GeneEvolve 0.74 User's Manual

Written by Matthew C Keller. Beyond sharing with people in your lab, please do not distribute this beta version without permission from the author. Thank you Mike Neale, Lindon Eaves, and Hermine Maes for your patience, support, and help throughout.

Obtaining

Download: http://matthewckeller.com/html/program_code.html
Requires: *R* statistical program (http://cran.r-project.org)
Optional: Mx statistical program (http://www.vcu.edui/mx)
plotrix *R* library (http://cran.r-project.org/src/contrib/PACKAGES.html)

Intro

GeneEvolve is simply a script written in *R* that accurately simulates genetically informative data, thereby offering an independent check of how our models are performing. You'll therefore need *R* on your system to run it, but beyond that, *you do not have to have proficiency in R to use GeneEvolve*. Depending on the options you feed it, GeneEvolve may call scripts written in Mx; therefore you might download and install Mx. The newest scripts and documentation are available at my website.

GeneEvolve allows you to simulate various types of pedigree data (e.g., twin plus parent data, extended twin family design data, classical twin design data, etc…) given user input parameters (the canonical A, D, C, and E parameters, plus many more). The name "GeneEvolve" is short for "Pedigree Evolver". In order to accurately simulate the environmental and genetic dynamics that occur in populations across time (e.g., the increase in additive genetic variation that occurs in the presence of assortative mating), it is necessary to allow populations to evolve (meet, mate, and have offspring, who meet, mate, and have offspring, etc…) for many generations, until the parameters reach an equilibrium state. Thus the name "GeneEvolve".

Why simulate pedigree data? The main reason is that we get to "play God", such that we know *exactly* what is causing the variation in the phenotype we are modeling. With complicated models, it is difficult if not impossible to find expected equilibrium parameter values analytically. Doing so through simulation, however, is straight-forward. In other words, GeneEvolve allows us to understand:

a) if we accurately estimate the parameters that are—and are not—in the data
b) if we have sufficient information and power to model additional parameters (e.g., Age*A interactions)
c) how unmodeled factors (e.g., recent changes in assortative mating, different models of vertical transmission or assortative mating, genes of major effect, cohort effects, natural selection, etc…) affect our conclusions.

GeneEvolve can also be used as a pedagogical tool for understanding the strengths and weaknesses of different designs, and for understanding population genetics concepts (e.g., the reasons for passive gene-environment correlations, the loss of alleles and heterozygosity due to random genetic drift, the increase in homozygosity with assortative mating, etc.)

Intro (cont.)

GeneEvolve is useful for simulating any of the flavors of genetically informative designs, from the Classical Twin Design to the Extended Twin Family Designs (see Truett et al, 1994, *Behavior Genetics*, *24*, 35-49) and everything in between. The ability to simulate adoption data will be available soon. GeneEvolve also allows you to simulate repeated-measures data, but the ability to model other types of multivariate data has not yet been implemented.

Finally, GeneEvolve is written in *R* and is open-source. This not only allows people to look into the black box and figure out exactly how it works, it also allows people to modify the script to their own liking. In the near future, I will write a short *GeneEvolve Programmers Manual* that should help people understand how the script is written and give pointers for modifying it to your liking. I would ask that you send any modifications to the script to me (matthew.c.keller@gmail.com), both as a means of quality control, and so that alternative versions of GeneEvolve can be found at one central location. I do not see the need right now to make GeneEvolve into an *R* library, but this may change in the future.

How to use it

1) GeneEvolve.zip comes with four files: GeneEvolve47.R, Fullcor.mx, UserManual47.pdf, and PE47.ExampleGraphics.pdf. From my website, you can downlod the zip file to obtain all the files, or download each individually. Only the first file is required to run the script – the two Mx scripts might be called, depending on the options you specify (see below). You can place GeneEvolve47.R script anywhere you want, but the two Mx files must be placed in a folder you designate in the script (the 'working directory'). GeneEvolve will place all simulated datasets and graphics in the working directory. Obviously, if you want to run multiple scripts, you'll need multiple working directories and multiple copies of the Mx files in each.

