



Computer Note

The G-matrix Simulator Family: Software for Research and Teaching

Adam G. Jones, Reinhard Bürger, and Stevan J. Arnold

From the Department of Biological Sciences, University of Idaho, Moscow, ID 83844 (Jones); the Faculty of Mathematics, University of Vienna, Oskar-Morgenstern-Platz 1, A-1090 Vienna, Austria (Bürger); and the Department of Integrative Biology, Oregon State University, Corvallis, OR 97331 (Arnold).

Address correspondence to A. G. Jones at the address above, or e-mail: adamjones@uidaho.edu.

Received August 10, 2018; First decision September 21, 2018; Accepted October 4, 2018.

Corresponding Editor: Scott Hodges

Abstract

Genetic variation plays a fundamental role in all models of evolution. For phenotypes composed of multiple quantitative traits, genetic variation is best quantified as additive genetic variances and covariances, as these values determine the rate and trajectory of evolution. Additive genetic variances and covariances are often summarized conveniently in the **G**-matrix, which has additive genetic variances for each trait on the diagonal and additive genetic covariances as its off-diagonal elements. The evolution of the **G**-matrix is an interesting topic in its own right, because the processes that affect trait means also affect the distribution of standing genetic variation, which, in turn, feeds back to affect the rate of change of trait means. Theoretical studies of the **G**-matrix have profitably employed simulation-based models because the topic is often too complex to yield meaningful analytical results. Here, we present a series of **G**-matrix simulation software packages, which have emerged from about 15 years of research on this topic. These simulation models are useful for research and for building intuition regarding the evolution of the **G**-matrix under a wide variety of circumstances. A tutorial and source code also provide a foundation upon which future models can be built. These tools will be useful to students as well as researchers.

Subject area: Quantitative genetics and Mendelian inheritance, Bioinformatics and computational genetics

Keywords: breeder's equation, C++, complex traits, quantitative genetics, simulations, theory

The additive genetic variance is a key facet of evolutionary quantitative genetics. The importance of additive genetic variance is apparent from the breeder's equation: $R = h^2S = (V_A/V_P)S$, where the additive genetic variance, V_A , plays a direct role in determining the response to directional selection (h^2 is the heritability, which is defined as the additive genetic variance, V_A , divided by the total phenotypic variance, V_P ; R is the response to selection; and S is the selection differential; Falconer and Mackay 1996). For a phenotype encompassing multiple traits, we use the multivariate breeder's equation ($\Delta\bar{z} = \mathbf{G}\boldsymbol{\beta}$), and the additive genetic variance is replaced by **G**, a matrix containing additive genetic variances and additive genetic covariances

(Lande 1979; Lande and Arnold 1983). In the multivariate breeder's equation, $\Delta\bar{z}$ is a vector of changes in trait means, and $\boldsymbol{\beta}$ is a vector of selection gradients (Lande and Arnold 1983). Thus, given knowledge of the nature of selection acting on the traits (i.e., $\boldsymbol{\beta}$), the response to selection can be predicted only if **G**, often somewhat redundantly called "the G-matrix," is also known.

While the contemporary G-matrix is knowable in principle, backward- or forward-looking evolutionary analyses require that **G** be known over the timeframe of interest (Lande 1979; Arnold et al. 2001). This realization has spawned a subdiscipline of comparative quantitative genetics (Steppan et al. 2002), which is concerned with how the

G-matrix changes over evolutionary time. As this field has matured, it has embraced the evolution of the G-matrix as an interesting topic in its own right rather than as an irritating source of noise interfering with the quest to resolve patterns of selection (Draghi and Whitlock 2012; Aguirre et al. 2014; Wood and Brodie 2015). Numerous empirical and theoretical studies have taken aim at the evolution of G (reviewed in Stepan et al. 2002; Arnold et al. 2008). On the empirical front, most studies involve estimating G-matrices and comparing them among populations within a species (Johansson et al. 2012; Puentes et al. 2016) or species within a genus (Roff et al. 1999; Bégin and Roff 2001), although some comparisons involve even larger taxonomic distances (Kohn and Atchley 1988). With respect to theoretical approaches, analytical theory quickly reached an impasse on G-matrix stability (Lande 1980; Turelli 1985, 1988; Turelli and Barton 1990), so simulation-based studies were embraced starting near the turn of the millennium (Jones et al. 2003, 2004, 2012; Bürger and Krall 2004; Krall 2005; Guillaume and Whitlock 2007; Revell 2007).

