GenomeMixer 0.54c (March 30, 2004)

http://www.nervenet.org/genome_mixer/
rwilliam@nb.utmem.edu

1. What is GenomeMixer?

GenomeMixer is a cross-platform program that allows you to simulate complex multi-generational breeding schemes for up to several hundred cages (or plots) and several hundred generations. GenomeMixer will handle from 2 to 20,000 parental strains. The number and length of chromosomes is fully editable, and you should be able to model almost any sexually reproducing diploid species.

---

2. What do I need to run GenomeMixer?

GenomeMixer is available for Mac OS X, Windows, and Unix/Linux, from <http://www.nervenet.org/genome_mixer/>. If you are running on a Macintosh or Windows system, you should be able to download GenomeMixer as a double-clickable application. On Linux, you can download an x86 binary, but if you are not able to find a binary for your particular system, you can compile GenomeMixer from the source.

You can also compile GenomeMixer from the source on a Macintosh or PC. For compiling the source on any platform, however, you will need to download Trolltech Qt (http://www.trolltech.com) first. Be aware that Qt is over a hundred megabytes when installed, and takes several hours to compile.

> **2b. Technical Question: I want to compile GenomeMixer from source. Once I have installed Qt, how do I compile the GenomeMixer source?**
> Once you have installed a copy of Qt, you should then be able to `cd` to the GenomeMixer directory, type `qmake`, then type `make` once that is finished. Aft er that, there should be a GenomeMixer executable program for you to run with "`./GenomeMixer`".

> **2c. Technical Question: How do I compile Qt in the first place?**
> Unfortunately, that is beyond the scope of this file. We suggest that you look for help on the Trolltech web site.

3. How to run GenomeMixer:

> **1.** Use a web browser to download GenomeMixer from the links at <http://www.nervenet.org/genome_mixer/>.

> **2.** Decompress the downloaded file with a utility like StuffIt Expander.

> **3.** On Mac OS X or Windows, you should now be able to double-click the GenomeMixer icon to launch it. On Linux, you can run the program from the command line, but you must have the capability of viewing an X-Windows display on your terminal (text-only will not work).

Mac OS X and Linux are also capable of using X11 to run GenomeMixer from a (perhaps faster) remote machine. If you have GenomeMixer installed on one Linux machine, or under the X11 environment on Mac OS X, you can run copies of it remotely on other machines (you will normally have to login with the "`-X`" option, indicating that the computer you logged into should export the display over the connection to your machine: for example, "`ssh -X myname@mixer.computer.xyz`".

Now that you have GenomeMixer running properly, you should see a window that looks something like one of these:
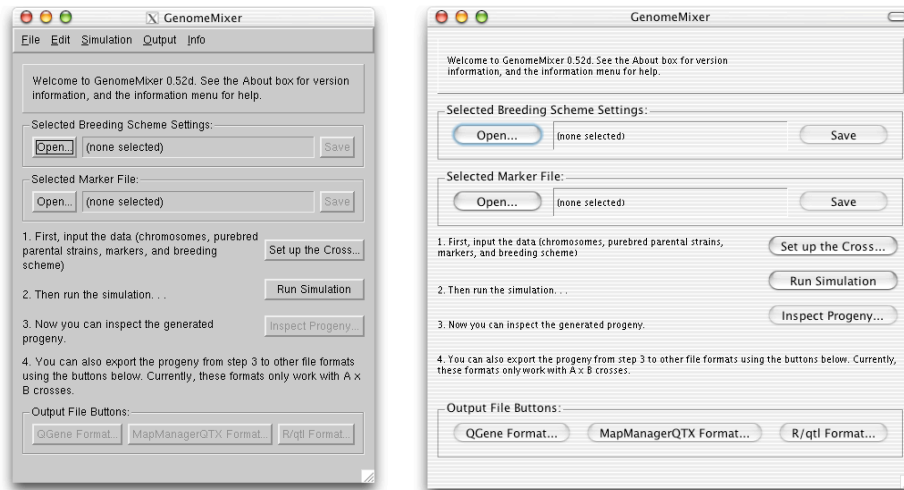
The window to the left is an example of using GenomeMixer over the network; the screenshot was taken on a Mac OS X system using Apple X11 (notice the Aqua buttons at the top), but it is actually running *on* a Linux machine (hence the appearance of the dialog itself). The version on the right is running natively on OS X.

A. The Menu bar

Here is a basic overview of the menu options. Note that if you are using the Mac OS X version, the menu bar will be at the top of the screen instead of being attached directly to the GenomeMixer window.

i. File Menu

Contains options for creating new Settings and Markers. The "Settings" file contains all of the data required for a cross except for the markers on each chromosome, which are stored separately in the Markers file. The Quickload item immediately loads in files from the Saved_Settings folder with the special names `quickmarkers.mar` and `quicksettings.gmx`. This saves you the trouble of using the "open" dialogs to choose your settings each time, if you save your files (or, copies of your files—safer, since there is no way to accidentally overwrite the originals) as `quickmarkers.mar` and `quicksettings.gmx` inside the Saved_Settings folder. For more information about the markers and settings files, see below.

ii. Simulation Menu

Contains the steps, in order, for running the simulation. First, you enter the markers and settings (this is the same as the "Set up the Cross..." button). Next, you run the simulation, and finally you inspect the resulting progeny.

iii. Output Menu

The options in this menu correspond to the various output file types. They are disabled (grayed out) until you actually run a simulation and have progeny to export. The three "linebreak" options at the top allow you to specify what operating system's linebreak format should be followed. More information about the output file formats is available in the "output file types" section below.

iv. Info Menu

The info menu contains a limited amount of online help. In general, it is less thorough than this readme file.

