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

Drew Hart, Ian Wang

**TODO**

* **continue hammering out the Yosemite example and plotting (maybe include elev as proxy var, to improve SDM?)**
* **fix other minor details in main figures**
* **write abstract**
* **take care of other yellow-highlighted bits**

# Abstract

WRITE ME

# Intro/Background

There is ever-growing interest in understanding and even predicting the genomic evolution of complex study systems on complex and changing landscapes. These systems might include one or multiple populations or species that are not at demographic equilibrium. These species might inhabit complex, multivariate, and even changing landscapes. And they may be undergoing neutral evolution as well as natural selection, often on multiple traits of variable genetic architecture. The field of landscape genomics studies the ways in which ecological and evolutionary processes playing out on real landscapes generate geographical patterns of genomic diversity, Landscape genomics studies frequently feature analysis of data collected from study systems of such genomic and geospatial complexity (Crossley et al., 2017; Harris and Munshi-South, 2017; Lind et al., 2017; Mastretta-Yanes et al., 2018). Study of such systems is crucial for developing evolutionary and ecological theory (Barrett et al., 2019; Pelletier, 2019), improving our knowledge of the past states and present functionality of ecosystems (Mastretta-Yanes et al., 2018), anticipating ecological futures in the Anthropocene (Bay et al., 2018), and informing conservation and management (Crossley et al., 2017; Lind et al., 2017).

The complex genomics of such systems are beyond the reach of analytical population genetics, and their spatial complexity and multifaceted 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 (Peng and Kimmel, 2005), NEMO (Guillaume and Rougemont, 2006) and QuantiNemo (Neuenschwander et al., 2008)), yet incorporate only rudimentary spatial components or none at all. Other programs are designed specifically for landscape-genetic simulations (e.g. CDPOP (Landguth and Cushman, 2010), CDMetaPOP (Landguth et al., 2017), SimAdapt (Rebaudo et al., 2013)), but are limited in their genomic complexity. For instance, many programs are incapable of modeling simultaneous selection on numerous multigenic traits. To our knowledge, SLiM (Haller and Messer, 2017, 2019; Messer, 2013) 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 (National Cancer Institute, 2019). (Indeed, the generalizability and complexity of which SLiM is capable far exceeds that of Geonomics.)

We developed Geonomics to provide a scriptable framework for building complex landscape-genomic simulations with minimal effort. Geonomics models are parameterized by way of an informatively annotated parameters file that provides the user a straightforward means of building models of arbitrary complexity, yet offers reasonable default settings and/or ‘off switches’ for parameters and components that are not the focus the user’s interest. Models consist of 1.) a landscape, composed of one or more environmental layers, and 2.) one or more species, each described by a variety of life-history parameters and a genomic architecture of arbitrary complexity that may be mapped onto one or more traits under selection. Landscape layers may be used to control various aspects of a species’ functionality, including carrying capacity, movement and offspring dispersal, and natural selection). Landscape layers and species can undergo one or more environmental and demographic change events over the course of a model run. At each timestep, a species undergoes movement (in continuous space), mating (using realistic genomics and generating additive phenotypes), mortality (by both density-dependence and selection), and change events (at the timesteps for which they are scheduled). Thus, Geonomics models are non-Wright-Fisher models, in which individuals live and move in continuous space, mating and mortality occur locally, and many species-level phenomena (e.g. population size(s), migration rates between landscape regions) are emergent properties of a model’s parameterization. For this reason, Geonomics produces realistic, spatially explicit, ecological-genomic datasets that should be useful for a wide variety of theoretical and empirical purposes.

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 selection in SLiM 2’s recipe book; e.g. section 14.11, page 288; (Haller and Messer, 2019)), but can be built in Geonomics with an edited parameters file and as little as three lines of code. 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, for a species moving realistically across that landscape, and then run this model an arbitrary number of times, collecting data heterochronously during each run — all of this by doing nothing more than templating a parameters file, making alterations, loading the file, and running the resulting model object.

What’s more, Geonomics is written and run entirely in Python, a broad-purpose and popular programming language that is already familiar to most researchers 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 within the range of reasonable parameter values (see section on runtime analysis, below), we do not expect runtime to be a major constraint for the sorts of models for which Geonomics was designed. And what Geonomics sacrifices in performance, it gains in flexibility, extensibility, and accessibility. The basic user needn’t even know how to write full Python scripts; users can build their own models by recycling and tweaking a minimal amount of code (available within the documentation and at the Geonomics homepage). However, advanced users wishing to code their own extensions or customizations have broad opportunity to do so, because Geonomics is a Python package that seamlessly integrates into the universe of Python functionality.

# Model description

## Components

A Geonomics model consists of two core components: the species and the landscape.

### Species

A species consists of an arbitrary number of individuals, which do not belong to discrete populations but instead are distributed in continuous space upon the landscape. A species is described by a wide variety of 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; 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 demographic and life-history parameters.

Each individual in a species is described by 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 objects called communities, which for most models will consist only of a single species, but which provides a framework for the advanced user to code inter-species interactions for multi-species models; this is a functionality that we hope to build into future versions of the software.)

Each individual has a diploid genome consisting of diallelic loci. These loci can be treated as representing either a contiguous haplotype block or a set of discrete markers, depending on the map of recombination rates assigned to the genome. (For simplicity’s sake, we refer to these loci herein, and in the software generally, as ‘the genome’.) Genomes are initially assigned based on a species’ genomic architecture—an object containing parameters describing all simulated loci. These parameters include the starting allele frequencies and dominance values for all loci, the inter-locus recombination rates across the genome. Separate chromosomes can be modeled by providing a list of chromosome lengths, between which the recombination rate is set to 0.5, ensuring independent assortment.

A genomic architecture can also stipulate any number of traits for a species. Traits can be mono- or multigenic, and are quantitative and continuously valued. 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, which universally decrease an individual’s fitness); and trait-affecting, which influence an individual’s phenotype, with the resulting fitness effect determined by the individual’s local environment. All three mutation types 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 provide a CSV-formatted file defining the architecture locus by locus. For details, see the documentation.)

### Landscape

Aside from the species, the other core component in a Geonomics model is the landscape. A landscape is a stack of an arbitrary number of layers (i.e. environmental variables), each represented by a raster of normalized values (). 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 resistance surface controlling the realistic movement 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 a landscape can be programmed to undergo any number of arbitrarily complex environmental change events during each model run. The changes they produce will in turn affect the simulation by way of any species on the landscape for which that layer plays a role in its population dynamics, movement, dispersal, or natural selection. Each event is defined by the timesteps over which it unfolds, and either the terminal raster of the event (with intermediate rasters being linearly interpolated) or a directory containing the stepwise time-series of rasters pertaining to the event. The latter option makes it extremely easy to simulate evolution on real-world rasters undergoing real, non-linear, spatially heterogeneous environmental change.

