Geonomics: A Python package for building agent-based, spatially explicit, and arbitrarily complex landscape-genomic simulations

Drew Hart

January 18, 2019

1 Abstract

TO BE COMPLETED

2 Intro/Background

There is ever-growing interest in understanding and even predicting the genomic evolution of complex study systems on complex and changing landscapes. Such systems might include one or multiple populations or species that are not at equilibrium, inhabiting complex, multivariate, and changing landscapes, and undergoing both neutral evolution and natural selection (often on multiple traits). Landscape genomics studies the ways in which ecological and evolutionary processes playing out on real landscapes generate geographical patterns of genomic diversity CITE, and the field frequently features analysis of data collected from study systems of such genomic and geospatial complexity CITATIONS. Study of such systems is crucial for improving our understanding of real-world systems, CITE PERSPECTIVE ON WILD MOUSE EVOLUTIONARY STUDY, 02/01/19 issue of Science, and for better anticipating evolutionary responses to climate change CITATIONS and other sources of environmental change in the Anthropocene CITATIONS.

But the complex genomics of such systems are beyond the reach of analytical population genetics, and their spatial complexity and non-neutral evolutionary dynamics make them intractable for coalescent simulation. This hinders not only our understanding of many empirical systems and our ability to unambiguously interpret analytical results, but also our ability to predict those systems' dynamics, and thus to manage them appropriately. Thus, as is increasingly the case in many fields, forward-time simulation is a crucial tool for dissecting the evolutionary dynamics of complex study systems in landscape genomics. However, the current suite of forward-time genomic simulators, however numerous, is still of limited use for such work. Most available software is limited, either genomically or geospatially, in the complexity it can model. Many programs can model systems of considerable genomic complexity (e.g. simuPOP, NEMO and QuantiNemo), yet incorporate no or only rudimentary spatial components. Various other programs are designed specifically for landscape-genetic simulations (e.g. CDPOP, CDmetaPOP, SimAdapt), but are limited in their genomic complexity (i.e. are incapable of modeling simultaneous selection on numerous multigenic traits).

To our knowledge, SLiM is the only package currently capable of simulating scenarios that are both as genomically and as geospatially complex as those for which Geonomics is designed need to cite https://popmodels.cancerco (Indeed, the complexity of which SLiM models are capable far exceeds that of Geonomics.) However, SLiM is not designed first and foremost as a landscape-genomic simulator. Thus the complex landscape-genomic models for which Geonomics is designed would require a considerable amount of work to script in SLiM's Eidos framework (see examples of spatialized selecition in SLiM 2's recipe book; e.g. section 14.11, page 288 CITE THIS). For example, a Geonomics user could build a model of evolution with natural selection on multiple multigenic traits, on a multivariate landscape undergoing spatially inhomogeneous environmental change for certain landscape layers, in a species moving realistically across that landscape, and then run this

model an arbitrary number of times, collecting data at various points during each run — all of this by doing nothing more than templating, editing, and reading in an informatively annotated parameters file. Indeed, the generation of that template parameters file, the instantiation of a model from it, and the running of that model will cost a mere three lines of code.

What's more, Geonomics is written and run entirely in Python, a broad-purpose and popular programming language that is already familiar to most people with exposure to bioinformatics. This of course makes Geonomics considerably slower than its brethren that are written in compiled languages such as C++ (e.g. SLiM). But run time is not expected to be a major constraint for the sorts of models for which Geonomics was built (see runtime analyses, below). And what Geonomics sacrifices in performance, it gains in flexibility, extensibility, and accessibility. In fact, the basic user needn't even need to know how to write full Python scripts to build a Geonomics model; users can build their own models by recycling and tweaking a minimal amount of code (available within the documentation and at the Geonomics homepage). On the other hand, advanced users wishing to code their own extensions or customizations have broad opportunity to do so, because Geonomics is a Python package that is seamlessly integrated into the universe of Python functionality.

3 Model overview

3.1 Components

A Geonomics model consists of two core components, the first of which is the species. Each species consists of an arbitrary number of individuals, which do not pertain to populations but instead are distributed in continuous space upon the landscape. A species is described by a wide variety of import demographic and life-history parameters that determine its behavior in the model (e.g. intrinsic growth rate, mate-search radius, mean number of offspring per mating event, reproductive age and maximum age, and so on; for a detailed discussion and these and all other model parameters, see the documentation). Each species can also undergo any number of arbitrarily complex demographic changes during each model run, including both population-size changes (which can be exponential, cyclical, stochastic, or custom-defined) and changes to various demographic and life-history parameters.

Each individual in a species has an x,y location, a sex, an age (or stage), a genome (optionally), and a phenotype for any traits assigned to the species. (It is worth noting that species are collected into communities. For most purposes a community will consist of only a single species. But the community framework gives the advanced user the potential to code inter-species interactions for multi-species models, a functionality that we hope to build into future versions of Geonomics.)

In simulations that use genomes, each individual has a diploid genome consisting of a L diallelic loci, where L is the genomic length being simulated. These loci can be treated as representing either a contiguous haplotype block or a set of discrete markers, depending on the (homo- or heterogeneous) recombination rates. (For simplicity's sake, we refer to these loci herein, and in the software generally, as 'the genome' of an individual.) The genomic architecture is a set of parameters describing all simulated loci, for each of which each individual's genome has a genotype. These parameters include the starting allele frequencies and dominance values for all loci, the inter-locus recombination rates across the genome, and the lengths of all simulated chromosomes (i.e. sections of the genome separated by 0.5 recombination rates)., A genomic architecture can also stipulate any number of mono- or multigenic traits for a species. Each trait is defined by a set of loci that underpin it, the effect sizes of those loci, and a selection coefficient (which can be heterogeneous or homogeneous in both space and time). Mutations, which are of three types — neutral, deleterious, and trait-affecting — are controlled by type-specific mutation rates (additional parameters within the genomic architecture, any or all of which can be set to zero). (To simulate complex, specific genomic architectures, users can feed into Geonomics a CSV-formatted file defining the architecture locus by locus. For details, see the documentation.)

The second core component of every Geonomics model is the landscape. Each landscape consists of a stack of an arbitrary number of layers (i.e. variables), each represented by a normalized ($0 \leq z \leq 1$) raster.

Each layer can be programmed to serve as the basis for any of a number of model components: 1.) the raster of cell-wise carrying-capacities controlling the population density of a species; 2.) the conductance surface controlling the realistic movemement of individuals and/or dispersal of offspring across the landscape; 3.) the selective force acting on one or more traits of one or more species. Each layer of the landscape can undergo its own, arbitrarily complex environmental change event during each model run, which will in turn affect the dynamics of any species on the landscape for which that layer plays a role in its population dynamics, movement or dispersal, and/or natural selection.

3.2 Operations

A Geonomics model can be run for an arbitrary number of runs. At the start of each run, the must be burned in. This is accomplished by running the model (without the genomic or selective components) until a series of statistical tests determines that each species' population size and spatial distribution has reached dynamic equilibrium. Then, if genomes are being used, each individual has a genome randomyl drawn (according to its genomic architecture) and assigned, such that the main phase of each model run begins in the absence of any population structure. At this point, the 'main' phase of the run can run for any number of timesteps. Each timestep is composed of a series of actions, some requisite, some optional (see Fig. 1 for details):

1. age/stage incrementation (requisite);
2. movement (optional);
3. mate-finding and mating (requisite);
4. gamete production (optional, because use of genomes is optional);
5. offspring dispersal (requisite);
6. mortality (due to the combination of density-dependence [requisite] and natural selection [optional]);
7. demographic change events (optional);
8. landscape-change events (optional);
9. recording of data and statistics (optional).

Each individual's age/stage increments at each timestep. A number of parameters can be set so as to modify a model's behavior on the basis of this attribute (including the minimum age of reproduction and the maximum age of a species).

Movement is in continuous space. Individuals' distances and directions are drawn separately, then composed into movement vectors. Distances are Wald-distributed (and the distributional parameters, as with nearly all distributions used in the model, can be set by the user within a Geonomics parameters file). Directions can either be drawn from a uniform distribution on the unit circle, resulting in isotropic movement (the default behavior); or they can be drawn from a 'movement surface' — an array of uni- or multimodal Von Mises distributions, derived from a landscape layer that serves as a conductance surface, which generates realistic, anisotropic movment across an environment of heterogenous habitat quality.

Mating pairs are chosen from among all pairs of individuals within the species' mate-search radius (based on eligibility by age and sex, and decided by a Bernoulli draw with probability equal to the species' intrinsic birth rate). For each mating pair a number of offspring is chosen (from a Poisson distribution with lambda equal to the species' mean number of offspring, unless the user fixes the number of offspring lambda). Each parent produces one gamete for each of its offspring, by recombination (at the inter-locus rates defined by the species' genomic architecture) and Mendelian segregation. Offspring individuals are created and then dispersed to a new location (where, as with movement, the directions of their dispersal vectors can be drawn either isotropically or anisotropically, the latter using a 'dispersal surface').

