Estimating selection gradients and divergence

The Evolution of Phenotypic Integration: How directional selection reshapes covariation in mice - Supporting Information -

Introduction

In this document, we explore the difficulties in estimating selection gradients from information on evolutionary divergence and extant covariation matrices using the Lande equation (Lande, R. 1979 "Quantitative Genetic Analysis of Multivariate Evolution, Applied to Brain: Body Size Allometry" - Evolution 33 (1): 402-416). Some of this material is discussed in Marroig, G.; Melo, D. and Garcia, G. (2012) "Modularity, Noise, and Natural Selection." - Evolution 66 (5): 1506–24.

Selection gradient estimation

The Lande equation estates that the evolutionary divergence (Δz) in a population will be proportional to the current additive genetic covariation (G-matrix) and the current selection gradient (β). If we measure Δz and the G-matrix we can invert the Lande equation and solve for β, inferring the direction of selection the population was subjected to. However, this procedure is extremely sensitive to errors in the G-matrix estimate, which can lead to β estimates that are wildly different from the real selection gradient. Here we use a simulation approach to show the effects of inverting the Lande equation to estimate β, discuss alternative methods to account for errors in the G-matrix, and give a more detailed explanation about the rationale behind using the direction of phenotypic divergence δz as an estimate of the direction of selection.

A known selection gradient

First, let's set up a simulated example using the mean pooled within-groups G-matrix from our experiment. This matrix was obtained from the BSFG model using all of the individuals and controlling for differences in means between the lines and between sexes.

```
G = readRDS("./G.rds")
G[1:3, 1:3]
```

```
##             [,1]        [,2]        [,3]
## [1,] 0.01032852 0.001279840 0.015369099
## [2,] 0.00127984 0.001508106 0.003901662
## [3,] 0.01536910 0.003901662 0.039194288
```

Now we simulate a known selection gradient which is highly correlated with cranial size, by perturbing the isometric vector with a small multivariate normal perturbation vector. This is reasonable, since we are dealing with overall size selection. We can control the size of the perturbation via the variance of the perturbation.

```
library(mvtnorm)
library(plyr)
library(evolqg)

vectorCor = function(x, y) t(Normalize(x)) %*% Normalize(y)

isometric = rep(1, 35)
beta_var = 0.5
beta = isometric + rnorm(35, 0, beta_var)
beta
```

```
##  [1]  1.3108842  0.6878299  2.1946133 -0.1000453  0.7155187  0.6017696
##  [7]  0.5590794  0.7660347  1.5250748  1.3983766  0.9494388  0.7927704
## [13]  0.7553928  1.1742697  0.3771082  0.9599153  1.3433819  1.4485024
## [19]  1.6677837  1.3723182  1.2546585  1.3174694  0.7324032  1.9297082
## [25]  0.5107780  0.5184618  0.4690690  1.5301183  0.4408164  1.5350409
## [31]  0.9792143  0.4085291  0.5563581  1.0071311  1.7600911
```

Now we can use this selection gradient to obtain a response to selection:

```
DeltaZ = G %*% beta
DeltaZ[1:3]
```

```
## [1] 0.22851818 0.04815061 0.51695550
```

If we have an error-free estimate of the G-matrix, we can use the Δz to reconstruct β exactly:

```
recons_beta = solve(G, DeltaZ)
```

```
## Are they numericaly the same?
all(abs(recons_beta - beta) < 10e-9)
```

```
## [1] TRUE
```

```
## Are they pointing in the same direction?
vectorCor(recons_beta, beta)
```

```
##      [,1]
## [1,]    1
```