2) You must specify your input parameters (e.g., A, C, D, E, etc…) in the script before you run GeneEvolve. To do this, simply open GeneEvolve.R in your favorite text editor (or in the *R* gui) and alter the 31 parameters (at the top of the script) to your liking.

3) Once you have modified the input parameters, there are many ways to run GeneEvolve.

a) Perhaps the simplest is to open *R* and, at the command prompt (>) , type:

```
> source("PATH/GeneEvolve47.R")
```

where "PATH" is the path to the folder where you placed GeneEvolve47.R (e.g., on my windows machine, its at C:/Matts Folder/RESEARCH/GeneEvolve). Be careful to use forward slashes (/) , not the backwards ones(\).

b) Or just open *R*, go to File -> Open Script, and locate GeneEvolve47.R and open it. Then highlight the entire script and run it (by hitting the middle button, "run line or selection").

c) Alternatively, just type in the system command that you would need to run any *R* script. For example, if you are working on the UNIX server at VIPBG, type:

```
> qR PATH/GeneEvolve47.R
```

Options

At the top of the GeneEvolve47.R script, the user needs to specify options for any particular run of GeneEvolve. You should not have to change anything else in the script to successfully simulate data. Below, I explain each of the 31 parameters that can be altered in GeneEvolve:

#BASICS
working.directory: (ex: “"/home/mkeller/GeneEvolve/P46/one"). This is the folder where everything created by GeneEvolve will be stored.

save.gen.data: (“yes” “no”) You can have GeneEvolve save the dataset – everyone in the generation and their complete data (genes, phenotypic values, etc) - for each generation. Unless there is some reason you want to have this data, I recommend against it because the datasets can take up a lot of space.

save.objects: (“max” “min” “none”) Do you want to save the R objects in a .RData file? Answer "min" or "none" unless you are debugging or working through script ("yes" makes the program a memory hog, which can crash the script unless you have enough RAM).

name.object: (ex: “mydata.Rdata”). Name of the .RData file (a file created by R which stores all the objects created during the simulation), followed by ".RData".

run.cor: ("yes" “no”) Do you want MX to compute ML correlation matrices for all relatives & present results graphically? If you answer “yes”, you must have "Fullcor.mx" file in working directory & you must give the location (below) of the MX program.

mx.location (ex: "/usr/local/bin/mxt166b"). This is the full path of location of mx (unix server example).

#DEMOGRAPHIC DETAILS:
number.runs: (ex: 100). Number of generations to evolve before population makes twin/spouse parents, twins, sibs, spouses, & twin children.

start.pop.size: (ex: 20000). Breeding population size at start of simulation; should be an absolute minimum of 99; population sizes greater than 1000 insure stability.

pop.growth: (ex: rep(1,number.runs)) Vector of length number.runs that tells how big each generation is relative to the one before it; for no growth: rep(1,number.runs); for constant 5% growth: rep(1.05,number.runs).

#MODEL DETAILS:
gene.model: ("rare" "common") Unless you are running GeneEvolve to study the dynamics of allele frequencies, you should choose “rare” here. Under the “rare” model, everyone in the population begins with a unique allele, whereas in the “common” model, everyone has one of just a few alleles that exist at each locus (specified under allele.number). If only a few alleles (e.g., 2) exist per locus, it is not possible to specify the genetic variance components up front (e.g., Va, Vd), because these values depend on the particular allelic effects taken individually

& in combination, as well as on the allele frequencies. To get around this problem, the "rare" option models an infinite allele system: all IBD relationships remain the same, but variance components can be correctly specified up front.

number.genes: (ex: 10). Number of genes affecting phenotype. Putting a large number of genes in here will slow down the program and provide no benefit.

number.alleles: (ex: 2) In the "common" model ONLY, how many alleles per locus?