## Operations

A Geonomics model can be run for any number of runs, with each run creating its own separate subdirectory of output. At the start of each run, the model is burned in. This is accomplished by iterating timesteps, without genomes and natural selection, until a series of statistical tests for temporal and spatial stability in population size is passed. These tests include a time-lagged t-test of mean population size; an augmented Dickey-Fuller test of population size; and a STILL NEED TO ADD A SPATIAL TEST. Then, if genomes are being used, each individual has its genome randomly drawn according to the species’ genomic architecture, such that the ‘main’ phase of each run begins without population structure. The main phase of the run can run for any number of timesteps. Each timestep is composed of a series of four core operations, some requisite, some optional (see Figure 1 for depictions):

1. **movement** (optional);
2. **mating** (requisite; include mate search, mate choice, offspring creationg, and offspring dispersal);
3. **mortality** (due to the combination of density-dependence [requisite] and natural selection [optional]);
4. **change events** (both environmental and demographic; optional);

Each individual’s age/stage increments at the beginning of each timestep, and data and statstics are saved at the end when necessary.

Movement takes place in continuous space, such that individuals are not arbitrarily restricted to grid cells. 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’. A movement surface is an array of uni- or multimodal Von Mises distributions, derived from a landscape layer that serves as a resistance surface. On a unimodal surface, each cell is assigned a Von Mises distribution that has equal to the direction of the centerpoint of the highest-valued neighboring cell (using an 8-cell neighborhood) and has set by the parameters file. On a multimodal surface, each cell’s mixture distribution is a weighted sum of eight such unimodal distributions, one for each cell in its neighborhood (with weights equal to the neighboring cells’ values divided by their sum). This is, to our knowledge, a novel approach to simulating movement. It generates realistic, anisotropic movment across an environment of heterogenous habitat quality, while avoiding the use of time-consuming neighborhood-querying functions during model runtime (see Figure 2).

Mating pairs are chosen from among all eligible pairs of individuals within the species’ mate-search radius. Eligibility can be based on age and sex. Pair decisions are 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 equal to the species’ mean number of offspring), unless it is set to a fixed value by the user. 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’ that is structurally identical to a movement surface).

Mating is followed by mortality. Deaths are Bernoulli draws, using individual-wise death probabilities that are a combination of the probabilities of death by density-dependence (from a spatialized logistic-growth model) and by natural selection (on any number of traits simultaneously). This is calculated as:

where is individual ’s probability of death by density-dependence and is individual ’s fitness for trait (and only factors in if natural selection is being used). The probability of density-dependent death at location is calculated as:

where is the expected number of deaths at the individual’s location on the landscape; is the population density (expressed as individuals per cell-area) at the location; is the expected number of births at the location; and is the logistic population growth rate at the location. Individual ’s fitness for trait is calculated as:

where is the selection coefficient on trait at the individual’s location; is the environmental value of the layer that serves as the selective force for trait i, defines the curvature of the fitness function for trait , and is individual ’s phenotype for trait . And individual’s phenotype is a result of the additive effects of that individual’s genotypes at all influencing loci—Geonomics does not model epistasis—and is calculated as:

where is the number of loci, is a locus’ effect size, is the individual’s genotype, and is 0 for monogenic traits, 0.5 for polygenic traits.

Environmental and demographic change events can be scheduled to occur over any desired portion of the total time of a run. Each event unfolds through a series of incremental changes, which occur at the timesteps specified in the parameters file. In similar fashion, statistics are calculated and data are collected at timesteps for which they are parameterized.

Geonomics is designed as an object-oriented scripting framework with both basic and advanced 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. The advanced mode allows users to write more detailed scripts that make modifications to the components of their models, or to collect custom data from their model, by calling custom functions before a model is run, between any timesteps, or after a run is complete. (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 operates on their Geonomics objects.)

## Validation

We have run a series of tests to statistically heuristically validate Geonomics’ full range of functionality. All test results matched expectations based on population-genetic and genomic theory. (For details, see Supplements.)

# Applications

## Example 1: Isolation by distance (IBD) and by isolation by environment (IBE)

Many real-world populations inhabit complex, heterogeneous landscapes. Landscape heterogeneity can constrain movement, and thus gene flow, when the probability of an individual moving across each area of a landscape is conditional on that area’s environment. Ecologists commonly use resistance surfaces to describe such movement (McRae et al., 2008). Geonomics allows a user to load resistance surface layers, then use them as the basis for creating movement and/or dispersal surfaces produce realistic movememnt patterns (see Figure 2). In a Geonomics model containing such a landscape, realistically simulated movement should generate a realistic pattern of isolation by distance (IBD): Pairwise genetic distances between individuals, populations, or landscape regions, should be positively correlated with pairwise geographic distances.

Landscape heterogeneity can also underlie processes that generate a pattern of isolation by environment (IBE). This is a pattern in which pairwise genetic distances between individuals, populations, or regions correlate positively with pairwise environmental distances, independently of pairwise geographic distances. Geonomics’ implementation of natural selection allows the selective regime to vary across the landscape (both in its optimal phenotype and its strength of selection). This models a key process that should produce realistic patterns of IBE: Spatially divergent selection.

To test Geonomics’ ability to generate patterns of both IBD and IBE, we constructed a model of a single species (roughly 2450 individuals) undergoing both neutral and selective evolution on a landscape of two layers. The first layer, which serves as the movement surface and the carrying-capacity raster, consists of a central, vertical barrier separating equal-area sides within which movement is unconstrained. The second layer, which serves as the selection layer, consists of two environmental gradients of inverse directionality, one on each side of the first layer’s barrier. As data, we collected the species’ full set of genomes both at the beginning of the model (just after the burn-in) and after the model ran to completion (1000 timesteps). We ran genetic Principal Components Analysis (PCA) on these two datasets. From each PCA, we extracted the first three principal components (PCs), scaled them to , then used the resulting numbers to assign red, green, and blue (RGB) values to each individual. We used each individual’s value for those three values to create a colored scatterplot of the species both before and after evolution. We paired these with plots of the species, colored by their phenotypes for the trait under selection (using Geonomics’ ‘model.plot\_phenotype()‘ method). We then also calculated and plotted correlations of pairwise (Euclidean) geographic and environmental distances with scaled pairwise genetic distances, and used logistic regression (with a complementary log log, or ClogLog, link function) to assess patterns of IBD and IBE.

