Multiple-Peak Selection Surface Simulation Methods

Diogo, Miudo, Gabriel, (não sei mais quem)

24/06/2020

Summary

Applying the dynamics predicted by the Lande equation, we use numeric simulations to investigate how a population with a fixed G-matrix evolves on a complex fitness landscape. We show that under certain conditions related to the number and position of the adaptive peaks, the G-matrix can alter the macroevolutionary history of a population. In particular, in a multiple peak landscape, the final phenotype of the population can be altered by G-matrix, and this effect produces an observable signature in the divergence between species.

Introduction

This document is a supporting information annex to "Bridging the gap...". Here, we describe in detail the simulations used to study the interaction between differing levels of genetic integration and complex selective surfaces. Our objective here is to study how these two parameters, genetic variation and selective surface, affect the relation between vector of phenotypic change and the genetic covariation in the population. There are two main elements to the simulations: (1) The level of integration in the genetic covariance matrix of the population under selection; and (2) the ruggedness, or number of peaks, of the selective surface. We begin by developing a strategy to simulate simple and complex selective surfaces.

The selective surface

We are interested in creating both very simple selective surfaces, with only a single phenotypic optimum for the population, and complex surfaces, with several optima distributed across the phenotypic space. In order to achieve this, we use a mixture of Gaussian functions. For the single peak landscape, we can simply sample a multivariate mean θ from some random distribution and use the diagonal matrix (Σ) as a covariance for the multivariate normal[1]. This induces the following mean fitness surface, for a k-dimensional phenotype x:

[1]The Σ parameter could be any positive definite matrix, but we do not explore this parameter here, and use the identity matrix in all simulations.

$$\overline{W}(x) = \mathcal{N}(x|\theta, \Sigma) = det(2\pi\Sigma)^{-\frac{1}{2}} e^{-\frac{1}{2}(x-\theta)^T \Sigma^{-1}(x-\theta)}$$

For a bivariate phenotype, we can visualize this surface using a color gradient (fig. 1). The triangle marks the position of the θ parameter. The origin of the coordinate system marks where our populations will be in relation to the selective surface.

Creating more complex surfaces can follow basically the same strategy, but using several overlapping Gaussian functions. For this, we have to sample several θ parameters and define the fitness as the sum of all the Gaussian functions. For a set of N Gaussian, each with a different θ and the same Σ, we have:

$$\overline{W}(x) = \sum_{i=1}^{N} \mathcal{N}(x|\theta_i, \Sigma) = det(2\pi\Sigma)^{-\frac{1}{2}} \sum_{i=1}^{N} e^{-\frac{1}{2}(x-\theta_i)^T \Sigma^{-1}(x-\theta_i)}$$

In order to sample the θ parameters for the Gaussian functions, we could use uniform distributions in some interval to sample the (x, y) components for several optima, like in fig. 2. However, this causes problems in relation to the origin, as some directions have more peaks than others. The directions along the diagonals are longer than along the two axes, and so we end up with more peaks along the diagonals. One solution is to sample the (x, y) components from a normal distribution, which is spherically symmetrical, and then scale the resulting vectors to a magnitude sampled from a uniform distribution. This gives us control over the distance of the peaks from the origin and guarantees that all directions are equally likely, as in fig. 3. This procedure also generalizes to more dimensions, and avoids the problem of having all the peaks in a thin shell in multivariate space that would happen if we were to only sample all coordinates from a normal distribution and not sample the magnitudes separately.

The final restriction on the multiple peaks is to impose a minimum distance from the origin. This is necessary so that our population can actually evolve, and not start the simulation already at a phenotypic optima. We do this by restricting the distance of the Gaussian optima to be larger than some minimum distance. A set of θ parameters that follow these rules is displayed in fig. 4, and this particular set of θ results in the complex surface shown in fig. 5. We can see that the optima for the full surface don't necessarily coincide with one particular θ and that this method can create complex surfaces, with several peaks, valleys, ridges, and saddle points.

Interaction between genetic covariation and the selective surface

After we have created a surface, we can place a population at any point and use quantitative genetics theory to predict how the mean phenotype of the population will change over time. Assuming the additive genetic covariance matrix of the population (G-matrix) stays constant, at each generation the change in the mean phenotype Δz will be given by the Lande equation:

$$\Delta z = G\beta = G\frac{1}{\overline{W}}\nabla\overline{W} = G\nabla ln\overline{W}$$