Mating is followed by mortality. Deaths are drawn binomially, based on individual-wise death probabilities, which are calculated as a combination of a combination of the probability of death by density-dependence

(from a spatialized logistic-growth model) and the probability of death by natural selection (on any number of traits simultaneously). This is calculated as:

$$P(d_i) = 1 - (1 - P(d_{x,y})) \prod_{p=1}^{m} \omega_{i,p} \tag{1}$$

where $P(d_{x,y})$ is individual i's probability of death by density-dependence and $\omega_{i,p}$ is individual i's fitness for trait p (and only factors in if natural selection is being used). The probability of density-dependent death at location x, y is calculated as:

$$P(d_{x,y}) = E[N_{d;x,y}]/N_{x,y} = \frac{E[N_{b;x,y}] - \frac{\mathrm{d}N_{x,y}}{\mathrm{d}t}}{N_{x,y}} \tag{2}$$

where $E[N_{d;x,y}]$ is the expected number of deaths at the individual's x, y location on the landscape; $N_{x,y}$ is the population density at the location; $E[N_{b;x,y}$ is the expected number of births at the location; and $\frac{\mathrm{d}N_{x,y}}{\mathrm{d}t}$ is the logistic population growth rate at the location. Individual i's fitness for trait p is calculated as:

$$\omega_{i,p} = 1 - \phi_{p;x,y}(\mid e_{p;x,y} - z_{i;p} \mid)^{\gamma_p} \tag{3}$$

where $\phi_{p;x,y}$ is the selection coefficient on trait p at the individual's location; $e_{p;x,y}$ is the environmental value of the layer that serves as the selective force for trait pi, γ_p defines the curvature of the fitness function for trait p, and $z_{i;p}$ is individual i's phenotype for trait p, which is calculated from the additive effects of the individual's genotypes at all influencing loci (Geonomics does not model epistasis) as:

$$z_{i;p} = baseline\_genotype + \sum_{l=0}^{n} \alpha_{p,l} g_{i,l} \tag{4}$$

where n is the number of loci, α is a locus' effect size, g is the individual's genotype, and *baseline_genotype* is 0 for monogenic traits, 0.5 for polygenic traits.

Geonomics is designed as an object-oriented scripting framework with both basic and advanced use-modes. The basic mode simply requires users to call a first command to create a template parameters file (which they must then edit as desired), a second command to create a model object from that parameters file, and a third command to then run the desired number of runs for that model (see Fig. 1). The advanced mode allows users to make modifications to the components of their models, or to collect custom data from their model, by calling additional functions before a model is run, between the timesteps of a run, or after a run is complete. This can be done using both built-in Geonomics functions and homespun Python code. (This is what makes Geonomics so extensible — because it is a Python package, users can call on the full spectrum of Python functionality to design custom code that can interact with Geonomics objects.)

Landscape and demographic change events unfold over some portion of the total time of a run. The changes are made incrementally, with each incremental change being made during each of the timesteps in the series of timesteps defined *a priori* by the user (in the parameters file). Likewise, statistics and data are calculated and collected at each of the set of timesteps defined *a prior* by the user.

4 Validations tests

- same as before, but: - update figures - improve captions (e.g. include all necessary, e.g. include the avg pop size and recombination rate in the sweep model) so that they stand on their own

5 Validations tests OLD

We have run a series of tests to statistically and heuristically validate the full range of functionality available in Geonomics. In this section, we briefly review the reasoning and setup for each test, then we present

the results. All results are as expected, demonstrating a robust ability to reproduce population-genetic and population-genomic findings. Some results show minor deviations, attributable to the fact that these tests are using a simulation framework designed for complex, spatially explicit models to approximate much simpler mathematical models. These deviations are discussed where applicable. (The code for all tests is available in the './tests/validation' directory of the package.)

5.1 Wright-Fisher test: genetic drift

The Wright-Fisher model of genetic drift models a fixed-size haploid population that turns over completely at each timestep (i.e. generation). The population can have any number of independent, biallelic genetic loci. For each locus, each generation's allele frequency is chosen as a binomial random variable, with the number of trials equal to the population size and the probability of success (i.e. of drawing the '1' allele) equal to the previous generation's '1'-allele frequency. The mean persistence time for an allele (i.e. the expected number of generations for which a locus remains segregating) is:

$$\bar{t}(p) = -4N[(1-p)ln(1-p) + pln(p)] \tag{5}$$

where $2N$ is the number of alleles in the population (such that N can represent the diploid population size) and p is the frequency of either allele at the locus large cite Hartl and Clark 2007

Clearly, the Wright-Fisher model is much simpler than the sorts of models for which Geonomics is designed (as are all of the following validations tests)—it is aspatial, panmictic, features fixed population sizes, models only neutral loci, and so forth. I parameterized Geonomics so as to approximate the model as closely as possible. To emulate aspatiality and panmixia, I used a population on a homogeneous landscape, with isotropic movement, and with movement and dispersal distributions and a mating radius that broadly encompass the diagonal width of the landscape. To enforce complete generational turnover, I set the maximum age parameter to 1 timestep. While Geonomics cannot maintain constant population size, I maintained the carrying-capacity raster at a constant, uniform value, thus maintaining a stationary mean population size. I simulated 100, independent neutral loci (by setting all interlocus recombination rates to 0.5), with starting '1'-allele frequencies of 0.5 (although the actual starting frequencies vary slightly around this value because of sampling error when all individuals' genotypes are drawn binomially). Of course, I did not implement natural selection or any environmental or demographic changes. (I set other parameter values reasonably, or left them at defaults; see `wf_params.py` for specifics.)

I ran the Wright-Fisher approximation test for three cell-values of the carrying-capacity raster (i.e. three values of 'K_factor'), hence for three mean population sizes. For each mean population size (calculated as the harmonic mean, to account for stochastic fluctations around the carrying capacity), I compared the mean persistence time to that expected by theory, according to the formula cited in the previous paragraph. As can be seen in figures 2 and 3, the results compare favorably to theory, although the observed mean persistence time undershoots the expected at all population sizes because of a variety of artefacts of the approximation. Despite the movement, dispersal, and mating-radius parameterizations, the simulated populations will still exhibit some neighborhood-mating behavior, generating a spatialized coalescent that reduces effective population size below the harmonic mean of the census size. Effective population size should also undershoot harmonic-mean size in Geonomics models because of the imperfection with which the "infinite gametes" assumption of the Wright-Fisher model's binomial draws of allele frequencies is modeled by random draws of alleles from each pair of mating parents. DOES THAT MAKE SENSE...? Other factors that would artificially reduce the time to fixation or loss of an allele include the fact that most alleles are not starting off at precisely 0.5 frequencies; and the fact that Geonomics uses a large, *a priori* sample of genome-wide recombination events to approximate realistic recombination at the stipulated recombination rates, such that sampling error could lead to artificially inflated linkage between some neighboring pairs of loci. Thus, overall, Geonomics demonstrates realistic Wright-Fisher-type drift, with persistence times characteristic of a population whose effective size undershoots its mean size.

5.2 Bottleneck test: population dynamics

Genetic drift is always operative in a population. Population dynamics can have marked effects on the rate of drift; because drift is a stronger evolutionary force in smaller populations, drift accelerates in shrinking populations. If a population undergoes a sudden reduction in size, followed by a quick recovery to its original size, i.e. a bottleneck event, the overall effect of drift on that population is expected to be larger than a constant-size population of equivalent starting size. In other words, the mean persistence time in that population should be shorter in the bottlenecked population that in the constant-size comparison population.

As with the Wright-Fisher model, I used a homogeneous landscape with broad distributions for movement and dispersal and with a mating radius that encompasses the full landscape to emulate aspatiality and panmixia. Geonomics features a variety of parameters for parameterizing demographic change events. To simulate a bottleneck event, I created a custom change event in which the population's carrying-capacity raster is reduced to 30% of its initial value for 50 timesteps (from the 200th to 250th), then returned to its initial value for the remainder of the simulation (through the 2500th timestep). (I set other parameter values reasonably, or left them at defaults; see `bottleneck_params.py` for specifics.)

Figure 4 shows a clear signal of drift acceleration during the bottleneck event.

