

Single-trait selection

Athens COST InsectIMP Course 2025

Leticia Aparecida de Castro Lara, Nelson Lubanga, and Gregor Gorjanc

2025-01-30

Introduction

In this vignette, we will learn about simulating selection to improve a single trait. We will achieve this by:

- Simulating a base population,
- Selecting parents of the next generation,
- Analysing response to selection between generations,
- EXTRA: Comparing the observed response to selection with a prediction from the Breeder's equation,
- EXTRA: Further analysis with the expanded Breeder's equation, and
- EXTRA: Selection over many generations.

Base population

We will start our simulation by simulating founder genomes and specifying a trait. Here we are simulating a maize example with 10 chromosomes and 100 founders. We are defining a trait that has a simple additive genetic architecture and is controlled by 100 loci on each chromosome (this is why we simulate 100 segregating sites in founder genomes). We set the mean of the trait to 10 units and genetic variance to 1 unit².

```
# Clean the working environment
rm(list = ls())

# Set the default plot layout
par(mfrow = c(1, 1))

# Load AlphaSimR, simulate founder genomes, set SP object, and define a trait
library(AlphaSimR)
```

```
## Loading required package: R6

# This runMacs() call will take quite a bit of time!
founderGenomes = runMacs(nInd = 100,
                        nChr = 10,
                        segSites = 100,
                        species = "MAIZE")

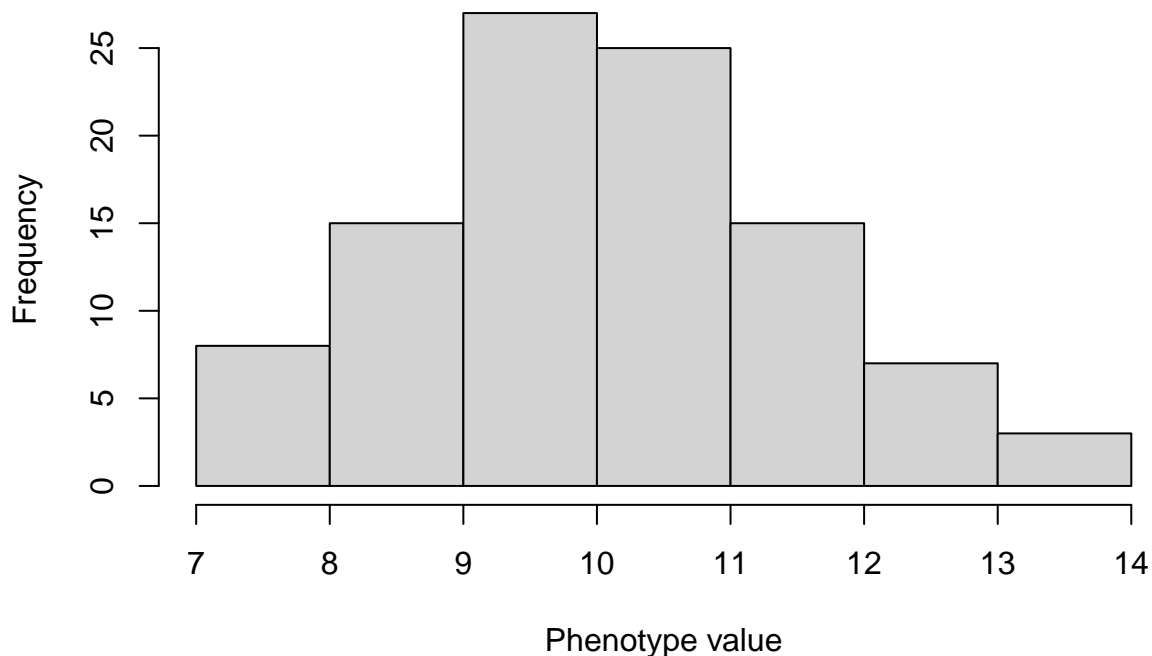
SP = SimParam$new(founderGenomes)
SP$addTraitA(nQtlPerChr = 100, mean = 10, var = 1)
```

With the simulated founder genomes and the defined trait we can now generate a base population. We also generate phenotypes for these individuals by assuming that heritability of these phenotypes is 0.5. We will save heritability in a variable so we can reuse it later.

```
# Base population and their phenotypes
basePop = newPop(founderGenomes)
heritability = 0.5
basePop = setPheno(basePop, h2 = heritability)
```

Let's summarise phenotype values in this base population.

```
# Histogram of phenotype values in the base population
hist(pheno(basePop), xlab = "Phenotype value", main = "")
```



```
# Mean of phenotype values in the base population
meanP(basePop)
```

```
## Trait1
## 10.06122
```

```
# Variance of phenotype values in the base population
varP(basePop)
```

```
## Trait1
## Trait1 2.027024
```