The results show a clear lack of spatial structure at the outset (Figure 3, row 1), because genomes were randomly drawn and assigned, so the population possessed no spatial structure. Spatial structure develops over evolutionary time (Figure 3, row 2, left) concordant with the development of local adaptation (Figure 3, row 2, right). At the end of the simulation, the species demonstrates a significant signal of both IBD and IBE (Figure 3, row 3).

## Example 2: Simultaneous selection

One of the powerful features of Geonomics is that it can simulate selection on numerous traits simultaneously, with the selection layer for each trait being separately specified. Thus, a species can experience multiple, distinct selective regimes simultaneously. This will prove useful for many landscape simulations, given that many natural systems are hypothesized to be locally adapted to multiple, divergent environmental gradients.

We simulated a scenario in which a population undergoes natural selection along two, orthogonally aligned environmental gradients. Each gradient serves as the selective force (s = 0.05) for a separate trait. Each trait is composed of 10 distinct, unlinked loci, all with effect sizes of 0.1 (i.e. one tenth of the distance between the two opposite, optimal phenotypes).

Results show a clear signal of local adaptation to both gradients, as can be see in maps of the final population colored according to their phenotypes for each of their two traits (Figure 4).

## Example 3: Polygenic adaptation to climate change in the Yosemite region

Geonomics’ full use-value is best demonstrated by walking through a complex, worked example. In this section, we explain a complex evolutionary scenario, explain the steps we used to simulate it, then present the results. (The source code for this example is available within the ./example/yosemite directory of the Geonomics package directory.)

There is growing interest in the evolutionary implications of climate change. Much of this interest focuses on species that are locally adapted along an environmental gradient that is expected to shift under climate change. Climatic shifts can be spatially non-homogeneous, because many cooler, higher-altitude regions are warming more quickly than warmer, low-altitude regions, and are projected to continue doing so (Mountain Research Initiative EDW Working Group, 2015; Rangwala et al., 2013; but see Oyler et al., 2015). Of particular interest in these situations is the ability of species to adapt to changing local conditions. Of the possible species responses to climate change—the others being range shifts, plasticity, tolerance, and local extinction—this is the response about which we know the least.

To show Geonomics’ utility for studying such a system, we simulate the evolutionary response to projected climate change of a continuously distributed, locally adapted species: the montane lizard *Sceloporus graciosus* ("sagebrush lizard"), in the Yosemite region of California (U.S.A.).

We began by preparing the raster layers to be used for the model’s landscape. These include a time-series of temperature rasters and a time-series of habitat-suitability rasters. (Note that both in the preparation of our temperature data and in the production of our SDM we use simplistic assumptions and minimal complexity, because the purpose of this example is only to demonstrate the utility of our software.) All data preparation was done using custom scripts (available in the supplements) in the R statistical programming language (R Core Team, 2019).

The first set of layers consists of current mean annual temperature and its projections through year 2100 (the latest available year). For current temperature we use PRISM data (Daly et al., 2008): 30-year normals (1981-2010), 800-m resolution, PRISM v.14.1. For future years, we use minimum and maximum annual temperature data at 5-year intervals (2015-2100) and 6-km resolution, downloaded from [Cal-Adapt](https://cal-adapt.org/help/faq/). This data consists of climate projections that were downscaled using the Localized Constructed Analogs, downscaling technique (LOCA; Pierce et al., 2014). The data are the minima and maxima observed across 32 LOCA-downscaled global climate models, using a conservative representative concentration pathway (RCP 4.5). We use this data to develop a time-series of future temperature layers at our 800-m current-temperature resolution, according to the following algorithm:

1. calculate the average of the minimum and maximum rasters for the first future year (2015);
2. extract the values of those two rasters for the top-left cell;
3. calculate the mean of those values;
4. find all cells in the 800-m PRISM raster that fall within this cell;
5. for each of those cells, add the difference between its current value and the average projected value to its current value;
6. repeat steps 2. to 5. for all cells in this year’s minimum and maximum rasters, gathering the resulting 800-m cells into a projected, 800-m raster;
7. repeat steps 1. to 6. for all remaining years, gathering the resulting rasters into a projected climate-change time-series;

The second set of layers consists of *S. graciosus* habitat-suitability layers, which we developed from a species distribution model (SDM) using a binomial genearlized linear model (GLM) with a logit link. Our SDM was constructed according to standard theory and practice (Peterson et al., 2011). We downloaded all geo-tagged occurrences of *S. graciosus* from [GBIF](https://www.gbif.org/) (accessed 10:33 a.m. PST on March 5, 2019). For presence data, we clipped those points to the area of California and Nevada, and subsampled to remove multiple points within the same raster cells. Per recommendation for regression-based SDMs, we generated pseudoabsence data by drawing random points in the California-Nevada region, outside of cells where presences were observed (Barbet‐Massin et al., 2012). We extracted the 30-year normals temperature data to these points, which we used as a predictor variable of presence/pseudoabsence in our GLM. We then projected that GLM onto the full current and future temperature rasters of our study region, producing a time-series of predicted habitat suitability across our region (where ). (We did not threshold these rasters for a minimum suitability value. We preferred to allow the product of these rasters and our species’ parameterized carrying-capacity constant to produce, in Geonomics, rasters of continually decreasing carrying capacity that would, in turn, produce a pattern of continually decreasing population density toward the species’ range edges.)

With these rasters prepared, the construction of our Geonomics model was straightforward. We generated a template Geonomics parameters file (using gnx.make\_parameters\_file(…)), then edited it to match our study system. We needed a parameters file that could parameterize: 2 ‘file’-type Layers (i.e. layers red in from files; see the Geonomics documentation for details), both of which would undergo landscape-change; 1 Species, with movement, a movement-surface, and genomes with 1 trait; and data-collection. The Geonomics command we ran to generate a template Geonomics parameters file with those components 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)  
We edited the resulting file, setting the appropriate parameter values to indicate the locations of our starting temperature and habitat-suitability rasters and the locations of our directories of change rasters for those layers. We set other life-history and demographic parameters to reasonable values, based on current knowledge of *S. graciosus* biology (see parameters file, in Supplements, for full details).

Then, in a Python script (also in the Supplements), we used our parameters file to create a Geonomics model object (using model = gnx.make\_model(…)), then ran that model. We used model.walk(…) to run the burn-in, run the main phase for 500 pre-climate change timesteps (to develop a pattern of local adaptation), then run the main phase for an additional 100 timesteps of climate change. The model.walk function is what we refer to as Geonomics’ ‘advanced’ mode above; as opposed to model.run, it allows the user to run the model for only a certain number of timesteps, then pause it to call their own custom code. In our case, we stopped the model at timesteps 500 (before climate change) and 600 (after climate change), each time plotting the landscape layers, the population, and a neighborhood-meaned raster of the population’s phenotypes (using a custom function written in-script).