5.3 Stepping-stone test: population subdivision and genetic differentiation

The stepping-stone model, or one-dimensional island model, is a spatially implicit model. It models a series of subpopulations, arranged along a straight line, with migration between all neighboring pairs. (I chose to validate Geonomics using the one-dimensional island model rather than the basic island model, in which all island pairs have equal migration rates, because of the impossibility of arranging more than three islands in two-dimensional space such that all island pairs are equidistant, and thus have equal migration rates; this is fine, however, because for the same reason it is unclear how applicable the basic island model is to real-world systems.) As a combined result of divergence by drift and homogenization by effective migration, subpopulations reach a stationary level of genetic differentiation—called migration-drift equilibrium. Theory provides the expected pairwise genetic differentiation between a pair of subpopulations at migration-drift equilibrium as:

$$F_{ST} = \frac{1}{1 + 4Nm} \tag{6}$$

where N is the population size and m is the per-generation migration rate, such that Nm can be interpreted as the per-generation number of migrant individuals cite Hartl and Clark 2007

To approximate the stepping-stone model, I created a Landscape Layer with a diagonal of six equally spaced islands (1.0-valued cells) embedded in a 'sea' of 0.0-valued cells. I used this layer as the carrying-capacity raster. I set the mating radius to encompass an individual's current island, but no neighboring islands. I parameterized dispersal to be very local to parents' midpoints, and parameterized movement to have a long right-skewed tail, such that long-distance movement (and hence potentially migration) events are uncommon. (I set other parameter values reasonably, or left them at defaults; see `stepping_stone_params.py` for specifics.) I ran the simulation for 5000 timesteps. Because Geonomics cannot implement stipulated migration rates between express portions of its continuous Landscape, I manually tracked the number of migration events during each timestep, for all possible directional migration events (i.e. for all permutations of two islands). I used that data to solve the equation cited in the previous paragraph, and compared the resulting F_{ST} expectations to the observed values (calculated from the simulated date using two common methods; see fig. 7 for details).

Results demonstrate that the model has approached reached migration-drift equilibrium (fig. 5), and that all island populations were at dynamic equilibria around the same mean size (fig. 6). Estimated migration rates and F_{ST} values qualitatively match theoretical expectations: islands greater than one step apart drop off precipitously and then continually decrease in mean migration rate, and genetic differentiation increases toward a saturating level of F_{ST}. Values of F_{ST} consistently undershoot the values expected based on

estimated migration rates, however, likely because the subpopulations have yet to approach fixation at most loci (i.e. the overall population has approached but still not reached migration-drift equilibrium).

5.4 Contrasting-habitat test: adaptive divergence

If a population is divided into two subpopulations which inhabit contrasting environments exerting opposing selective forces, and there exists standing genetic variation for a biallelic locus underlying a trait that is responsive to those selective forces, then theory predicts that the two subpopulations will diverge at that locus as each population moves toward its respective adaptive peak. Analogously to the stepping-stone model, the rate at which that divergence will occur, in each of the subpopulations, depends on the relative strengths of two opposing evolutionary forces: the strength of natural selection, which causes divergence, and the strength of gene flow from migration, which homogenizes the two subpopulations. The rate of allele frequency change in either subpopulation at timestep t is expressed as:

$$\delta q = \frac{-spq[q + h(p - q)]}{1 - sq(2hp + q)} + m_i q^* - m_o q \tag{7}$$

where q and p are the frequencies of the deleterious and beneficial alleles (with respect to the subpopulation being analyzed), s is the selection coefficient against the homozygous recessive phenotype, h is the degree of dominance of the recessive allele, m_i and m^o are the migration rates into and out of the subpopulation being analyzed, and q^* is the frequency of the recessive allele in the alternative subpopulation cite Hartl and Clark 2007.

This model, much like the stepping-stone model, is spatially implicit (except in cases where it is used to represent sympatric ecological isolation). To approximate this, I created a Landscape with two Layers. The first was divided into two equal-sized halves, valued at 0.0 and 1.0, and was used as the environmental variable driving natural selection (such that 0.0 and 1.0 phenotypes were most fit, respectively). The second was valued uniformly at 1.0, and was used as the habitat-quality layer, which served as the carrying-capacity raster (thus setting uniform population density across the Landscape and determining, in sum, the overall carrying capacity of the landscape). I created one monogenic trait whose locus was randomly chosen within a genomic architecture of 100 independent (i.e. unlinked) loci. This trait was selected upon the first landscape layer. (I set other parameter values reasonably, or left them at defaults ;see `divergence_params.py` for specifics.) I ran the model for 1000 timesteps for each of three values of phi (which for a monogenic trait is identical to s, the selection coefficient in classical population genetics): 0.1, 0.05, and 0.01. Given that Geonomics does not employ express migration rates, I tracked the number of migration events during each timestep and used that data to solve the equation cited in the previous paragraph.

Results depict clear local adaptation to each of the two halves of the landscape, with opposite-phenotype bleedover and heterozygote births occuring mainly along the border between the two habitats (fig. 8, right). Allele trajectories in each half of the environment follow qualitatively the increasing and saturating trajectories expected by theory, but reach consistently more extreme allele frequencies than expected (fig. 8, left). This is likely because of WHAT?

5.5 Cline test: local adaptation

Similar to the contrasting-habitat model of adaptive divergence, but perhaps more to most real-world cases of local adaptation, is the cline model. In this model, a population adapts locally across a continuous environmental gradient, which is characterized by the extremes of its environmental values and its steepness (i.e. the instantaneous rate of environmental change along it). Local adaptation across this gradient will generate a cline, i.e. a geographic gradient in allele frequency (though of course natural selection is not the only evolutionary force that can generate a cline). In the clinally adapted population, loci that underlie the trait undergoing clinal selection are expected to exhibit clinal variation in allele frequency that mirrors the environmental gradient driving selection, whereas loci unlinked to those loci have no long-term expectation of concordant clinal variation (though they could initially be swept along with the selective locus if the beginning stages of clinal adaptation, and any number could continue to show spurious concordant clinal

variation by chance). To detect clinally adapted loci, we can fit cline curves to the spatial allele-frequency variation at all loci, with the expectation that the clines fit to adaptive will match the environmental gradient. Numerous equations have been used to fit clines, but classical models include a sigmoidal *tanh* function of the following form:

$$p_x = \frac{1}{2}(1 + \tanh[\frac{2(x-c)}{w}]) \tag{8}$$

where p is the frequency of the reference allele at position x along the cline, c is the centerpoint of the cline (such that $p_{x=c} = 0.5$), and w is the 'width', which is defined as $w = \frac{1}{slope}$ at centerpoint c cite Porter ClineFit manual.

To implement the cline model in Geonomics, I created a landscape with two layers. Similarly to the landscape used in the divergence model (see previous section), the landscape consisted of two layers, the first being an environmental layer (in this case a symmetrical non-linear gradient between 0-valued and 1-valued halves, rather than two discrete 0- and 1-valued halves), the second being a uniformly valued habitat-quality layer (used to set a uniform population density and thus determine the global carrying capacity). Also similarly to the divergence model, I created a monogenic trait whose locus was randomly placed within a genomic architecture of 100 independent loci. The trait had a *phi* (i.e. s) of 0.01, and was selected upon by the gradient layer. (I set other parameter values reasonably, or left them at defaults; see `divergence_params.py` for specifics.)

I ran the cline model for 2500 timesteps, then used a numerical optimization function in Python's `scipy` package to fit *tanh* clines to all loci. I plotted all fitted clines on top of the first landscape layer, with the cline for the one selective locus highlighted. The selective locus consistently and clearly stands out as the only locus with a cline matching the expectation (i.e. mirroring the environmental gradient; fig. 9, left), and results show an obviously locally adapted population, with a zone of hybridization and phenotypic spillover surround the clinal center (fig. 9, right). Furthermore, for a family of regression models of environmental value on genotype for all loci, after correction for multiple testing, the selective locus consistently stands out as the most significantly correlated locus (though numerous other loci also give false-positive results, albeit with considerably larger p-values). SHOW THESE RESULTS!

5.6 Selective sweep test: genetic hitchhiking

While classical population genetics provides us with a great deal of understanding under scenarios of selection on unlinked loci, population genomics attempts to reckon with the reality that all loci under selection are actually embedded within a contiguous genome, and thus are tightly linked to a block neighboring loci. Consideration of genomic context and linkage considerably complicates the study of molecular adaptive evolution, but is essential for understanding and interpreting data in the genomic age. The most basic model of selection with linkage is that of the selective sweep: a beneficial mutation occurs in a population, falling on a random genomic background, and then rises in frequency because of its selective advantage until it becomes fixed, pulling up the frequency of its haplotype as it does so. But even as the haplotype increases in frequency it is continually subject to recombination, which gradually erodes the haplotype block, causing it to contract around the beneficial mutaiton and freeing its loci from linkage to the mutation. Thus the model predicts that once a beneficial mutation occurs—as long as it is not lost early on by chance— it and a haplotype block around it will rise in frequency, the mutation will eventually fix, and the haplotype block will continually fade over time. The haplotype block should be clearly visible in genomic data, where it will manifest as a genomic region of reduced diversity and heterozygosity, centered on the mutation.

To implement the selective sweep model in Geonomics, I again created a model approximating an aspatial, panmicitic population (with movement and dispersal distributions and mating-radius broadly encompassing the landscape, as for the Wright-Fisher and bottleneck tests, above). I created a single, monogenic trait in this population, with a *phi* of 0.15, and with its locus manually set to 50, and thus centered within the 101-locus genomic architecture. I manually set the starting '1'-allele frequency at this locus to 0.0, but set the trait to selected upon by the landscape's first and only layer, a uniformly 1-valued raster, such that all

individuals began the model equally ill-fit (i.e. with a fitness value of $1 - \phi = 0.85$). I burned the model in. Then I iteratively chose a random individual, introduced a '1'-mutation in their genome at locus 50, ran the model for 50 timesteps, and checked whether the '1' allele had reached a frequency greater than 0.05 by that time. I iterated until that check was passed, at which point I declared the mutant allele 'established' and continued to run the model until 2500 timesteps after the mutant allele had fixed, calculating genome-wide nucleotide diversity at three points during the model.