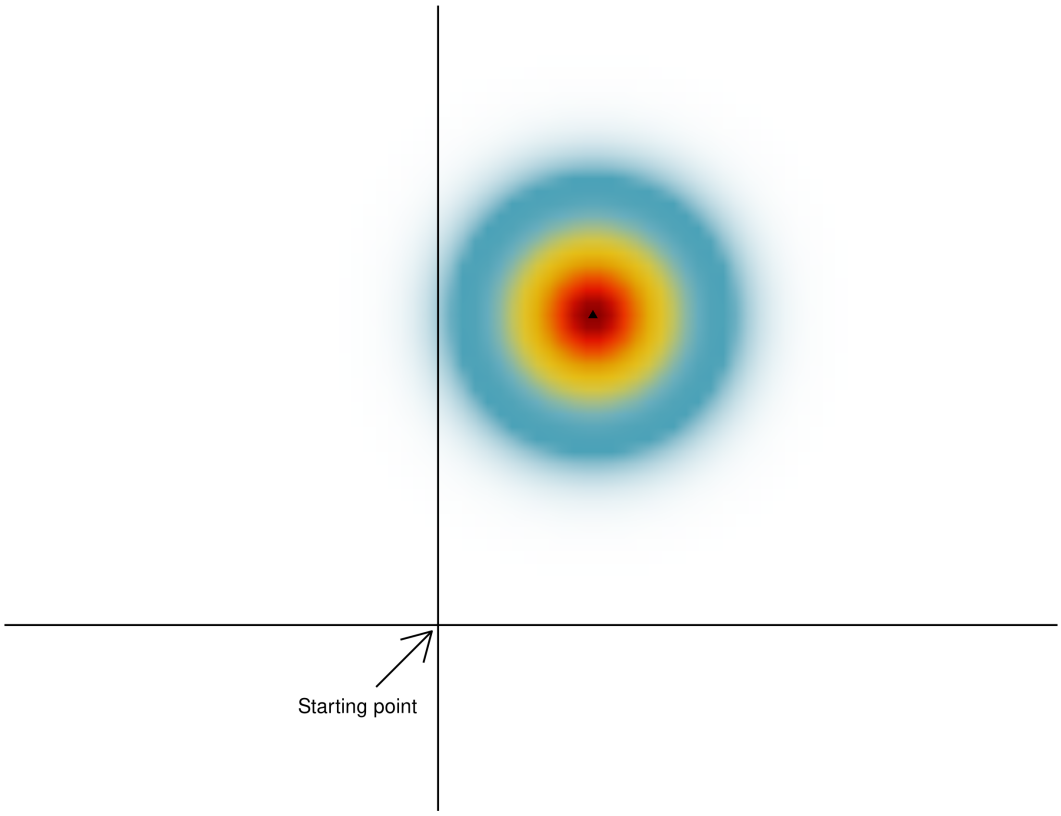

Figure 1: Single peak surface. The triangle marks the position of the θ parameter. The origin of the coordinate system marks where our populations will be in relation to the selective surface.

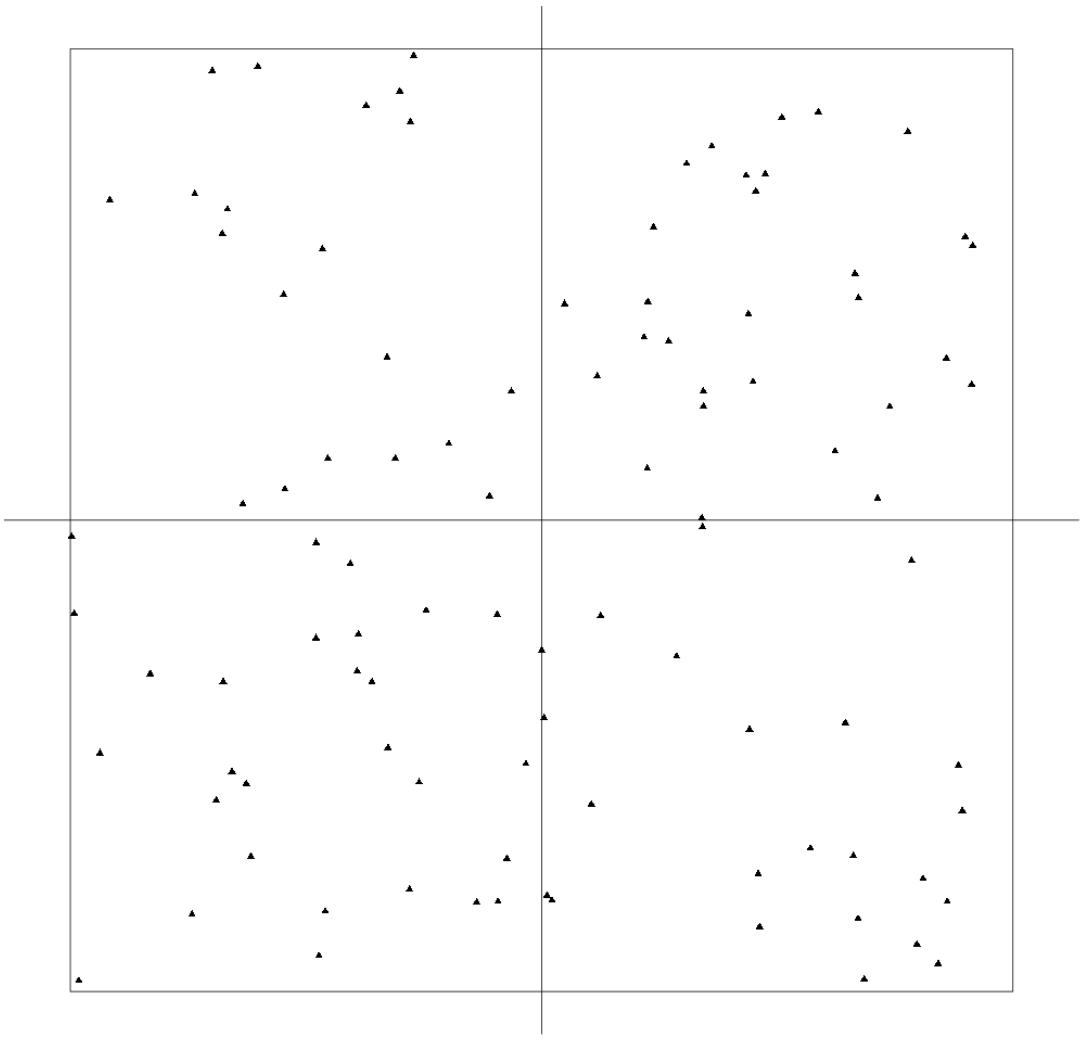

Figure 2: Multiple Gaussian optima with uniform coordinates. All the peaks are restricted to be inside the square, but this causes a bias in direction in relation to the origin. Directions along the diagonal are longer, and so have more peaks along them.

