GAMETES User's Guide

Version 2.1 Beta

Ryan J Urbanowicz[1] and Geoffrey Kiralis[2] and Jonathan M Fisher[3] and Jason H Moore[4]

March 29, 2014

[1]ryan.j.urbanowicz@dartmouth.edu - Algorithm and Software Development
[2]geoffrey.kiralis@dartmouth.edu - Algorithm Development
[3]jonathan.fisher@dartmouth.edu - Software Development
[4]jason.h.moore@dartmouth.edu - Principle Investigator

Contents

Chapter 1

Introduction

1.1 What is GAMETES?

Genetic Architecture Model Emulator for Testing and Evaluating Software (GAMETES) is a fast direct algorithm for the generation of complex single nucleotide polymorphism (SNP) models for simulation studies. In particular, GAMETES is designed to generate epistatic models which we refer to as pure and strict. Purely and strictly epistatic models constitute the worst-case in terms of detecting disease associations, since such associations may only be observed if all n-loci are included in the disease model. This makes them an attractive gold standard for simulation studies considering complex multi-locus effects. The user friendly GAMETES software affords users the ability to rapidly and precisely generate epistatic multi-locus models, as well as the option to generate simulated datasets based on these models. In version 2, we have added the ability to generate heterogeneous datasets by applying multiple independent models to different subsets of the simulated data. Additionally we have added a custom model generation feature, so that users may directly specify and examine the properties of any 2 or 3 locus SNP model. Simple main effect models (i.e Mendelian) may also be generated with this feature.

1.2 Algorithm Overview

The core GAMETES algorithm provides a direct approach for the simulation of biallelic n-locus epistatic models which may be used in conjunction with any sample generation strategy. Each n-locus model is generated deterministically, based on a set of random parameters, a randomly selected direction, and specified values of heritability, minor allele frequencies, and population disease prevalence. For valid combinations of these model constraints, GAMETES attempts to generate a population of model architectures. We use the term *architecture* to reference the unique composition of a model (i.e. the penetrance values and arrangement of those values across genotypes). This algorithm was

designed to maximize the randomness of model generation, given a desired set of genetic constraints.

1.3 Further Reading

- For a complete description of the GAMETES algorithm and an example simulation study, see [7].
- For a complete description of the difficulty metric adopted by GAMETES for model selection, see [6].
- For early applications of the GAMETES model/data simulation strategy to a simulation study, see [4, 3].
- For a classification and characterization of 2-locus models, an examination of the relationship between model 'shape' and model detection difficulty, and an exploration of how model population size and population prevalence impact model diversity in GAMETES, see [5]

1.4 Obtaining the Software

GAMETES 2.1 is available as an open-source (GPL) software package. It is a cross-platform program written entirely in Java. It is freely available for download from http://sourceforge.net. You may also contact Dr. Jason Moore for a copy of the software or source code if you experience difficulties downloading it from the web site.

1.5 Minimum System Requirements

- Java Runtime Environment, version 5.0 or higher (http://www.java.com/).
- 1 GHz processor
- 256 MB Ram
- 800x600 screen resolution

1.6 GAMETES Versions

Version 1.0: Made available on 1/16/2012. Original version of algorithm/software.

Version 2.0: Made available on 2/20/2014. Expanded version of GAMETES.

- Added ability to combine genetic models and generate heterogeneous datasets.
- Added window to GUI to allow user to specify custom 2 or 3 locus models, or to edit existing 2 or 3 locus models.

Version 2.1: Made available on 3/29/2014. Fixed dataset generation limitation.

- In GAMETES 2.0, simulated datasets were being stored in memory during generation, limiting the algorithm's ability to generate datasets with greater than approximately 10,000 attributes. This version avoids storing the entire dataset in memory, and thus has a much smaller memory footprint when generating large datasets.

1.7 Starting the Program

After downloading the file, there will be a file called `GAMETES_2.1.jar`. Under most operating systems, simply double-clicking this file will be sufficient to start the program. However, there are reasons a user may wish to start the program from the command line. To do so, open a command shell and navigate to the directory containing `GAMETES_2.1.jar`. Issue the command:

```
java -jar GAMETES_2.1.jar
```

To open the command line help for GAMETES, give the command:

```
java -jar GAMETES_2.1.jar -h
```

Running GAMETES from the command line requires arguments which are discussed in section 2.8. The option to run GAMETES from the command line with no graphical user interface (GUI) facilitates the generation of an extensive model/dataset archive.

Chapter 2

Using GAMETES

2.1 GAMETES GUI Overview

When you open the GAMETES GUI, you will see the window given in Figure 2.1. The GUI is divided into a top half, dedicated to model generation, and a bottom half dedicated to dataset generation. The GAMETES GUI may be used to (1) generate new models, (2) generate datasets from an existing model, or (3) both tasks. As seen in Figure 2.1, when you first start GAMETES, the `Model Construction` box is empty (i.e. there are no model files opened). Whether your task is model or dataset generation, begin by opening a model in the `Model Construction` box. This is accomplished by clicking one of three buttons at the top of the GUI: (1) `Generate Model` (2) `Create Model`, or (3) `Load Model`. The `Generate Model` button will open a secondary window for applying the GAMETES algorithm to generate a model as discussed in section 2.2. Once a new model has been saved it will automatically open in the `Model Construction` box (see Figure 2.2). The `Create Model` button will open a secondary window for specifying or editing a custom 2 or 3 locus model. Similarly, a model saved from this window will automatically open in the `Model Construction` box. The `Load Model` button brings up a file browser, which allows you to navigate to and select a previously saved model file. Note that model files saved in GAMETES have file names with '_Models' added to distinguish them as a properly formatted model file.