However, if we only have a sample of the population to estimate the G-matrix, things change. Even a reasonably large sample of 100 individuals will carry considerable noise, and the β estimate will suffer.

```
## One hundred individuals sampled from G
pop = rmvnorm(100, mean = rep(0, 35), sigma = G)

## The estimated covariance matrix
G_estimated = cov(pop)

## check that G and G_estimated are similar
MatrixCompare(G, G_estimated)
```

```
##        correlation probability    .id
## RS       0.9765115 0.000000000     RS
## Mantel   0.9473408 0.000999001 Mantel
## Krz      0.8805953          NA    Krz
## PCA      0.9844630          NA    PCA
```

The matrices are fairly similar, but let's check the estimated β, using the actual Δz:

```
beta_estimated = solve(G_estimated, DeltaZ)
vectorCor(beta, beta_estimated)
```

```
##           [,1]
## [1,] 0.1531601
```

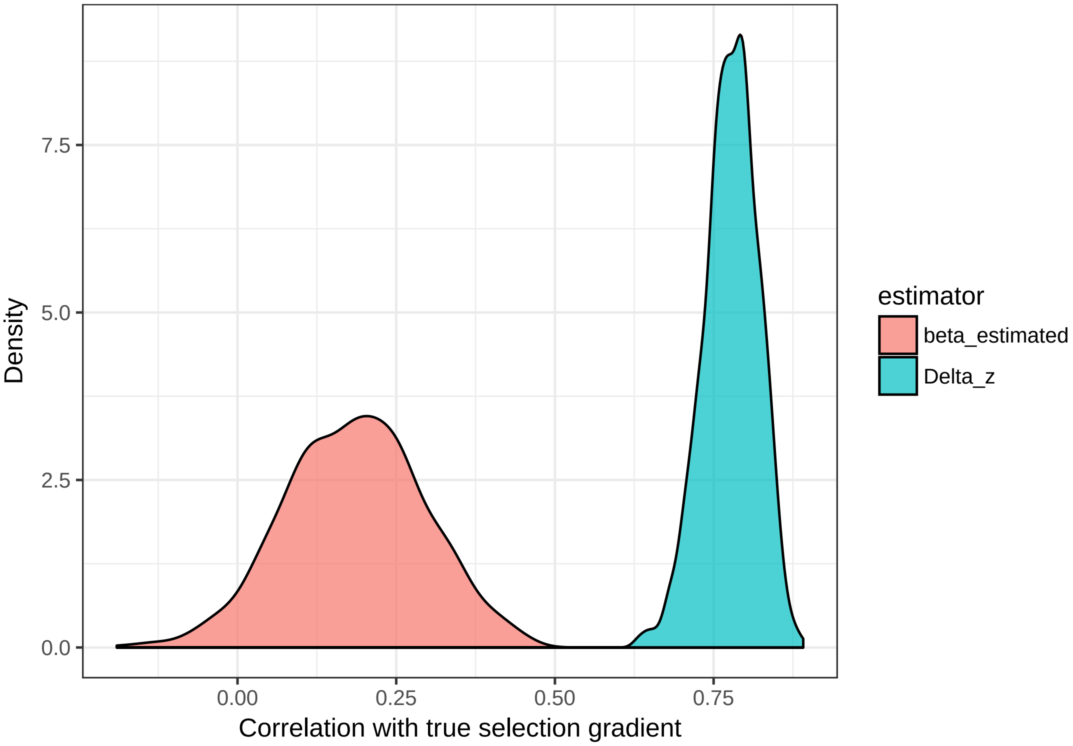
The estimated β is completely different from the actual β, and their correlation is about what we would expect for random vectors in 35 dimensions. If fact, the Δz is a better predictor of the direction of the actual β than the estimated selection gradient we obtain from solving the Lande equation.

```
vectorCor(DeltaZ, beta)
```

```
##           [,1]
## [1,] 0.7797053
```