Here, we introduce a set of software packages and a tutorial designed to overcome 2 major barriers to entry into the realm of G-matrix evolution. The first hurdle for students of the G-matrix is to gain an intuitive understanding of how the G-matrix changes over evolutionary time and how it responds to various evolutionary pressures. To overcome this first difficulty, we present 2 versions of a Windows-based G-matrix simulator with a graphical interface. The graphical interface permits easy manipulation of parameter values, and the output includes an animated, graphical representation of the evolving G-matrix. The second difficulty for scientists interested in G-matrix evolution is that the development of G-matrix models can be difficult, requiring advanced programming skills and myriad decisions regarding how nature should best be translated into an *in silico* representation. To overcome this problem, we provide a command-line version of the G-matrix simulator, along with the source code and a step-by-step tutorial documenting the development of the program (Jones 2018). This program and the approach it exemplifies provide a starting place for simulation-based models of the evolution of complex traits. These base models provide a solid foundation that can be modified for future, more realistic studies, as our empirical understanding of complex traits expands. Overall, these tools will be useful for teaching quantitative genetics and for the development of future quantitative genetic models by interested researchers.

Methods

Here, we briefly introduce the basic model before describing each program. These software packages are capable of producing the full suite of results described in our articles that focused on G-matrix evolution *per se* (e.g., Jones et al. 2003, 2004, 2012).

The model is an individual-based Monte Carlo simulation built on models developed by Bürger et al. (1989) and Bürger and Lande (1994) to study the evolution of the additive genetic variance for a single trait. Our model extends the univariate model by adding a second trait. This 2-trait model adds additional complexity but retains intuitive appeal because the selection surface and G-matrix can easily be visualized in the 2-trait case. The traits are assumed to be determined by a number of pleiotropic, additive loci, so an individual's breeding value is calculated by simply summing across loci. The life cycle includes monogamous mating, mutation and recombination, natural selection, and population regulation. In this version of the model, loci are unlinked, so they recombine freely. Natural selection occurs according to a bivariate Gaussian individual selection surface (Lande 1979), which allows stabilizing selection, correlational

selection, and directional selection, depending on the shape of the selection surface and the position of the bivariate optimum. When a mutation occurs, mutational effects are drawn from a bivariate Gaussian distribution with specified variances and covariance. These new effects are added to existing effects, following the continuum of alleles model (Crow and Kimura 1964). After natural selection, population regulation is imposed on the survivors. Individuals are culled at random until the number of surviving adults equals the carrying capacity. Further details of the model are provided by Jones et al. (2003, 2004, 2012).

G-matrix Simulator 2014 Version

This version is available at <https://github.com/JonesLabIdaho/GmatrixSimulator2014>. The 2014 version includes the original model that was used to produce the data for Jones et al. (2003, 2004, 2012). Thus, it permits a stationary optimum or a moving optimum, as well as stochastic or episodic movement of the optimum. The output includes every variable calculated in these studies, and the documentation describes the meaning of each important variable. This program is provided as a compiled Windows executable, which should run on any version of Windows since Windows XP (including Windows 7 and 10). To run the program, the user simply needs to unzip the archive (if the download is compressed) and double click on the executable icon. The program has a graphical user interface, which allows the parameter values to be set easily (Figure 1).

G-matrix Simulator Home Version (2016)

The 2016 version of the G-matrix Simulator, also known as the "home version" (available at <https://github.com/JonesLabIdaho/GmatrixHomeVersion>), has been adapted for use in the Evolutionary Quantitative Genetics short course offered annually by Stevan J. Arnold and Joseph Felsenstein. This version of the model is streamlined to make it more effective for teaching purposes. For instance, the output is reduced to include a smaller set of key variables, such as the genetic variances and covariances, population means, angle of the leading eigenvector of the G-matrix (i.e., g_{max} ; Schluter 1996), and several other useful summary statistics. The user interface is also updated to provide pop-up tips for each parameter on a mouse-over. Like the 2014 version, this program runs on all modern versions of Windows.

G-matrix Simulator Command Line

The G-matrix Simulator Command Line program is a completely rewritten version of the G-matrix Simulator designed to compile and run on any operating system. The source code is available at <https://github.com/JonesLabIdaho/GmatrixCommandLine>, and any modern C++ compiler can be used to compile it. The tutorial *C++ for Biologists: Evolutionary Models* (Jones 2018) describes, in a step-by-step fashion, the development of the object-oriented source code for this project. The tutorial is available for free on the Jones Lab website (<https://pipefishguysite.wordpress.com/jones-lab-publications/>). The GitHub repository also includes example BASH shell scripts for running the compiled program on Linux and a sample R script for visualizing the G-matrix at the end of a simulation run.