B. The Output File Types

The output menu contains a list of all of the file formats that GenomeMixer can write to. The results of a cross can be exported to various other formats, so that you can take advantage of other programs' analysis features on data exported from GenomeMixer. Most of these programs handle only simple 2-way (A x B) crosses.

i. QGene Format: http://www.qgene.org/

QGene is a Macintosh application (OS 9 only, as of August 2003) by J. C. Nelson. It has excellent support for visualization of marker data.

ii. MapManager QTX: http://mapmgr.roswellpark.org/mmQTX.html

MapManager QTX (Manly, KF, Cudmore, Jr, RH, Meer, JM) runs under Mac OS X, OS 9, and Windows, and is used for complex trait analysis. More information on how to import a QTX file is available below.

iii. R/qtl: http://www.biostat.jhsph.edu/~kbroman/qtl/

R/qtl by Karl Broman is a mapping program that runs as a module to the R statistical programming environment. R is a free download for nearly all modern computer systems, and R/qtl can run on any computer that supports R.

iv. Plain Table format:

This export format is just a simple two-dimensional table, with markers at the top of each column (markers from *all* chromosomes), and each individual as two rows of data (two rows of marker data, rather than one, because the first row lists marker data for the maternal chromosomes of the individual in question, and the second row lists marker data for the paternal chromosomes).

> **v. HAPPY:** http://www.well.ox.ac.uk/happy/formats.shtml
>
> No support is available yet for HAPPY by Richard Mott, but this program can support up to 8-way crosses (8 parental strains). In the future, exporting to this format may be extremely useful for complex GenomeMixer crosses.

C. Settings and Running the Simulation

The "Open" buttons let you specify an already-existing GenomeMixer settings file or markers file. You can open the "quickload" markers and settings that come with this distribution, if you like, or select "New..." from the File menu and make your own. The three buttons to the right let you edit the current settings, run a simulation using the current settings, and inspect the generated progeny (if any: this button is disabled if there are no progeny to inspect). We discuss the settings in further detail below.

4. Setting up the simulation:

The best way to figure out how to edit the GenomeMixer settings is probably to load in a test file and change the settings until you have a reasonable idea of what is going on. You can load the sample settings that come in the `Saved_Settings` folder (or you can select "Quickload" from the File menu to automatically load the files in the `Saved_Settings` folder named `quickmarkers.mar` and `quicksettings.gmx`).

If you click on the "Set up the Cross" button (or if you select "Edit Markers and Settings" from the Simulation menu), a new window will appear with four tabs at the top: Chromosome, Markers, Purebreds, and Breeding Scheme. Here is an overview of each of the four tabs:

Chromosome Tab (Mac interface):
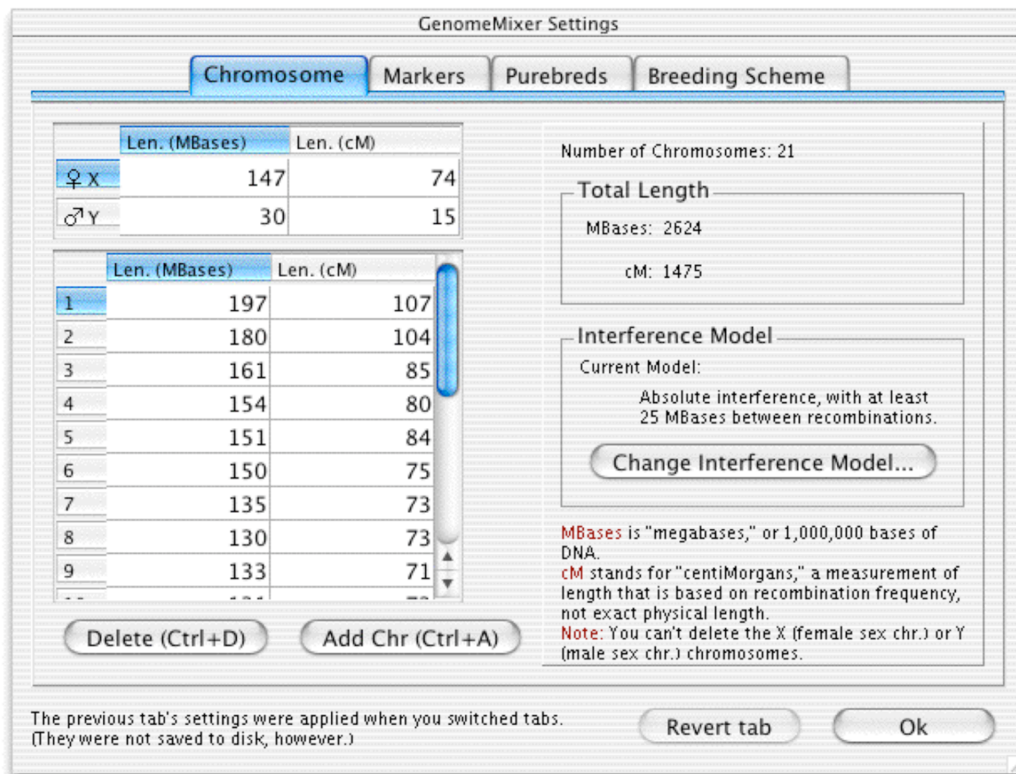
In the chromosome tab, you specify the lengths of each chromosome in megabases (millions of base pairs) and centimorgans (cM: a measurement of length based on recombination frequency). The values that come in the default settings files are values for the mouse.