vt.model: (ex: "I") "I"= vertical transmission from parental phenotypes to offspring. "II" = vertical transmission from parental "F" to offspring "F" vertical transmission is an environmental model of parent-offspring resemblance. If vt.model is "I", parents "pass on" their phenotype to their children's phenotype (e.g., perhaps children mimic their parents). If vt.model is "II", parents' C is passed to the children's C (e.g., perhaps wealth affects the phenotype and parents pass wealth to offspring).

am.model: (ex: "I") "I" = "primary phenotypic assortment" - correlation b/w mates due to their choosing similar phenotypes. "II" = "social homogamy" - mating similarity comes through correlated environmental factors. "III" = "familial social homogamy" - mating similarity comes only through correlated "F" (familial) factor. "IV" = "genetic homogamy" – mating similarity comes only through correlated "A" (additive genetic) factor.

#VARIANCES:
Variance terms of the first generation are specified by the user. These variance components will change thereafter due to stochastic processes or to evolutionary dynamics (e.g., the increase in additive genetic variation that stems from primary phenotypic assortative mating). Its nice, but unnecessary, for the variance components to sum to 1.

Genetic Factors:
A: (ex: .20) Variance accounted for by *additive genetic effects*; also, variance of the intercepts (or levels) of A (see path diagrams, below).

AA: (ex: .20) Variance accounted for by *additive-by-additive epistasis*. number.genes must be greater than 1 if AA is greater than 0.

D: (ex: .20) Variance accounted for by *dominance genetic effects*.

Environmental Factors:
U: (ex: .20) Variance accounted for by *unique experiences*. Note: in non-twins, E (eg, the E from ACE models)=U+MZ+T. In other words, U, MZ, and T split the unique environment (E) into three components. U is the aspect unique to any individual. MZ is the part of E that is unique to everyone except MZ twins, with whom this aspect of the environment is shared (e.g., perhaps MZ twins share peer groups more often, and peer groups affect the phenotype). Finally, TW is the part of E that is unique to everyone except twins (both DZ and MZ), with whom this aspect of the environment is shared. No, I'm not making a statement about whether such "special twin environments" are problems in twin designs; they are included for completeness.

F: (ex: .20) Variance accounted for by the *familial environment*. This is passed from parent to offspring, as specified by "vt.model".

S: (ex: .20) Variance accounted for by the *sibling environment*. Note: in only children, this is part of E.

MZ: (ex: .20) Variance accounted for by *special MZ twin environment* (this goes into E for non-mz individuals).

TW: (ex: .20) Variance accounted for by *special twin environment* (this goes into E for non-twin individuals).

Covariates & Moderators

AGE: (ex: .20) Variance accounted for by *age*. This term can be negative; the square root of its absolute values is the change in phenotype over 1 SD of age.

AGE.by.A: (ex: .20) Variance accounted for by *interaction between A & Age* (assuming an age range of 15-70). Please note: different age distributions will give different variances accounted for by this term. Age.by.A allows for a non-scalar Age-by-Additive genetic interaction (i.e., different genes 'turning on' at different ages). Each allele conveys a level effect (A_L) which does not change across age (the variance of which is A), and a slope effect (A_S), which is the effect that depends on age and which, in combination with age, creates the Age.by.A variance. Also, there can be a correlation between A_L and A_S (R.Alevel.Aslope) – for example, if people with the highest levels also tend to increase the most across age, this correlation would be positive. The diagrams at the end of this manual illustrate the full GeneEvolve design as well as this non-scalar interaction.

#COVARIANCES:

latent.AM: (ex: rep(.0,number.runs)). This is a vector of length = number.runs. It specifies the correlation between spouses' latent mating phenotype each generation, as determined by "am.model". If "am.model = "I" (primary phenotypic assortment), the actual phenotypic correlation between spouses is equal to this number; otherwise, it is less than this number because the mating phenotype is but a part of the whole phenotype (see path diagrams below).

sibling.age.cor: (ex: .5). This allows control over how closely spaced siblings are in age. Higher correlation imply less spacing between siblings.