The `Edit Model`, and `Delete Model` buttons only become available once a model has been loaded into the `Model Construction` box, and the user has selected the check box for the model they wish to edit or delete. The `Edit Model` button opens the same secondary window as `Create Model`, but instead of a blank model, the selected model is opened for editing. `Edit Model` is only available for models with 2 or 3 loci (loci are also referred to as attributes), and that have a respective model file with a `Quantile Count` of 1. Of note, GAMETES model files can contain multiple models (i.e. penetrance functions) based on the `Quantile Count` specified by the user. `Quantile Count` is discussed further in

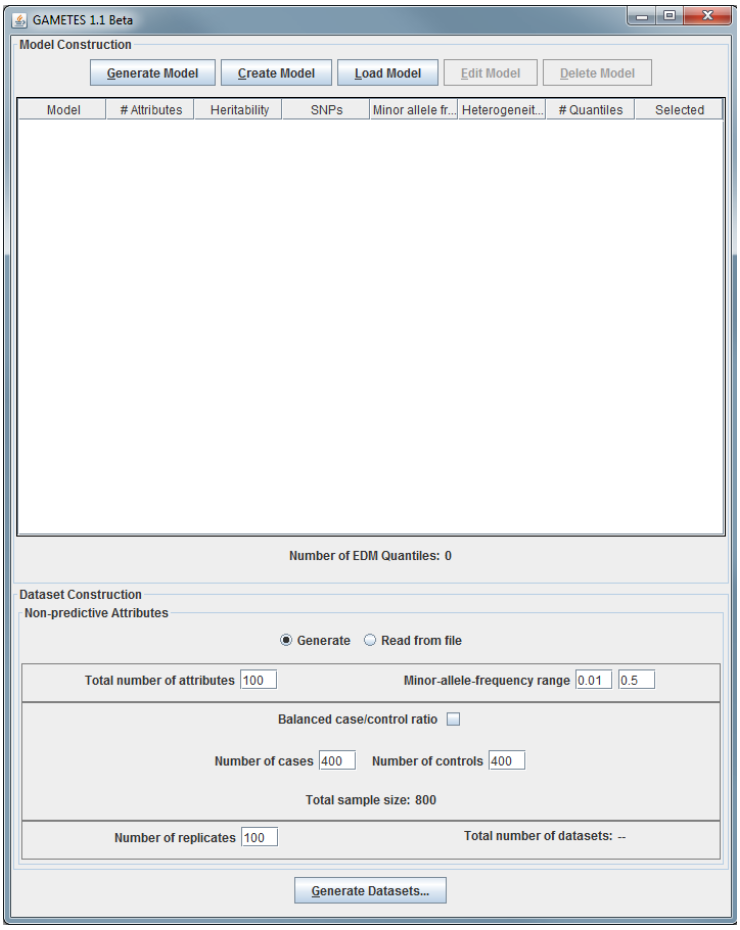

Figure 2.1: GAMETES GUI screenshot.

section 2.2.5. Lastly, the `Delete Model` button clears a selected model file from the `Model Construction` box, but does not delete the respective model .txt file that was also generated.

Towards the top of the `Model Construction` box, notice the column headers which describe the properties of the opened model file(s). These include (1) `Model`, an arbitrary model identifier, (2) `# Attributes`, (3) `Heritability`, (4) `SNPs`, the number of loci included in the model, (5) `Minor allele frequencies`, for each attribute, and (6) `Heterogeneity proportion`, used in generating heterogeneous data, (7) `# Quantiles`, the number of models in the respective model file, and (8) `Selected`, a checkbox for selecting specific models loaded in the window. Of note, `Heterogeneity proportion` and `Selected` are the only columns in the `Model Construction` box with an editable field.

The next three sections details the major components of the GUI as follows:

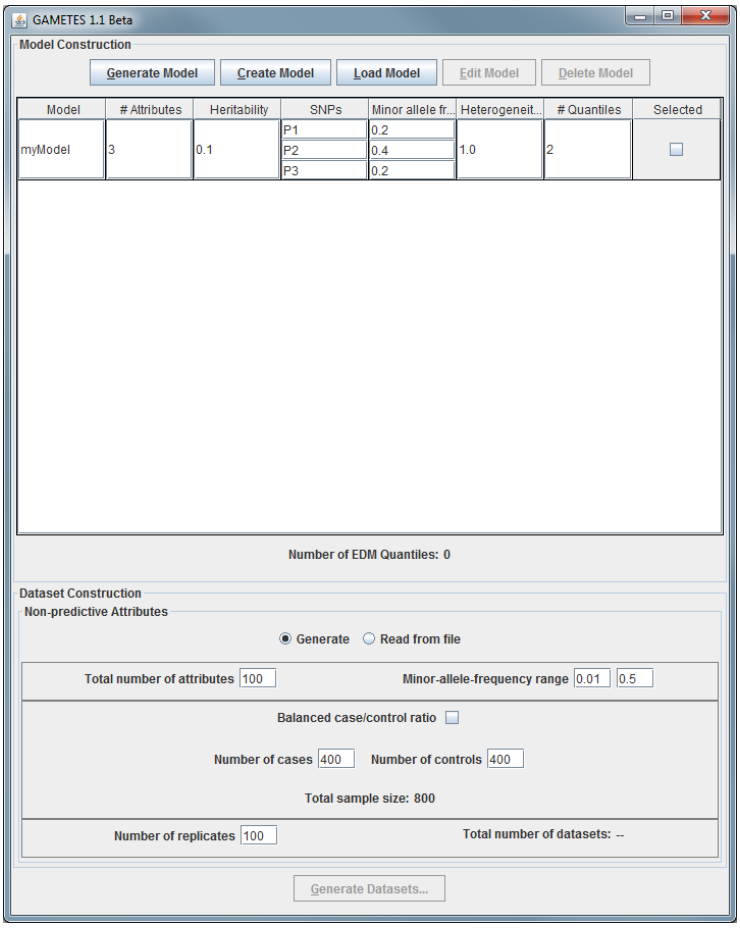

Figure 2.2: Model opened in the model construction box.

(1) the generation of models using the core GAMETES algorithm (section 2.2), (2) the creation and editing of custom models (section 2.3), and (3) dataset generation, including the generation of heterogeneous datasets (section 2.4).