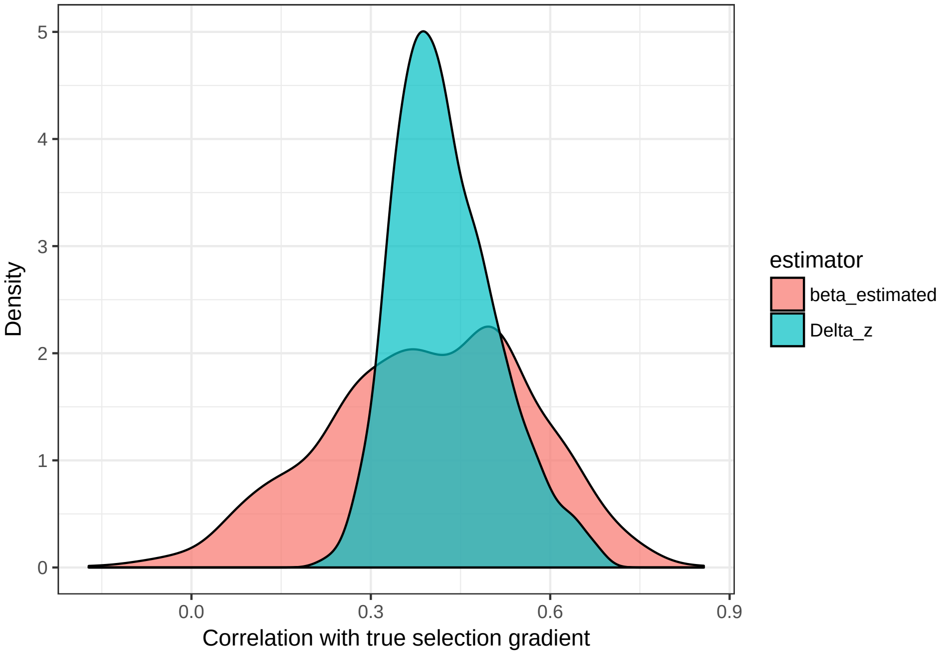
We can repeat this many times and see if this was a fluke:

```
library(tidyr)
library(plyr)
library(ggplot2)
simulateBetaEstimation = function(beta_center, G, beta_var = 0.5, Gsample = 100) {
  beta = beta_center + rnorm(35, 0, beta_var)
  Delta_z = G %*% beta
  pop = rmvnorm(Gsample, mean = rep(0, 35), sigma = G)
  G_estimated = cov(pop)
  beta_estimated = solve(G_estimated, Delta_z)
  data.frame(beta_estimated = vectorCor(beta_estimated, beta),
             Delta_z = vectorCor(Delta_z, beta))
}
result = rdply(1000, simulateBetaEstimation(isometric, G))
m_result = gather(result, estimator, value, beta_estimated:Delta_z)
ggplot(m_result, aes(value, fill = estimator)) + geom_density(alpha = 0.7) +
  labs(x = "Correlation with true selection gradient", y = "Density") + theme_bw()
```