R.Alevel.Aslope: (ex: .6). This is the correlation between the intercepts and slopes of A - e.g., do people with the highest A values also tend to have the highest positive changes in A across age? If this is 1, AGE.by.A becomes the scalar interaction coefficient (the "Purcell" model).

#DATASET PARAMETERS

percent.mz: (ex: .50). The percent of twins who are MZ in the last generation.

min.age: (ex: 10). Minimum age for inclusion of family members in final dataset.

max.age: (ex: 90). Maximum age for inclusion in final dataset. Regardless of your entry, people die w/ increasing likelihood as they age, and so older people are increasingly unlikely to be in the datasets created by GeneEvolve.

range.twin.age: (ex: c(14,14,18,70)). Range of twin ages in final dataset at each timepoint. This is a vector twice as long as # of measurements. In the current example, twins would be measured between 14 and 14 for the first measurement (i.e., all twins would be exactly 14), and between 18 and 70 for the second measurement.

What you get

Once GeneEvolve has finished running, you will always obtain the following files in your working directory (additional files may also be created depending on options specified).

Simulated Datasets

a) PedigreeData.Full: the full dataset, containing all family members and their IDs, maternal and paternal allele names, sex, age, true variance components (e.g., A, D, C, etc…), and phenotypic values.
b) Five extended pedigree files (MZM, MZF, DZM, DZF, DZO) that are in a format readable by the Cascade.mx script written by Medland, Keller, & Hatemi. The order of the variables are: Famid, tw1, tw2, fa, mo, 2 brothers, 2 sisters, spouse twin 1, 2 sons twin 1, 2 daughters twin 1, 2 sons twin 2, and 2 daughters twin 2. Age follows this, in the same order. Missing data is coded as -999. GeneEvolve creates one set of files for each measurement you specify: repeated measures will create multiple files.
c) TwinsOnly: dataset that only contains twins and their phenotypic values and ages for each measurement period.

Other information

a) Track.Changes: A rectangular file containing variance parameters from each generation of your simulation.
b) <name you supply>.Rdata: a file containing all the objects that you have saved from the script (see options below)
c) GeneEvolveResults.pdf: a graphical representation of your data and the evolution of your pedigree. An example PDF is included with GeneEvolve (PE47.ExampleGraphics.pdf.). Most of the pdf file should be self-explanatory, with a few exceptions, described below.