> A. The sex chromosomes are at the top left. In our model, females have two X chromosomes, and males have an X and a Y chromosome. If the species you want to model works differently from this, you will have to keep that in mind.
>
> B. In the table, the values on the left—under "Len (Mbases)"—are the lengths in megabases (MBases). This is not really important as far as recombination is

concerned, but it is used internally in a limited way, so using approximately correct values is best. Values from 50 to 200 are not unusual, but of course this will vary depending on the species being modeled.

C. In the table, the values on the right—under "Len (cM)"—are lengths in centimorgans (cM). The greater this value, the more recombinations we expect, on average, in the chromosome. This value is not based directly on the physical length of the chromosome. (It is possible to model recombination hotspots by specifying a large cM interval between two markers, in the markers tab.)

D. The "Change Interference Model" button allows you to decide what kind of effect a recombination has on the probability of another recombination in an adjacent region of the chromosome. You can also turn interference off, so that recombinations become Poisson-distributed events. Currently, only absolute interference (requiring at least X bases between recombination events) is implemented. Recombinations are randomly placed based on centiMorgan values between markers (not base locations). Thus, a recombination hotspot can be modeled by a sequence of markers that are close together in bases, but with a relatively large centiMorgan distance between them. Each centiMorgan has a 1% chance of containing the site of a recombination, disregarding interference effects.
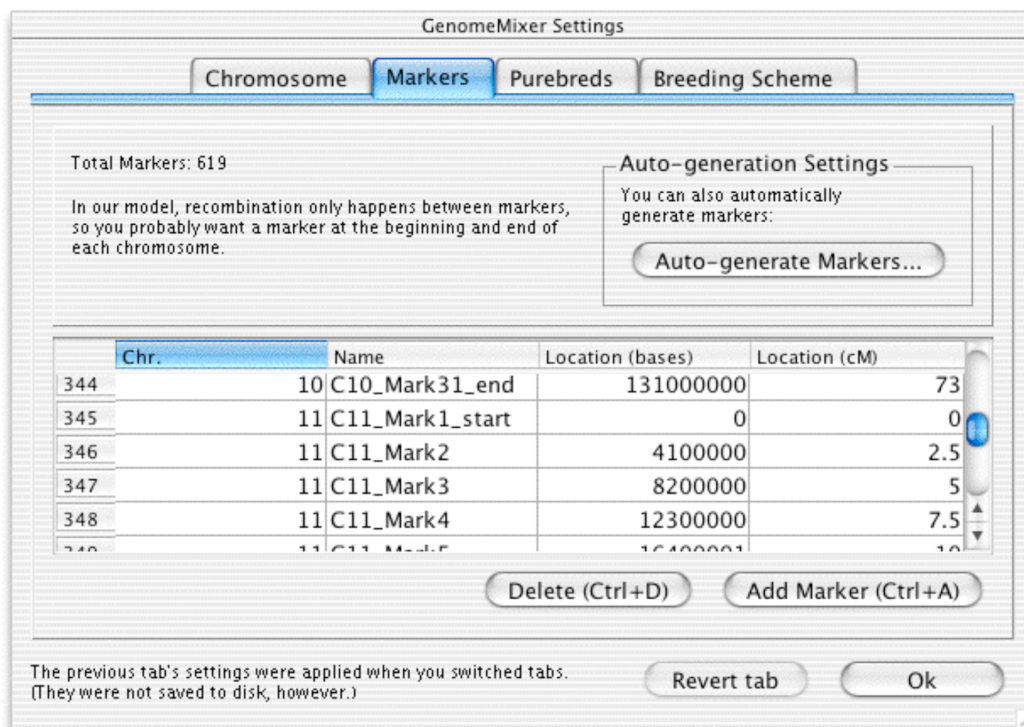
Markers Tab (Mac interface)

In the markers tab, you specify the lengths of each chromosome in megabases (millions of base pairs) and centimorgans (cM: a measurement of length based on recombination frequency). The values that come in the default settings files are (approximately correct)

values for the mouse.

    A. The add button adds a new marker after the currently selected one. The delete button deletes the currently selected record. Recombination only occurs between markers in our model, and two adjacent markers may only have one recombination between them per generation at most. It is possible to model recombination hotspots by specifying a large cM interval between two adjacent markers.

    B. The auto-generate markers button brings up a new dialog box that allows you to automatically fill in the marker table with markers spaced evenly over a certain interval. The only exception to the even spacing is that auto-generate will always create a marker at the beginning and end of each chromosome.

    C. There is no "Cancel" button, only an "Ok" button and a "Revert" button. The "Ok" this button will apply your changes and return to you the main GenomeMixer window. Since there is no cancel button, you can revert the settings to the way they were when you last selected the current tab by clicking the "Revert tab" button. *Note that switching from one tab to another applies your changes.* Note that "Ok" does not save your changes to disk. If you want to save changes to disk, you will have to select Save from the menu, after clicking "Ok."