This produced the results seen in Figure 4. 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 adaptation exhibits a pronounced upslope shift in response to the period of climate change. These results are visible both heuristically (from the individuals’ phenotypes plotted before and after climate change) and analytically (from the neighborhood-meaned phenotype rasters plotted at the same timepoints). Population size is roughly 34,000 before climate change, declining to roughly 28,000 at the end of the simulation.

# Discussion

Both our validations tests (Supplements) and our three example simulations demonstrate that Geonomics is capable of simulating population-genomic and landscape-genomic scenarios of varying complexity, generating accurate and realistic spatial genomic datasets. Importantly, it does this in a software package that multiple benefits unavailable in other similar tools. Geonomics provides a scripting framework embedded in Python, one of the world’s most popular programming languages and a language already familiar to many researchers who use bioinformatics methods. It makes the creation of arbitrarily complex models quick and easy, without even requiring much Python experience. Simultaneously, it provides advanced users with easy access to the model objects and their components, enabling broad customization and extension.

We believe Geonomics will be useful for many purposes. Many theoretical questions in population genomics necessitate explicitly spatial study methods, which Geonomics will make more tractable than ever before. Landscape genomics studies draw conclusions about complex, real-world systems, sometimes with directly implications for conservation and management. Yet these results are often obtained in the absence of adequate evolutionary null models – as opposed to the statistical null models used in outlier tests, for example – and thus without sufficient knowledge about the validity and power of common methods. Geonomics will not only enable the generation of simulated datasets specific to certain study systems—parameterized from field data or from the literature, and simulated under multiple, competing hypothetical scenarios. It will also aid the development and testing of landscape-genomic analytical methods, strengthening our ability to draw accurate and actionable inferences from real-world data.

Of course, all tools have limitations and caveats. These are discussed in the following two sections.

## Runtime and memory

As mentioned above, Geonomics models run considerably slower than models such as SLiM (Haller and Messer, 2017, 2019; Messer, 2013) that are written in compiled languages. However, for users whose scenarios are well served by the design and affordances of Geonomics, we believe that what is sacrificed in runtime will be made up for in flexibility, customizability, and ease of use. With a reasonably powerful computer, we believe that most users will not find runtime a major limitation. (Indeed, our three examples were all all run on a modern laptop with 8Gb of RAM and an Intel® Core™ i5-8250U quad-core processor. They ran in an average of 270 s for the IBD/IBE model, 135 s for the simultaneous-selection model, and 5400 s (1.5 hr) for the Yosemite model. These are all overestimates of the true runtime for just the models, because these figures include the non-negligible runtime of code used to calculate and produce data used for plots.)

Given the complexity of Geonomics and the number of parameters a user can modify, there are numerous parameters and parameter-combinations which could have a significant effect of a model’s average runtime. It would be impossible to summarize them all here. However, we provide a rudimentary runtime analysis (Figure 6), run on the same 8Gb, quad-core laptop as the examples above. This analysis highlights some basic parameters that are likely to influence a model’s average runtime per timestep. These are:

‘**K\_factor**’: The factor by which a species’ carrying capacity raster is multiplied to determine its local carrying capacities and thus, in sum, its mean total population size;

‘**n\_births\_distr\_lambda**’: The parameter on the Poisson distribution from which the number of offspring per mating event is drawn;

‘**dim**’: The dimensions of the simulated landscape;

‘**L**’: The length of the simulated genome;

(It is worth noting that runtime increases with landscape dimension both because of functions whose runtimes scale with the landscape size directly and because of functions whose runtime scales with total population size—which can estimated as the sum of the carrying-capacity raster, the size of which always matches the landscape size. So, the difference between the ‘K\_factor’ and ‘dim’ lines in Figure 6 can be taken as an indication of the runtime cost of landscape dimension above and beyond population-size effects.)

Memory usage could likewise pose real limitations for some users modeling some systems. However, we believe it is unlikely to do so for most scenarios within the range of parameterizatons that are also unlikely to pose runtime limitations. There are, however, two potential memory limitations of special concern. These relate to the topics discussed in the following section, so are explained therein.

## Caveats

Geonomics uses two unconventional approximations—to make complex models tractable in a slower language within reasonable compute-time—which are worth special consideration. The first is the approximation used to model recombination. Precise modeling of recombination between all neighboring loci for each gamete produced would require an extremely large and time-consuming number of Bernoulli draws during a model run. To avoid this, when a model is first created, Geonomics generates and saves (as binary arrays) a large collection of recombination paths (the number of which can be parameterized by the user). Each path is just a genome-length array containing ones at interlocus-positions where recombinations occur and zeros elsewhere. The path can then quickly be used to subset an individual’s genome to produce a gamete. As a model runs, these paths are repeatedly shuffled and drawn through, like a deck of cards, as gametes are created.

In this way, Geonomics approximates free and potentially heterogeneous recombination. (Importantly, for long genomes and large numbers of paths, memory limitations will occur, because the data structure containing the paths is essentially a two-dimensional binary array whose size is the product of these values.) For genomic architectures with homogeneous recombination rates, Geonomics provides the option to use an alternative recombination mechanism, which precisely simulates recombination on the fly but does so at a cost of increased average runtime per timestep. (For details, see the documentation.)

We feel that this is a reasonable approach that poses little risk of generating systematic artefacts. It balances the needs of the system being modeled with the constraints of the interpreted language being used. However, the user should be aware of one concern: The total number of recombination paths generated determines the minimum recombination rate that can be modeled (because all loci with non-zero recombination rates are forced to have at least one recombination event among the set of paths generated) and the precision to which recombination rates are approximated and at which near rates can be differentiated. (For example, the smallest recombination rate that can be modeled when 10000 recombination paths are used is , such that a locus with a nominal recombination rate of would be inaccurately modeled as recombining with a probability of , and any two locations with recombination rates less than different will necessarily be modeled as having identical rates.) When necessary, Geonomics raises relevant warnings about this at the time a model is created.

The other approximation of which users should be aware is that of directional movement across a movement or dispersal surface. Conceptually, a movement or dispersal surface is an array of Von Mises distributions (as explained in the ’Operations’ section, above). Each distribution in a surface is represented by an approximation vector of direction draw, which is drawn once when a model is built. Random draws are then made from these vectors during model runs. This increases computational efficiency by avoiding large numbers of calls to random number generators during runtime. But it means that movement and dispersal surfaces, rather than being two-dimensional grids of random-number functions, are actually numpy arrays, where and are the landscape dimensions and is the length of the approximation vectors (a value controlled from the Geonomics parameters file).

