User's Guide to the Program `NewHybrids` Version 1.0

Eric C. Anderson*

April 8, 2002

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 `GFMCMC_UserGuide.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 2). After that I go into more detail on the required data file format, program options, and the graphical features unique to `NewHybrids`.

This version (1.0) is the first distributed version. It lacks some features that I hope to add to the software very soon. In my mind, the greatest deficiency is that I have not added any features whereby the priors can be appropriately manipulated, or where samples of known individuals can be used to generate an appropriate prior. 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

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

1 Components of the Distribution

The distributions of `NewHybrids` Version 1.0 for the Mac and Windows come with executable programs `NewHybrids_Mac_1.0` and `NewHybrids_PC_1.0` respectively. They 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.txt` 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. `TwoGensGtypFreq.txt` The file that holds the definitions of the genotype frequency classes possible after two generations of mating between two species.
5. `NewHybrids_PreDefdViews.txt` A file used by the 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.

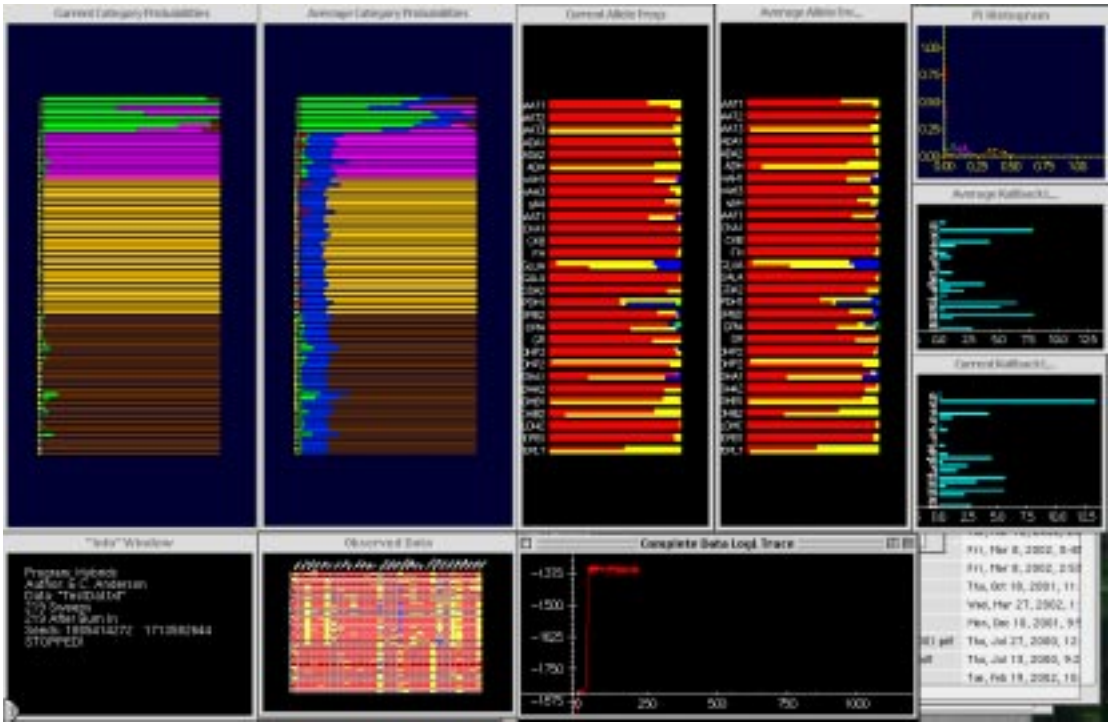

Figure 1: Screen grab of `NewHybrids`' first predefined view.

2. 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_1.0` or `NewHybrids_PC_1.0`, and follow the directions to use the TestDat.txt and the TwoGensGtypFreq.txt files. This involves entering zeroes when prompted to.

4. Enter two integers for random seeds (when prompted to)

5. 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 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.

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.

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
```

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 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
1_Bx      0.00000    0.250000    0.250000     0.50000
0_Bx      0.50000    0.250000    0.250000     0.00000
F2        0.25000    0.250000    0.250000     0.25000
F1        0.00000      .5          .5         0.00000
Pure_1    0.00000    0.00000     0.00000      1.00000
Pure_0    1.00000    0.00000     0.00000      0.00000
```

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.

3.3 Priors

Currently, the only priors supported in the documented user's version of the program are the "Jeffreys-like" prior for the mixing proportions and the allele frequencies. It's just going to be a matter of me getting around to writing a few lines of code to change that, but I want to get this first version out now.

3.4 Program and Individual Options

Currently there are no major program options or "individual-level" options. In the future I plan to add support for including individuals of known origin or category in the data file and then converting the data in them into an updated prior for the mixing proportions or allele frequencies in user-specifiable ways. But that will be in a later version also.

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 *GFMCMC Guide* to learn how to open these windows, *etc.*

4.1 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*.

4.2 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 inviduals into separate columns. See the *GFMCMC Guide* (sections on the Legend and "Controlling Column Numbers" for instructions on how to do those things.

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.

4.3 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.

4.4 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.

4.5 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.

4.6 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.

4.7 Category Prob Histograms

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

4.8 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 alelle 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.

8 Software Agreement

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.