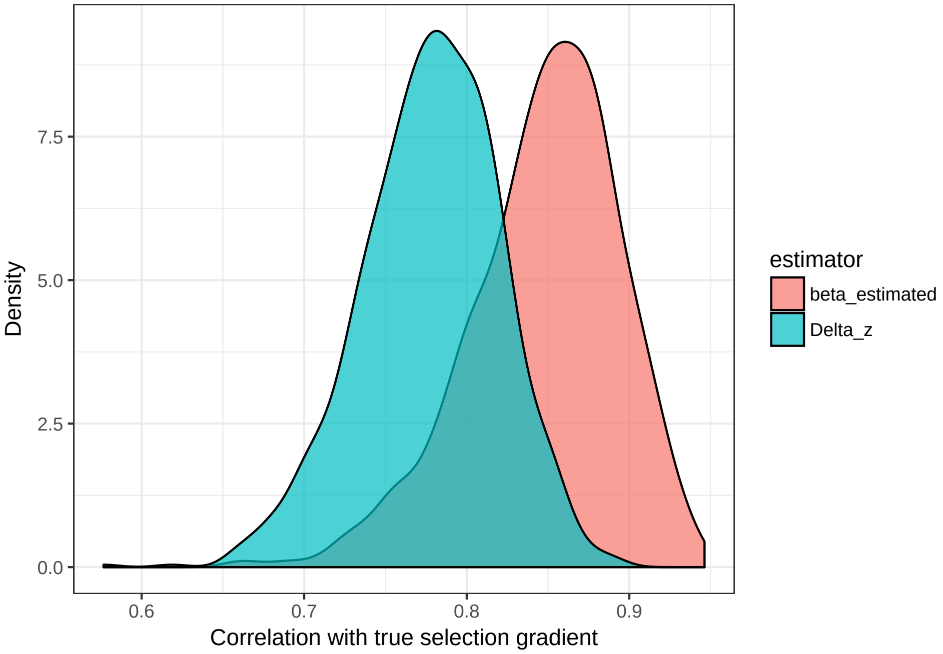
While this is surprising, it make sense if we understand that β estimation is very sensitive to noise in the G-matrix, and that selection in the direction of size (the first eigenvector) will tend to lead to responses also aligned with size. If we increase the variance in the selection gradient simulation, and selection is no longer in the size direction, Δz becomes an estimator that is frequently worst than the selection gradient:

```
result = rdply(1000, simulateBetaEstimation(isometric, G, beta_var = 3))
m_result = gather(result, estimator, value, beta_estimated:Delta_z)
ggplot(m_result, aes(value, fill = estimator)) + geom_density(alpha = 0.7) +
  labs(x = "Correlation with true selection gradient", y = "Density") + theme_bw()
```



We can also improve the estimation of the selection gradient by having extremely large sample sizes and, consequently, well estimated matrices:

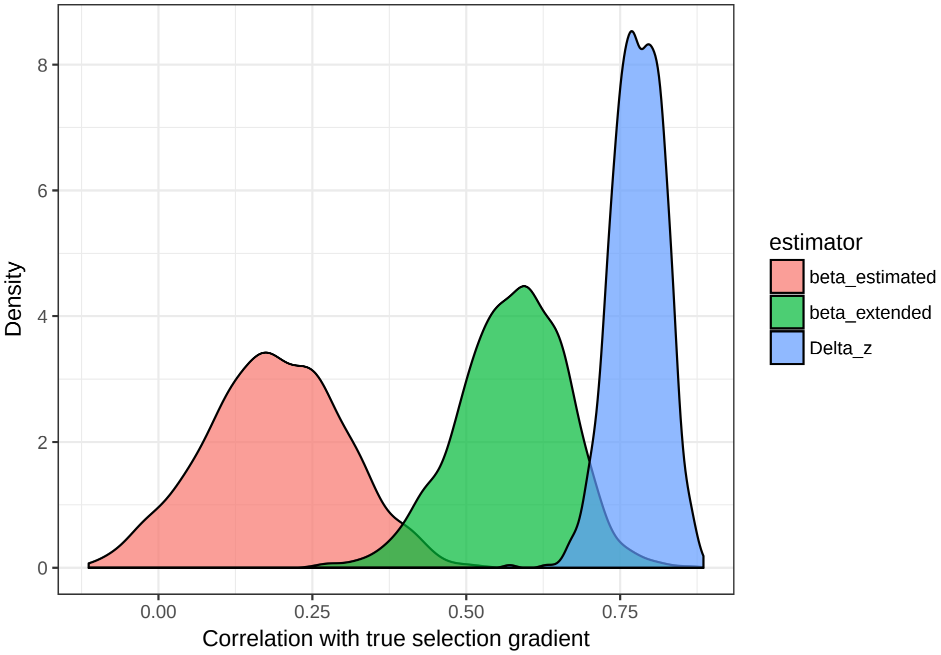
```
result = rdply(1000, simulateBetaEstimation(isometric, G, Gsample = 5000))
m_result = gather(result, estimator, value, beta_estimated:Delta_z)
ggplot(m_result, aes(value, fill = estimator)) + geom_density(alpha = 0.7) +
  labs(x = "Correlation with true selection gradient", y = "Density") + theme_bw()
```