Clearly, the accuracy of these approximations of the true, continuous distributions they are drawn from is a function of the length of the approximation vectors. This length will usually not be so constrained that it significantly impacts the accuracy of these approximations. Nonetheless, users may wish to check accuracy, which they can do by using built-in functions that visualize the composition and behavior of movement and dispersal surfaces.

Notably, a memory limitation could occur if the landscape layer that serves as the basis for a movement or dispersal surface also undergoes environmental change. In this case, the movement surfaces corresponding to each step of this change event will be created and stored when the model is first created. The series of three-dimensional arrays that would need to be stored upfront for a large landscape with long approximation vectors and a many-step environmental change event will demand a lot of memory. One way to handle this problem would be to decrease the temporal resolution of the environmental change event. (For details, see the documentation.)

# Getting started

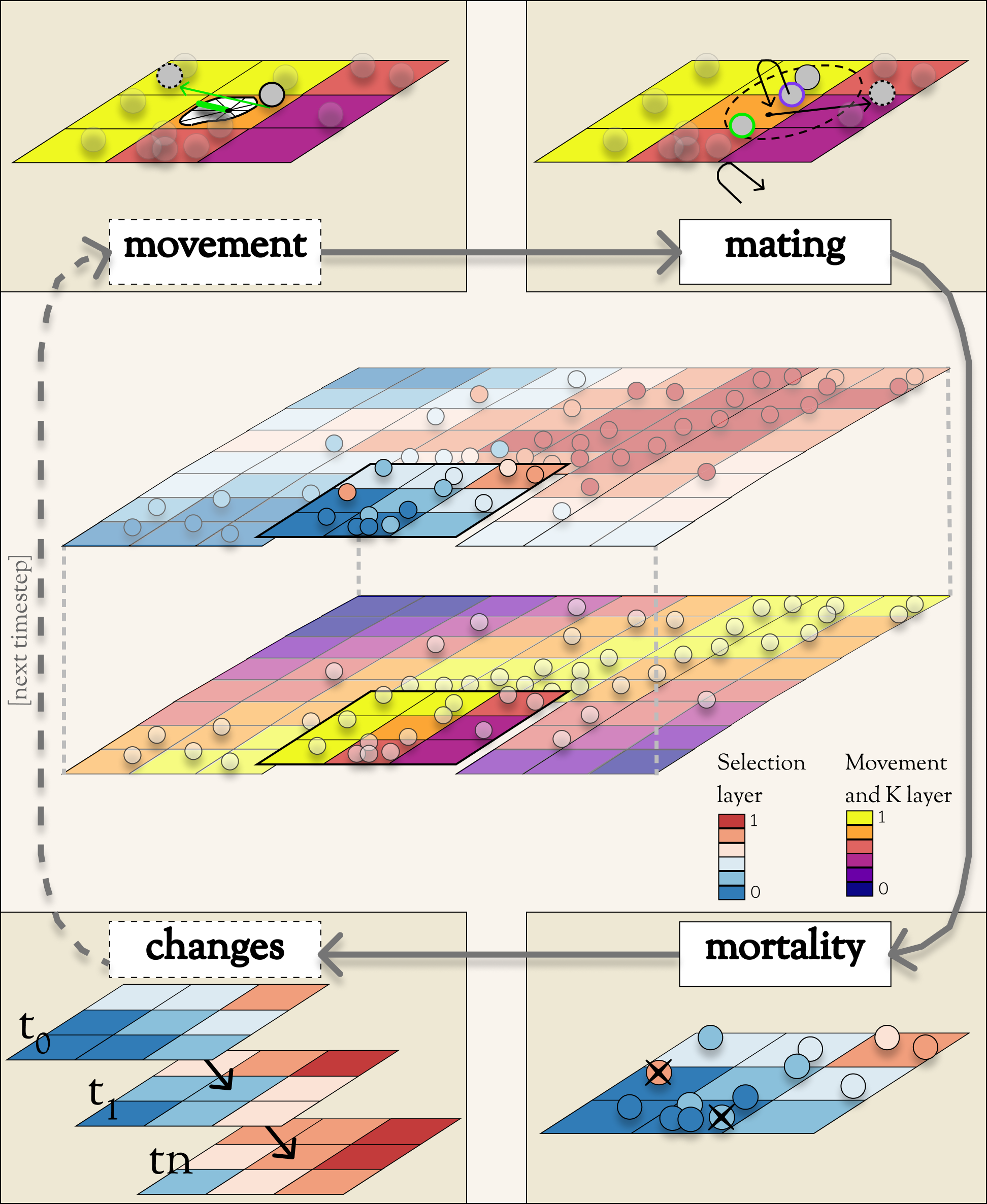
For those interested in using Geonomics, the simplest way is to install via pip or conda. (Note that Geonomics only uses common, well-established Python packages as required dependencies: Numpy (Oliphant, 2006; van der Walt *et al.*, 2011), Matplotlib (Hunter, 2007), Pandas (McKinney, 2010), Shapely (Gillies *et al*., 2007), Scipy (Jones *et al.*, 2001), Scikit-learn (Pedregosa *et al., 2011*), Statsmodels (Seabold and Perktold, 2010), PyVCF (Dougherty *et al*., 2011), and Bitarray (Schnell 2011).) The source code is also publicly available on GitHub ([URL\_HERE](HART_Geonomics_methods_paper_DRAFT.odt/URL_HERE)), where it is actively maintained and developed.

The documentation is comprehensive (and will continue to expand as new worked examples are offered and new functionality is added to the package), and is available [online](http://htmlpreview.github.io/?https://github.com/drewhart/geonomics_docs/blob/master/built/doc.html). The documentation includes much greater technical detail on all components of the model, including a section explaining the meaning, default values, and usage for every parameter in a Geonomics parameter file. It also includes a number of worked examples, from the most basic to the more advanced. Details about the usage of each function are available both in the documentation and in the functions’ docstrings (the latter of which can be accessed by by calling Python’s help() function on a function of interest).

# Conclusion

Geonomics is a Python package designed to make it simple and quick to create and run complex landscape-genomic models. At the same time, it provides a flexible scripting framework that will allow advanced user to customize and extend its functionality. (Indeed, the package is designed with the intention of adding extensions and new functionalities into future versions.) We believe Geonomics will prove highly useful for theoretical, empirical, methodological, and applied research in population genomics, molecular ecology, global change biology, and conservation.