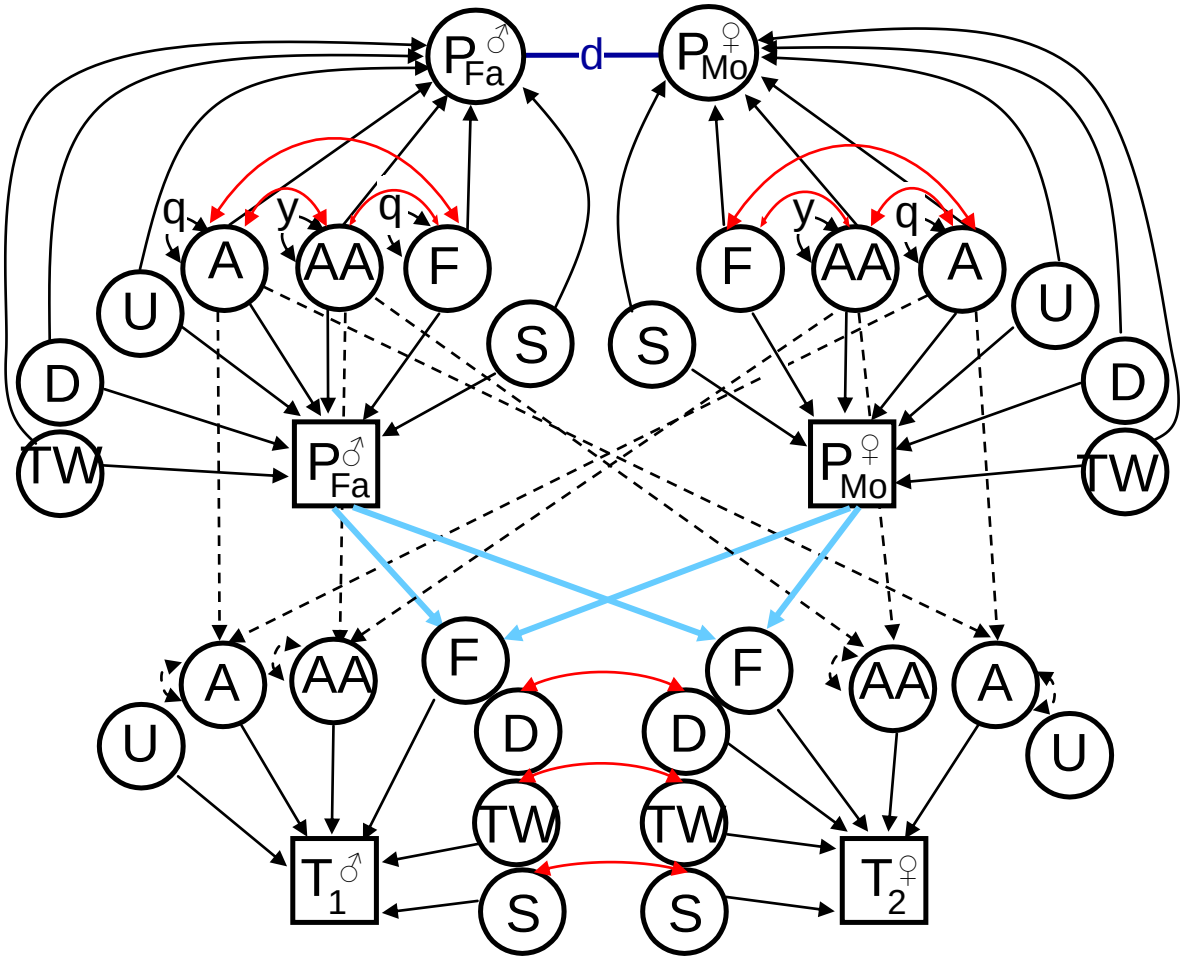
   First, the graphics show variance parameters for “First Generation”, “Final Generation”, and “Dataset”, as well as for “CTD Estimates” and, depending on if you ran the Stealth model, for “Stealth Estimates”. The variance and covariance parameters are those for the entire adult population – roughly ages 15 to 80—with one exception: assorative mating. In “First Generation” through “Final Generation”, assorative mating refers to the correlation between mates *at the time of mating*. In “Dataset”, assortative mating refers to the correlation between mates *at the time of data collection*. Therefore, if there are age effects, people may become more or less related to one another after they mate, and thus the “Dataset” assortative mating coefficient may differ from the “Final Generation” coefficient.

Along these lines, the "Final Generation" variance parameters are for the final generation, and include the parents of twin spouses and parents of twins. The "Dataset" refers to twins and the twins' parents, siblings, spouses, and offspring. If there are age effects, we might expect the "Dataset" variance parameters to be slightly different than the "Final Generation" variance parameters because the age ranges are likely to be somewhat different between the two. Thus, the correct comparison is between the estimated variance parameters (from the CTD or Stealth models) and the "Dataset" variance parameters.

Last, "COV's" refer to two times the sum of all the parameter covariances in the model – which reflects how these covariance terms affect the overall phenotypic variance. Thus, COV is a mishmash of many different covariance effects.

GeneEvolve Path Models

Diagram of the core part of the GeneEvolve model. Shown are parents and offspring only. P' represents the phenotype on which parents are assorting on, and thus the path coefficients going to them are either equal to those to the phenotype or are equalt to zero. The main effects of age and details of the Age-by-A interaction are shown in the subsequent graph. MZ latent factor not shown.



GeneEvolve Path Models (cont.)

Diagram of the Age-by-Additive Genetic Non-Scalar Interaction.