Geonomics very realistically emulated the behavior of a selective sweep. The adaptive phenotype (the '1'—'1' genotype, plotted as white on a white environmental background; in fig. 10, top row) clearly emerges in a region surrounding the mutation's origin, then spreads rapidly throughout the population from there. The population's mean fitness increases quickly from 0.85 (the universal fitness value before the mutation is introduced) to 1.00 (the universal fitness value after the sweep is complete; fig. 10, bottom row, leftmost plot). And there is a clear region of depressed nucleotide diversity immediately around the selective locus, which becomes more pronounced once the sweep goes toward completion, then slowly erodes over further time as a result of recombination of the mutant haplotype's alleles onto non-mutant backgrounds (fig. 10, bottom row, second, third, and fourth plots from the left).

5.7 PCA test: isolation by resistance

Many real-world populations inhabit landscapes with complex patterns of hetereogeneous habitat suitability. The probability of an individual moving across each part of these landscapes is a function of that part's habitat. Ecologists often use resistance surfaces (or their reciprocal, conductance surfaces) to describe movement across such landscapes. Geonomics' movement surfaces and dispersal surfaces both model such movement (for individuals' timestep-to-timestep movement and for the dispersal of new offspring, respectively). In a population evolving on such a landscape, gene flow between any two locatons on the landscape is expected to be an inverse function of the resistance-distance between the locations. As a result of this, a pattern of isolation by resistance (IBR) is expected to develop: pairwise genetic distances between different populations, or different regions, should be positively correlated with pairwise resistance distances.

To test this, we constructed a Geonomics model of a single species evolving neutrally for 1000 timesteps (i.e. 1000 rounds of mating) on a randomly generated complex landscape layer, with that layer serving as the basis for the species' movement surface. We ran a genetic Principal Components Analysis (PCA) on the full species' simulated genomes, both after the burn-in (i.e. when genomes had just been randomly drawn and assigned to all individuals) and after the model had run. For each PCA, we extracted the first three principal components (PCs), scaled them to $0 \leq value \leq$, then used the resulting numerals to determine a the red, green, and blue (RGB) values for the color of each individual. We used those colors to color each of the individuals in a plot of the full species (using the `mod.plot()` command).

The results (in figure 11) show a clear lack of spatial structure at the outset; because genomes were randomly drawn and assigned, the population possessed no spatial structure. But more importantly, the results demonstrate significant spatial strucutre, which maps onto the movement surface's landscape layer exactly as one would expect. Neutral evolution with realistic movement across this landscape generated a clear and strong signal of IBR.

5.8 simultaneous-selection test: selection on multiple traits

FIGURE OUT WHY THESE RESULTS DON'T LOOK RIGHT One of the powerful features of Geonomics is that it can simulate on numerous traits simultaneously, each trait being selected upon by a distinct, and potentially spatially diferentially distributed selective force. When a population is undergoing selection of this nature, the evolutionary outcome should be a function of the genomic architecture of the traits (i.e. how many loci underlie them, and whether or not their loci are linked), and of the correlation of the two traits' selective forces in space.

We simulated two scenarios for 1000 timesteps each. Both scenarios are situated on the same landscape—a landscape with two distinct and uncorrelated selective-force layers: one a symmetric, horizontal environmental gradient from 0 to 1, the other a an identical but vertical gradient. In both scenarios, the simulated

species has a 20-length genome, with 10 distinct loci underpinning each of the two traits in the simulation (i.e. there is no pleiotropy, hence no overlap in loci between the two traits). However, all loci are independent in the first scenario (recombination rate = 0.5), whereas all loci are linked in the second scenario (recombination rate ≈ 0.05).

Theory suggests that local adaptation should proceed successfully in the first scenario, but should be hindered in the second scenario because of genomic conflict resulting from opposing selection on closely linked loci. Heuristically, our results corroborate these expectations, with scenario 1 showing a clear signal of local adaptation (fig. 12) whereas scenario 2 shows a much messier and more complicated spatial distribution of phenotypes for each of the two traits (fig. 13).

6 Example use-case: Polygenic adaptation to climate change in the Yosemite region

- good for now, but reread and edit anyhow

7 Example use-case: Polygenic adaptation to climate change in the Yosemite region OLD

Perhaps Genomics' use-value is best demonstrated by way of a worked example. In this section, I explain a complex evolutionary scenario for which simulated data is desirable, explain the steps by which I have simulated the scenario in Geonomics, and present the simulation results. (The source code for this example is available within the `./example/yosemite` directory of the Geonomics repository.)

There is growing interest in the evolutionary implications of climate change. Much of this interest focuses on species that are locally adapted along some environmental gradient that is expected to shift, perhaps non-linearly, under climate change, and particularly with the potential for such species to respond adaptively to the change.

Here I simulate an example of such a study system: the adaptive response of a continuously disributed, locally adapted species to non-linear climate change in the Yosemite region. To begin with, I have downloaded a raster dataset of 30-year temperature normals of the Yosemite region (from Cal-Adapt; `www.caladapt.org`), cropped it to a 90-cell by 90-cell window around the Yosmite valley, and saved the cropped file. I then wrote a couple quick functions to process that raster into the full set of rasters I needed for my simulation.

Temperature is the environmental variable driving natural selection. In order to model adaptive responses to change in this variable, I needed a raster of future temperatures, to feed into a landscape-change event as the endpoint raster. (I could have downloaded a series of future-projection rasters, and loaded them in as the stepwise changes for the environmental-change event that I will create in this Geonomics model, because Geonomics is designed to accept a directory of such files as the steps in a change event. However, to keep things simple I just processed the 30-year normals raster with a simple, heuristic function.) So I wrote a function that adds to the 30-year normals raster a fixed temperature increase (2 degrees) plus an additional, elevation-dependent fraction of that increase (where there fraction varies from 0.0 at the hottest cell to 1.0 at the coldest cell). This function emulates the fact that warming due to climate change appears to be quickest at high elevations. I saved the resulting future-temperature array to a new raster file.

I also needed habitat-quality rasters for the model. These should reflect the fact that species tend to find highest-quality habitat, and thus exist at their highest densities, in the core of their range, and that edge habitat becomes increasingly marginal. Thus I wrote a function that accepts the temperature raster as input and produces from it a second, core-edge-type habitat-quality raster (where cells with temperature values between 7 and 11 degrees are assigned a 1.0 quality value, and cells outside that range are assigned quality values that linearly decrease to 0.0 in the hottest and coldest cells in the raster). I fed both the 30-year normals raster and the future temperature raster through this function, thereby generating both current and future habitat-quality rasters, which I saved to new raster files.

With these rasters prepared, I then created a Geonomics parameters file. I needed to parameterize a model with 2 'file'-type Layers, both of which would undergo landscape-change; with 1 Species, with movement, a movement-surface, and genomes with 1 trait; and with data-collection. The command to do this was:

```
>>> gnx.make_parameters_file(filepath='yosemite_params.py',
layers=[{'type': 'file', 'change': True},
{'type': 'file', 'change': True}],
species=[{'movement': True, 'movement_surface': True,
'genomes': True, 'n_traits': 1}],
data=True)
```

After running this command, I opened the resulting, auto-generated parameters file (`yosemite_params.py`) and edited the parameters as needed. In the Layer 'init' and 'change' parameters secitons I replaced the placeholder filenames with the names of the four temperature-derived rasters whose creation I explained above. I parameterized my landscape-change events: both Layers would change from their original rasters to their final rasters in 20 stepwise changes over 1000 timesteps (i.e. 1000 years), starting on timestep 499 and finishing on timestep 1499. The habitat-quality layer would serve as the basis for the movement-surface. The model would include no mutation. I stipulated the selection coefficient (*phi*) on my trait (0.05), and the number of loci underlying it (100 randomly-selected loci). Other parameters were left at default values.

Finally, I wrote a short script to create a model from the parameters ('mod = gnx.make_model('./yosemite_params.py')'), and then to manually burn in and run the model (by using the `mod.walk` function to run the model in either 'burn' or 'main' mode for the desired number of timesteps, and using `mod.plot_<_>` functions or manual `matplotlib` code to plot the model at various timesteps during its progress).

The model generates a clear and realistic pattern of polygenic adaptation to the elevation-based temperature gradient in the Yosemite region, and that gradient of local adpatation exhibits a pronounced upslope shift in response to the period of climate change. These results are visible both heuristically (from the individuals' phenotypes plotted across the three discrete timesteps' columns in fig. 14) and analytically (from the neighborhood-meaned phenotype rasters plotted across fig. 14, row 3).