Noise control methods

Another option is to improve the G-matrix estimate using statistical methods. Marroig et. al. (2012) showed that the extended covariance matrix estimator could vastly improve selection gradient estimation. Let's try with our simulated example, using the ExtendMatrix function from the EvolQG package:

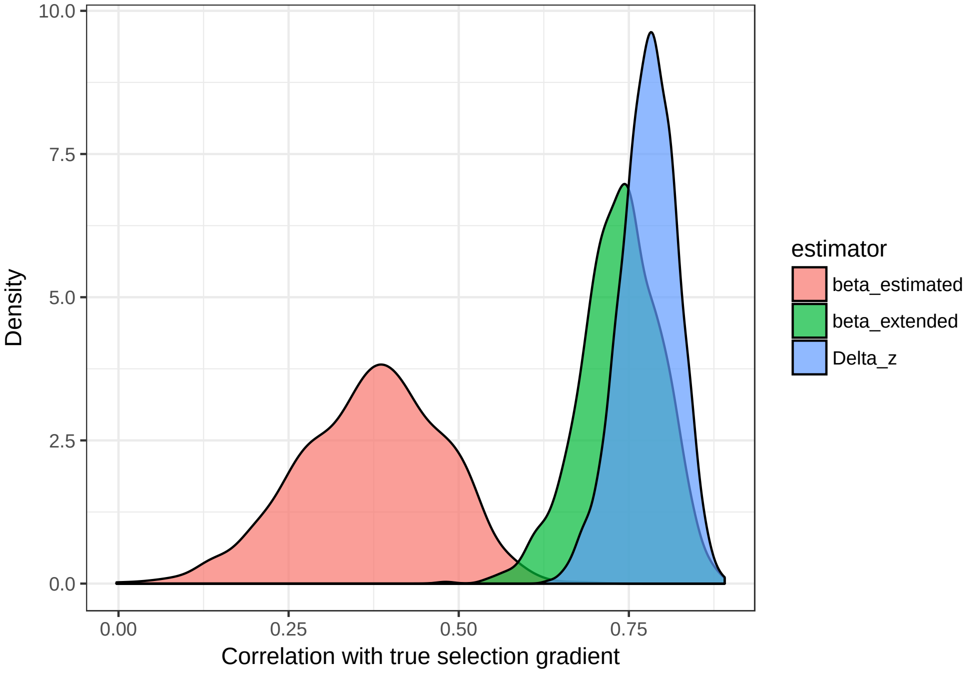
```
simulateBetaEstimation = function(beta_center, G, beta_var = 0.5, Gsample = 100) {
  beta = beta_center + rnorm(35, 0, beta_var)
  Delta_z = G %*% beta
  pop = rmvnorm(Gsample, mean = rep(0, 35), sigma = G)
  G_estimated = cov(pop)
  beta_estimated = solve(G_estimated, Delta_z)
  beta_extended = solve(ExtendMatrix(G_estimated)[[1]], Delta_z)
  data.frame(beta_estimated = vectorCor(beta_estimated, beta),
           beta_extended = vectorCor(beta_extended, beta),
             Delta_z = vectorCor(Delta_z, beta))

}
result = rdply(1000, simulateBetaEstimation(isometric, G, 0.5))
m_result = gather(result, estimator, value, beta_estimated:Delta_z)
ggplot(m_result, aes(value, fill = estimator)) + geom_density(alpha = 0.7) +
  labs(x = "Correlation with true selection gradient", y = "Density") + theme_bw()
```



With a sample of 100 for the G-matrix, Δz is a better estimator of β under size selection, but the matrix extension helps with the selection gradient estimate.