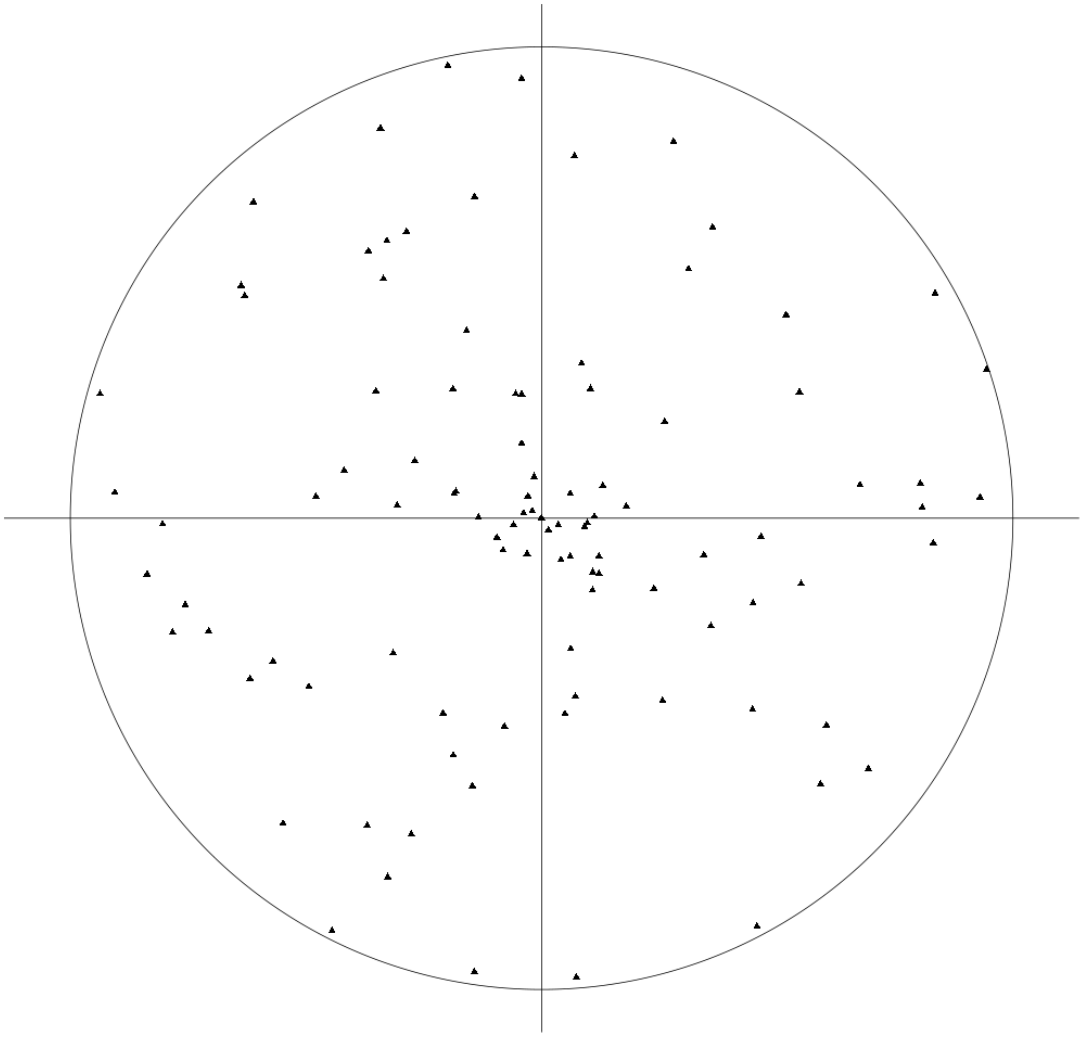

Figure 3: Multiple Gaussian optima with spherical symmetry and uniform distance from origin.

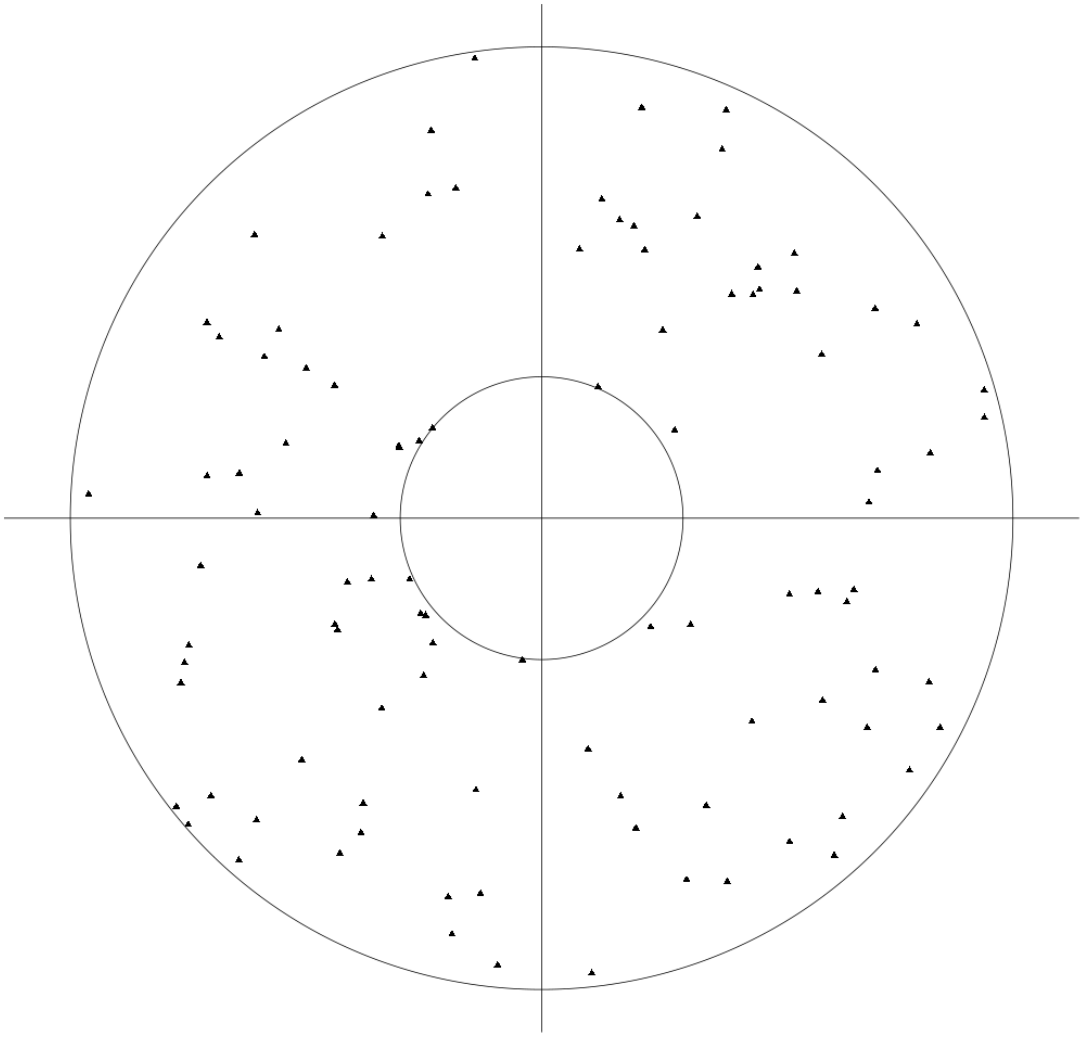

Figure 4: Multiple Gaussian optima with spherical symmetry and a minimum distance from the origin.

