must prepare a data file with the relevant information. To do this:
 - (a) Start the program and give it the name of your data file.
 - (b) When the program asks you to “Enter the name of a file holding prior allele frequency information or just type 0 to not include any prior information,” you should enter a 0.
 - (c) Proceed running the program as long or as little as desired. Then Quit out of the program.
2. At this point there will be a file called “aa-LociAndAlleles.txt” which lists the polymorphic loci in your data set and the alleles present at each of those loci.
3. The file “aa-LociAndAlleles.txt” will become the file that you will use to specify your allele frequency priors. Therefore, rename the file “aa-LociAndAlleles.txt”. Give it a name like “MyAllelePriors.txt” or something. For the remainder of this discussion we will assume it is named “MyAllelePriors.txt”.
4. Edit the file “MyAllelePriors.txt” to include the prior information that you have. This is done by adding to each allele line the number of alleles of that type reported in a previous study in species 0, followed by the number of that type of allele reported in species 1.
5. Then the next time you run the program when asked to “Enter the name of a file holding prior allele frequency information or just type 0 to not include any prior information,” you should enter the name “MyAllelePriors.txt”. The program will process the information in that file and will summarize the result to the screen.

In this case, the values that you specify for each allele at each locus will be added to the appropriate parameters of the Dirichlet prior distribution.

EXAMPLE I: SPECIFYING PRIORS IN A FILE: Let’s imagine you have a data set consisting of 100 birds typed at 3 loci, Loc1, Loc2, and Loc3, each having three alleles, coded as 1, 2, or 3. If you ran these data on `NewHybrids`, the file “aa-LociAndAlleles.txt” would be generated and would look something like:

```
3 polymorphic loci from 3 loci in data set
```

```

Locus Loc1
  2
  1
  3

```

```
Locus Loc2
```

```
3
1
2
```

```
Locus Loc3
```

```
1
2
3
```

(Notice that the alleles are listed in the order in which they appear in the data set, not in order of the numbers which specify them.) Now, suppose that your collection of birds consists of purebreds or hybrids of two species *A* and *B*. And let us imagine that a previous study genotyped 40 purebred individuals of species *A* and reported the following allele frequencies for locus *Loc2*: Allele 1 = .063, Allele 2 = .50, Allele 3 = .438. You don't have access to the original data, but you wish to include this information in the prior for species *A*. With 40 birds in this previous study, the allele frequencies reported probably came from a sample that had 5 alleles of type 1, 40 of type 2, and 35 of type 3 (because the numbers should sum to 80). But there may be some rounding error. To include this information in the prior for species *A*'s allele frequencies you would rename "aa-LociAndAlleles.txt" to something like "BirdPriors.txt" and then modify it to look like the following:

This line can be whatever as long as it doesn't start with "Locus"

```
Locus Loc1
```

```
2
1
3
```

```
Locus Loc2
```

```
3 35.04 0
1 5.04 0
2 40 0
```

```
Locus Loc3
```

```
1
2
3
```

(so long as the sample size is large, it should not matter too much whether or not you round to the nearest integer.) In this example, species *A* is the one that will be designated species 0 in the program, because the priors for its allele frequencies come immediately following the allele name on each line where a prior is given for *Loc2*. The column of zeroes under *Locus Loc2* are crucial. They indicate that there is not a previously obtained sample for Population B at *Locus Loc2*, and the 0's must be there for the program to properly read the data in.

To go just a little further, we could imagine that a previous sample of size 25 diploids was available for Species B (but not Species A) at locus *Loc3*, with estimated allele frequencies there as .37, .63, and 0.0 for alleles 1 to 3, respectively. In that case, you would want to change the file "BirdPriors.txt" to look like the following:

This line can be whatever as long as it doesn't start with "Locus"

```
Locus Loc1
```

```
2
1
```

3

Locus	Loc2	
3	35.04	0
1	5.04	0
2	40	0

Locus	Loc3	
1	0	9.25
2	0	15.75
3	0	0

3.5.2 Priors for species-specific alleles

Previous studies may have indicated that among the alleles of a particular locus, one of them is specific for one of the species, *i.e.*, it occurs in one species at some non-zero frequency but has never been reported in the other species. This type of situation can be dealt with by specifying allele frequency priors as above so long as one has a good idea of the sample sizes and estimated frequencies of the various alleles in the two different species. In such a case, it is still important to have estimates of the frequencies of all the alleles at a locus in the separate species. There is no facility in **NewHybrids** to declare just a single allele to be species-specific without having an estimate of all the allele frequencies at the locus.

3.6 Program-Wide Options

Will document these later.