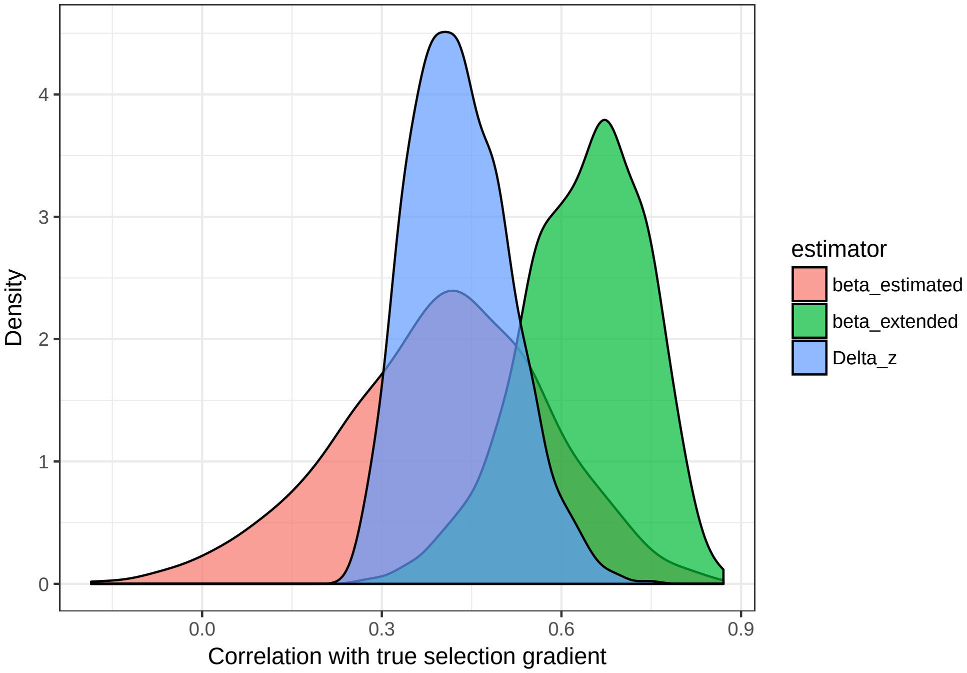
With a larger sample, the extended β is about the same as Δz, even for size selection.

```
result = rdply(1000, simulateBetaEstimation(isometric, G, 0.5, Gsample = 350))
m_result = gather(result, estimator, value, beta_estimated:Delta_z)
ggplot(m_result, aes(value, fill = estimator)) + geom_density(alpha = 0.7) +
  labs(x = "Correlation with true selection gradient", y = "Density") + theme_bw()
```



For selection that is not size related, the β estimator using the extended matrix is almost always the best:

```
result = rdply(1000, simulateBetaEstimation(isometric, G, 3, Gsample = 100))
m_result = gather(result, estimator, value, beta_estimated:Delta_z)
ggplot(m_result, aes(value, fill = estimator)) + geom_density(alpha = 0.7) +
  labs(x = "Correlation with true selection gradient", y = "Density") + theme_bw()
```