Parents of the next generation

Now we will identify 20 superior individuals in the base population and use them to generate a new, improved population. We will identify individuals as superior based on their phenotype values. To improve the trait we will be interested in identifying individuals with highest values. Here is a list of the top 20 individuals in the base populations, sorted by phenotype values.

```
# Collect data
basePopData = data.frame(id = basePop@id, pheno = pheno(basePop)[, 1])

# Save the number of selected individuals for later reuse
nSelected = 20
```

```
# Show top individuals by phenotype values
basePopData[order(basePopData$pheno, decreasing = TRUE), ][1:nSelected, ]
```

```
##      id      pheno
## 79 79 13.98034
## 64 64 13.76956
## 1   1 13.02453
## 43 43 12.77192
## 48 48 12.77016
## 24 24 12.70961
## 91 91 12.59856
## 20 20 12.33782
## 97 97 12.25852
## 3   3 12.07427
## 95 95 11.77692
## 14 14 11.68184
## 89 89 11.62074
## 21 21 11.50897
## 6   6 11.41826
## 81 81 11.35824
## 65 65 11.33986
## 49 49 11.31460
## 68 68 11.15686
## 77 77 11.12449
```

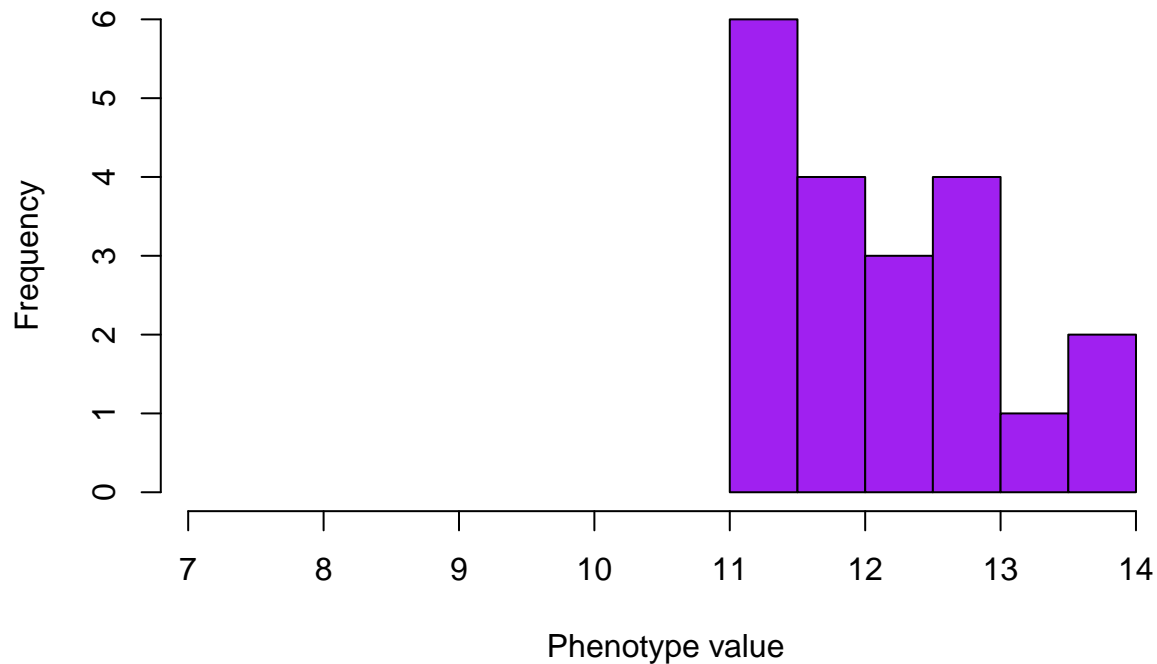
Since selection is a key operation in breeding, we have a dedicated function for selecting individuals from a population, `selectInd()`. Read more about the `selectInd()` function in its help page. In particular, pay attention to the `nInd` and `use` arguments.

```
help(selectInd)
```

We will now select the 20 superior individuals and summarise their phenotype values.

```
# Select the superior individuals
basePopSelected = selectInd(pop = basePop,
                           nInd = nSelected,
                           use = "pheno")

# Histogram of phenotype values in the selected part of the base population
hist(pheno(basePopSelected), xlim = range(pheno(basePop)), col = "purple",
     xlab = "Phenotype value", main = "")
```



```
# Mean of phenotype values in the selected part of the base population
meanP(basePopSelected)
```

```
## Trait1
## 12.1298
```

```
# Variance of phenotype values in the selected part of the base population
varP(basePopSelected)
```

```
## Trait1
## Trait1 0.683614
```

The superior individuals have, on average, higher phenotype values (10.0612161 vs 12.1298041). They also have lower variance (2.027024 vs 0.683614). A higher mean is expected because we have selected individuals with the highest phenotype values. The variance is lower because we have selected a non-random subset of individuals from the population.