# Figures



*Figure 1:* ***Geonomics conceptual flow diagram****: In the center is a species distributed across a multi-layer landscape---in this case, composed of two layers. The upper layer serves as a selection layer for a trait, with both the environmental (i.e. cell) values and individuals' trait-phenotype (i.e. circle) values indicated by the blue-red colormap. The lower layer serves as both a movement layer (which would be the basis for a movement surface) and a carrying-capacity (K) layer. Arranged in series around the landscape is a clockwise flow-diagram of the major operations that occur during each timestep. Each operation is depicted in its own labeled box (with dashed-line label-boxes indicating optional operations).* ***Movement****: An individual is displaced to a new location along a randomly drawn vector, composed of a random distance (drawn from a Wald distribution, which is not depicted) and a random direction (drawn from a circular distribution, which could be uniform, unimodal, or, in this case, a multimodal Von Mises mixture distribution determined by the relative permeability of the 8-cell neighborhood of the movement layer).* ***Mating****: An individual (outlined in purple) randomly chooses a mate (outlined in green) from among all potential mates (solid gray individuals) located within its mating radius (green dashed circle on the landscape). Both mates produce gametes, composing the genome of an offspring individual, which disperses from its parents' midpoint (black dot on the landscape) along a randomly drawn dispersal vector. (Dispersal vectors are drawn akin to, but independently of, movement vectors.)* ***Mortality****: Individuals' probabilities of death are a product of density-dependence (determined by the comparison between local density and local carrying capacity, following a spatialized logistic growth model) and selection (with fitness determined by the absolute difference between phenotypic values and environmental values for all pairs of trait and selection-layer). Death events are Bernoulli draws on these probabilities.* ***Changes****: Change events are composed of n stepwise changes, which occur at the ends of their scheduled timesteps (here depicted as ‘t1, t2, …, tn). These can include both environmental changes (i.e. substitutions of the rasters for certain layers, as depicted here) and demographic changes (not depicted).*

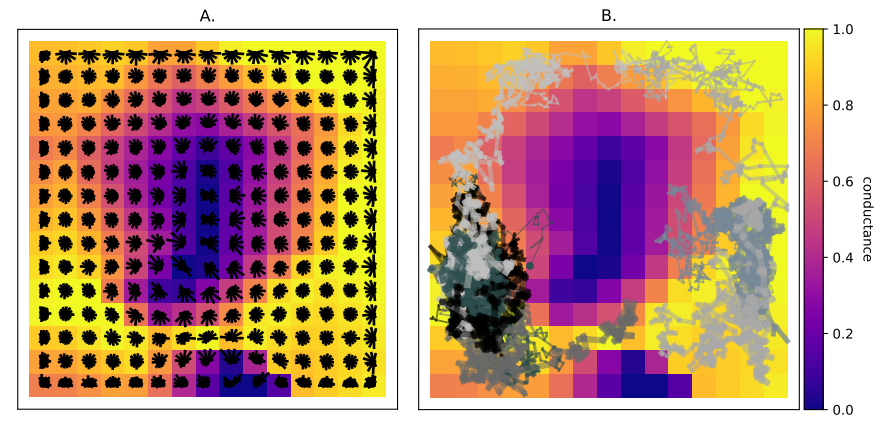
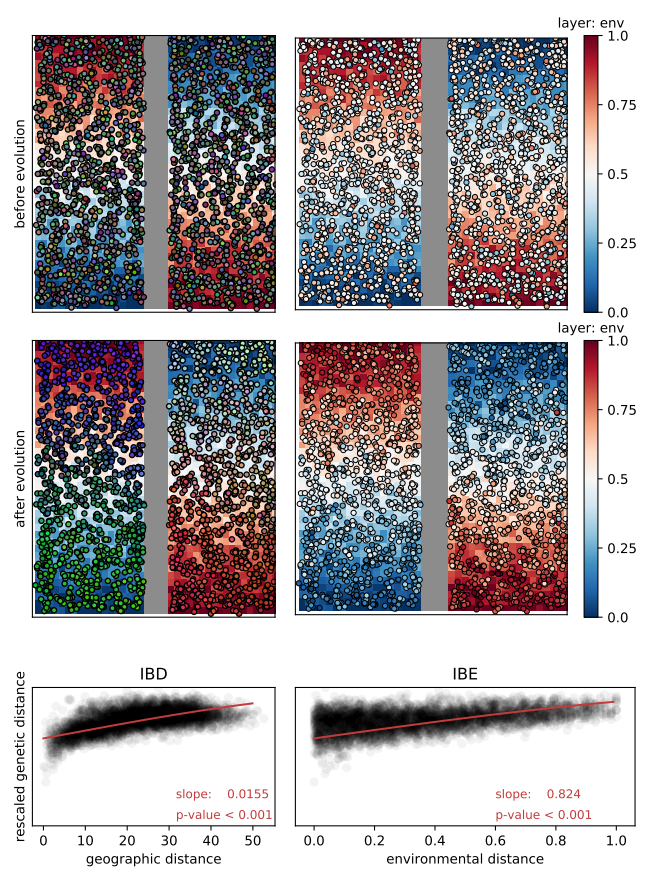


Figure 2: **Movement surface structure (A) and function (B):** A: Circularized histograms of the VonMises mixture-distribution approximations underlying a movement surface, plotted over a 15x15 landscape layer on which the surface is based. The taller the bar in a histogram (i.e. the further from the histogram’s center), the higher the probability that an individual located on that histogram’s cell would make a random draw that moves it in a direction within the angular neighborhood of that bar. (This plot was produed by the Geonomics method model.plot\_movement\_surface.) B: Examples of movement tracks for 25 individuals randomly selected from a species, moving across the movement surface depicted in A. Each track is 150 steps long, beginning at the individuals’ starting locations (plotted as dots) and thickening with each increasing timestep. Individuals’ preferential movement toward higher-suitability regions of the landscape (values with environmental values nearer 1) is evident. Occasional migrations between relatively isolated portions of the landscape are also visible. (This plot was produced using the Geonomics function gnx.help.param\_help.plot\_movement.)