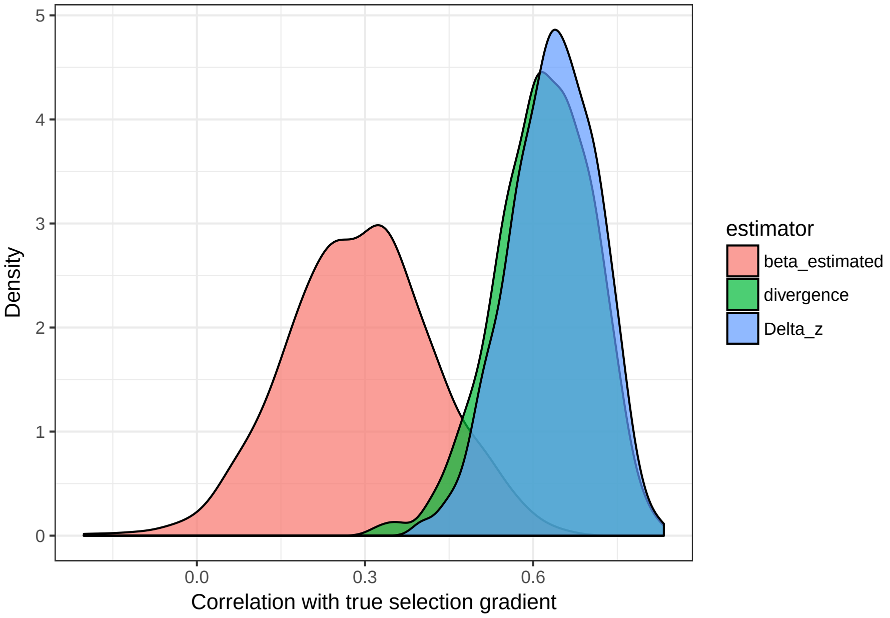


Divergence and variation

In our manuscript we are interested in exploring the non-isotropic changes in variation after an event of selection on overall size. To do so, we identify the direction that describes the divergence between the two-way selection δz and examine the amount of variation in this direction.

We can simulate the direction of divergence δz between two populations that were subject to opposite selection gradients (both related to overall size) and compare its alignment with the actual β. We use independent errors in both selection gradients, with a larger variance than in the other examples, to account for the fact that selection is not completely symmetrical:

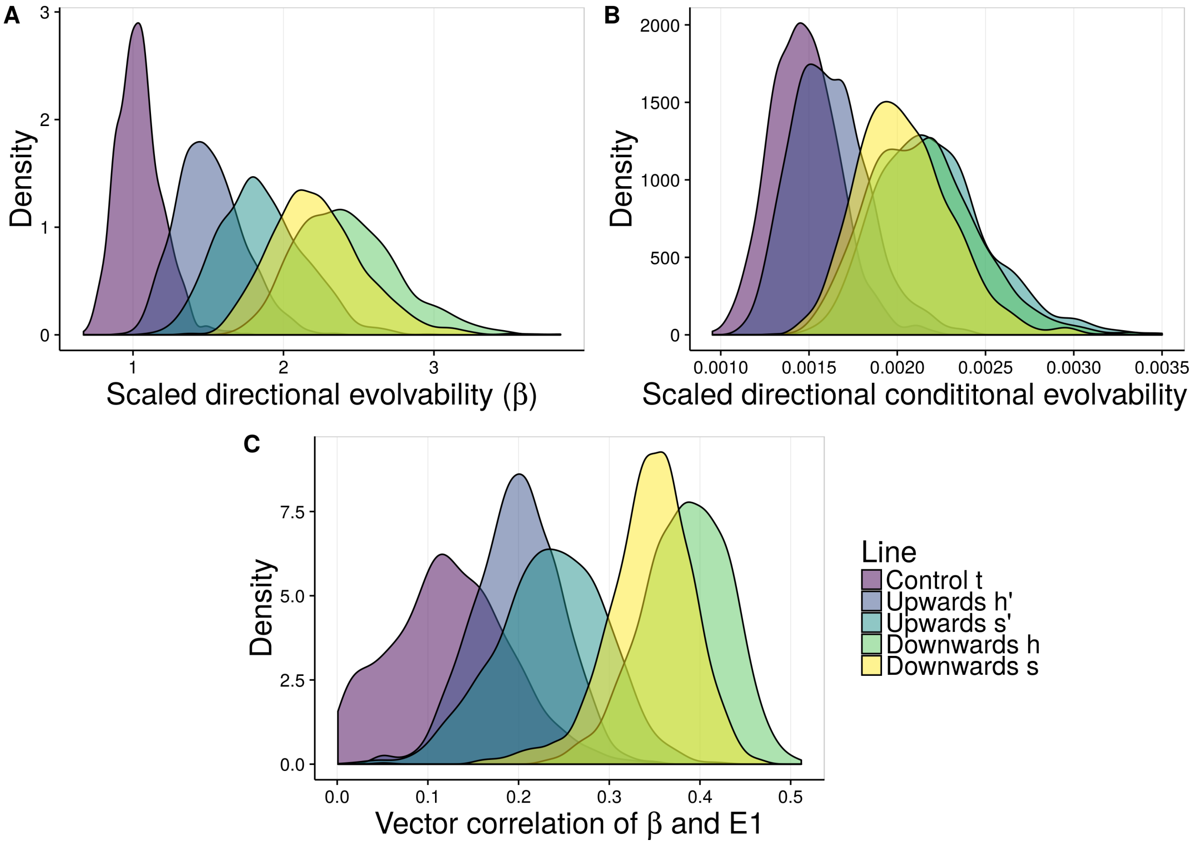
```
simulateBetaEstimation = function(beta_center, G, beta_var = 1, Gsample = 150) {
  pop = rmvnorm(Gsample, mean = rep(0, 35), sigma = G)
  G_estimated = cov(pop)
  ancestral_mean = colMeans(pop)

  beta = beta_center + rnorm(35, 0, beta_var)
  Delta_z = G %*% beta

  beta_opposite = - beta_center + rnorm(35, 0, beta_var)
  Delta_z_opposite = G %*% beta_opposite

  divergence = (ancestral_mean + Delta_z) - (ancestral_mean + Delta_z_opposite)

  beta_estimated = solve(G_estimated, Delta_z)
  data.frame(beta_estimated = vectorCor(beta_estimated, beta),
             divergence = vectorCor(divergence, beta),
             Delta_z = vectorCor(Delta_z, beta))
}
result = rdply(1000, simulateBetaEstimation(isometric, G))
m_result = gather(result, estimator, value, beta_estimated:Delta_z)
m_result$estimator = factor(m_result$estimator, levels = unique(m_result$estimator))
ggplot(m_result, aes(value, fill = estimator)) + geom_density(alpha = 0.7) +
  labs(x = "Correlation with true selection gradient", y = "Density") + theme_bw()
```