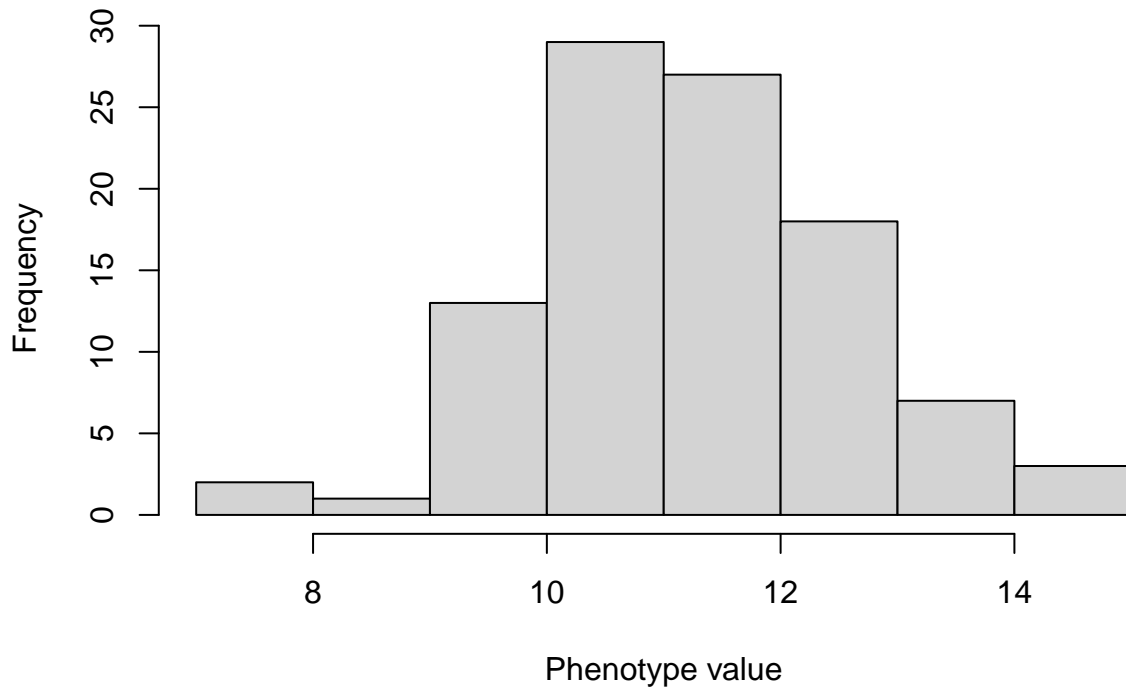
Now we will create a new population by randomly crossing the selected individuals. We will phenotype the progeny and quantify difference between the populations to evaluate what response to selection have we achieved.

```
# Study the randCross function
help(randCross)
```

```
# Cross selected individuals
newPop = randCross(pop = basePopSelected, nCrosses = nInd(basePop))
```

```
# Phenotype the progeny
newPop = setPheno(newPop, h2 = heritability)
```

```
# Histogram of phenotype values in the new population
hist(pheno(newPop), xlab = "Phenotype value", main = "")
```



```
# Mean of phenotype values in the new population
meanP(newPop)
```

```
## Trait1
## 11.24269
```

```
# Variance of phenotype values in the new population
varP(newPop)
```

```
## Trait1
## Trait1 1.876991
```

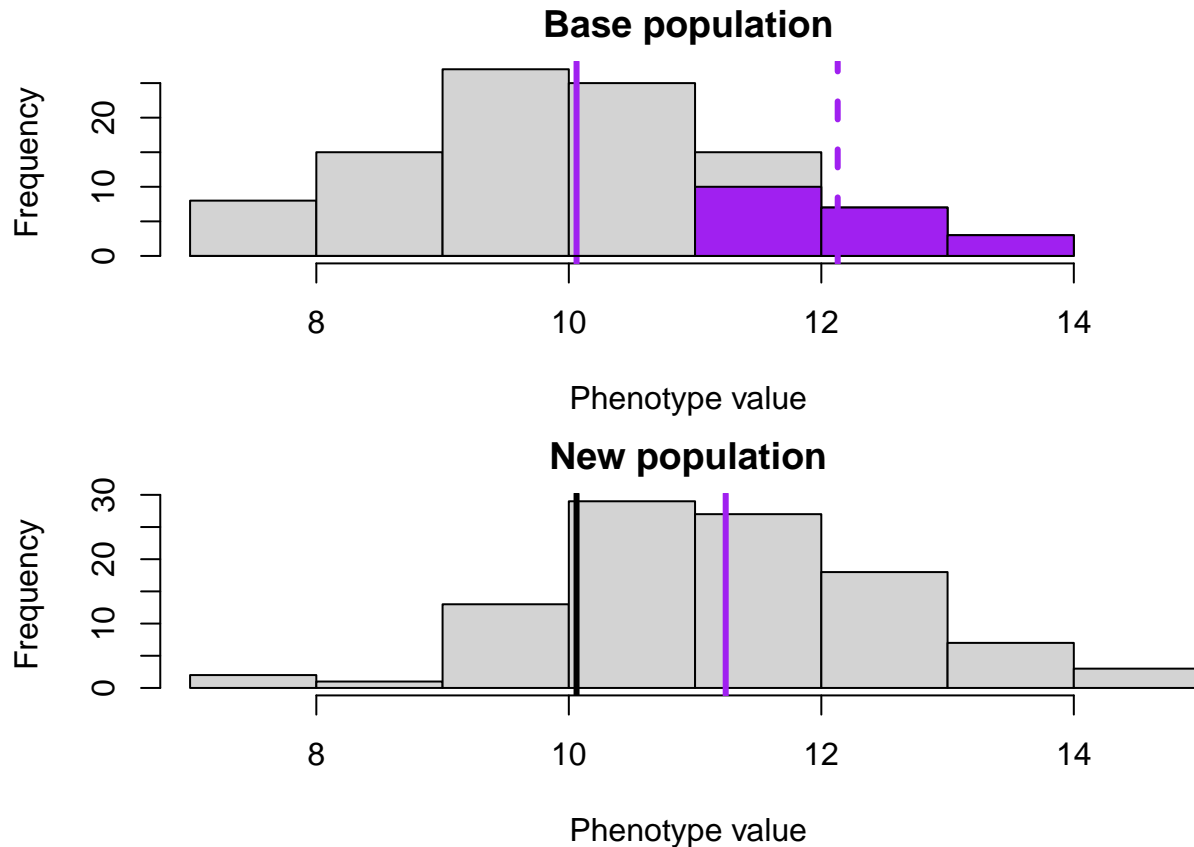
The mean difference in phenotype values between the two generations is 1.1814786. We can visualise these differences with the following code for plotting histogram of phenotypes in different generations. We will highlight with a purple solid line the overall mean of the population in both generations, with a purple dotted line the mean of the parents, and with a black line the overall mean of base population, so we can clearly see response to selection.

```
phenoRange = range(pheno(c(basePop, newPop)))
par(mfrow = c(2, 1),
    mar = c(4, 4, 2, 1))

# Base population
tmp = hist(pheno(basePop), xlim = phenoRange, xlab = "Phenotype value", main = "Base population")
abline(v = mean(pheno(basePop)), col = "purple", lty = 1, lwd = 3)

# Selected individuals
hist(pheno(basePopSelected), add = TRUE, col = "purple", breaks = tmp$breaks)
abline(v = mean(pheno(basePopSelected)), col = "purple", lty = 2, lwd = 3)

# New generation
hist(pheno(newPop), xlim = phenoRange, xlab = "Phenotype value", main = "New population")
abline(v = mean(pheno(basePop)), col = "black", lty = 1, lwd = 3)
abline(v = mean(pheno(newPop)), col = "purple", lty = 1, lwd = 3)
```