*Figure 3:* ***Isolation by distance (IBD) and isolation by environment (IBE):*** *A species evolving on a landscape consisting of 1.) a barrier layer that serves as the species’ movement surface (displayed as a gray band down the landscape’s center), and 2.) an environmental layer that serves as the selective surface for a 10-locus trait (displayed as the red-to-blue cell values of the landscape). Top row: The population before evolution began, colored by genetic distance (right; genetic-distance colors are derived from individuals’ scores on the first three PCs of a genetic PCA, with each score scaled to 0 ≤ score ≤ 255, then used to assign RGB values) and by phenotype (right; the most-fit individuals would be those whose phenotypic colors perfectly match the cells over which they are located). Middle row: Same information as the top row, but plotted after 1000 timesteps of evolution. Bottom row: Scatter plots of inter-individual pairwise genetic distance (Euclidean distance in genetic PC space, scaled to 0 ≤ distance ≤ 1) as a function of Euclidean geographic distance (left; IBD) and Euclidean environmental distance (right; IBE); slopes and p-values correspond to the red curves, which were fitted as binomial GLMs with a ClogLog link function (to allow a saturating functional form).*

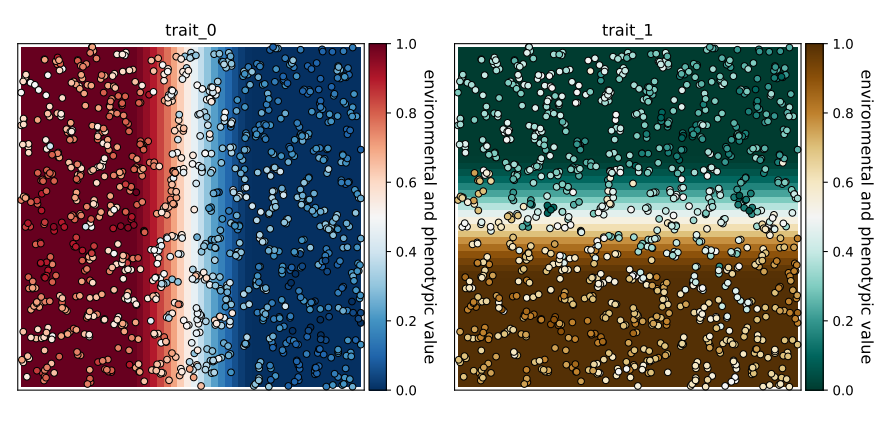
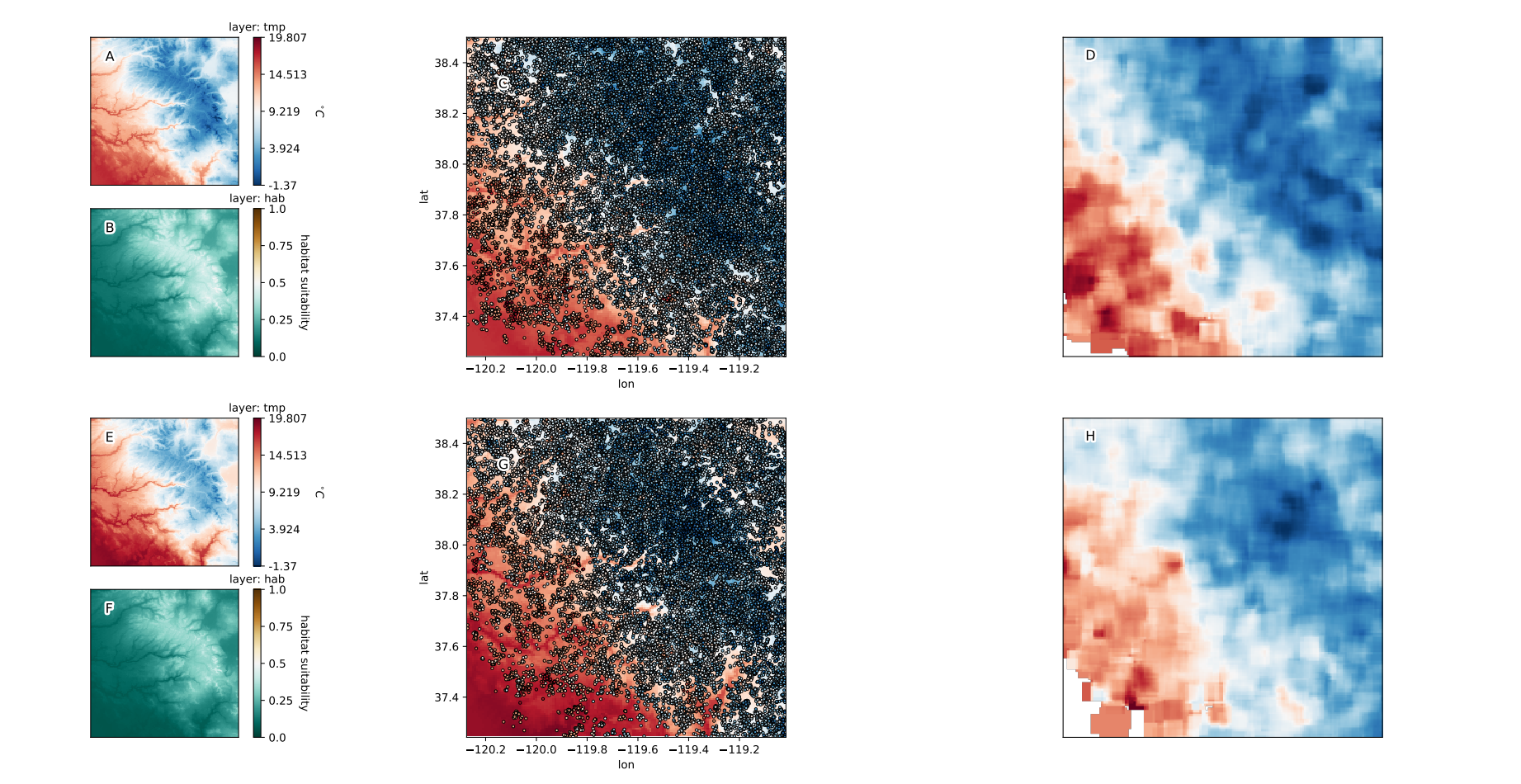


Figure 4: **Simultaneous selection**: Results of simultaneous selection on two traits with spatially distinct selective regimes. Each trait is controlled by 10 unlinked loci and has a selection coefficient of . Individuals are colored by phenotype, such that fitter individuals are those whose colors are more similar to their background environmental colors. (Plots were produced by the Geonomics method model.plot\_phenotype.)

Figure 5: **Polygenic adaptation to climate change in the Yosemite region:** Shown are the temperature rasters before (A) and after (E) climate change; the habitat-suitability rasters before (B) and after (F); the locally adapted population plotted on top of the temperature raster both before (C) and after (G), and neighborhood-meaned rasters of the population’s phenotypes, both before (D) and after (H). (Plots in the left and middle columns were produced using the Geonomics methods model.plot and model.plot\_phenotype respectively.)