Purebreds Tab (Mac interface)
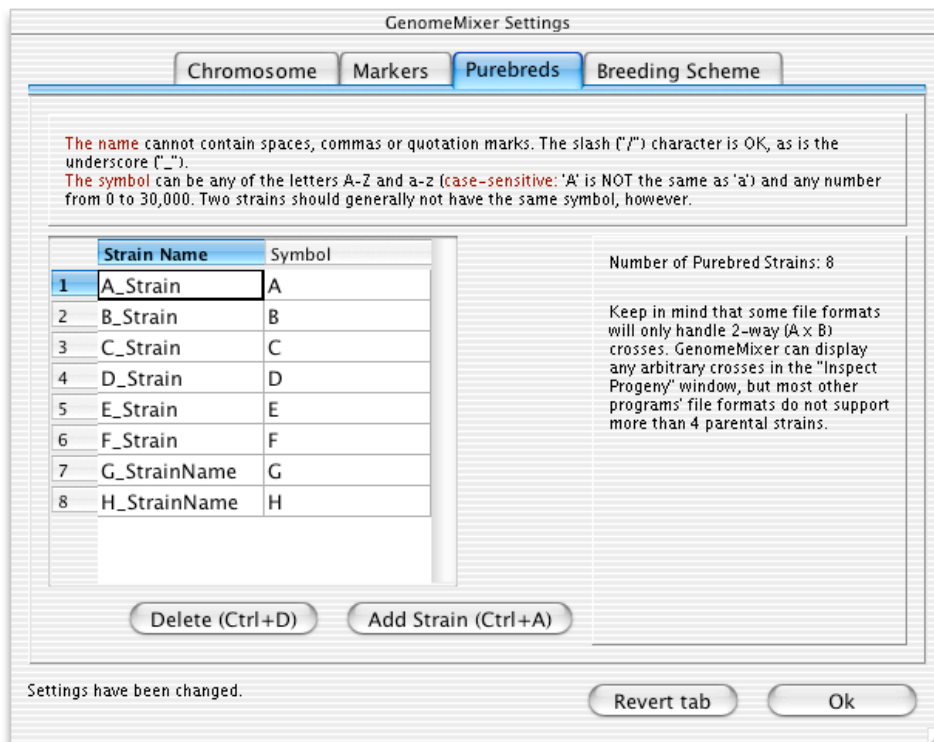
In the purebreds tab, you specify the names and symbols associated with the parental purebred strains.

A. The Strain Name field is for a reasonably human-readable name for each purebred strain. You can name your strains "AA" or "BXD32," or anything without a comma, space, or quotation marks. Note that the strain name must be a single word: so instead of using spaces, you should use the underscore character.

B. The Symbol field should be either a single character (A, a, B, b, etc…) or a number from 0 to 30,000. If you use a capital letter or a number from 0 to 9, then the Progeny Inspector dialog will be color-coded, so it is recommended that you avoid lower-case or numbers greater than 10 for the symbol if possible. If you need to, you can set different Strain Names to have the same symbol. Normally you probably would not want to do this, but if you want to export GenomeMixer data to an output format that only supports 'A' and 'B' parents, you might have to set some parental strains to have the same symbols.

Breeding Scheme Tab (Mac interface)

This is the tab in which you set up the actual breeding scheme. You will have to set up at least two parental strains in the purebreds tab before you can do any meaningful crosses in this tab.

A. Each cell in the breeding scheme table specifies a particular cage and a particular generation. You can use the controls at the bottom to edit the cell, or you can double-click on the cell and edit the text manually. To delete a cell, double-click it and delete all the text in it, and then when you click away from that cell, the data will be gone. **Important:** Note that when you are editing the parents, any changes you type in will not take effect until you hit return / enter. Selecting a different cell does *not* automatically save the changes in the parental fields.

B. The parental selector controls allow you to choose the parents for the progeny to be generated for the current cell. You can select "Purebred" from the pulldown menu, and it will bring up a list of valid purebred parents, or you can select "From Cell..." and then type in the identifier for a particular cell (for example, the top left cell is A1, as specified by the letter in the column header "(A) Gen 1" and the number from the row header "Cage (1)"). This is the same format that most spreadsheet programs use. Note that you cannot "forward reference" a cell; if you have selected cell B2 (Generation 2, Cage 2), then you cannot request that an individual from generation 3 (column C) be a parent, because the individuals in generation 3 would not exist at that point. All of the cells in the leftmost column (generation 1 (A)) should have purebred parents. Other cells may have purebred parents, or may specify parents from any of the cells to their left (that is, any earlier generation).

C. The ♀ symbol represents the female parent and the number of females to be generated. In the Progeny Inspector, it also indicates the chromosome inherited from the individual's mother. The ♂ symbol indicates the male / paternal versions of the same things.

D. The progeny controls let you decide how many individuals will be generated for this cell, from the mating specified by the parental selector controls. The default generation is 1 male and 1 female, but you can increase (or decrease) these numbers, and add individuals with randomly-assigned sex by incrementing the random counter.

5. Simulation and Results:

Once you have specified all of the data in the four-tabbed "Edit Markers and Breeding Scheme" window, you should click "Ok" and return to the main window. There, you will be able to click on the "Run Simulation" button (or select "Run Simulation" from the Simulation menu).

GenomeMixer will indicate that it has finished running the simulation by updating the general information text to say "Simulation finished, running time was X seconds." If an error occurs during the simulation, an error box will pop up to inform you of the error, in which case you can go back and edit the breeding scheme to fix the problem.