Analysing response to selection

We have observed that the average difference in phenotype values between the base and the new, improved population is $11.2426947 - 10.0612161 = 1.1814786$. However, the mean of phenotype values in the new population (11.2426947) is quite a bit smaller than the mean of phenotype values of the parents that have generated this new population (12.1298041).

To understand the discrepancy between the parental and progeny phenotype means, we have to recognise that parents pass DNA to the next generation, and not phenotypes. To this end, let's analyse mean phenotype and genetic values in the base population, the selected individuals, and the new population.

```
# Mean of phenotype and genetic values in the base population
c(meanP(basePop), meanG(basePop))
```

```
## Trait1 Trait1
## 10.06122 10.00000
```

```
# Mean of phenotype and genetic values in the selected part of the base population
c(meanP(basePopSelected), meanG(basePopSelected))
```

```
## Trait1 Trait1
## 12.12980 11.04494
```

```
# Mean of phenotype and genetic values in the new population
c(meanP(newPop), meanG(newPop))
```

```
## Trait1 Trait1
## 11.24269 11.02128
```

The above means show that selected individuals have higher genetic values than the average of their population (1.0449445 difference), as intended when we selected individuals with the highest phenotype values. However, comparison of their mean phenotype values (12.1298041) and mean genetic values (11.0449445) shows considerable discrepancy (1.0848596 difference). This discrepancy means that the selected individuals don't have as high genetic values as we might have hoped for from their phenotype values. Finally, comparing the mean genetic value in the selected individuals (11.0449445) and their progeny (11.0212806), we find a very good match, confirming that parents pass DNA and associated genetic values to the next generation.