8 Getting started

For those interested in using Geonomics, the simplest way is to install via `pip` (the most popular Python package-installation software), by calling `$ pip install geonomics`. (Note that Geonomics only uses common, well established Python packages as required dependencies: `numpy` `matplotlib`, `pandas`, `shapely`, `bitarray`, `scipy`, `sklearn`, `statsmodels`, and `vcf`.) The source code is also publicly available on GitHub (`URL\_HERE`), where it is actively maintained and developed.

Documentation is extensive (and continues to expand as new worked examples are offered and new functionality is added to the package), and is available online at `http://htmlpreview.github.io/?https://github.com/drewhart/geonomics\_docs/blob/master/built/doc.html`. The simplest and advanced use cases are explained above (see section *Design, structure, and function*), as well as in the documentation. The documentation provides a detailed section explaining the meaning, default values, and usage for every parameter in a Geonomics parameter file. Both the documentation and specific functions' docstrings (available by calling Python's `help()` on the function of interest) provide details on the usage of each function.

9 Caveats and considerations

- runtime analyses (basic), and explicitly point out the key concerns/parameters that users should think about for runtime and memory usage constraints (including, e.g., a layer that serves as basis for movement-surface and that changes a lot) - need to discuss the weird approach to mutation? - recombination is an

approximation (and how, briefly; see docs for details) - movement is an approximaton (and how, briefly; see docs for details)

10 Conclusion

- very very brief summary of, once again, Geonomics' design-purposed ability - think it should be very helpful for running simulations of high utility to climate change ecology, conservation, and comparison to many molecular ecology empirical datasets - should be valuable for theoretical, methodological, empirical, and applied research - expandibility planned in advance (e.g. multi-species models foreseen; MENTION AT END)

11 Works Cited

12 Figures and Captions

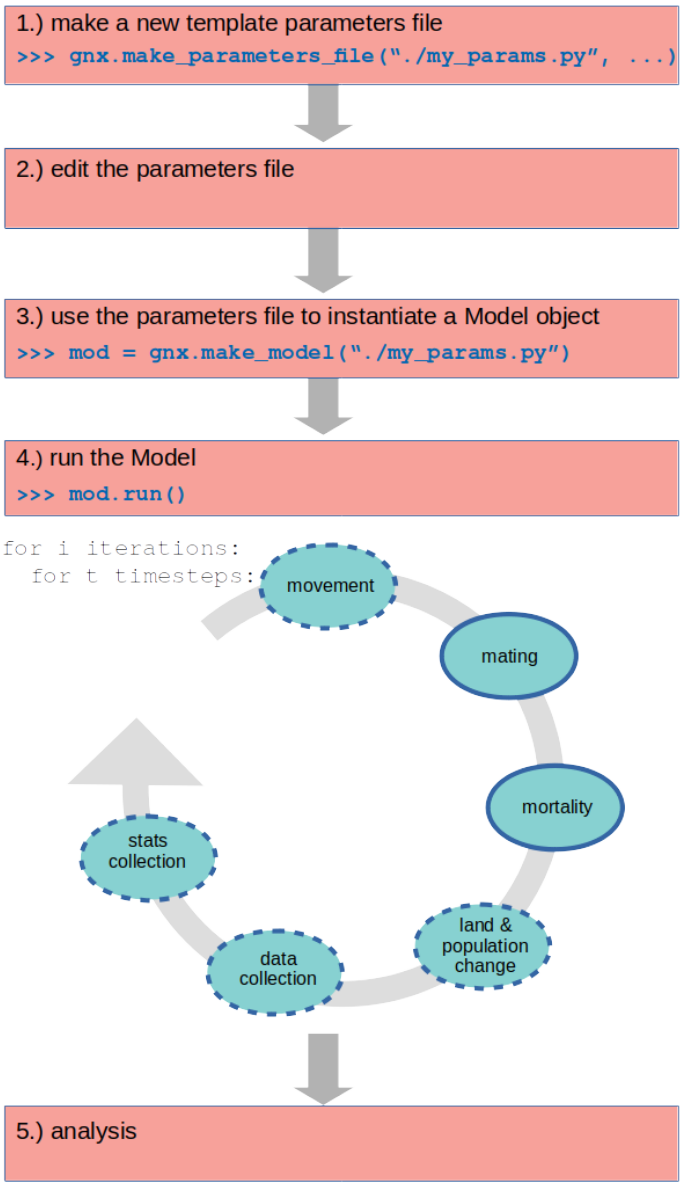

Figure 1: Geonomics flow diagram

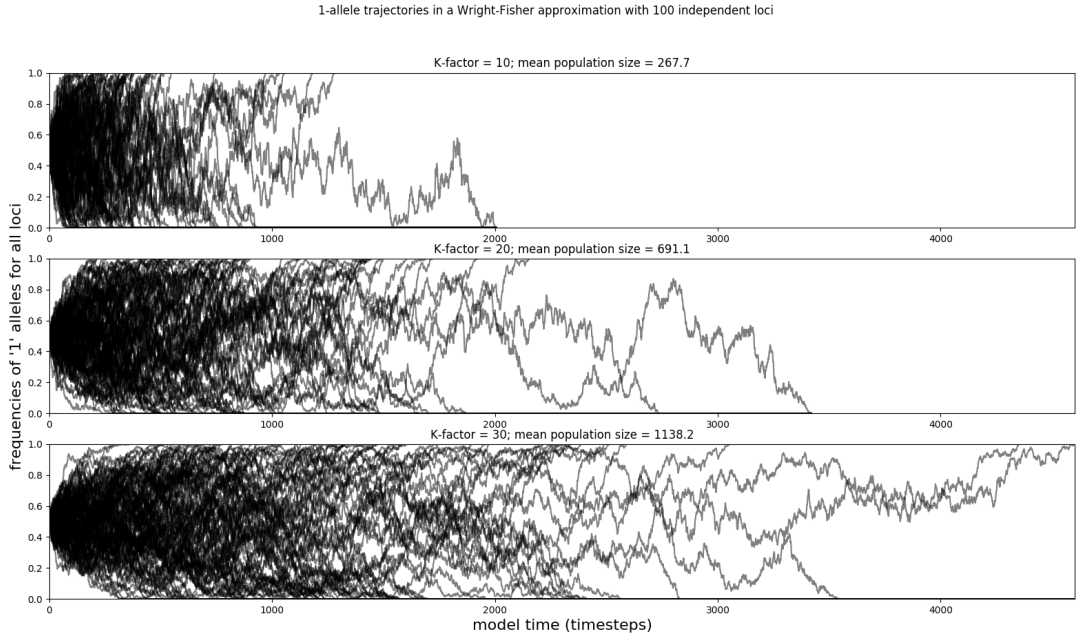

Figure 2: Wright-Fisher test: allele-frequency trajectories

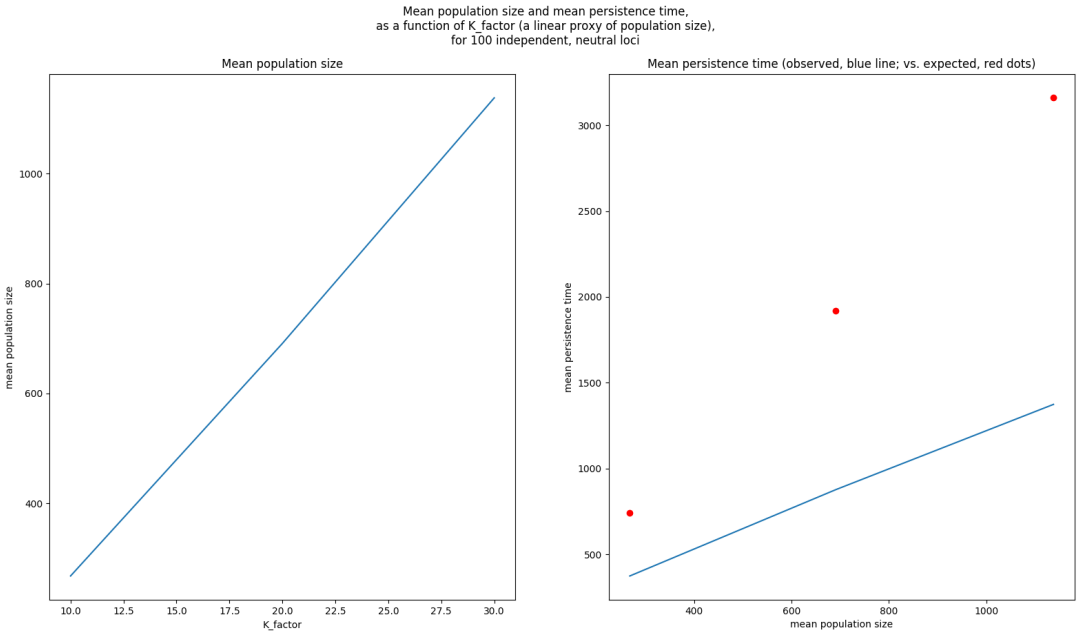

Figure 3: Wright-Fisher test: left: mean population size as a function of 'K_factor' (multiplicative factor determining a species' spatialized carrying capacity and thus equilibrium population size; right: mean persistence time for segregating sites as a function of mean population size