If the simulation completed without any errors, you can now press the "Inspect Progeny" button (or select "Inspect Progeny" from the Simulation menu). A new window like the one below will appear.
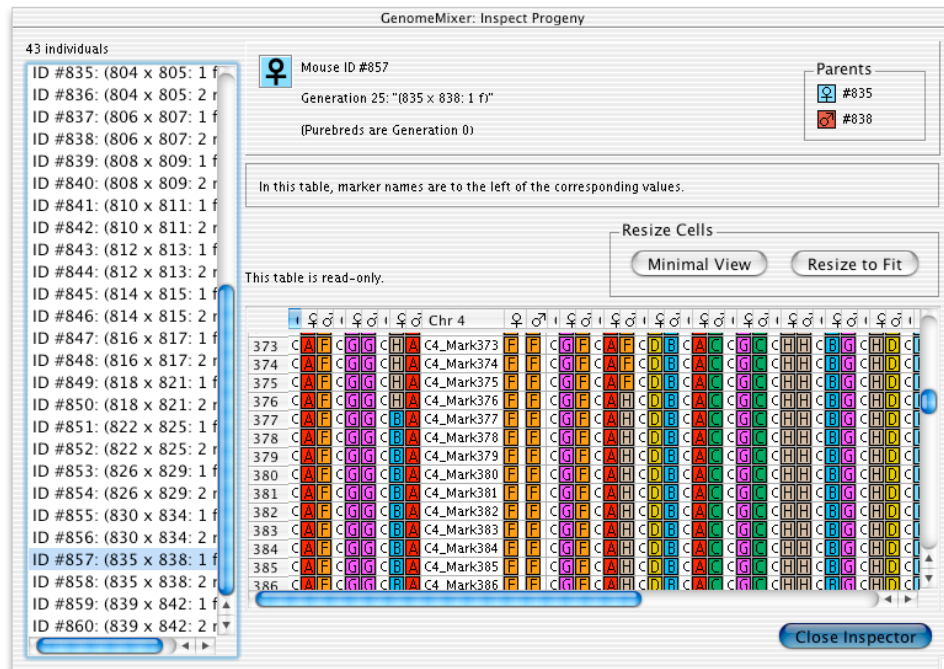
This window is the "Progeny Inspector," where you can look at the genotype at each marker location for each individual. Note that although this view only allows you to inspect chromosomes at marker resolution, recombinations are handled internally at the base pair level. If the marker density is not high enough, there is the possibility of recombinations occurring but remaining undetected.

    A. The left side of this window is a list of all of the individuals from cells that were tagged as "to be exported" in the breeding scheme dialog. The list is ordered by ID, which is determined by the order in which the individuals were generated. Purebreds have IDs as well, but they are not exported, which is why the list does not necessarily start at ID #1.

B. The "Parents" region in the top-right corner indicates which individuals were mated to produce this offspring. Each individual has a unique ID number, so it can be easily determined if two individuals have the same parents, for example.

C. The two "resize cells" buttons automatically resize all of the columns. Note that you can also manually resize columns and rows (that is why the columns in the sample picture are different sizes).

D. The inspection table itself, in the lower-right section of the window, gives you an overview of a selected individual's genotype. The genotype of whichever individual that was selected from the list at the left will appear in this section. Each group of three columns represents a chromosome; one column is labeled with the chromosome number (e.g. "Chr 13"). This column contains the names of all the markers. To the right of each marker is the genotype at that marker, from the chromosome inherited from the mother first (with the female symbol as the column header), and then from the chromosome inherited from the father (with the male symbol as the column header).
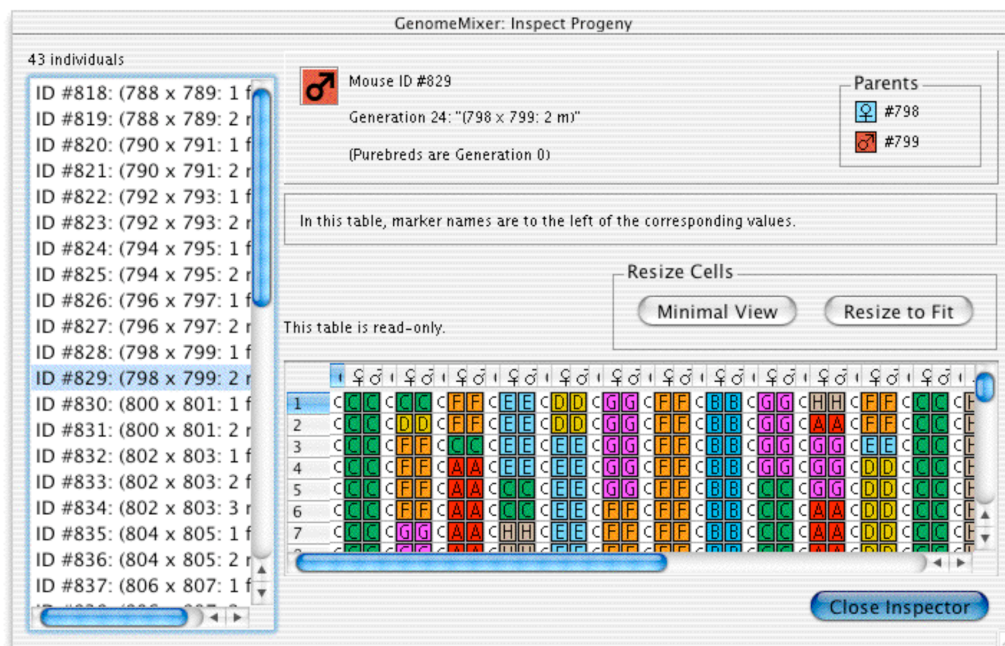
Here, the Inspect Progeny window is displaying information for individual #829, which is a recombinant inbred male (evident in the homozygous but non-uniform genome). Using a higher-density marker map, we would find that the genome is not 100% homozygous, even though it is extremely close. This individual was created in the 24th generation, and its parents were #798 (female) and #799 (male).