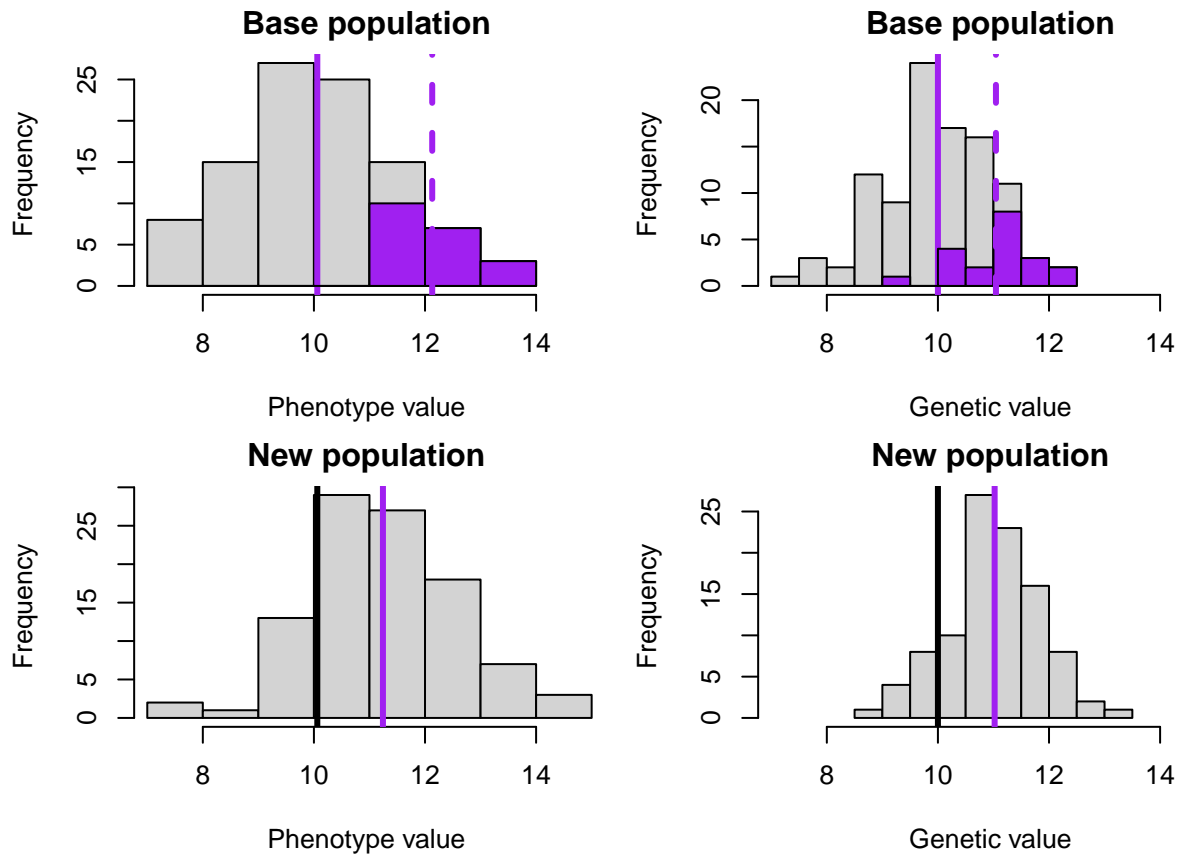
We will now visualise these differences in phenotype and genetic values and their means, for all three groups of individuals in each generation. As before, we will highlight with a purple solid line the overall mean of the population in both generations, with a purple dotted line the mean of the parents, and with a black line the overall mean of base population, so we can clearly see response to selection.

```
par(mfrow = c(2, 2),
    mar = c(4, 4, 2, 1))
# Phenotype values in the base population and the selected individuals
tmp = hist(pheno(basePop), xlim = phenoRange, xlab = "Phenotype value", main = "Base population")
abline(v = mean(pheno(basePop)), col = "purple", lty = 1, lwd = 3)
hist(pheno(basePopSelected), col = "purple", breaks = tmp$breaks, add = TRUE)
abline(v = mean(pheno(basePopSelected)), col = "purple", lty = 2, lwd = 3)

# Genetic values in the base population and the selected individuals
tmp = hist(gv(basePop), xlim = phenoRange, xlab = "Genetic value", main = "Base population")
abline(v = mean(gv(basePop)), col = "purple", lty = 1, lwd = 3)
hist(gv(basePopSelected), col = "purple", breaks = tmp$breaks, add = TRUE)
abline(v = mean(gv(basePopSelected)), col = "purple", lty = 2, lwd = 3)

# Phenotype values in the new population
hist(pheno(newPop), xlim = phenoRange, xlab = "Phenotype value", main = "New population")
abline(v = mean(pheno(basePop)), col = "black", lty = 1, lwd = 3)
abline(v = mean(pheno(newPop)), col = "purple", lty = 1, lwd = 3)

# Genetic values in the new population
hist(gv(newPop), xlim = phenoRange, xlab = "Genetic value", main = "New population")
abline(v = mean(gv(basePop)), col = "black", lty = 1, lwd = 3)
abline(v = mean(gv(newPop)), col = "purple", lty = 1, lwd = 3)
```



The resulting histograms clearly show the discrepancy in the distribution of phenotype values and genetic values for the selected individuals (in purple).

EXTRA: Breeder's equation

The above analysis can be wrapped in a simple formula called the breeder's equation (Lush, 1937; Kelly, 2011). This equation predicts the expected change in mean genetic value due to selection (ΔG) by multiplying selection differential (SD) with heritability (h^2); that is, only a fraction of the phenotype superiority is due to genetics:

$$\Delta G = SD \times h^2.$$

Selection differential (SD) is defined as the difference in mean phenotype values between selected individuals (\bar{P}_{sel}) and all individuals of a population (\bar{P}), $SD = \bar{P}_{sel} - \bar{P}$. Since the mean genetic value of the selected individuals is also the expected mean genetic value of their progeny, the breeder's equation can also predict the expected change in phenotype values between the generations. However, when working across generations we can only predict changes due the additive component of genetic values, which is what we have assumed in this simulation.