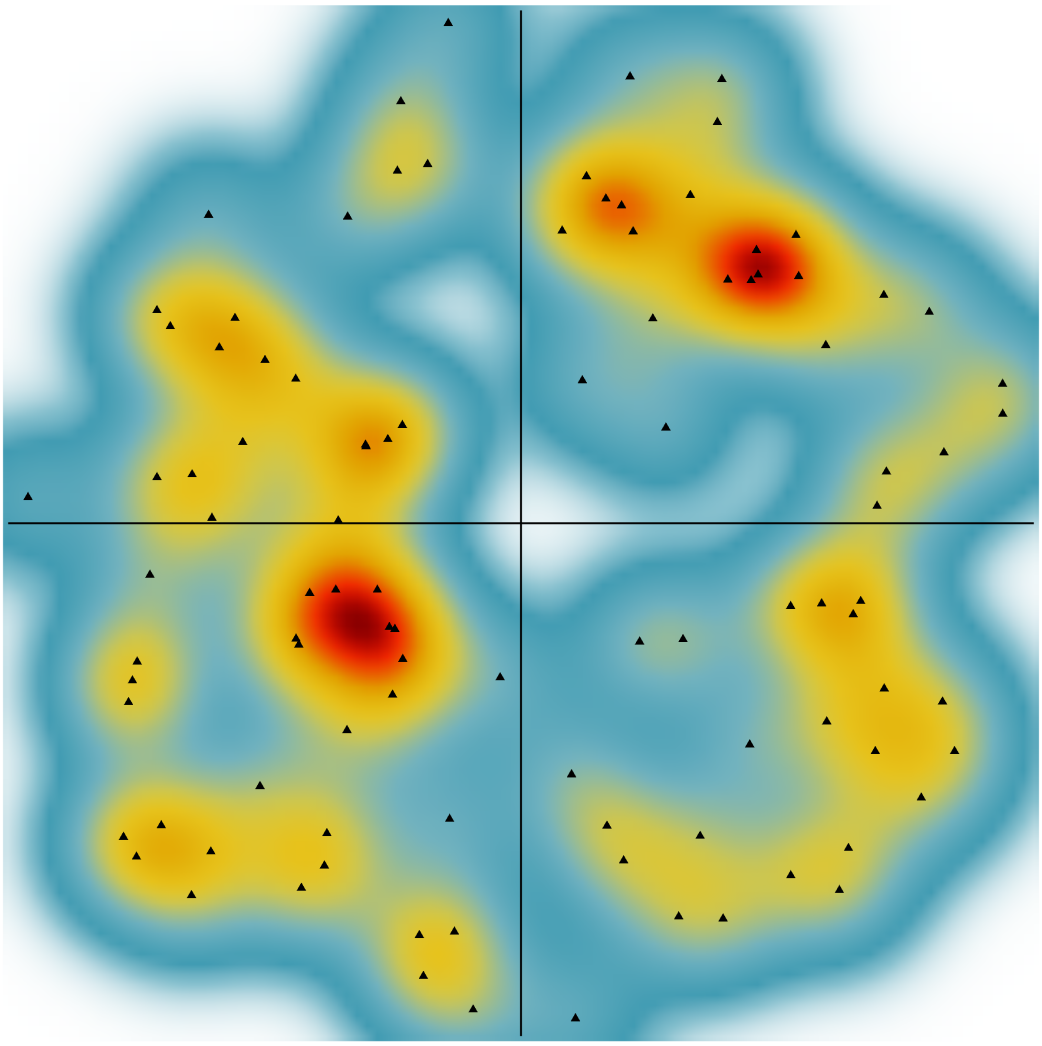

Figure 5: Surface generated by the multiple Gaussian optima in fig. 4.

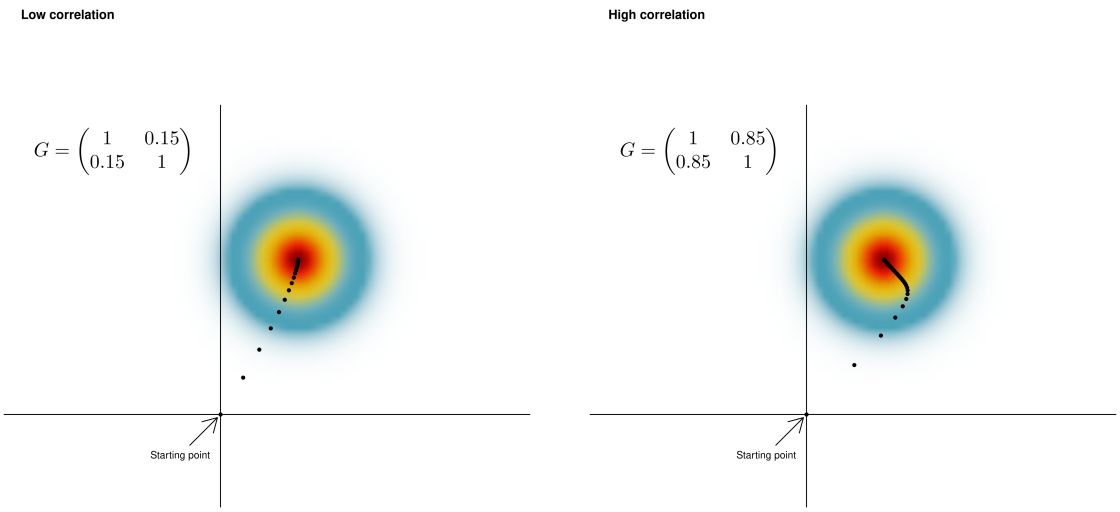

Figure 6: Different G-matrices produce different trajectories for the surface in fig. 1.

In this equation, the selection vector (β) is given by the gradient of the natural logarithm of the selective surface. The choice of modeling the selective surface with a sum of Gaussian functions allows for an analytical calculation of the gradient, which greatly improves the efficiency of the simulations. Using numerical gradients would make simulations in high dimensional cases impossible.