6. Importing / Exporting:

Importing Markers in large numbers (from Excel, or text):

GenomeMixer's built-in marker editor is suitable for editing small numbers of markers by hand and for automatically generating markers at evenly-spaced intervals. However, entering large numbers of markers by hand is tedious, and if you already have markers in another format, you will want to import them instead of re-entering them.

GenomeMixer marker (`.mar`) files consist of the following:

*Regular text comments at the top of the file (**warning**: if you overwrite this marker file, GenomeMixer will not save the comments).*

<START MARKERS> (*must be alone at the very start of a line*)

<MARKER> (`tab`) **Name** (`tab`) **Chr_Number** (`tab`) **Loc_in_bases** (`tab`) **Loc_in_cM** (newline*)

<MARKER> (`tab`) **Name** (`tab`) **Chr_Number** (`tab`) **Loc_in_bases** (`tab`) **Loc_in_cM** (newline*)

…

<MARKER> (`tab`) **Name** (`tab`) **Chr_Number** (`tab`) **Loc_in_bases** (`tab`) **Loc_in_cM** (newline*)

(*end of file: nothing special at the end of the file*)

*Newline: If you are on a Unix system, then "newline" is the default line ending. However, if you are on a Mac or Windows system, the default line ending contains a "carriage return." The "newline" is represented by "\n", and the "carriage return" is represented by "\r". If you are on a Mac or Windows system, you will need to use a utility (such as BBEdit Lite or Josh Aas' "LineBreak" for the Macintosh) to convert your linebreaks to "Unix format," or GenomeMixer will not be able to read them.

Chr_number should be the number of the chromosome this marker is on, from 1 onwards. This should always be a number, and never "X" or "Y". The X chromosome is considered the next-to-last chromosome, and Y is considered the last chromosome. So if you have a species with 19 autosomes plus X and Y, "X" should be chromosome 20, and "Y" should be chromosome 21.

Loc_in_bases should be an integer. It is common for the location of a marker in bases to be up to several hundred million. If a chromosome is 150 megabases long, then the last marker could be located at 150000000. You cannot use commas in the numbers (although you could type the locations out with commas and then do a search-and-replace to delete them all in the end), and the loc_in_bases cannot be a decimal. Two markers on the same chromosome should not share the same location in bases.

Loc_in_cM, on the other hand, will frequently have a decimal point. The length of a chromosome in cM (centimorgans) could be 85.3, 120.425, or any other floating-point (or integer) number. This should probably be no greater than 1,000, and in most cases will probably be no greater than 100. If two markers on the same chromosome share the same cM value, then there is no chance of a recombination occurring between them.

None of these four parameters can be blank. Each marker must have a name, chromosome number, location in bases, and location in centimorgans. Location in centimorgans is the most important as far as recombination simulation is concerned; if you don't know a location in bases, you can just put in a reasonable estimation.

Here is an example of a short, but completely valid, marker file:

```
Some_Random_Marker_Comments_That_Won't_Be_Saved
<START MARKERS>
<MARKER>    MarkerA      1    0            0
<MARKER>    TheMarker    1    100002       10
<MARKER>    AloneMarker  2    70000000     20.51
```

"MarkerA" is at the start of Chromosome 1; hence, it's "loc_in_bases" is zero, as is its "loc_in_cM.." In most cases, there should be a marker at the very beginning (0 bases, 0 cM) and very end of every chromosome.

Also, notice that in the above example, no recombinations will occur on chromosome 2, because only one marker (the "AloneMarker") is defined on chromosome 2, and in our model, recombinations only occur *between* markers. If the marker density is too low (< 5 per chromosome), you may get unrealistic recombination effects, because only one recombination can occur between markers (although it can be located anywhere between them—unless the chromosomes are < 1 megabase in length, there is virtually no chance of one recombination occurring and then, in a later generation, a subsequent recombination occurring in exactly the same spot).

Here is the step-by-step procedure for converting an Excel or text document into a GenomeMixer marker file:

1. Edit marker file in spreadsheet program or text editor.

2. Save in **plain text** format (not Excel format, for example)

3. Use a program like BBEdit to convert the linebreaks to "Unix format."

4. Rename the file so that it has the extension "`.mar`"

5. Import the file into GenomeMixer by selecting "Open" and choosing this marker file.

6. When you go to "Edit Markers / Breeding Scheme" and select the "Markers" tab, verify that the imported markers look the way you expected them to.

If the markers do not appear in step 6, or if there is only one marker, there are a number of possibilities. Perhaps `<START MARKERS>` is not at the beginning of its own line, so GenomeMixer just never read any markers in the file. Perhaps `<MARKER> (tab)` is not at the beginning of every line, or perhaps the file is not Unix-linebreak formatted (check the troubleshooting section for linebreak information). It is also possible that the file was saved in some other format that contains extra data (Excel `.xls`, for example), instead of plain tab-delimited text.