2.2 Model Generation

When the `Generate Model` button is selected, the window like the one shown in Figure 2.3 will appear. This window allows the user to specify constraints of the genetic model(s) they wish to generate. By default, this window will open with simple model constraints specified that will yield valid 2-locus models (heritability of 0.2, and minor allele frequencies of 0.2). The user has the option to edit the following items; (1) number of attributes, (2) heritability, (3) minor allele frequencies, (4) population prevalence, (5) the name of the SNP, (6) the

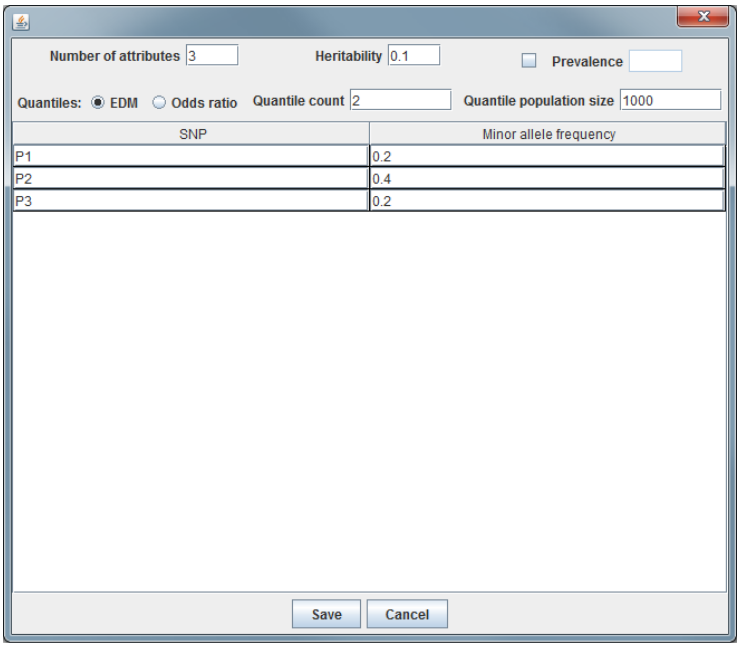

Figure 2.3: Model generation window.

model difficulty metric used to identify quantiles, (7) the quantile count, and (8) quantile population size. These are all discussed below. To modify one of the default model constraints, click on the respective box and type in a new value. Double clicking one of these boxes will highlight the contents of the box and allow the user to directly replace the existing value. Figure 2.3 gives a model generation window that has been edited to specify a model with 3 attributes, a heritability of 0.1, and minor allele frequencies of 0.2, 0.4, and 0.2 respectively.

2.2.1 Number of attributes

`Number of attributes` refers to the number of SNPs which are to be included in the model. After changing the number of attributes, click on any other white space in the window for the change to take effect. You will see additional rows appear in the model window as you increase this value.

2.2.2 Heritability

`Heritability`, or the (broad-sense) heritability of a genetic model is the proportion of observable differences between individuals that is due to genetic differences. Keep in mind that as you select higher values of heritability it becomes less likely that GAMETES will be able to randomly generate such models since GAMETES has the ability to scale a model down to the desired heritability,

but not up. See [7] for more details.

2.2.3 Population Prevalence

Population prevalence (K) is the proportion of individuals in a population that have the disease of interest. Note that GAMETES gives the user the option to leave K unspecified, in which case the K of the models that are generated will vary. To specify K, select the check-box to the left of `Prevalence` and then specify a value of K between 0 and 1. Previous analysis of simulated models suggests that K has a negligible impact on model difficulty. Allowing K to vary also facilitates GAMETES's ability to find models with a precise set of model constraints (particularly important for models with a higher heritability or larger number of attributes) [5]. Additionally, dataset samples are generated according to user defined quantities of cases and controls, and thus the K of a model will not impact the case/control balance in the dataset. For these three reasons, we suggest leaving K unspecified in most situations.

2.2.4 Minor Allele Frequency

The minor allele frequency (MAF) is the frequency at which the less common allele occurs in a given population. Once the user has specified number of attributes, adjust the MAF for every SNP in the model to the desired value between 0 and 0.5.

2.2.5 Other Parameters

While GAMETES gives a default name for each SNP in the model (e.g. P1, which stands for Predictive SNP 1) you may modify each name, again by clicking on the respective cell and typing in a new name. The remaining parameters deal with the population of models to be generated, and how a model or models will be selected and saved. First, `Quantiles` is an option with two radio buttons indicating 'EDM' or 'Odds Ratio'. This simply selects the difficulty metric that will be used to rank models in the population to be generated. See Section 2.5 for more on these metrics. Second, `Quantile Count` refers to the number of random model architectures you wish to save to the model output file (which may subsequently used to generate simulated datasets). Third, `Quantile Population Size` refers to the number of random model architectures you want GAMETES to try and generate for the given model constraints. If you wish to generate a single random model (with the specified model constraints) set `Quantile Count` and `Quantile Population Size` both to 1. Each model that is successfully generated will satisfy the model constraints given, but will possess a unique random architecture. If you want GAMETES to save all model architectures it generates, set `Quantile Count` equal to `Quantile Population Size`. Setting `Quantile Count` to a lower value than `Quantile Population Size` will direct GAMETES to choose a subset of the random models to be saved. This selection process is based on a model difficulty metric discussed

in section 2.5. In short, N random models (where N = `Quantile Population Size`) are ordered by their difficulty and X models are reported/saved (where X = `Quantile Count`). `Quantile Count` may not be larger than `Quantile Population Size`. If `Quantile Count` = 1, the model with the median "difficulty" is reported. If `Quantile Count` = 2, the models with the maximum and minimum "difficulties" are reported. If `Quantile Count` = 3, the maximum, minimum, and median models are reported. Larger values of `Quantile Count` select models at evenly spaced intervals.

2.2.6 Saving Models

Once all parameters have been specified, click `Save`. This will bring up a file browser, which allows you to select the name and destination for your model file. Model files are saved as `FileName_Models.txt`. In addition to the models file, a secondary .txt file will be saved that gives the model difficulty scores for all N models generated for the given combination of model constraints. These files are saved as `FileName_EDM_Scores.txt` or `FileName_OddsRatio_Scores.txt` depending on the difficulty metric you select. When searching for models, a progress bar will appear which indicates the proportion of search attempts GAMETES has made. This does not necessarily reflect the number of models it has successfully found during the course of running.