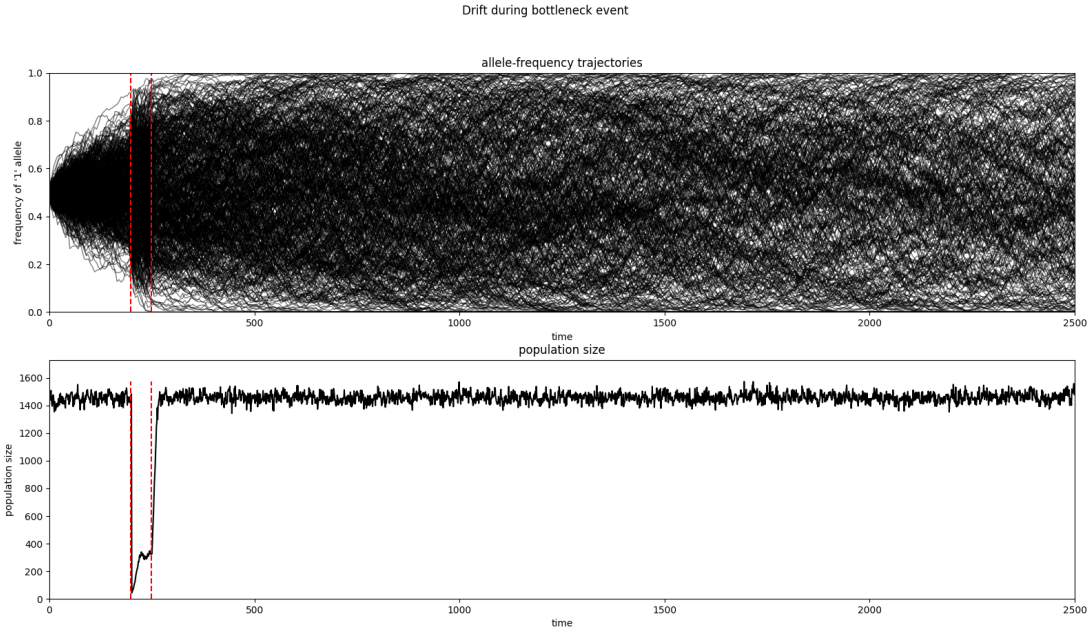

Figure 4: Bottleneck test: 1-allele frequencies (top) and population size (bottom) as a function of time

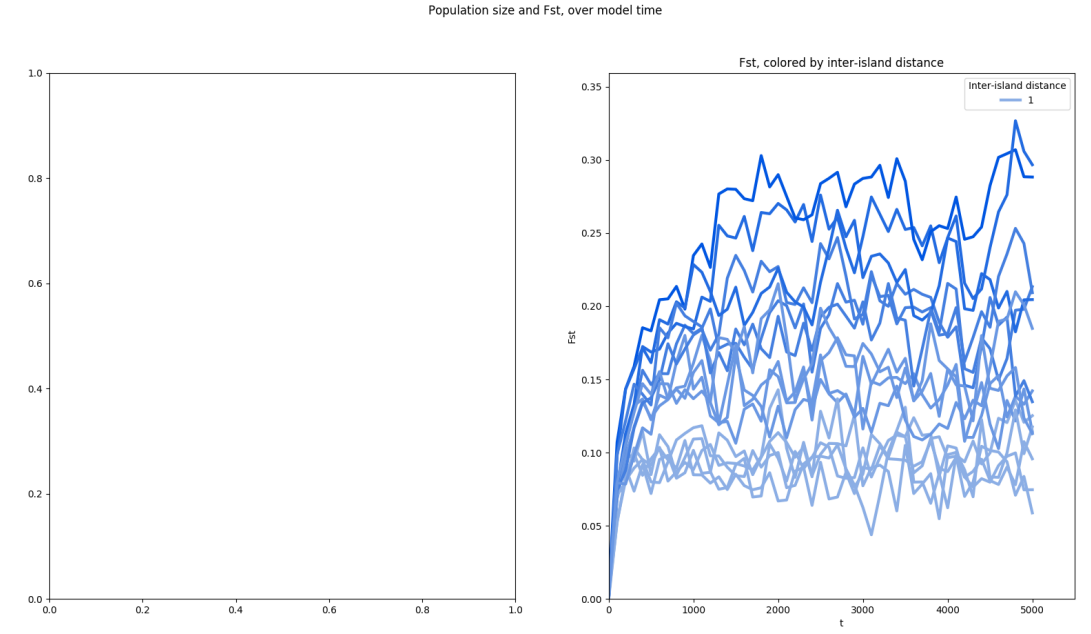

Figure 5: Stepping-stone test: F_{ST} as a function of model time, across increasing inter-island distances (gradually darker shades of blue)

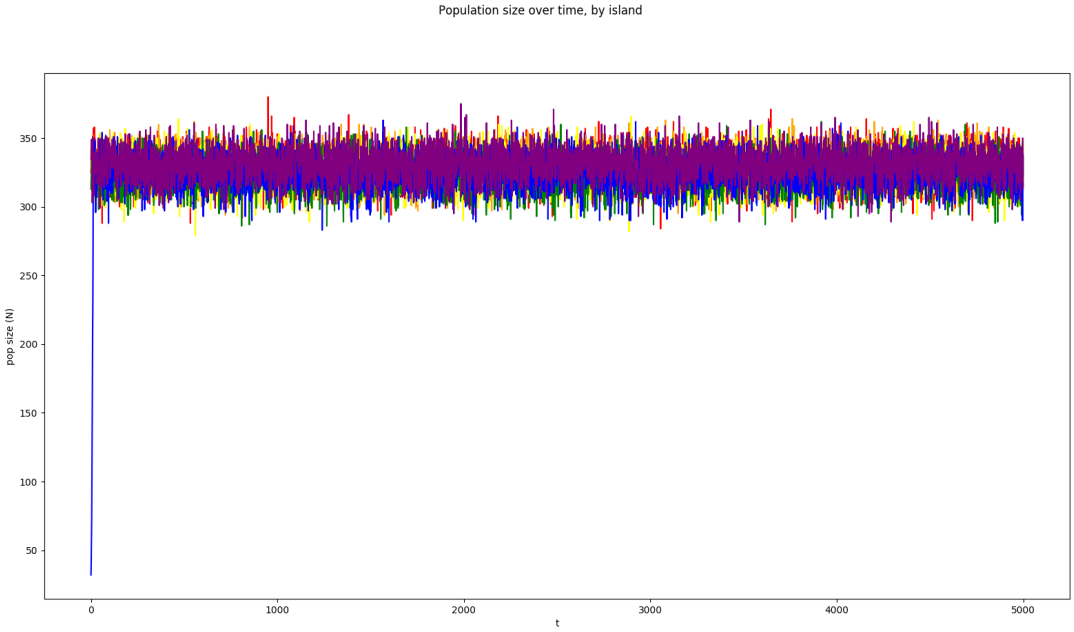

Figure 6: Stepping-stone test: Population size as a function of time, for all 6 islands' populations

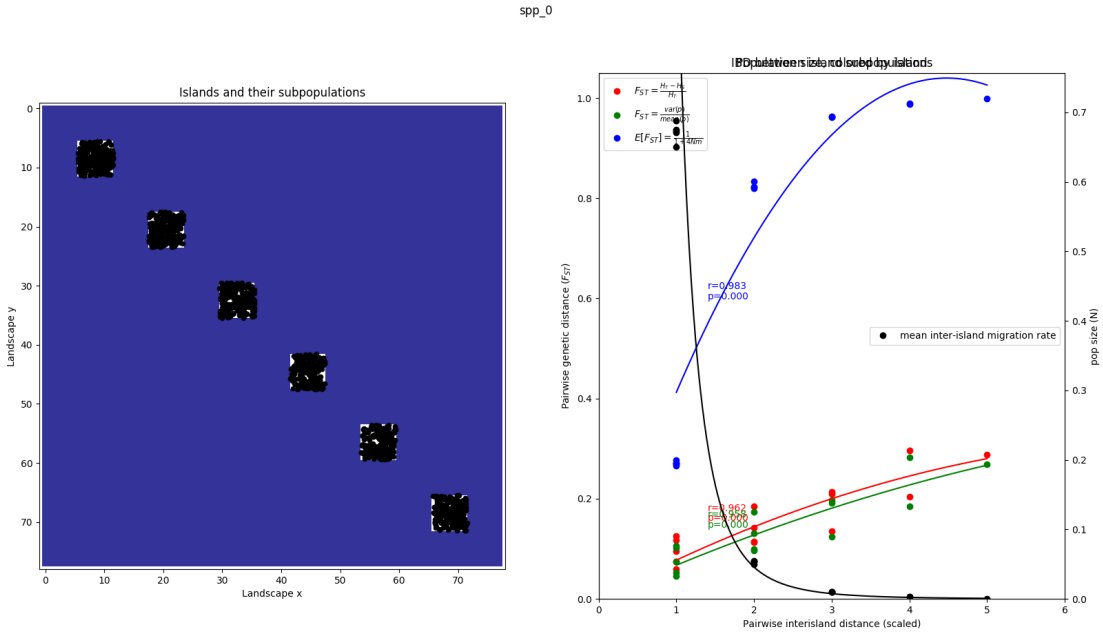

Figure 7: Stepping-stone test: left: Map of all 6 islands' populations at the end of the simulation; right: pairwise F_{ST} (left y-axis; calculated by 3 different formulae) and inter-island migration rate (right y-axis) as a function of inter-island distance