Exporting data for MapManager QTX:

These are the steps required to make use of exported data in MapManager. After you run a simulation, you can save the exported progeny in MapManager QTX format by selecting the MapManager option from the output menu. A few steps are still required before MapManager can make use of the data.

1. Take note of the "`(n=…)`" part of the filename in the MapManager-format file you just saved. If the file is named "`mm_export_(n=23)`", then there are 23 individuals in the file.

2. Launch MapManager QTX, and from the File menu, select "Import -> Text..." A new window will appear, with the following data at the top:

3. Set the "# Progeny" to the same number as the "`n=`" in the filename. In our example, it is 23, because the sample filename was "`mm_export_(n=23)`".

4. After changing the "# Progeny," you must click "Apply."

5. After you click the apply button, click "OK," and proceed to a dialog that asks you to "Enter a file to import." Select the correct text file, and click open.

6. Now, a dialog named "Import Text" will appear. You should set the options as they appear in the image below:

7. Once you have made sure the correct options on the left are selected, with 8 checked (Dataset: dataset, progeny names, name. Chr: chr, name. Locus: name, alias, geno) and 11 unchecked, plus "tab" between items, "line feed" between records, and "none" between genotypes, then you should click "OK." MapManager should open up a new file with the imported information.

After you have successfully imported the data into MapManager, you can save it as a native MapManager file, so that you will not need to follow these steps again to open that file.

X. Troubleshooting

After I edited a file with another program, GenomeMixer can no longer read it

There several possibilities here: one is that the file is simply using the wrong linebreak format. Check "Unix-style linebreaks," below, for a solution to that problem.

If you edited the file using a word processor such as Microsoft Word, it probably saved a lot of "hidden" data in the file (formatting, margins, "track changes," etc.), which is invisible within the word processor, but interferes with GenomeMixer's ability to read the file. You can tell if there is "hidden" data in a file by opening it in a plain text editor, like TextEdit or BBEdit for the Mac, or Notepad for Windows. If there are extra characters besides the bare GenomeMixer data, you will have to re-open the file with the word processor it was edited with, save it as "plain text" format, and then fix the linebreak format again (again, under "Unix-style linebreaks").

If there are no strange characters, and the linebreaks are already Unix-format, then it is likely that an important line or character was inadvertently deleted, or there is an extra space somewhere (especially in the purebred names, which cannot have spaces). For example, in the breeding scheme table, all of the entries must be surrounded in double-quotes, and it is easy to leave these out (also, note that "straight quotes" must be used—"curly quotes" will not be read correctly. If you use Word to edit a GenomeMixer file, you may have to temporarily disable the "smart quotes" feature).

Unix-style Linebreaks (problems with Mac / Win GenomeMixer files)

GenomeMixer can only read files that use Unix linebreaks, and although GenomeMixer automatically saves files in this format, if you used another Mac or Windows program to manually edit a file, it is likely that the editor changed the linebreak format. The solution to this problem is to use a utility that can convert linebreaks to Unix format: the freeware Mac OS X drag-and-drop program "LineBreak" by Josh Aas is one such program. BBEdit and BBEdit Lite also have the ability to convert linebreaks. Simply use one of these applications to change the linebreak format, save your changes, and reload the now-properly-formatted file.

Quickload doesn't work

The "Quickload" menu option depends on two files, named "`quickmarkers.mar`" and "`quicksettings.gmx`" (case-sensitive), being located inside the "`Saved_Settings`" folder (case-sensitive, also note the underscore). The `Saved_Settings` folder must be located in the same folder as the GenomeMixer application. If Quickload gives an error, check to make sure that valid markers and settings file are in fact in the `Saved_Settings` folder, that the capitalization of file names is correct, and that you are running the correct copy of GenomeMixer, in case you have more than one copy.

If everything seems ok, but the files still do not load, check to make sure that they have Unix-style linebreaks (see "Unix-style linebreaks" above).

Appendix I:

Specification of the GenomeMixer breeding scheme file:

The GenomeMixer breeding scheme file (".gmx") contains all of the information required to specify the type of organism (number of chromosomes, for example) and the breeding scheme used to produce output progeny. (Marker information is stored in a separate ".mar" marker file.)

Several example breeding scheme files are included in the Saved_Settings folder in the default GenomeMixer distribution.

Note that under normal circumstances, you will not want to edit the breeding scheme file directly, but if desired, Excel or another spreadsheet program can be used to edit it (provided that the file is saved in plain tab-delimited text format afterward).

Here is a sample GenomeMixer beeding scheme (.gmx) file, with annotation:

`<NAME>     This_is_a_configuration_file.`

> The name of the file. This currently has no significance other than to save the name of a particular breeding scheme in a location separate from the filename, in case the file is renamed, for example.

`<VERSION> 1.01`

> If there were multiple versions of GenomeMixer breeding scheme input, the version number would allow GenomeMixer to easily distinguish between them for backwards compatibility.

`<CHR>      Chr01 197 MB 107 cM`

`<CHR>      Chr02 180 MB 104 cM`

`<CHR>      Chr_My_Name 161 MB 85 cM`

> The autosomes, given with an arbitrary name ("Chr01"), a length in megabases, and a length in centiMorgans.