The direction of divergence δz describes the same direction as Δz, and when the selection gradient is related to size, it is also a better predictor of the actual direction of selection than the estimated β.

In our figure 4, we used the direction of phenotypic divergence between the upward and downward selected lines (instead of estimating beta) because in order to have a meaningful direction in which to explore the differences in variation we must maximize our power at every step of the way. Our approach (i.e., calculate the difference between the means of selected lines) is based on the fact that the lines started at the same point, changed in opposite directions (as shown in figures S1 and S3), and this direction is highly correlated with overall size (correlation of 0.82 with the isometric vector). Using the difference between variant lines is a common practice in studies where there is no data about the ancestral population (e.g. Bolstad et al., 2014 - doi:10.1098/rstb.2013.0255), which is the case here. Thus, δz gives us a widely used quantity that reflects the differences caused by the selective regimes, and can plausibly be a better estimator of the direction of selection than the selection gradient, as shown here. Moreover, estimating each of the four directions of divergence (Δz for each individual line) would substantially reduce our sample sizes, impairing estimates due to increased noise. Furthermore, our control t line does not correspond to the ancestral population (i.e., it was sampled in generation 55), and therefore may already have accumulated changes due to drift.

Another option that would be reasonable given our experiment and sample sizes would be to use the direction of divergence and the extended pooled within-groups G- or P-matrix (an estimate of the ancestral matrix) to estimate a single selection gradient, and then use this gradient to explore the directional variation in the covariance matrices. Using this approach, we get the following version for figure 4:



This is somewhat different from the figure in the manuscript, as is expected by the difficulty in β estimation, but the results are largely consistent. Overall, the effects in the direction of divergence seem larger, consistent with the expectation that δz is a better estimate of the direction of selection under size selection and given our sample sizes.