2.3 Custom Models

When either the `Create Model` or `Edit Model` button is selected, a window like the one shown in Figure 2.4 will appear. This window allows the user to specify or edit a custom model. This feature is only available for generating or editing model files with a `Quantile Count` of 1. If `Create Model` is selected, the initial model will be blank (i.e. all penetrance values will be zero), while if `Edit Model` is selected, the the selected model will be opened for editing (however for readability, only a few decimal places from the original model file will be preserved). This window allows the user to edit three aspects of a given model: (1) the model order, i.e. the number of attributes in the model (limited to 2 or 3 attributes), (2) the penetrance values of the penetrance function (penetrance is the probability of disease if a individual has a particular genotype combination), and (3) the minor allele frequencies of each attribute. To change model order, click on the appropriate radio button. Note that when the 3-locus option is selected the dimensionality of the penetrance function is updated accordingly (see Figure 2.5). Also, note that genotypes for each attribute are either labeled ([AA, Aa, aa], [BB, Bb, bb], or [CC, Cc, cc]). Attributes themselves are labeled by default as P1,P2,or P3 for each 'predictive' SNP in the model. To modify penetrance or minor allele frequency values, click on the respective box and type in a new value. Double clicking one of these boxes will highlight the contents of the box and allow the user to directly replace the existing value. The user may also hit the Tab key to shift from one box to the next.

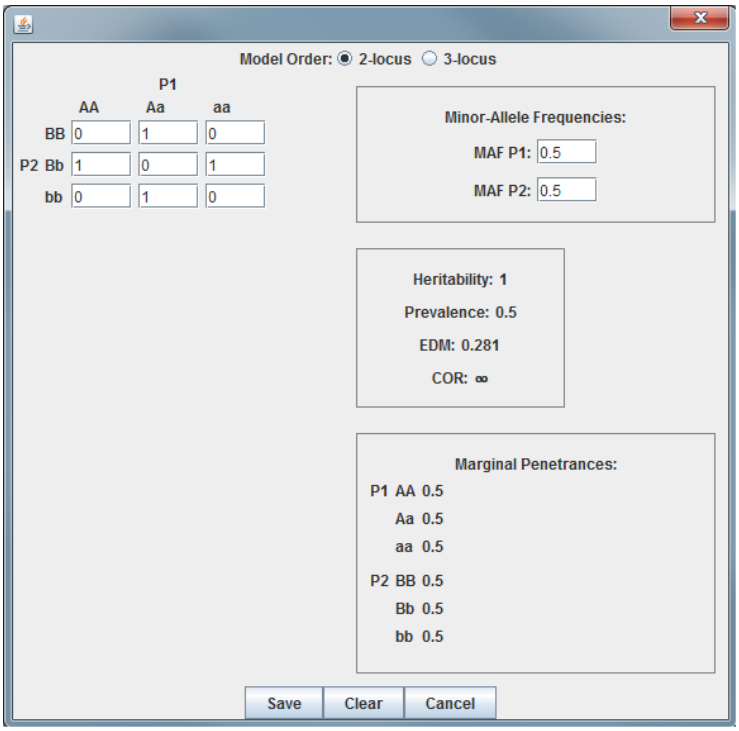

Figure 2.4: Custom model window (2-locus model).

The custom model window also includes feedback on the properties of the currently specified model. This includes heritability, population prevalence, two difficulty metrics (EDM and COR discussed further in section 2.5), and the marginal penetrances of the model. Marginal penetrances give the probability of disease for each geneotype when when considering only a single attribute at a time.

Saving custom models is achieved as described in section 2.2.6, however it is important to keep in mind that when creating or editing custom models, all saved model files will include only that single model, giving all such model files a `Quantile Count` of 1.

Also, please note that based on how the COR difficulty metric is calculated, it is possible for the calculation of the ratio to become mathematically impossible, at which point a value of infinity is assigned within the `Create Model` window. This occurs when either the number of false positives, or the number of false negatives in the model is zero. This is true for all fully penetrant models (models where all penetrance values are either 0 or 1). To clarify further, a fully penetrant genotype (i.e. cell in the model) with a value of 1 yields no false negatives, while a genotype with a penetrance of 0 yields no false positives. If a cell has a penetrance larger the the population prevalence, than that cell con-

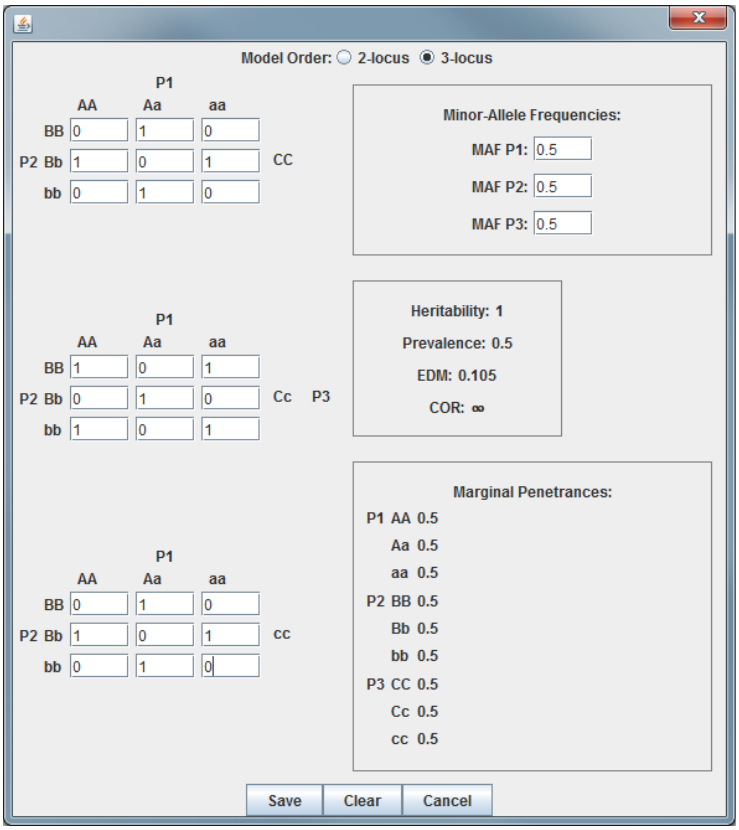

Figure 2.5: Custom model window (3-locus model).

tributes either false negatives or true positives. If a cell has a penetrance smaler than the population prevalence, then the cell contributes either false positives or true negatives. For more information on the COR metric, please see [6].

2.4 Dataset Generation

We begin by describing how to generate datasets from a single model. Later in section 2.4.3 we will explain how to combine multiple models in order to generate heterogeneous datsets.