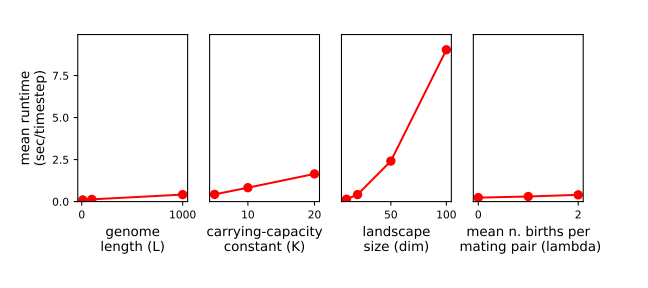


Figure 6: Runtime analysis: Average runtime per timestep (in seconds), as a function of various parameters. Left to right: ‘**genome length (L)**’: The length of the simulated genome; **‘carrying-capacity constant (K)’**: The factor by which a species’ carrying capacity raster is multiplied to determine its local carrying capacities and thus, in sum, its mean total population size; **‘landscape size (dim)’**: The dimensions of the simualted landscape; **‘n\_births\_distr\_lambda**’: The parameter of the Poisson distribution from which the number of offspring in each mating event is drawn.

# Works Cited

Barbet‐Massin, M., Jiguet, F., Albert, C.H., and Thuiller, W. (2012). Selecting pseudo-absences for species distribution models: how, where and how many? Methods Ecol. Evol. *3*, 327–338.

Barrett, R.D.H., Laurent, S., Mallarino, R., Pfeifer, S.P., Xu, C.C.Y., Foll, M., Wakamatsu, K., Duke-Cohan, J.S., Jensen, J.D., and Hoekstra, H.E. (2019). Linking a mutation to survival in wild mice. Science *363*, 499–504.

Bay, R.A., Harrigan, R.J., Underwood, V.L., Gibbs, H.L., Smith, T.B., and Ruegg, K. (2018). Genomic signals of selection predict climate-driven population declines in a migratory bird. Science *359*, 83–86.

Crossley, M.S., Chen, Y.H., Groves, R.L., and Schoville, S.D. (2017). Landscape genomics of Colorado potato beetle provides evidence of polygenic adaptation to insecticides. Mol. Ecol. *26*, 6284–6300.

Daly, C., Halbleib, M., Smith, J.I., Gibson, W.P., Doggett, M.K., Taylor, G.H., Curtis, J., and Pasteris, P.P. (2008). Physiographically sensitive mapping of climatological temperature and precipitation across the conterminous United States. Int. J. Climatol. *28*, 2031–2064.

Guillaume, F., and Rougemont, J. (2006). Nemo: an evolutionary and population genetics programming framework. Bioinformatics *22*, 2556–2557.

Haller, B.C., and Messer, P.W. (2017). SLiM 2: Flexible, Interactive Forward Genetic Simulations. Mol. Biol. Evol. *34*, 230–240.

Haller, B.C., and Messer, P.W. (2019). SLiM 3: Forward Genetic Simulations Beyond the Wright–Fisher Model. Mol. Biol. Evol. *36*, 632–637.

Haller, B.C., and Messer, P.W. SLiM: An Evolutionary Simulation Framework. 517.

Harris, S.E., and Munshi-South, J. (2017). Signatures of positive selection and local adaptation to urbanization in white-footed mice ( *Peromyscus leucopus* ). Mol. Ecol. *26*, 6336–6350.

Landguth, E.L., and Cushman, S.A. (2010). cdpop: A spatially explicit cost distance population genetics program: COMPUTER PROGRAM NOTE. Mol. Ecol. Resour. *10*, 156–161.

Landguth, E.L., Bearlin, A., Day, C.C., and Dunham, J. (2017). CDMetaPOP: an individual-based, eco-evolutionary model for spatially explicit simulation of landscape demogenetics. Methods Ecol. Evol. *8*, 4–11.

Lind, B.M., Friedline, C.J., Wegrzyn, J.L., Maloney, P.E., Vogler, D.R., Neale, D.B., and Eckert, A.J. (2017). Water availability drives signatures of local adaptation in whitebark pine ( *Pinus albicaulis* Engelm.) across fine spatial scales of the Lake Tahoe Basin, USA. Mol. Ecol. *26*, 3168–3185.

Mastretta-Yanes, A., Xue, A.T., Moreno-Letelier, A., Jorgensen, T.H., Alvarez, N., Piñero, D., and Emerson, B.C. (2018). Long-term in situ persistence of biodiversity in tropical sky islands revealed by landscape genomics. Mol. Ecol. *27*, 432–448.

McRae, B.H., Dickson, B.G., Keitt, T.H., and Shah, V.B. (2008). USING CIRCUIT THEORY TO MODEL CONNECTIVITY IN ECOLOGY, EVOLUTION, AND CONSERVATION. Ecology *89*, 2712–2724.

Messer, P.W. (2013). SLiM: Simulating Evolution with Selection and Linkage. Genetics *194*, 1037–1039.

Mountain Research Initiative EDW Working Group (2015). Elevation-dependent warming in mountain regions of the world. Nat. Clim. Change *5*, 424–430.

Neuenschwander, S., Hospital, F., Guillaume, F., and Goudet, J. (2008). quantiNemo: an individual-based program to simulate quantitative traits with explicit genetic architecture in a dynamic metapopulation. Bioinformatics *24*, 1552–1553.

Oyler, J.W., Dobrowski, S.Z., Ballantyne, A.P., Klene, A.E., and Running, S.W. (2015). Artificial amplification of warming trends across the mountains of the western United States. Geophys. Res. Lett. *42*, 153–161.

Pelletier, F. (2019). Testing evolutionary predictions in wild mice. Science *363*, 452–453.

Peng, B., and Kimmel, M. (2005). simuPOP: a forward-time population genetics simulation environment. Bioinformatics *21*, 3686–3687.

Peterson, A.T., Soberón, J., Pearson, R.G., Anderson, R.P., Martínez-Meyer, E., Nakamura, M., and Bastos Araújo, M. (2011). Ecological niches and geographic distributions (Princeton, N.J).

Pierce, D.W., Cayan, D.R., and Thrasher, B.L. (2014). Statistical Downscaling Using Localized Constructed Analogs (LOCA)\*. J. Hydrometeorol. *15*, 2558–2585.

Rangwala, I., Sinsky, E., and Miller, J.R. (2013). Amplified warming projections for high altitude regions of the northern hemisphere mid-latitudes from CMIP5 models. Environ. Res. Lett. *8*, 024040.

Rebaudo, F., Le Rouzic, A., Dupas, S., Silvain, J.-F., Harry, M., and Dangles, O. (2013). SimAdapt: an individual-based genetic model for simulating landscape management impacts on populations. Methods Ecol. Evol. *4*, 595–600.