For a single multivariate Gaussian function $p(x|\theta, \Sigma)$, with $x \in \mathbb{R}^k$, the gradient of $p(x|\theta, \Sigma)$ is given by:

$$\frac{\partial p(x|\theta, \Sigma)}{\partial x} = -p(x)\Sigma^{-1}(x - \theta)$$

For a complex surface defined as the sum of several Gaussian functions, with $p_i(x) = p(x|\theta_i, \Sigma)$, $\overline{W}(x|\theta_1, \cdots, \theta_n, \Sigma) = \sum_{i=1}^{n} p_i(x)$, the gradient of the logarithm of $\overline{W}(x)$ is given by:

$$\nabla ln\overline{W}(x|\theta_1, \cdots, \theta_n, \Sigma) = \frac{1}{\overline{W}(x)} \sum_{i=1}^{n} -p_i(x)\Sigma^{-1}(x - \theta_i)$$

Multiplying the gradient at the position the population currently occupies by the G-matrix gives the phenotypic change, which can be added to the current position to obtain the new position of the population after selection. Iterating this process gives a trajectory in phenotype space, which stops when the gradient is zero. Different covariance matrices will create different trajectories. For example, fig. 6 shows the difference between the trajectories of two populations with high and low integration. In this example, both populations end up on the same optimum. However, if the surface is more complex, with several optima, they can end up on different optima. Figure 7 shows a slightly more complex surface where the different correlations lead the populations to different optima on the same surface. As the surfaces become more complex, these cases become more frequent.

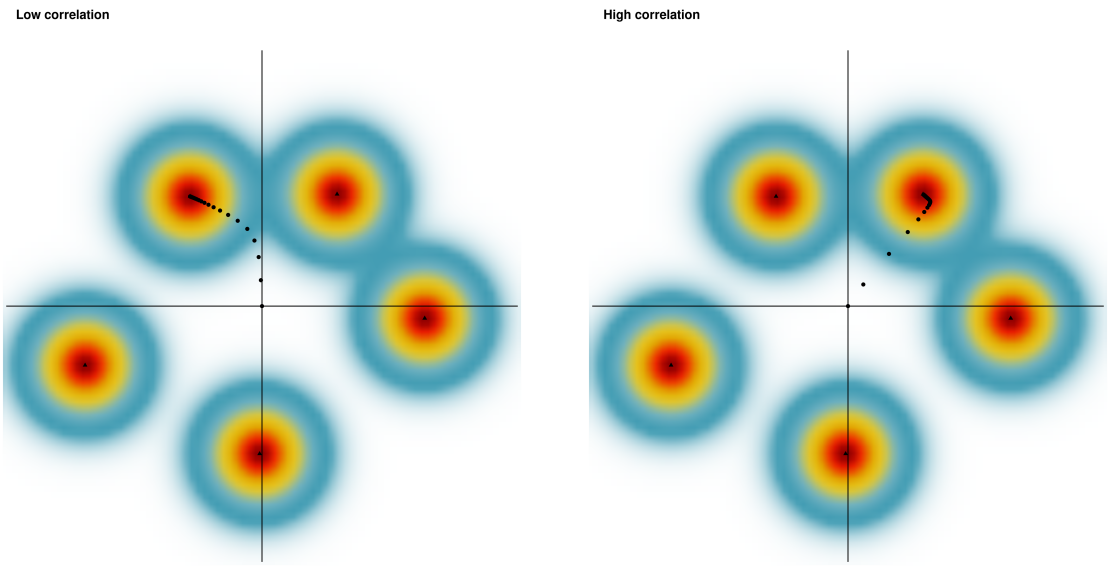

Figure 7: Different G-matrices can alter the final position in the landscape.

Simulations

We use this framework of generating random selective surfaces to investigate the relation between the magnitude of the total phenotypic change ($||\Delta z||$) and the alignment between the direction of phenotypic change and the main axis of genetic variation in the populations (given by the cosine of the angle between these vectors: $Cos(g_{max}, \Delta z)$, see fig. 8).

We use 4 different scenarios, presenting high and low integration populations with single- and multi-peaked selective surfaces. By generating several different surfaces, we can investigate the relation between these variables.

Covariance matrices

In order to generate simulations that are comparable to the empirical results, we used one of the empirical P-matrices as a hypothetical G-matrix in our simulations. We use a very highly integrated P-matrix, from the marsupial order Lutreolina, as the **integrated** G-matrix in our simulations. To create a comparable low integration matrix, we start with the integrated matrix and alter the distribution of variance along the eigenvalues in order to modify the integration, while keeping the eigenvectors and total variation constant. This is done by exponentiating the eigenvalues by some constant p. If G_H is the original high-integration matrix, λ_{G_H} and Λ_{G_H} are respectively its eigenvalues and eigenvectors, the new low integration matrix has the form:

$$G_{new} = \Lambda_{G_H} \lambda_{G_H}^{p} \Lambda_{G_H}^{-1}$$

If p is greater than one, the variance in the eigenvalues increases and consequently the overall magnitude of genetic integration also increases. If p is less than one, the variance of the eigenvalues is reduced, and