Let's test the breeder's equation on our simulated example.

```
# Selection differential
(selDiff = meanP(basePopSelected) - meanP(basePop))
```

```
## Trait1
## 2.068588
```

```
# Expected change in the mean of genetic values according to the Breeder's equation
(deltaGExpected = selDiff * heritability)
```

```
## Trait1
## 1.034294
```

```
# Observed change in the mean of genetic values from the simulation
(deltaGObserved = meanG(newPop) - meanG(basePop))
```

```
## Trait1
## 1.021281
```

The expected and observed change in mean genetic value due to selection did not match perfectly, but the difference isn't too large. You can convince yourself by running this vignette multiple times and compare the observed and expected values. And this is where breeding simulations shine. Namely, formulae such as the breeder's equation rest on certain assumptions. For example, it assumes that we are working with a large population and polygenic traits. With simulations we can test how sensitive such formulae are when we violate these assumptions. Furthermore, some formulae can be rather involved, while we can always summarise phenotype and genetic values in a simulation. For example, we can evaluate variance of phenotype and genotypic values in all three groups of individuals with the following code.

```
# Variance of phenotype and genetic values in the base population
c(varP(basePop), varG(basePop))
```

```
## [1] 2.027024 1.000000
```

```
# Variance of phenotype and genetic values in the parents
c(varP(basePopSelected), varG(basePopSelected))
```

```
## [1] 0.6836140 0.5112169
```

```
# Variance of phenotype and genetic values in the new population
c(varP(newPop), varG(newPop))
```

```
## [1] 1.8769913 0.6537975
```

EXTRA: Expanded breeder's equation

Breeder's equation can be expanded further to gain insight into the driving factors of selection. The expanded breeder's equation predicts the expected change in mean genetic value due to selection per generation (ΔG) by multiplying the intensity of selection (i), accuracy of selection (r), and standard deviation of genetic values (S_G ; when working across generations we can only predict change in additive genetic values and have to use standard deviation of breeding values S_A) (Lush, 1937):

$$\Delta G = i \times r \times S_G.$$

Intensity of selection (i) is defined as selection differential (SD) expressed in units of phenotype standard deviation (S_P), $i = SD/S_P$. Accuracy of selection (r) is defined as Pearson correlation between true genetic values (G) and estimated genetic values (\hat{G}), $cor(G, \hat{G})$. Estimated genetic values are given by the criterion we use to rank selection candidates. Here we used phenotype values.

Lets evaluate these components and expected change in mean genetic value.

```
# Intensity of selection
(i = selDiff / sqrt(varP(basePop)))
```

```
## Trait1
## Trait1 1.45293
```

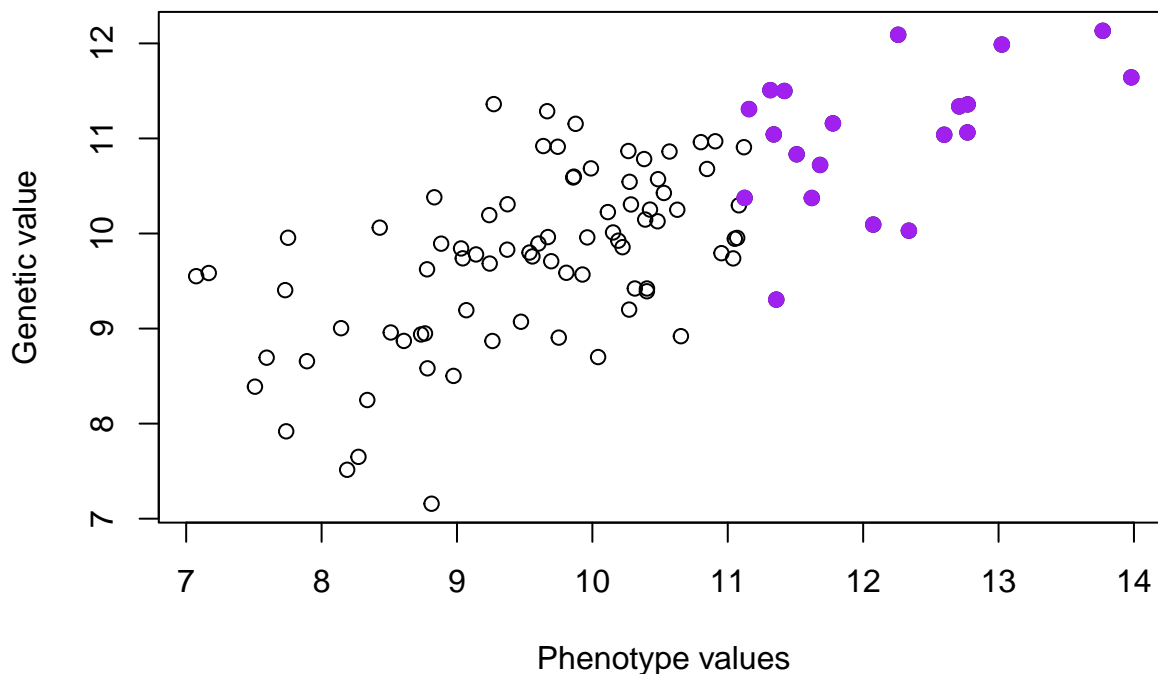
```
# alternatively we can use the AlphaSimR's selInt() function that is parametrised
# with the proportion of selected individuals and assumes a normal distribution
# for phenotype values (so you might get different values!)
selInt(p = nSelected / nInd(basePop))
```

```
## [1] 1.39981
```

```
# Accuracy of selection
(r = cor(gv(basePop), pheno(basePop)))
```

```
##          Trait1
## Trait1 0.6968421
```

```
par(mfrow = c(1, 1))
plot(y = gv(basePop), x = pheno(basePop),
     ylab = "Genetic value", xlab = "Phenotype values")
points(y = gv(basePopSelected), x = pheno(basePopSelected), col = "purple", pch = 19)
```



```
# Expected change in the mean of genetic values according to the expanded Breeder's equation
(deltaGExpected = i * r * sqrt(varG(basePop)))
```

```
##          Trait1
## Trait1 1.012462
```

```
# Observed change in the mean of genetic values from the simulation
(deltaGObserved = meanG(newPop) - meanG(basePop))
```

```
##          Trait1
## 1.021281
```