`<FEMALE_SEX_CHR>    Chr20X 147 MB 74 cM`

`<MALE_SEX_CHR>     Chr21Y 30 MB 15 cM`

> The format is the same for the sex chromosomes, except that they are preceded by the tag "< . . . _SEX_CHR>" instead of simply "<CHR>".

`<DEFAULT_LITTER_SIZE>     10`

> When a new cell is made to specify a mating, if the user does not explicitly specify a particular number of progeny from the mating, this default value will be used.

`<DEFAULT_LITTER_STDEV>     0`

> Standard deviation is not currently implemented, but it would allow some randomness to be incorporated into litter size determination.

```
<ABSOLUTE_INTERFERENCE>   25000000
```

> Specifies the recombination method used in this breeding scheme. In this case, around every breakpoint there is an absolute threshold of 25,000,000 bases in which no other breakpoint may occur.

```
<PUREBRED>      AA          A

<PUREBRED>      BB          B

<PUREBRED>      TwoTails   C
```

> This is a list of the purebred ancestral strains from which matings can be specified. In this example, there are four distinct ancestral strains. The first parameter ("TwoTails") is an arbitrary name, but it will appear frequently in the breeding scheme table, so it should be short and contain no spaces or special characters. The second parameter is a letter or number that is used to represent the purebred strain when exporting to other formats ("A" and "B" are good choices for this letter, since some other programs expect A and B parental strains). Note that the letter is **case-sensitive**: "A" and "a" are different. At least two purebred strains will be needed to do any meaningful simulations (if there is only one purebred strain, all progeny will be clones of the purebred strain individuals, no matter how the cross is set up).

```
<GENERATIONS 3>       1     2     3
```

> This is the header of the table of GENERATIONS x CAGES (for plant genetics, each cage may be thought of as "plot." In the future, it should be possible to specify the terminology here). Note that there is a number inside the <GENERATIONS **N**> tag, which must be the same as the number of generation numbers (separated by tabs) to the right. So if 8 generations were desired, the line would read <GENERATIONS 8> 1 2 3 . . . 7 8.

<CAGE 1>	"exp p(AA) x p(BB) r=0 m=1 f=1"	"exp A1 x A1 r=0 m=1 f=1"	"exp B1 x B3 r=0 m=1 f=1"
<CAGE 2>	"exp p(BB) x p(AA) r=0 m=1 f=1"	"#exp A1 x A2 r=1 m=1 f=1"	""
<CAGE 3>	""	"A2 x A2 r=2 m=1 f=1"	""
<CAGE 4>	""	"#A1 x A1 r=5 m=0 f=0"	""

> Each cage / plot may either contain mating information (surrounded by double-quotes, and separated by tabs) or be blank, which is indicated by either a plain tab or by two consecutive double-quotes ("") with nothing in between. If you are going to edit the files manually, make sure you have **plain quotes**, as fancy curly quotes / smart quotes (" ") may cause confusion when importing.

> Cells are numbered down, starting from 1, and "lettered" across, starting from

"A." So the top-left cell is A1, the cell to its right is B1, and the cell below A1 is A2. The number always indicates the cage, and the letter indications the generation. "Z" is the 26th generation, and "AA" is the 27th. "AB" is the 28th, and so on. The cell immediately to the right of "AZ99" would be cell "BA99." This is the same format used by nearly every spreadsheet program.

The format of the entries is as follows, where items in brackets [ ] are optional:

[#] [exp] *mother* x *father* r=NUMBER m=NUMBER f=NUMBER

A hash (#) indicates a temporarily-disabled cell. No progeny will be generated by any matings specified in a disabled cell. In this example, the second-generation cells for cage 2 and cage 4 are disabled (i.e., cells B2 and B4 are disabled).

"exp" means "**exp**ort the progeny generated in this cell to the output file." If a cell does not have "exp" in it, the progeny from that mating *will* still be calculated, but those individuals will *not* be written to a file. This is useful if you only want the final results of a breeding scheme; the F1 generation, for example, might not contain the "exp" keyword, if those individuals were genetically uninteresting.

r=NUMBER: specifies the number of progeny whose sex will be determined **r**andomly. Any non-negative integer is a possible value. Note that "r" is completely independent of the "m" and "f" parameters. "r=10" will result in 10 individuals of random (50/50 chance, for now) sex, whereas "r=10 m=10" will result in 10 random individuals plus 10 males.

m=NUMBER: specifies the number of male progeny to generate in this cell, in addition to any progeny of randomly-chosen sex. If this number is 0, then there is no guarantee that any male progeny will be generated in this cell, which could potentially be a problem if a cell in a later generation requests a male individual for a mating.

f=NUMBER: specifies the number of female progeny to generate in this cell, in addition to any progeny of randomly-chosen sex. If this number is 0, then there is no guarantee that any female progeny will be generated in this cell, which could potentially be a problem if a cell in a later generation requests a female individual for a mating.

The *mother* and *father* parameters can either be specified by requesting an individual from a previous cell ("A1" will get an individual of the appropriate sex from whatever progeny were produced in cell A1), or by requesting a purebred parental strain.

 "p(NAME)" means "get the purebred strain with name NAME." In the example above, "p(AA) x p(BB)" obtains a female from the "A" purebred population, and a male from the "B" population.

`<END_DATA>`

This is simply an indication that there is no more breeding scheme information. Comments may be written below this line. Note that if a file is opened in GenomeMixer, edited, and saved again, **then the comments will not be saved.**