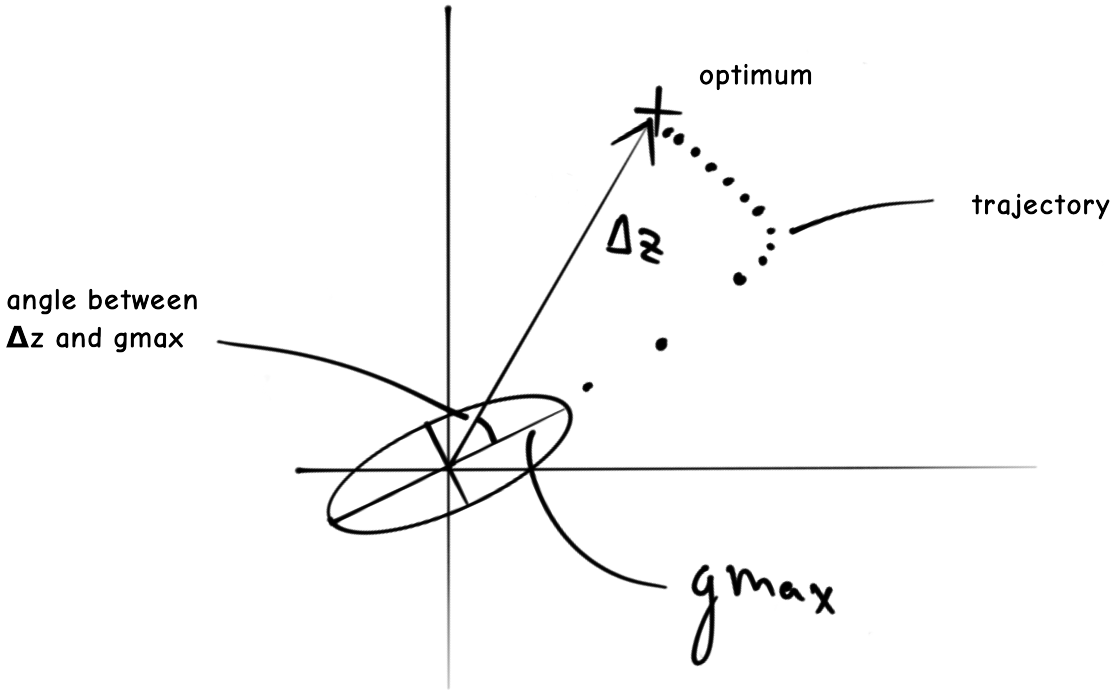

Figure 8: Scheme of the measurements in a simulation run. The total length of the phenotypic change is given by the norm of the Δz vector. The angle between the direction of most genetic variation (g_{max}) and the Δz vector measures the alignment between phenotypic change and genetic variation.

so is the integration. We chose the value of p by targeting an integration value around 0.02, which is lower than most empirical matrices. This value is calculated using the standardized eigenvalue variance described in Machado et al. 2019, using the CalcEigenVar function in the evolqg package. This new matrix is then scaled to have the same variances, taken from the empirical matrix (Lutreolina). This guarantees that both matrices have the same amount of total variation.

Random peaks

We start the simulation by using randomly generated peaks, as shown in fig. 5. For the single peak landscapes, there is no difference between Integrated and Diagonal matrices, and $||\Delta z||$ values are widely distributed for both matrices, with large and small amounts of phenotypic divergence being equally likely. We observe no relation between $||\Delta z||$ and $Cor(\Delta z, g_{max})$ (fig. 9). The multi peak landscape leads to much less variation in $||\Delta z||$, with most values clustering around the minimum peak distance for both matrices. Again, there is no influence of $Cor(\Delta z, g_{max})$ on $||\Delta z||$.

There is a conspicuous absence of $Cor(\Delta z, g_{max})$ values above 0.6. This is a problem because this is exactly the parameter region we expect to lead to differences in $||\Delta z||$. The lack of any peaks in this region of the phenotypic space can be explained by the high dimensionality of the system. The probability that random vectors in high dimensional space will have a correlation above 0.6 is vanishingly small. We address this by slightly modifying the distribution of peaks.

Enriched peaks

To effectively sample the region of phenotype space that is aligned with g_{max}, we developed a sampling procedure that enriches the peaks in the direction of g_{max} to a customizable amount. The idea is to sample random vectors in a way that produces a set of peaks that are in directions correlated with g_{max} according to a predetermined target distribution of $Cor(\Delta z, g_{max})$. To create this target distribution, we start by establishing the distribution of $Cor(\Delta z, g_{max})$ under a random sampling of peaks. This random distribution can be adequately represented by a beta distribution $Beta(a, b)$, which we fit to the observed random correlations using maximum likelihood, obtaining parameters a_r and b_r. This fitted beta distribution is then used in a mixture with a beta with parameters $a = 1$ and $b = 1$ and a mixture parameter ρ, which results in a target distribution of $Cor(\Delta z, g_{max})$ given by:

$$P(Cor(\Delta z, g_{max})) = \rho Beta(1, 1) + (1 - \rho)Beta(a_r, b_r)$$

Because $Beta(1, 1)$ is essentially flat in the $[0, 1]$ interval, this produces a small amount of vectors with correlations above 0.6, and this amount can be tuned by changing a and b or ρ. We select these parameters to produce a small amount of vectors in this region above 0.6, and to not change the rest of the distribution significantly. All three distributions (random, fitted beta, and enriched mixture) can be seen in fig. 10.

After determining the target distribution, we have to effectively sample the random peaks that follow this distribution. This is done using rejection sampling in three steps. First, we establish the number of random vectors we want to sample and the precision we want the target distribution to be followed. This