Once a single model has been opened and selected in the `Model Construction` box, you may generate simulated datasets derived from that model. No matter how many models are loaded in the `Model Construction` box, only the one that is 'selected' will be used to generate datesets. Notice that once a model is 'selected' the `Number of Quantiles` for the loaded model is displayed at the bottom of the `Model Construction` box. Datasets simulated using GAMETES have two types of attributes/SNPs, (1) predictive attributes, and (2)

non-predictive attributes. Predictive attributes are those specified in the genetic model. Non-predictive attributes are all other attributes which have no specified association with affection status (i.e. case or control). The first step is to decide how to include non-predictive attributes. Select the radio button, `Generate` to randomly generate genotypes for all non-predictive attributes. Select the radio button `Read from file` to use an existing real or simulated SNP dataset as the non-predictive attributes.

2.4.1 Randomly Generate Non-Predictive Attributes

If `Generate` is selected the GAMETES window will appear as in Figure 2.2. If `Generate` is selected then you must also specify the `Total number of attributes`. This is the total number of SNPs which will appear in each simulated dataset. This number includes all predictive attributes included in the model. E.g. If you have a 3-locus model, and you specify 100 total attributes, 97 will be non-predictive. Additionally, select the `Minor-allele-frequency range`. The MAF of each non-predictive attribute is randomly selected from within this range with uniform probability. Next if an equal number of cases and controls is desired, check the box for `Balanced case/control ratio`. This will restrict the number of controls, such that it is the same as the number of cases. Adjust `Number of cases` and `Number of controls` by double-clicking the respective box and entering the desired value. If `Balanced case/control ratio` is checked, a change in `Number of cases` will appear in `Number of controls` when you click in the white space of a separate box. Once you have done so, the `Total sample size` will update it self to reflect the correct total. Finally, specify the `Number of replicates`. This is the number of randomly seeded simulated datasets which will be generated for each of the genetic models. `Total number of datasets` indicates the number of datsets which will be generated when the user clicks `Generate Datasets...` at the bottom of the window. If the user has selected 3 model quantiles, and 100 replicates, a total of 300 datasets will be generated (100 for each of three models with different architectures, but the same set of model constraints). Next, click `Generate Datasets...`. This will bring up a file browser, which allows you to select the name and destination for folders which will contain your saved datasets (one folder for each quantile). Finally, click `Save` and a progress bar will appear as datasets are generated and saved. Notice that by default, the datasets are saved such that the first columns include non-predictive attributes, followed by predictive attributes, and lastly the class status. Non-predictive attributes are labeled in the dataset with the prefix 'N' (e.g. N38). Predictive attribute labels begin with a simple model identifier (useful when modeling heterogeneity), with the prefix 'M' for model, and then the prefix 'P' for predictive attribute (e.g. M0P4).

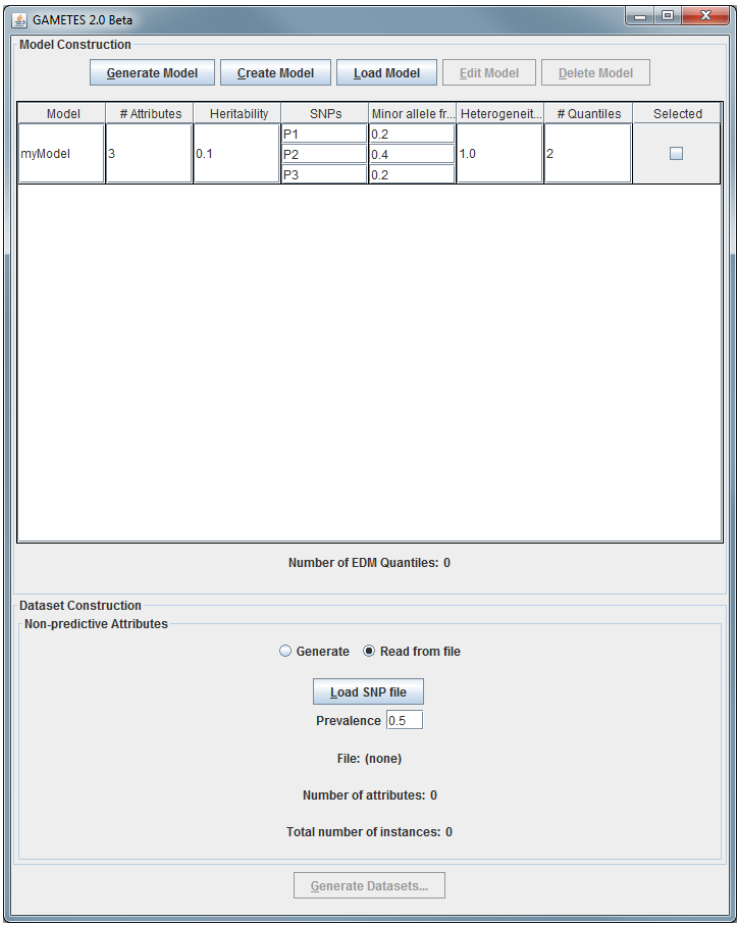

Figure 2.6: Read from file options

2.4.2 Load Non-Predictive Attributes

If you select `Read from file`, the options in the lower half of the GAMETES window change (see Figure 2.6). Selecting this option allows a user to load an existing file (such as a real SNP dataset) to be used as the non-predictive attributes. Click `Load SNP file` to bring up a file browser, which allows you to navigate to and select an existing dataset. The current implementation of GAMETES requires the following format for loaded datasets:

- Files should be tab-delimited text files.
- The first row of the file should contain header labels for all attributes/SNPs.
- Rows should denote subjects, and columns should denote attributes/SNPs.

- The three biallelic SNP genotypes should be encoded as 0,1,or 2. There should be no other attribute values or missing data.
- The file should include attribute data only (no class label or status for any subject).

Once loaded, the `Number of attributes` and `Total number of instances` will be automatically determined, based on the dataset that has been loaded. `Number of attributes` will equal the number of attributes in the loaded dataset only. The number of attributes in the datasets that will be generated is `Number of attributes` plus the number of attributes in the loaded genetic model. `Total number of instances` will be equal to the number of instances in the loaded dataset. A small update in GAMETES 2.0, we have added the ability to adjust the case/control ratio when generating datasets in this manner. To adjust the ratio of cases to controls in the dataset to be generated, adjust the `Prevalence` value between 0 and 1. If the user sets `Prevalence` to 0.5, GAMETES will generate a balanced dataset (i.e. 50% cases and 50% controls) assuming the loaded dataset contained an even number of samples (if not GAMETES will generate a dataset as close to balanced as possible). A lower value will yield more controls, and a higher value will yield more cases. Lastly, click `Generate Datasets...` to generate datasets as described above. Notice that a model from the `Model Construction` box must be selected in order to generated datasets.