Results and Discussion

Here, we present a family of G-matrix simulation software applications. Each of these programs fills a slightly different niche. In each case, user-friendliness is a key component. None of the software packages requires dependencies or multistep installations. Each

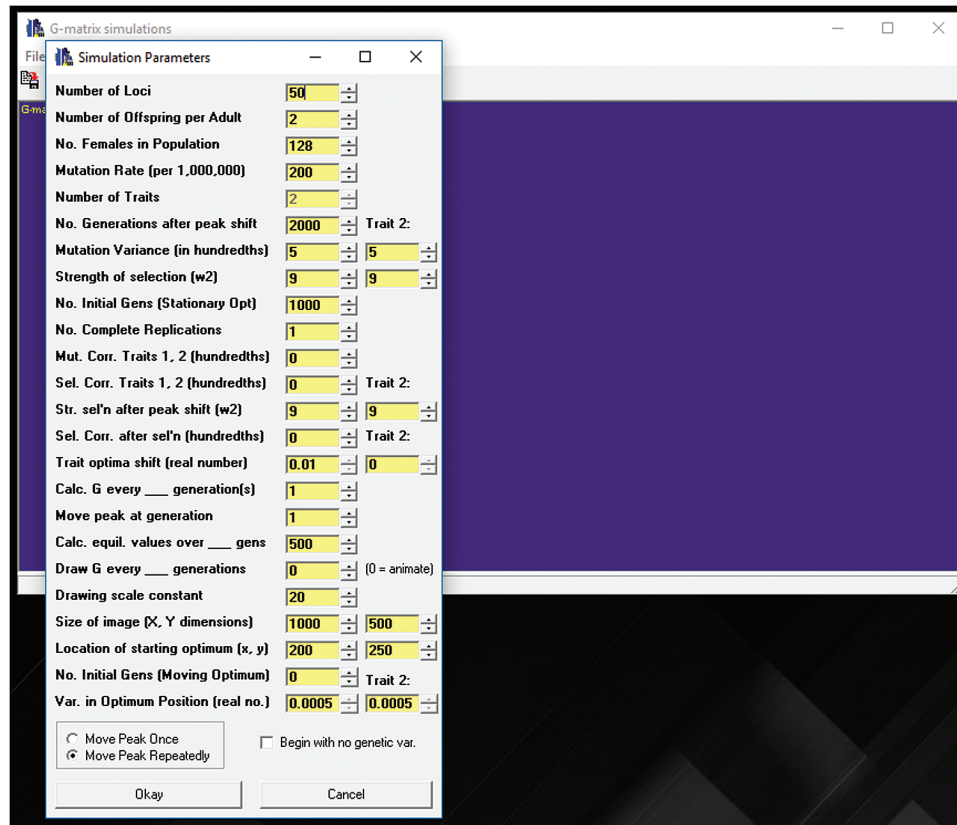


Figure 1. The parameter-control window in the graphical user interface for the **G**-matrix Simulator 2014 and 2016 versions. Parameter names are mostly self-explanatory, and they are also described in detail in the user manual.