4 Graphical Output of NewHybrids

We employ Markov chain Monte Carlo (MCMC) sampling to compute the desired posterior probabilities in this problem. MCMC samplers are prone to “mixing problems” in which the chain gets stuck in locally maximal portions of the parameter space, and do not “move” as well as they should. If you are dealing with genetically well-separated populations, then the chain should mix well. But there are cases in which it might get stuck in local maxima, etc. Being able to diagnose mixing problems is then of critical importance. For this reason, I have integrated **NewHybrids** into a graphical environment I developed for watching MCMC simulations unfold. Being able to observe all the variables involved helps considerably in observing how long it takes the chain to arrive in a region of the space that has reasonable probability under the target distribution, and also how vulnerable it is getting caught in different places.

A description of the different windows available to watch the variables involved in the simulations is given here. Consult the *GFMC* *Guide* to learn how to open these windows, *etc.*

4.1 “Info” Window

This window, which must remain open the entire time the program is executing, shows general information relevant to the run, *i.e.*,

- Program name
- Program author

- Name of data file being analyzed
- Number of sweeps that the MCMC analysis has been running
- Number of sweeps since burn-in (*i.e.*, the number of sweeps over which the Monte Carlo averages have been accumulating)
- The random number seeds
- Whether the simulation is running or stopped
- The type of prior in use for the allele frequencies, Θ
- The type of prior in use for the mixing proportions, π .

When the “Info” Window is the active window, hitting the “t” key will change the prior assumed for the allele frequencies from Jeffreys-like to Uniform, or vice-versa. Hitting the “p” key will do the same for the prior assumed on the mixing proportions. This is convenient—it allows you to see quickly how much influence the different prior types have on the result. Remember, if you are doing a long run to collect precise averages, you should reset all the averages (by hitting the “e” key) after selecting which prior to use.

4.2 Observed Data

This window shows the observed data. Each row is an individual (numbered on the left side), and each column is a locus (with the name appearing in text at the top of the column). The two gene copies carried by an individual are represented by two rectangles. Different colors refer to different allelic types. Boxes which are just outlines (*i.e.*, a frame box over the background color) denote missing data at that locus. For help with zooming in or out in the window, see the *GFMCMC Guide*.

4.3 Category Probs

At every iteration of the algorithm, the probability that each individual belongs to any of the different categories given the current values of the allele frequencies and the mixing proportions is computed. These are the quantities given in Equation 10 of ANDERSON and THOMPSON (2002). These current values (also called the “full-conditional” probabilities) are displayed in the window with the title “Current Category Probabilities.” The running average of these values is shown in the window titled “Average Category Probabilities.” These are the MCMC estimates of the posterior probabilities that are probably of greatest interest to the user—the posterior probability that each member of the sample falls into each of the different genotype frequency categories.

The appearance of each window is the same—individuals are represented by horizontal bars. The total length of the bar represents a probability of 1.0, and the length of bar occupied by each color represents the probability that the individual belongs to the genotype frequency category denoted by that color. To learn the meaning of the different colors, you need to view the legend for the window. Also, if you have many individuals in your sample, you will want to split the individuals into separate columns. See the *GFMCMC Guide* (sections on the Legend and “Controlling Column Numbers”) for instructions on how to do those things.

The text labels to the left of each bar carry information on the options which have been applied to particular individuals in the sample. See Page 9 for a discussion of that.) Note that, even though an individual may have the **z** option applied to it, the full conditional probabilities that it falls into different categories still get computed by default. [I may make this a user-controllable option at

some point to increase program speed when many individuals have the **z** option. However, in the meantime, it is quite informative to witness how well (or poorly) you can classify individuals of known category with the data you have collected.]

Another thing you can do from this window is select particular individuals (see the section on “Selecting Items” in the *GFMCMC Guide*) and then do a center mouse click (this might not work on some versions of Windows, and it is option-click on the Mac) to bring up a menu allowing you to open a histogram view of the Category Probs for the selected individual.

4.4 Allele Frequencies

The current allele frequencies and the posterior mean allele frequencies (estimated over the run of the chain since averages were reset) can also be viewed in windows named accordingly. Each bar represents a locus. Once again, the full length of a bar represents frequency of 1.0. The length of different colors on the top half of the bar represent the frequency of the different alleles at that locus in Population 0, and on the bottom half of the bar in Population 1. Again, loci can be selected in this window and center-clicked to access a menu giving you the choice of opening a histogram view for those allele frequencies.

4.5 Complete Data LogL Trace

IMPORTANT NOTE! In order to view the contents of this window, open the window and make it the current window, then hit the lowercase “v” key once the simulation has started running.

This window shows the last 1,000 values (with a line connecting them) of the complete-data log likelihood. This is the quantity given in Equation 5 in ANDERSON and THOMPSON (2002). This is a fairly informative quantity about how high the probability is in the region of the space in which the MCMC sampler is in. It is most useful for comparing the complete-data log likelihood between different locales where the sampler may get “stuck” for some period of time. For example, running the “TestDat.txt” data set from particular starting values, the chain may initially get caught for hundreds of iterations in a state where every individual is considered to be an F2 individual with high probability. When the sampler finally finds the right part of the space, the complete-data log likelihood is seen to increase by about 500 units. This is a massive increase in complete-data log-likelihood, so we can be fairly confident that the sampler was stuck in the earlier phase (all individuals as F2’s) merely because it was stuck there, not because that region of the space has globally high probability.