As before, we see some deviation between the expected value (according to the theory that rests on assumptions) and the observed value. A keen observer might have noticed that our predictions with the two versions of the Breeder's equation differ: 1.034294 for the basic Breeder's equation and 1.0124625 for the expanded Breeder's equation. The reason is again in the assumptions. In the first instance we used assumed heritability, which we have used as a parameter for the simulation. But, this need not be exactly the realised heritability in each

simulated population! Furthermore, we estimated accuracy of selection by correlating phenotype and genetic values. This again shows the benefit of simulations compared to deterministic predictions.

```
# Observed genetic gain
meanG(newPop) - meanG(basePop)

##      Trait1
## 1.021281

# Basic Breeder's equation with the assumed heritability
selDiff * heritability

##      Trait1
## 1.034294

# Basic Breeder's equation with a realised heritability
selDiff * varA(basePop) / varP(basePop)

##              Trait1
## Trait1 1.020505

# Expanded Breeder's equation with a realised genetic variance and estimated accuracy
i * r * sqrt(varG(basePop))

##              Trait1
## Trait1 1.012462

# Expanded Breeder's equation with a realised genetic variance and assumed accuracy
i * (sqrt(varG(basePop)) / sqrt(varP(basePop))) * sqrt(varG(basePop))

##              Trait1
## Trait1 1.020505
```

EXTRA: Selection over many generations

Finally, we can repeat the selection of superior individuals for many generations and track changes. To do this, we will first allocate vectors to store summaries for each generation. Let's perform selection over 10 generations and save mean and variance of phenotype and genetic values for all and selected individuals in each generation.

```
# Allocate vectors
nGenerations = 10 + 1 # +1 to store the starting generation
meanPA11 = numeric(nGenerations)
varPA11 = numeric(nGenerations)
meanGA11 = numeric(nGenerations)
varGA11 = numeric(nGenerations)
meanPSelected = numeric(nGenerations)
varPSelected = numeric(nGenerations)
meanGSelected = numeric(nGenerations)
varGSelected = numeric(nGenerations)

# Save the starting values
meanPA11[1] = meanP(basePop)
varPA11[1] = varP(basePop)
meanGA11[1] = meanG(basePop)
varGA11[1] = varG(basePop)
meanPSelected[1] = meanP(basePopSelected)
varPSelected[1] = varP(basePopSelected)
```

```
meanGSelected[1] = meanG(basePopSelected)
varGSelected[1] = varG(basePopSelected)
```

Now, we will repeat selection and crossing as we have done before between the `basePop` and `newPop`, but this time we will use R's for loop to simulate across many generations.

```
# To make the for loop below simpler we will make a copy of the object basePopSelected
newPopSelected = basePopSelected

# Selection over many generations
for (generation in 1:(nGenerations - 1)) {
  # Cross parents, phenotype progeny, and select new parents
  newPop = randCross(pop = newPopSelected, nCrosses = nInd(basePop))
  newPop = setPheno(newPop, h2 = heritability)
  newPopSelected = selectInd(pop = newPop,
                             nInd = nSelected,
                             use = "pheno")

  # Save summaries
  meanPAll[1 + generation] = meanP(newPop)
  varPAll[1 + generation] = varP(newPop)
  meanGAll[1 + generation] = meanG(newPop)
  varGAll[1 + generation] = varG(newPop)
  meanPSelected[1 + generation] = meanP(newPopSelected)
  varPSelected[1 + generation] = varP(newPopSelected)
  meanGSelected[1 + generation] = meanG(newPopSelected)
  varGSelected[1 + generation] = varG(newPopSelected)
}
```

Take a quick look at mean phenotype values.

```
meanPAll
```

```
## [1] 10.06122 10.94583 11.91749 12.61823 13.57164 14.20702 14.68455 15.63605
## [9] 16.04340 16.52045 17.51062
```