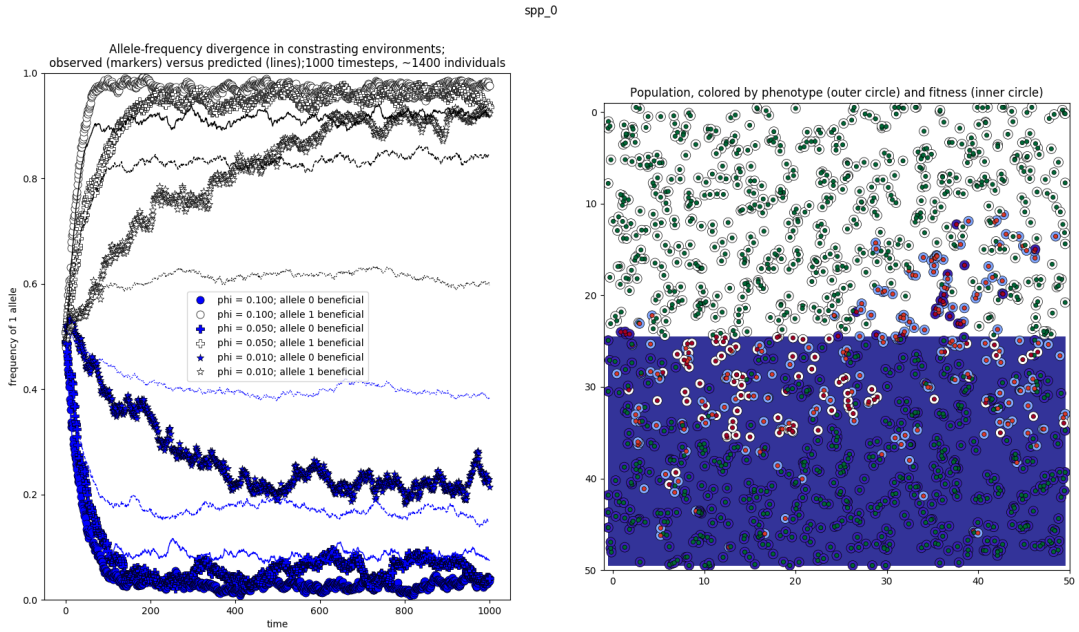

Figure 8: Contrasting-habitat test: left: Observed (markers) versus predicted (lines) allele-frequency trajectories for two contrasting habitats (blue = 0.0-valued; white = 1.0-valued), across three selection coefficients ($\phi = 0.01$: stars; 0.05: crosses; 0.10: circles); right: map of the population after spatially divergent selection at $\phi = 0.10$, with individuals, colored by phenotype (outer circles) and fitness (inner circles), plotted on top of the selective landscape layer (horizontally divided into white and blue halves)

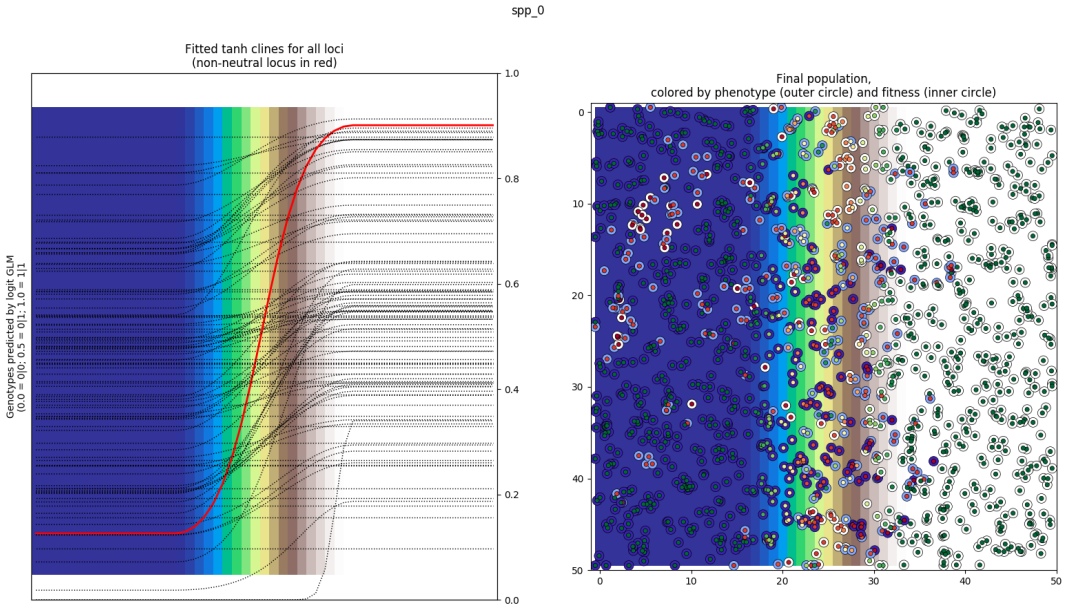

Figure 9: Cline test: left: plot of allele-frequency clines (neutral loci in black, selective locus in bold red) against the selective landscape layer (horizontal gradient from blue to white); right: map of the final population on top of the selective lanscape layer, with individuals colored by phenotype (out circles) and fitness (inner circles)

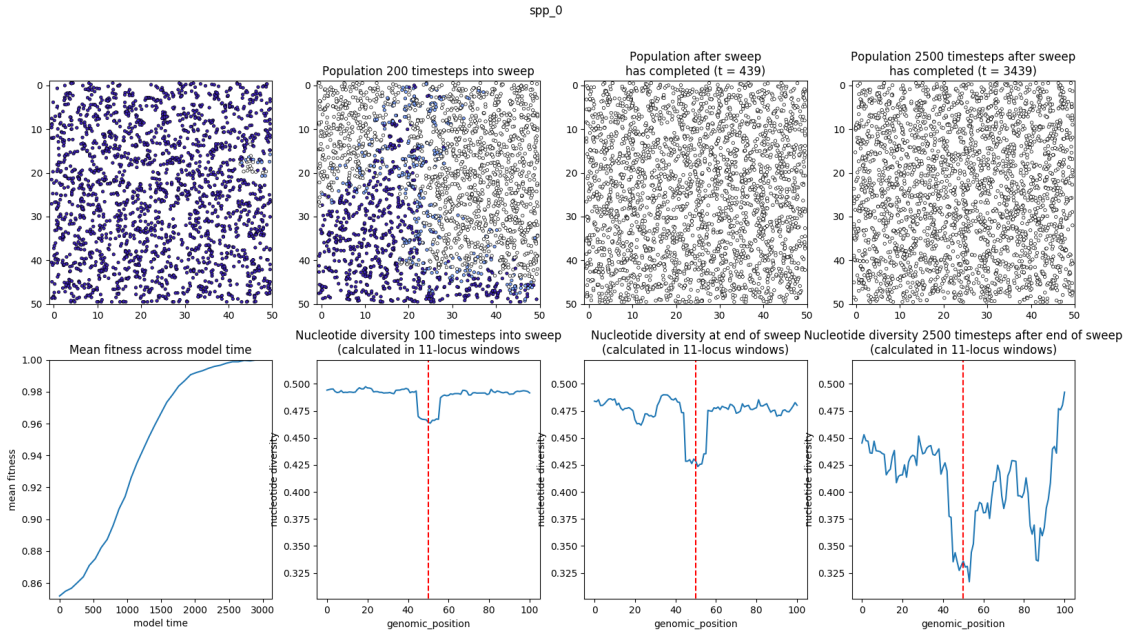

Figure 10: Selective sweep test: top row: Maps of population (colored by phenotype) at various points in model time (from left to right: timestep 0, timestep 200, after completion of sweep, and 2500 timesteps after completion of sweep); bottom row: mean fitness as a function of model time (first plot on left) and genome-wide nucleotide diversity at timestep 200, immediately after completion of sweep, and 2500 timesteps after completion of sweep (second, third, and fourth plots from left)