2.4.3 Heterogeneous Datasets

The first step to generate heterogeneous data is to load and select two or more models into the `Model Construction` box. A new model may be added to this window as previously described by either generating, creating, or loading a model. Note that while any number of models may be loaded into the `Model Construction` box, only models that are selected using the check boxes will be used to generate datasets. Additionally, all selected model files must have the same number of quantiles (i.e. the same `Quantile Count`). For example, if two models files are selected (each with two quantiles), heterogeneous datasets will be generated for both the high and low difficulty quantiles (where the high difficulty models are paired to make heterogeneous datasets, and the low difficulty models are similarly paired to make separate heterogeneous datasets. Next, within this same window, adjust the values for the `Heterogeneity proportion` as desired. This value indicates the relative proportion of the dataset which will be simulated based on the given model. Any values entered into this field for respective models will be normalized with respect to each other. For example if this value is set the same for each model, then GAMETES will attempt to generate a dataset with an equal number of samples having been generated from each model. As another example, if the user has selected two models, and wants 75% of the data from one model and 25% from the other, they could simply specify a `Heterogeneity proportion` of 75 for the first and 25 for the second. This is illustrated in Figure 2.7. When only a single model is selected in the

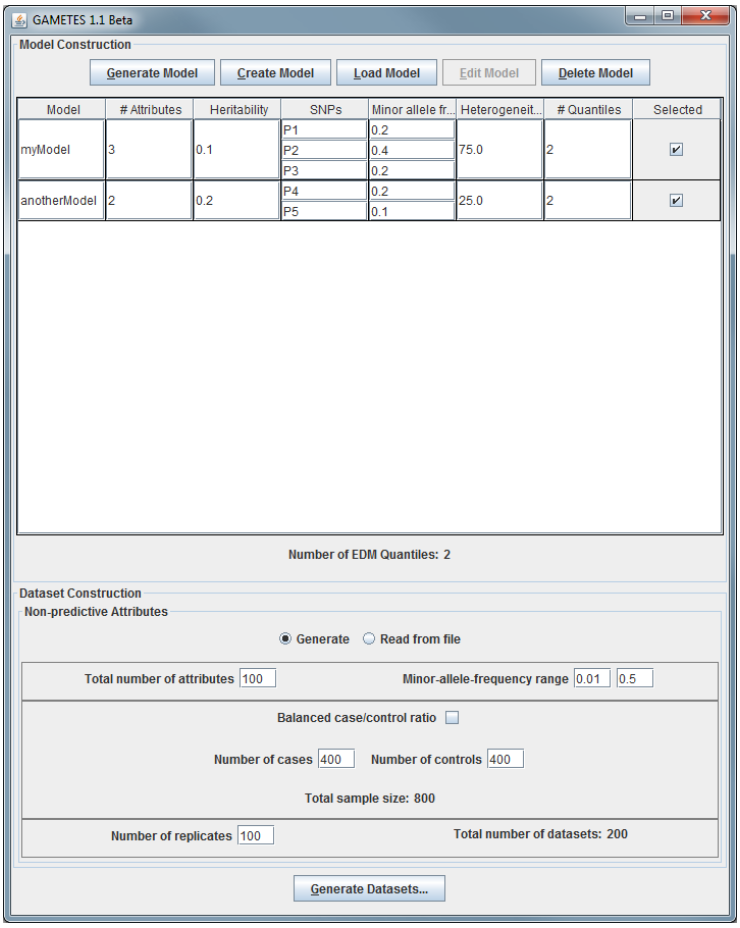

Figure 2.7: Generating heterogeneous datasets

`Model Construction` box, the `Heterogeneity proportion` has no impact on the datasets that are simulated. This value is arbitrarily set as to 1.0 initially for each model in the window. In order to generate and save the simulated datasets, carry on as previously described in the previous subsections. Note that within saved heterogeneous datasets, predictive attribute names are modified with an arbitrary identifier that indicates which model the respective attributes came from. For example instead of attributes from the first model being named P0 and P1, they will be named M0P0, and M0P1 while attributes from a second model would have the prefix 'M1'.

2.5 EDM and COR

A more recent feature to have been added to GAMETES is the ability to select a subset of representative models from a population of models with random architectures. In developing and testing GAMETES, we observed that despite keeping all previously mentioned model constraints constant, an algorithm's detection proportion could vary greatly. Detection proportion refers to the proportion of datasets within which the correct underlying model was identified. One would observe higher proportions for detecting an "easier" model, than a more "difficult" model. While some variation can be explained by the probabilistic translation of models into randomly seeded datasets, the rest can logically be attributed to subtle differences in model architecture. GAMETES has two difficulty metrics implemented as options for model selection. These include a customized odds ratio (COR), informally utilized in [2] and [1], as well as our Ease of Detection Measure (EDM), formally introduced and evaluated in [6]. Both metrics are calculated directly from the genetic model, and were found to demonstrate a strong, significant correlation with a given model's detectability. Both metrics offer a viable method for model selection with no significant difference between them. Also, for both metrics, a larger respective value indicates that the model should be easier to detect than a model with a lower value.

2.6 Model Output Files

Text files in which models are saved provide the following information for each model selected as a quantile by GAMETES: attribute names, MAFs, K, heritability, and both the EDM and COR score for the model. Additionally the penetrance values for every genotype combination in the model are given as a penetrance function. These penetrance functions can become difficult for a user to easily interpret as the number of loci (n) is greater than 3. If n is 2, the genotypes of the first SNP are the rows of the penetrance function, while the genotypes of the second SNP are the columns. If n is 3, these positions shift. The genotypes of first SNP are represented by the three 2-locus penetrance functions, the genotypes of the second SNP are the rows, and the genotypes of the third SNP are the columns. If n is 4, the genotypes of first SNP are now represented by the three, 3-locus penetrance functions. This pattern continues as n continues to increase, where the last of n SNPs is always represented by the columns. See Table 2.1 below for an example of a 2-locus penetrance function, where the function includes 9, or 3^n, penetrance values. Notice here how the rows represent genotypes of SNP 1, while columns represent genotypes of SNP 2.