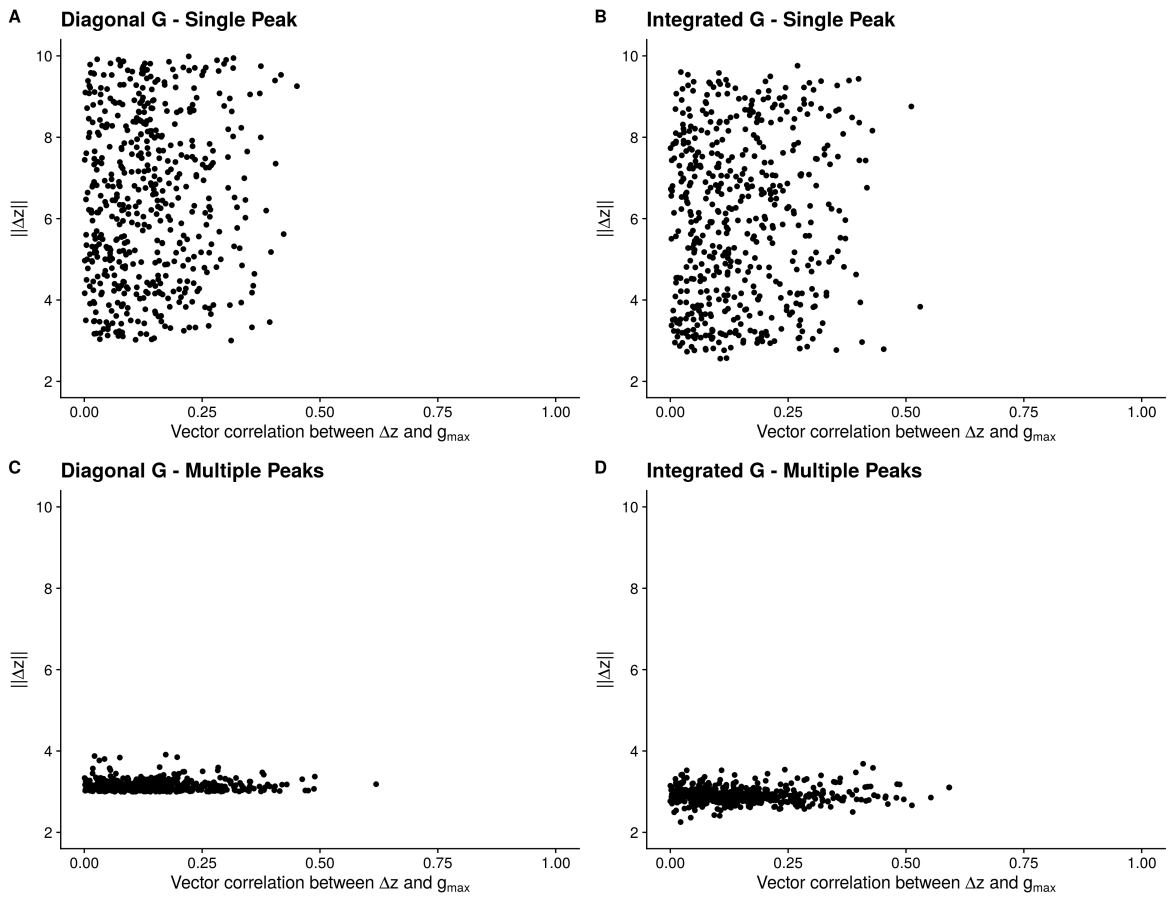

Figure 9: Simulations using random peaks. Integrated matrix is the Lutreolina P-matrix, while the diagonal matrix is a modification of the integrated matrix that reduces correlations while maintaining eigenvectors and total variance. We see a clear lack of Δz with high correlation to g_max. In this simulation, dimensionality of the phenotype is 35, and in the multi-peak landscape we have N = 50 peaks. Simulations are repeated 1000 times.

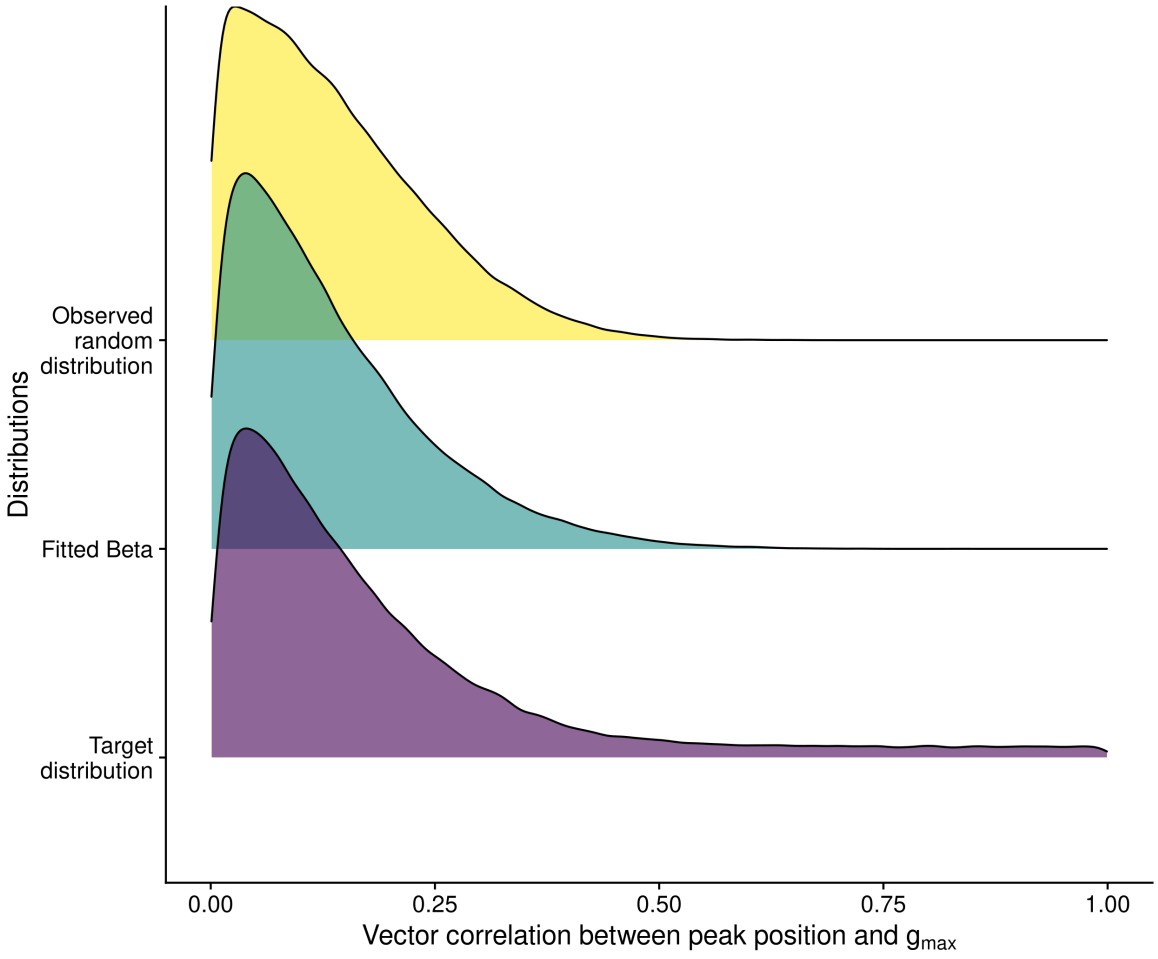

Figure 10: Comparison of the distributions of the correlations between peak position and g_{max} under the random distribution, the beta distribution fitted to this random distribution, and the target distribution using a mixture of betas. In this example: $a_r = 1.25$, $b_r = 8.05$, $\rho = 0.15$.