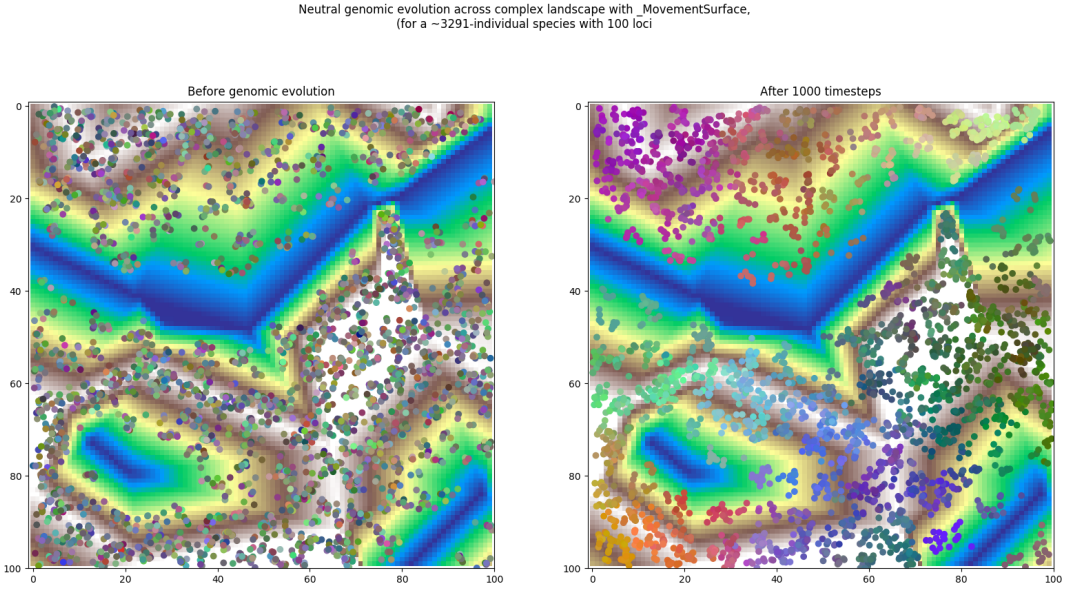

Figure 11: Spatial structure in a species evolving on a complex landscape layer serving as a movement surface, before (left) and after (right) 1000 timesteps of neutral evolution. Individuals' colors are derived from their values for the first three PCs of a genetic PCA (each PC scaled to $0 \leq value \leq$, then used to assign RGB values)

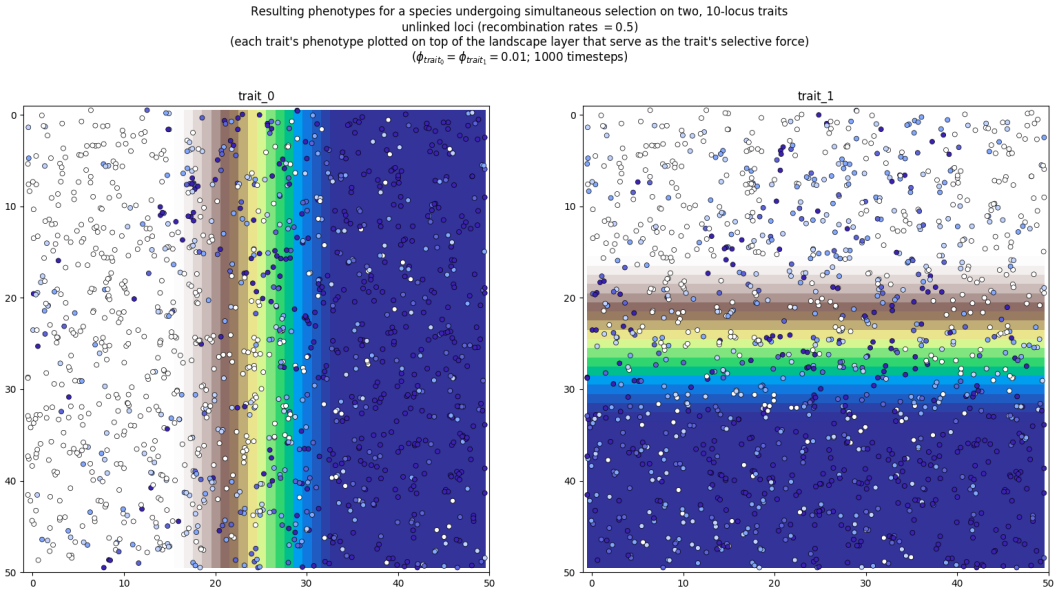

Figure 12: Results of simultaneous selection on two traits with divergent maps of selective force. Each trait has 10 unlinked loci and a selection coefficient of $\phi = 0.05$

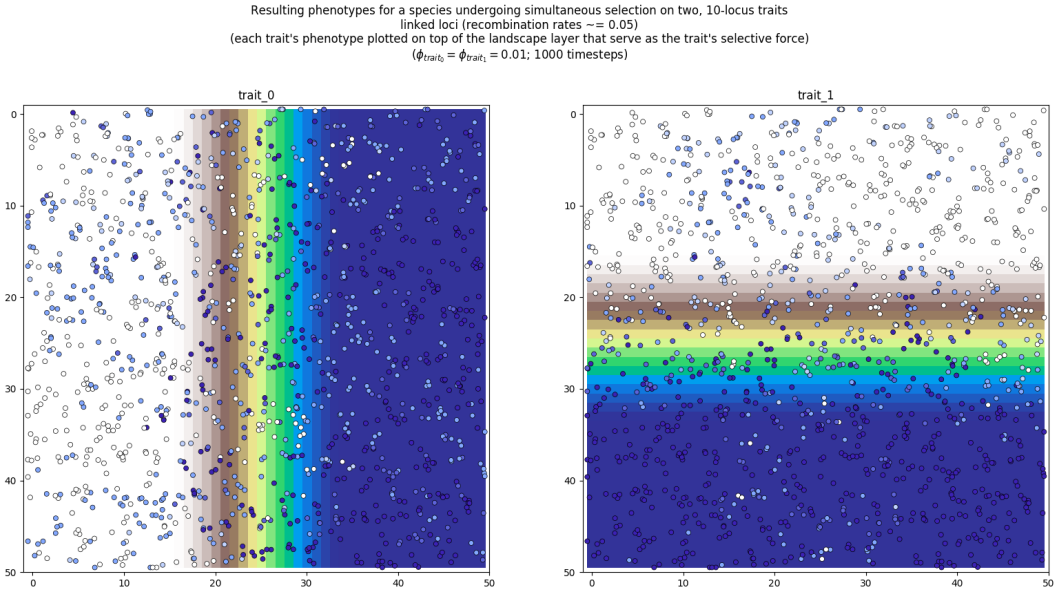

Figure 13: Results of simultaneous selection on two traits with divergent maps of selective force. Each trait has 10 linked loci (recombination rate = 0.05) and a selection coefficient of $\phi = 0.05$

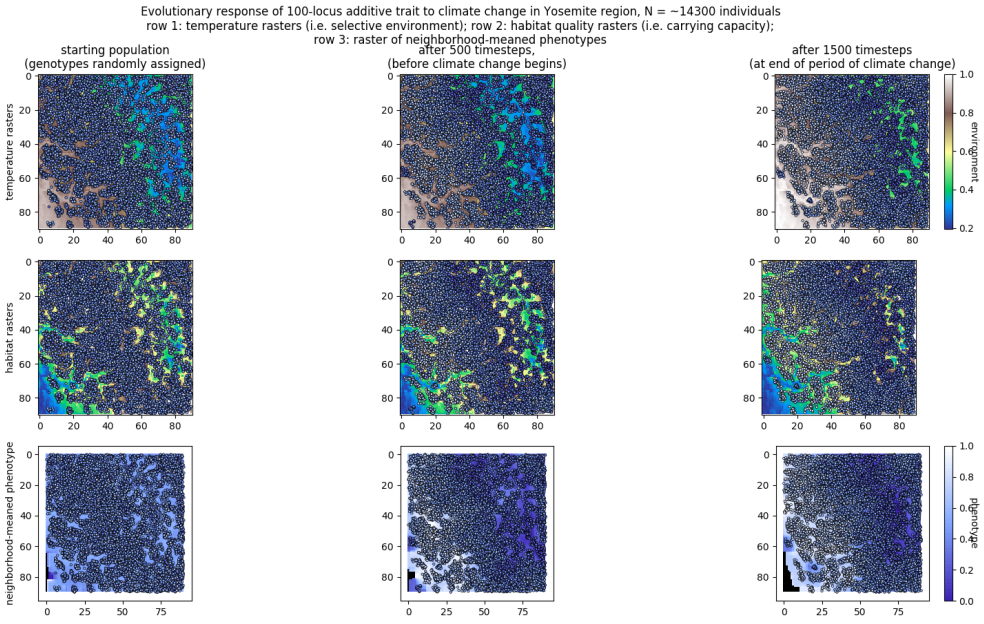

Figure 14: Temperature rasters (top row), habitat rasters (middle row), rasters of neighborhood-meaned phenotype (bottom row) at timesteps 0 (left column), 500 (before beginning of climate change; center column), and 1500 (after climate change; right column) for a species with a 100-gene trait adapted to temperature