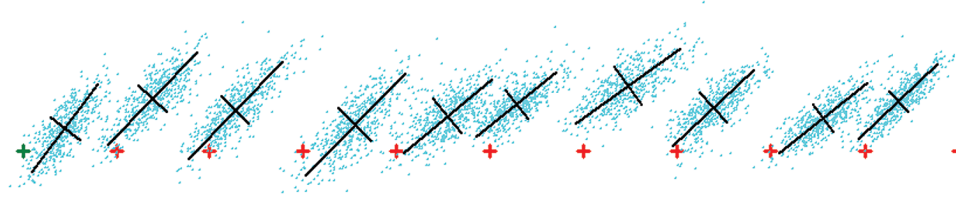


Figure 2. Graphical output of the **G**-matrix Simulator Home version. Here, the **G**-matrix is drawn every 200 generations during a 2000 generation simulation run. The leftmost cross is the position of the starting optimum (and the **G**-matrix corresponding to this generation is not shown). All other crosses represent the position of the optimum as the simulation runs, drawn at 200-generation intervals. The optimum is moving to the right, indicating that the trait 1 optimum (on the x-axis) is changing while the trait 2 optimum is stationary (y-axis). The **G**-matrix is lagging up and to the left of the optimum. The x-axis lag occurs because the population mean cannot keep up with the moving optimum. The y-axis lag is caused by a correlated response to selection on trait 2, as a consequence of a strongly positive genetic correlation (evidenced by the 45-degree orientation of **G**), an example of the “flying-kite effect” (Jones et al. 2004). For each depiction of **G**, the 2 lines represent the eigenvectors, scaled by the corresponding eigenvalues, and the small dots are the breeding values of the individuals in the population. In an actual simulation run, the **G**-matrices are drawn in real time, and the program also includes an option to animate the **G**-matrix by drawing and erasing it every generation. This latter feature can be very revealing regarding the dynamics of the genetic variance-covariance structure over time.

package can also be used to visualize the **G**-matrix over evolutionary time, thus allowing users to build an intuitive understanding of how the **G**-matrix behaves under various selection scenarios. In the cases of **G**-matrix Simulator 2014 and the **G**-matrix Simulator Home Version, the graphical output is produced automatically as part of the user interface during each simulation run (Figure 2). In the case of **G**-matrix Simulator Command Line, the output can be loaded into R, and a simple script can be used to draw the **G**-matrix over user-specified time intervals (Figure 3).

These programs have been developed for 2 specific purposes. The first purpose is related to instruction, primarily at the graduate level. The **G**-matrix Simulator Home Version provides a very simple way for students to explore **G**-matrix evolution under various

evolutionary scenarios. The skills required to run the program are minimal, as it involves downloading the program from GitHub and double-clicking on the executable. The parameters are then easily altered through a graphical user interface. The running program produces a real-time graphical depiction of the **G**-matrix evolving in phenotypic space, and the program provides text output of per-generation summary statistics. One limitation is that this program runs only on Windows machines, or other operating systems with some sort of Windows emulator installed. For non-Windows users, the **G**-matrix Simulator Command Line program can be used with the provided R script for **G**-matrix plotting. This program is slightly more difficult to use, as it requires at least minimal familiarity with the command line.

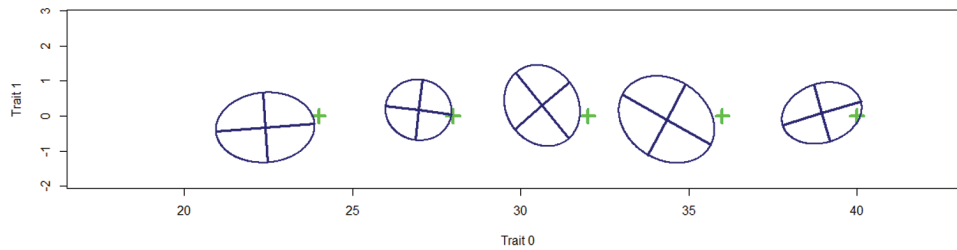


Figure 3. An example of the results of the **G**-matrix drawing script in R applied to the output of **G**-matrix Simulator Command Line. Here, the **G**-matrix is drawn as an ellipse (with the crossbars representing the scaled eigenvectors). The **G**-matrix is depicted every 400 generations, and the position of the moving optimum is shown as a cross. In this case, the genetic correlation is not strong, so the **G**-matrix lags directly behind the moving optimum. One advantage of using R to depict the **G**-matrix is that the script can be modified easily to suit the needs of the user.

The second goal of the **G**-matrix Simulator software family is to allow other researchers to get started developing individual-based quantitative genetic simulations. Due to computational speed limitations, development of these models almost always requires knowledge of a high-performance, compiled programming language, such as C++. Models in interpreted languages, such as Python or R, will almost always be prohibitively slow. In addition, the development of the model requires multiple decisions regarding the structure of underlying algorithms. Thus, our second goal is to demystify this process. We provide the source files for the **G**-matrix Command Line Simulator and a free tutorial that documents the step-by-step process of coding an object-oriented **G**-matrix simulator in C++ (Jones 2018). This tutorial assumes no starting knowledge of C++, so it is suitable for anyone wishing to embark on this research avenue. A key step in development is to ensure that the new code base can replicate results of previous models, and the various simulators we provide here provide useful tools for this validation step.