produces a target histogram for the random vectors, with a target number of vectors in each histogram bin. Table 1 shows an example of a histogram with 5 equally spaced bins and the corresponding number of vectors in each level of correlation that would need to be sampled. Next, we sample random vectors in the usual way, from a multivariate normal distribution, and fill as much of the target distribution as possible. Vectors whose correlation to g_{max} falls within one of the target bins that still have not been filled are accepted, while vectors falling into completely filled bins are discarded. This procedure mostly fills the bins with correlations lower than 0.5, but vectors with correlations above 0.5 are almost never observed. After this initial round of sampling, we generate random vectors with higher correlations to g_{max} by using perturbations. A perturbation vector is sampled from a multivariate normal distribution, and this perturbation vector is added to g_{max}. This produces a new random vector that can have its correlation to g_{max} tunned by altering the variance of the distribution that generates the perturbation. Small variances produce random vectors with higher correlation to g_{max}, while larger variances produce random vectors with lower correlations to g_{max}. By dynamically modifying this variance, we can complete the sampling of the target distribution and fill the bins with higher correlation to g_{max}. This pool of random directions that have been enriched to sample the region of high correlation with g_{max} is used to create the selective surface. The final position of the peaks is determined by sampling the distance of each peak from the origin from an uniform distribution, as described previously.

Table 1: Example target for the number of vectors in each correlation interval, for 5 intervals and 1000 vectors, using the distribution in fig. 10. In practice we use many more intervals (30) and vectors (100000).

Correlation Interval	0-0.2	0.2-0.4	0.4-0.6	0.6-0.8	0.8-1
Number of vectors	684	212	58	22	24

After enriching the peaks that are aligned with g_{max}, we can effectively investigate how the presence of these peaks affects the interaction between more or less integrated G-matrices and the selective surface. For the single peak landscapes, the results are similar to the random peaks, with the difference that we now observe some $Cor(\Delta z, g_{max})$ above 0.6, as expected. Still, for the single peak landscapes we see no difference between diagonal and integrated G-matrices, and no relation between $Cor(\Delta z, g_{max})$ and Δz. For the multiple peak landscapes, the results are different, with the integrated G-matrix now showing an increase in $||\Delta z||$ for higher values of $Cor(\Delta z, g_{max})$. This pattern is similar to the one observed in natural populations.

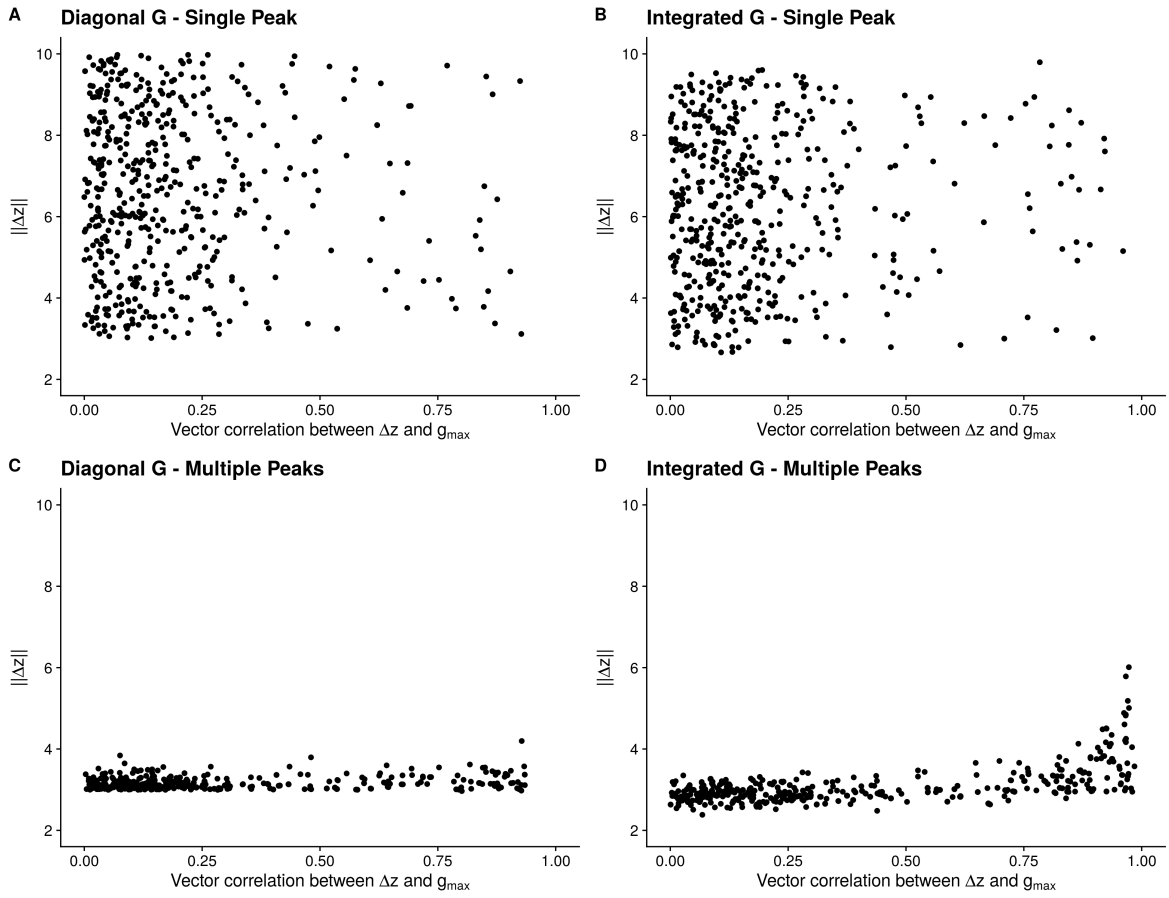

Figure 11: Simulations using enriched peaks. We now see the increase in $||\Delta z||$ for high values of $Cor(\Delta z, g_{max})$ in the integrated matrix. In this simulation, dimensionality of the phenotype is 35, and in the multi peak landscape we have 50 peaks. Simulations are repeated 1000 times.