2.7 Limits of GAMETES

In [7] we discuss specific limits of the GAMETES software. In particular users should be aware that GAMETES's ability to generate genetic models is lim-

Table 2.1: A 2-locus purely epistatic penetrance function.

		SNP 2			Marginal
	Genotype	BB(.25)	Bb (.5)	bb(.25)	Penetrance
SNP 1	AA(.36)	.266	.764	.664	.614
	Aa (.48)	.928	.398	.733	.614
	aa(.16)	.456	.927	.147	.614
	Marginal Penetrance	.614	.614	.614	K = .614

ited by what constraint combinations are mathematically possible, as well as which ones have a reasonable probability of being generated by chance. We illustrate the limits of 2-locus model constraint combinations in Figure 2.8 published in [7]. Presently if you wish to know if GAMETES can generate models with a particular set of model constraints, the best strategy is trial and error. By default, the GAMETES GUI is alotted a `Number of Attempts` of 100,000 to find the desired number of models for `Quantile Population Size` requests up to 1000. If the user requests population sizes larger than 1000, the maximum `Number of Attempts` becomes 100X the requested `Quantile Population Size`. Alternatively, `Number of Attempts` may be precisely specified when running GAMETES from the command-line. If GAMETES fails to find the number of models specified in `Quantile Population Size`, but that number is larger than Quantile Count (the number of models the user wanted specified in the model file), a warning message will appear notifying the user of the number of models it was able to find. In this case a model file will still be generated, but GAMETES will select the number of models specified in Quantile Count, out of the population of models it was able to find. If GAMETES finds fewer models than Quantile Count, the following error message will appear: "Unable to generate desired number of table quantiles". In this situation GAMETES will not output a model file.

Generally speaking if GAMETES was unable to find models with the desired constraints, try using a larger `Number of Attempts`, especially when looking for models with higher heritabilities (e.g. > 0.4) and higher n. If this still does not work, either the combination requested is not mathematically possible, or the odds of randomly generating it are highly improbable.

2.8 Command-Line Operation

GAMETES may also be run from the command line. If your goal is to generate an archive of genetic models and simulated datasets, than it would be advantageous to code a wrapper script calling GAMETES from the command line. To obtain a list of commands for command line operation, type `java -jar GAMETES_2.1.jar -h`. Like the GUI, running GAMETES from the command line affords users the ability to independently generate genetic models and then

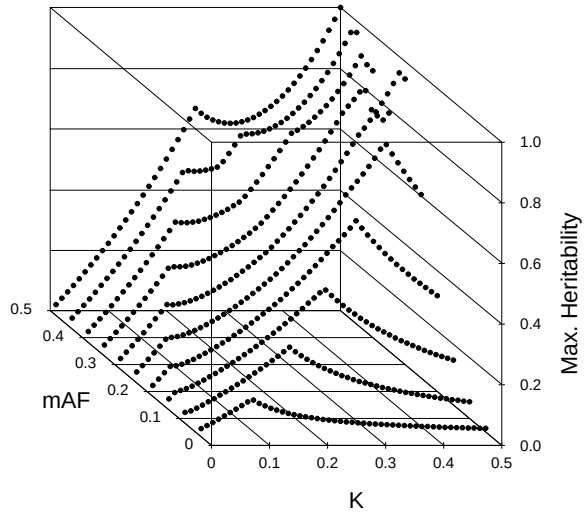

Figure 2.8: Plot of our maximum heritability estimates for pure, strict, 2-locus models.

separately generate simulated datasets. Note that custom models may not be generated from the command line. However, custom model files may be generated with the GUI and then used to generate datasets from the command line.

Below is the help printout for GAMETES specifying the various command line arguments.

```
Usage: java  -jar  gametes.jar  [-h, --help]
[{-M, --model} "[{-h, --heritability} float] [{-p, --prevalence} float]
[{-a, --attributeAlleleFrequency} float] [{-d, --useOddsRatio}] [{-o, --outputFile} filename]
[{-f, --proportion} float]"]

[{-q, --quantileCount} integer] [{-p, --populationCount} integer]
[{-t, --tryCount} integer]
[{-r, --randomSeed} integer]
[{-v, --predictiveInputFile} filename]
[{-z, --noiseInputFile} filename]
[{-i, --modelInputFile} filename]
[{-f, --modelInputFileFraction} float]

[{-D, --dataset} "[{-n, --alleleFrequencyMin} float] [{-x, --alleleFrequencyMax} float]
[{-a, --totalAttributeCount} integer] [{-s, --caseCount} integer]
[{-w, --controlCount} integer] [{-r, --replicateCount} integer]
[{-o, --datasetOutputFile} filename]"]
```

2.8.1 Generate a Single Model File

When running GAMETES from the command line, the user has a number of options. First we will explain the simplest scenario, where GAMETES is applied to generate models given certain model constraints. Consider the following example...

```
java -jar GAMETES_2.1.jar -M " -h 0.2 -p 0.3 -a 0.3 -a 0.2 -o
              myModel" -q 2 -p 1000 -t 100000
```

This command directs GAMETES to generate a 2-locus model with heritability (-h) of 0.2, population prevalence (-p) of 0.3, one SNP with MAF (-a) of 0.3, and the second with MAF (-a) of 0.2. The number of loci in the model is specified by the number of MAF values specified (e.g. if three -a constraints were added, GAMETES would search for 3-locus models). Note, that the model constraints are given following -M, with all model constraints given within quotes. Next, comes commands dealing with how models are generated, selected, and saved. In this example, GAMETES would output 2 model architectures (-q), selected as quantiles from the model population which are saved to an output file (-o) named "mySimulatedModel.txt" in the working directory. These models would be selected from a population (-p) of 1000 models generated by GAMETES with the above model constraints. Since GAMETES does not generate a successful model every attempt, in this example we have limited GAMETES to 100,000 model generation attempts (-t), at which point it will stop trying to reach its goal of 1000 models, and select the 2 specified model architectures, assuming it has found at least 2 in its search. By default, model generation from the command line uses the EDM difficulty metric to rank models. In order to use the COR metric simply add -d to the above command as follows:

```
java -jar GAMETES_2.1.jar -M " -h 0.2 -p 0.3 -a 0.3 -a 0.2 -d -o
            otherModel.txt" -q 2 -p 1000 -t 100000
```