Other Related Software

Our models are not the only ones capable of simulating the evolution of a phenotype composed of multiple quantitative traits. The most relevant similar programs are Nemo (Guillaume and Rougemont 2006) and QuantiNemo (Neuenschwander et al. 2008), with the latter built on the source code of the former. Nemo provides a flexible, feature-rich programming platform with many facets that are not available in our family of **G**-matrix simulator programs. In this sense, the programs we present here do not compete with Nemo. Rather, these various programs are complementary. With respect to classroom instruction, our **G**-matrix simulators with graphical interfaces allow the class to hit the ground running, quickly iterating between modifying parameter values and seeing the results of the simulations. At the other end of the spectrum, for developers, the **G**-matrix Simulator Command Line source code is a much simpler code base compared with that of Nemo, and our tutorial assumes no knowledge of C++ programming. Thus, Nemo is more feature-rich and harder to use, but the **G**-matrix simulator is more accessible for beginners, including students interested in the **G**-matrix and scientists new to programming. The **G**-matrix simulators we present here can be seen as a starting point, where individuals can gain some familiarity with the issues surrounding individual-based simulations and **G**-matrix evolution before graduating to the more sophisticated and powerful programming environment afforded by Nemo.

A handful of other programs possess some similarity to the **G**-matrix Simulator family of applications, but none of these other programs is capable of precisely replicating the approach used by Jones et al. (2003, 2004, 2012). For instance, FFPopSim provides a more

efficient method for simulating multi-locus genotypes in a forward-in-time model (Zanini and Neher 2012) but does not serve as a replacement for Nemo or the **G**-matrix simulators, as it does not explicitly model quantitative traits. Similarly, ALADYN (Schiffers et al. 2014) simulates a phenotype composed of 2 quantitative traits but does not allow pleiotropy and requires one trait's optimum to vary spatially while the other trait's optimum varies temporally. Thus, ALADYN cannot address precisely the same quantitative genetic scenarios implemented in the **G**-matrix simulator models. Many other population and quantitative genetic simulation packages are available (see reviews by Hoban et al. 2012 and Hoban 2014) but Nemo and QuantiNemo appear to be the only ones capable of multivariate simulations focusing on the **G**-matrix in the vein of Jones et al. (2003, 2004, 2012).

Conclusions

The **G**-matrix Simulator family of software applications provides a useful set of tools for teaching and research. These programs are especially useful for helping students develop intuition regarding the evolution of the **G**-matrix and quantitative traits under various evolutionary scenarios. The tools can also be used to address new topics, to ground-truth newly developed models, and to provide budding developers an introduction to C++ and the design of individual-based models of quantitative traits.

Funding

The development of very early versions of the **G**-matrix Simulator was supported by a grant from the National Institutes of Health (1F32GM020561-01 to A.G.J.) and the National Science Foundation (DEB-0448268 to A.G.J., S.J.A., and R.B.).

Conflict of Interest

None declared.

Acknowledgments

We are grateful for feedback from graduate students and postdocs taking our workshops and tutorials in evolutionary quantitative genetics (2011–2018) at NESCent, NimBios, and Friday Harbor.

Data Availability

All software referenced in this article can be obtained from GitHub (<https://github.com/JonesLabIdaho>).