4.6 Kullback Leibler Div By Locus

There are two windows that show values associated with the Kullback-Leibler divergence between the populations at each locus. These are the “Current” and the “Average” Kullback Leibler Div By Locus windows. The higher the Kullback-Leibler divergence, the more informative is the locus for determining the hybrid category of the individuals in the sample. The Current values are the values associated with the current values of the allele frequencies, and the Average values are averaged over the run of the chain.

4.7 Allele Frequency Histograms

For the selected locus, this shows a histogram (with 50 bins) of estimated frequencies of the different alleles. The solid lines pertain to Population 0 and the stippled lines to Population 1.

4.8 Category Prob Histograms

Histogram with 50 bins of the posterior probability that the selected individual belongs to each of the genotype frequency categories.

4.9 Pi Histograms

This window shows a histogram (separated into 50 bins) of the posterior probability of the mixing proportions.

5 Text Output of NewHybrids

5.1 Echoed Data

NewHybrids outputs several text files after it has read the data in. These files are just printouts of the data that NewHybrids has read and stored in memory. The first few times you use NewHybrids on a new data set you should always have a look at these files and verify that they indicate that the program read your data correctly. Note that NewHybrids does not have very advanced error checking when reading data in!

EchoedGtypData.txt lists the genotype data that NewHybrids has read in. Each row is an individual. The first number in each row is the index of that individual, and the rest of the numbers on each row denote the individual's genotypes. The format is either Lumped or NonLumped depending on how the data were input. But missing data is always denoted by a -1.

EchoedGtypFreqCats.txt lists the genotype frequency category proportions read and used by the program.

5.2 Program Results Output

There are also three text files that get rewritten every 300 iterations of the chain.

aa-Pi.hist is a text file that holds histograms for the mixing proportions.

aa-Theta.hist is a text file that holds histograms for the estimated allele frequencies.

aa-PofZ.txt is a text file that holds the estimated posterior probabilities that each individual belongs to each of the different genotype frequency categories.

These are all white-space delimited files and you should be able to open them with any text editor, or with a spreadsheet or graphing program. Their format should be self-evident.

6 Strategies for Use of NewHybrids

Run the thing from lots of different starting points and see if the sampler always ends up in the same part of the space. If so, run it till it gets to that part of the space, then reset all the averages, then close all the windows except the "Info Window" (rendering the graphics takes up some time, and things run more quickly with fewer windows closed) and run the thing for a long time.

Note!! Depending on the starting conditions, one population or another may get the label "0" (with the other being population "1"). Without prior information about the allele frequencies in the different populations that might be contributing to your sample, there is no way of specifying which will be labelled "0" and which will be labelled "1". So, when you start from different starting conditions and get the same looking result, but with the colors "reversed," this is essentially the same result!

I'll expand on more strategies later. For now, just give me a call at (510) 524-1831 between 9 AM and 5 PM Pacific Time and we'll talk.

7 Implementation Notes

7.1 Treatment of Missing Data

Currently handles missing loci by ignoring them. This is fine if the data are missing completely at random. May someday write some code to impute missing values, if I convince myself that it will make a big difference.

7.2 Implementation of the Individual-Specific Options

Gonna have to write this down soon, before it becomes mysterious even to me, the author of the code.

8 Software Agreement

Copyright ©. The Regents of the University of California (Regents). All Rights Reserved.

Permission to use, copy, modify, and distribute this software and its documentation for educational, research, and not-for-profit purposes, without fee and without a signed licensing agreement, is hereby granted, provided that the above copyright notice, this paragraph and the following two paragraphs appear in all copies, modifications, and distributions. Contact The Office of Technology Licensing, UC Berkeley, 2150 Shattuck Avenue, Suite 510, Berkeley, CA 94720-1620, (510) 643-7201, for commercial licensing opportunities. Created by Eric C. Anderson, Department of Integrative Biology, University of California, Berkeley.

IN NO EVENT SHALL REGENTS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF REGENTS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

REGENTS SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE AND ACCOMPANYING DOCUMENTATION, IF ANY, PROVIDED HEREUNDER IS PROVIDED "AS IS". REGENTS HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

References

ANDERSON, E. C. and E. A. THOMPSON, 2002 A model-based method for identifying species hybrids using multilocus genetic data. *Genetics* **160**: 1217–1229.

A Bug Fixes and Release History

2 DEC 2002 Found a bug with allocating memory to hold the LocusTypes. Was allocating enough for the number of individuals, but not the number of loci. Thanks to Igor Emelianov whose data set had a number of loci exceeding the number of individuals, and hence exposed this bug.