2.8.2 Generate Datasets From a Single Model File

Now, lets assume we want to generate simulated datasets from a single, previously saved model file. Consider the following example which generates datasets including randomly generated non-predictive attributes, from the models we generated in the first command line example above.

```
java -jar GAMETES_2.1.jar -i myModel_Models.txt -D " -n 0.01 -x
          0.5 -a 100 -s 500 -w 500 -r 100 -o myData"
```

This command would direct GAMETES to generate 100 replicate datasets (-r) from each model architecture found within the input model file (-i) named "mySimulatedModel.txt". In this example all dataset constraints are given following -D with all constraints given within quotes. For each of the datasets generated, non-predictive SNPs will have a minimum MAF (-n) of 0.01, and

a maximum MAF (-x) of 0.5. Each will also have a total of 100 attributes (-a), predictive and non-predictive combined, along with 500 cases (-s) and 500 controls (-w). (-o) gives the output file name for the dataset files.

2.8.3 Do Both: Generate a Model File and Associated Datasets

Instead of doing both tasks separately, you can generate a model as well as associated datasets in one command as follows...

```
java -jar GAMETES_2.1.jar -M " -h 0.2 -p 0.3 -a 0.3 -a 0.2 -o
myModel.txt" -q 2 -p 1000 -t 100000 -D " -n 0.01 -x 0.5 -a 100 -s
500 -w 500 -r 100 -o myData"
```

2.8.4 Generate Non-Predictive Attributes From A Loaded Dataset

Again here we are generating data from a single model file. However instead of randomly generating non-predictive attributes, lets assume that you wish to load an existing file (such as a real SNP dataset) to be used as the non-predictive attributes.

```
java -jar GAMETES_2.1.jar -i myModel_Models.txt -z myRealData.txt
-D " -p 0.5 -o myData"
```

This example loads myRealData.txt which must be formatted as described in section 2.4.2 and adds predictive attributes simulated from the model in myModel_Models.txt. If there are multiple quantile models in myModel_Models.txt, a single simulated dataset will be generated for each. In this example -p is no longer the prevalence of a model, rather the prevalence of sick/case samples. This give the user the option to adjust the case/control ratio within the simulated dataset. When loading a data for non-predictive attributes, the number of instances in the data is limited by the number of instances in the loaded dataset file. If the user sets -p to 0.5, GAMETES will generate a balanced dataset (i.e. 50% cases and 50% controls) assuming myRealData.txt contained an even number of samples. Also note that -n, -x, -a, -s, -w, and -r are no longer applicable since the non-predictive attributes are completely determined by the loaded file.

2.8.5 Generate Heterogeneous Datasets from Existing Model Files

Assuming that the user wants to generate heterogeneous datasets combining previously generated model files, this can be performed as in the following example...

```
java -jar GAMETES_2.1.jar -i myModel_Models.txt -f 75 -i
otherModel_Models.txt -f 25 -D "-n .05 -x 0.5 -a 100 -s 200 -w
200 -r 100 -o myHetData"
```

2.8.6 Do Both: Generate Multiple Models and Associated Heterogeneous Datasets

Again, both tasks can be combined as a single command. The following example will generate two model files and use them to generate heterogeneous datasets.

```
java -jar GAMETES_2.1.jar -M " -h 0.1 -p 0.5 -a 0.3 -a 0.1 -f 75
-o myModel.txt" -M " -h 0.03 -p 0.5 -a 0.5 -a 0.5 -a 0.5 -f 25 -o
otherModel.txt" -q 2 -p 1000 -t 100000 -D "-n 0.01 -x 0.5 -a 100
                -s 500 -w 500 -r 100 -o myHetData"
```

Chapter 3

Future GAMETES Expansions

The following sections describe expansions of the GAMETES software intended to be made available in future versions.

3.1 Impure and Nested Epistasis

While pure, strict epistasis makes a logical gold standard for complex multi-locus interaction models, users may want to generate models that don't meet these strict specifications. The GAMETES algorithm may be easily expanded to generate impure, nested epistatic models as well. *Impure* epistasis implies that one or more of the interacting loci have a main effect contributing to disease status. *Nested* refers to epistasis in which at least one proper subset of the loci also interact epistatically.

Bibliography

[1] T. Edwards, K. Lewis, T. Digna, R. Dudek, and M. Ritchie. Exploring the performance of multifactor dimensionality reduction in large scale SNP studies and in the presence of genetic heterogeneity among epistatic disease models. *Hum. Hered*, 67:183–192, 2009.

[2] A.A. Motsinger-Reif, D.M. Reif, T.J. Fanelli, and M.D. Ritchie. A comparison of analytical methods for genetic association studies. *Genetic epidemiology*, 32(8):767–778, 2008.

[3] R. Urbanowicz and J. Moore. The application of pittsburgh-style learning classifier systems to address genetic heterogeneity and epistasis in association studies. *Parallel Problem Solving from Nature–PPSN XI*, pages 404–413, 2011.

[4] R.J. Urbanowicz and J.H. Moore. The application of michigan-style learning classifier systems to address genetic heterogeneity and epistasis in association studies. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 195–202. ACM, 2010.

[5] Ryan J Urbanowicz, Ambrose LS Granizo-Mackenzie, Jeff Kiralis, and Jason H Moore. A classification and characterization of two-locus, pure, strict, epistatic models for simulation and detection. *BioData mining*, Submitted.

[6] Ryan J Urbanowicz, Jeff Kiralis, Jonathan M Fisher, and Jason H Moore. Predicting the difficulty of pure, strict, epistatic models: metrics for simulated model selection. *BioData mining*, 5(1):1–13, 2012.

[7] Ryan J Urbanowicz, Jeff Kiralis, Nicholas A Sinnott-Armstrong, Tamra Heberling, Jonathan M Fisher, and Jason H Moore. Gametes: a fast, direct algorithm for generating pure, strict, epistatic models with random architectures. *BioData mining*, 5(1):1–14, 2012.