Let's plot how the means and variances change over generations.

```
meanRanges = range(c(meanPAll, meanPSelected, meanGAll, meanGSelected))
varRanges = range(c(varPAll, varPSelected, varGAll, varGSelected))

par(mfrow = c(2, 2),
    mar = c(4, 4, 1, 1))

# Plot mean of phenotype values over time
plot(x = 1:nGenerations, y = meanPAll, type = "l", col = "black", lwd = 3,
     xlab = "Generation", ylab = "Mean of phenotype values", ylim = meanRanges)
lines(x = 1:nGenerations, y = meanPSelected, type = "l", col = "purple", lty = 2, lwd = 3)
legend(x = "topleft", legend = c("All", "Selected"),
      lwd = 3, lty = c(1, 2), col = c("black", "purple"), bty = "n")

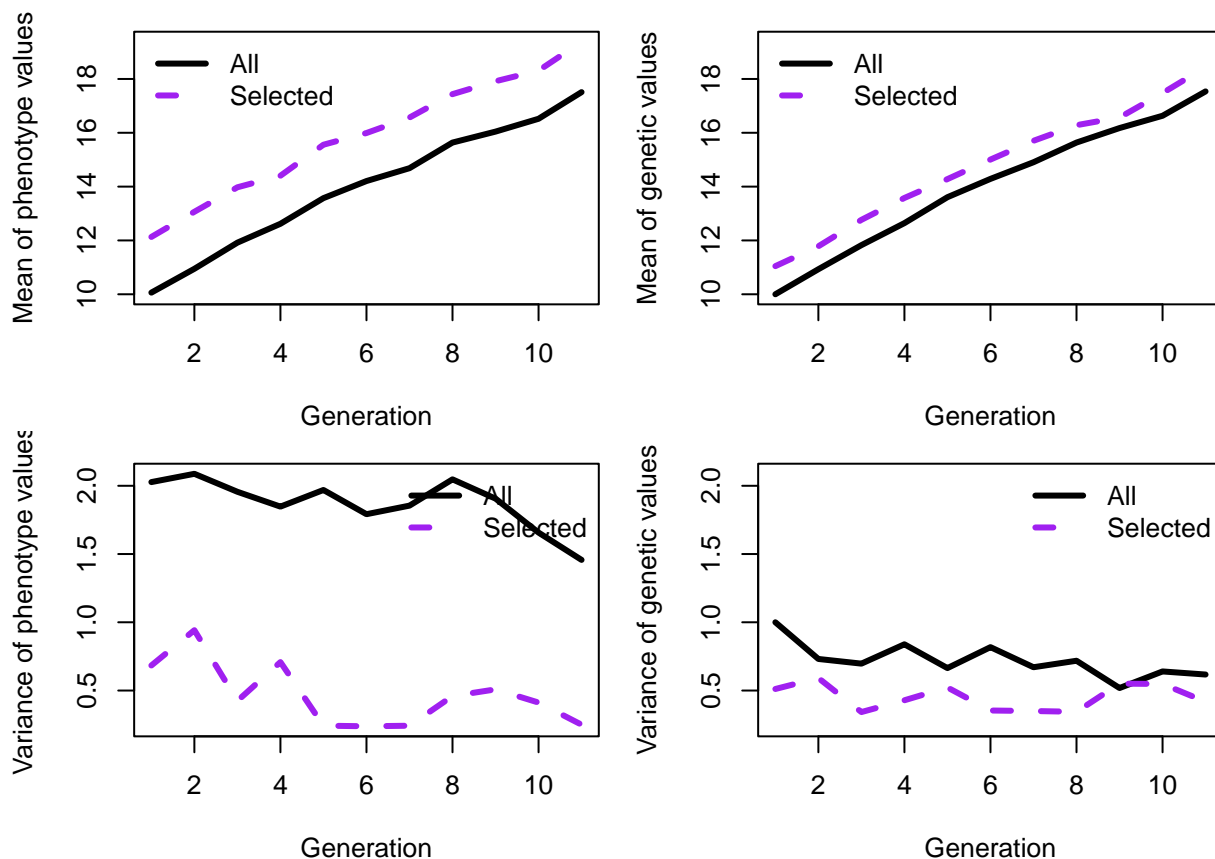
# Plot mean of genetic values over time
plot(x = 1:nGenerations, y = meanGAll, type = "l", col = "black", lwd = 3,
     xlab = "Generation", ylab = "Mean of genetic values", ylim = meanRanges)
lines(x = 1:nGenerations, y = meanGSelected, type = "l", col = "purple", lty = 2, lwd = 3)
legend(x = "topleft", legend = c("All", "Selected"),
      lwd = 3, lty = c(1, 2), col = c("black", "purple"), bty = "n")
```

```

# Plot variance of phenotype values over time
plot(x = 1:nGenerations, y = varPAll, type = "l", col = "black", lwd = 3,
     xlab = "Generation", ylab = "Variance of phenotype values", ylim = varRanges)
lines(x = 1:nGenerations, y = varPSelected, type = "l", col = "purple", lty = 2, lwd = 3)
legend(x = "topright", legend = c("All", "Selected"),
      lwd = 3, lty = c(1, 2), col = c("black", "purple"), bty = "n")

# Plot variance of genetic values over time
plot(x = 1:nGenerations, y = varGAll, type = "l", col = "black", lwd = 3,
     xlab = "Generation", ylab = "Variance of genetic values", ylim = varRanges)
lines(x = 1:nGenerations, y = varGSelected, type = "l", col = "purple", lty = 2, lwd = 3)
legend(x = "topright", legend = c("All", "Selected"),
      lwd = 3, lty = c(1, 2), col = c("black", "purple"), bty = "n")

```



We can see increase in the means of phenotype and genetic values over generations. We can also see decrease in variance of phenotype and genetic values over generations. As observed before, selected individuals (parents) have higher means and lower variances.

References

Kelly J.K. (2011) The breeder's equation. Nature Education Knowledge 4(5):5, <https://www.nature.com/scitable/knowledge/library/the-breeder-s-equation-24204828>

Lush J. L. (1937) Animal Breeding Plans. Iowa State Press. Ames, Iowa, <https://github.com/wintermind/Animal-Breeding-Plans>