References

- Aguirre JD, Hine E, McGuigan K, Blows MW. 2014. Comparing G: multivariate analysis of genetic variation in multiple populations. *Heredity (Edinb)*. 112:21–29.
- Arnold SJ, Bürger R, Hohenlohe PA, Ajie BC, Jones AG. 2008. Understanding the evolution and stability of the G-matrix. *Evolution*. 62:2451–2461.
- Arnold SJ, Pfrender ME, Jones AG. 2001. The adaptive landscape as a conceptual bridge between micro- and macroevolution. *Genetica*. 112–113:9–32.
- Bégin M, Roff DA. 2001. An analysis of G matrix variation in two closely related cricket species, *Gryllus firmus* and *G. pennsylvanicus*. *J Evol Biol*. 14:1–13.
- Bürger R, Lande R. 1994. On the distribution of the mean and variance of a quantitative trait under mutation-selection-drift balance. *Genetics*. 138:901–912.
- Bürger R, Krall C. 2004. Quantitative-genetic models and changing environments. In: Ferriere R, Dieckmann U, Couvet D, editors. Cambridge (UK): Cambridge University Press. p. 171–187.
- Bürger R, Wagner GP, Stettinger F. 1989. How much heritable variation can be maintained in finite populations by a mutation-selection balance? *Evolution*. 43:1748–1766.
- Crow JF, Kimura M. 1964. The theory of genetic loads. In: Geerts SJ, editor. *Proceedings of the XI international congress of genetics*. Oxford (UK): Pergamon. p. 495–505.
- Draghi JA, Whitlock MC. 2012. Phenotypic plasticity facilitates mutational variance, genetic variance, and evolvability along the major axis of environmental variation. *Evolution*. 66:2891–2902.
- Falconer DS, Mackay TFC. 1996. *Introduction to quantitative genetics*. 4th ed. London (UK): Pearson.
- Guillaume F, Rougemont J. 2006. Nemo: an evolutionary and population genetics programming framework. *Bioinformatics*. 22:2556–2557.
- Guillaume F, Whitlock MC. 2007. Effects of migration on the genetic covariance matrix. *Evolution*. 61:2398–2409.
- Hoban S. 2014. An overview of the utility of population simulation software in molecular ecology. *Mol Ecol*. 23:2383–2401.
- Hoban S, Bertorelle G, Gaggiotti OE. 2012. Computer simulations: tools for population and evolutionary genetics. *Nat Rev Genet*. 13:110–122.
- Johansson F, Lind MI, Ingvarsson PK, Bokma F. 2012. Evolution of the G-matrix in life history traits in the common frog during a recent colonisation of an island system. *Evol Ecol*. 26:863–878.
- Jones AG. 2018. *C++ for biologists: evolutionary models*. Moscow (ID): Independently published.
- Jones AG, Arnold SJ, Bürger R. 2003. Stability of the G-matrix in a population experiencing pleiotropic mutation, stabilizing selection, and genetic drift. *Evolution*. 57:1747–1760.
- Jones AG, Arnold SJ, Bürger R. 2004. Evolution and stability of the G-matrix on a landscape with a moving optimum. *Evolution*. 58:1639–1654.
- Jones AG, Bürger R, Arnold SJ, Hohenlohe PA, Uyeda JC. 2012. The effects of stochastic and episodic movement of the optimum on the evolution of the G-matrix and the response of the trait mean to selection. *J Evol Biol*. 25:2210–2231.
- Kohn LAP, Atchley WR. 1988. How similar are genetic correlation structures? Data from mice and rats. *Evolution*. 42:467–481.
- Krall C. 2005. *Effects of pleiotropy on the genetic variance and extinction times of small populations* [Dissertation]. Vienna, Austria: Universität Wien.
- Lande R. 1979. Quantitative genetic analysis of multivariate evolution, applied to brain:body size allometry. *Evolution*. 33:402–416.
- Lande R. 1980. The genetic covariance between characters maintained by pleiotropic mutations. *Genetics*. 94:203–215.
- Lande R, Arnold SJ. 1983. The measurement of selection on correlated characters. *Evolution*. 37:1210–1226.
- Neuenschwander S, Hospital F, Guillaume F, Goudet J. 2008. quantiNemo: an individual-based program to simulate quantitative traits with explicit genetic architecture in a dynamic metapopulation. *Bioinformatics*. 24:1552–1553.
- Puentes A, Granath G, Ågren J. 2016. Similarity in G matrix structure among natural populations of *Arabidopsis lyrata*. *Evolution*. 70:2370–2386.
- Revell LJ. 2007. The G matrix under fluctuating correlational mutation and selection. *Evolution*. 61:1857–1872.
- Roff DA, Mousseau TA, Howard DJ. 1999. Variation in genetic architecture of calling song among populations of *Allonemobius socius*, *A. fasciatus*, and a hybrid population: drift or selection? *Evolution*. 53:216–224.
- Schiffers KH, Travis JM. 2014. ALADYN—a spatially explicit, allelic model for simulating adaptive dynamics. *Ecography (Cop)*. 37:1288–1291.
- Schluter D. 1996. Adaptive radiation along genetic lines of least resistance. *Evolution*. 50:1766–1774.
- Steppan AJ, Phillips PC, Houle D. 2002. Comparative quantitative genetics: evolution of the G matrix. *Trends Ecol Evol*. 17:320–327.
- Turelli M. 1985. Effects of pleiotropy on predictions concerning mutation-selection balance for polygenic traits. *Genetics*. 111:165–195.
- Turelli M. 1988. Phenotypic evolution, constant covariances, and the maintenance of additive variance. *Evolution*. 42:1342–1347.
- Turelli M, Barton NH. 1990. Dynamics of polygenic characters under selection. *Theor Pop Biol*. 38:1–57.
- Wood CW, Brodie ED III. 2015. Environmental effects on the structure of the G-matrix. *Evolution*. 69:2927–2940.
- Zanini F, Neher RA. 2012. FFPopSim: an efficient forward simulation package for the evolution of large populations. *Bioinformatics*. 28:3332–3333.