Optimizing breeding schemes //

Manual / Index selection

Using a desired gains index for multi-trait selection with BLUEs

- This manual describes how to calculate and implement multi-trait selection for quantitative traits using a desired gains index with BLUEs or raw phenotypes.
- Published on 01/11/2023
- Excellenceinbreeding.org/toolbox/tools/eib-breeding-scheme-optimization-manuals

CGIAR | INITIATIVE ON Accelerated Breeding

cgiar.org/initiative/accelerated-breeding

Using a desired gains index for multi-trait selection with BLUEs

Author

Christian R. Werner, Dr / c.werner@cgiar.org

Scientist, Quantitative Genetics, Accelerated Breeding Initiative

Editor

Dorcus Gemenet, Dr / d.gemenet@cgiar.org

Quantitative Genetics Lead, Accelerated Breeding Initiative

Contributors

Giovanny Eduardo Covarrubias-Pazaran, Dr

Quantitative Genetics and Biometrics Lead, IRRI

Keith Gardner, Dr
Quantitative Geneticist, CIMMYT

Marlee Labroo, Dr
Principal Traited Genome Design Scientist, Bayer CropScience

CGIAR | INITIATIVE ON Accelerated Breeding

Table of Contents

Introduction

The desired gains index is an effective method for improving multiple quantitative traits at the same time. It allows breeders to specify the gain they want to make in each trait and achieve it within a minimum number of generations. The desired gains index does not require "relative economic weights", which are usually difficult to get for some or all of the traits of interest. Instead, index weights are calculated based on the specified desired gains (i.e., the breeder does not set the weights).

The amount of gain that a breeder would like to see in each trait (i.e., the desired gains) can be expressed in different ways. Breeders can, for example, define a Target Product Profile to set the minimum performance standards for a future variety. Alternatively, the breeding target can be defined in terms of standard deviations gain or percent gain per trait. We will demonstrate all three approaches in this document.

The desired gains index can be used with raw phenotypes, BLUEs, and BLUPs. Here, we explain how the desired gains index is generated with BLUEs, as described in the original paper by Pesek and Baker (1969). The same approach is used with raw phenotypes (although we do ***not*** recommend the use of uncorrected phenotypes for breeding purposes). The generation of the desired gains with BLUPs is covered in a separate manual.

In this manual, we demonstrate how to:

1. Define the breeding target and calculate the vector of desired gains (d).
2. Calculate the weights (w) for the desired gains index.
3. Calculate the index values for all individuals under selection.
4. Use the software DESIRE to tweak the desired gains vector d (optional).
5. Use the software DESIRE to derive the desired gains weights (w) without an a priori breeding target.

1. Define the breeding target and calculate the vector of desired gains (*d*)

To generate a desired gains index, we need 1) the BLUEs (or phenotypes) for all quantitative traits of interest, 2) the genetic variance-covariance matrix, and 3) a clearly defined breeding target. We use a data frame with BLUEs of 500 bean genotypes for yield (YLD), flowering time (FLT), and cooking time (CKT).

```
dim(blues) # Dimensions of the data frame: 500 genotypes, 3 traits.
```

```
## [1] 500   3
```

```
head(blues, 10)
```

```
##             YLD      FLT      CKT
## ID_1  0.6716844 34.93960 38.43541
## ID_2  0.6164249 33.99369 40.14006
## ID_3  0.9478266 35.30571 40.29665
## ID_4  0.6506884 33.61453 41.90746
## ID_5  0.8169898 34.02298 40.83401
## ID_6  0.7096956 37.09874 39.17443
## ID_7  0.7722052 30.94215 40.73555
## ID_8  0.9566101 35.33167 41.15993
## ID_9  0.8489520 33.65494 39.11739
## ID_10 0.9365495 39.75143 38.85600
```

Our breeding objectives are:

- Increase yield (t/ha),
- Stabilise flowering time at 35 days after sowing, and
- Reduce cooking time (min).

First, we need to calculate the population means and the genetic variance-covariance matrix (G). The genetic variance-covariance matrix must be obtained from a prior statistical analysis of the data set (with a linear mixed model). For demonstration, we obtained G in advance and omit the analysis here.

```
pop_means <- colMeans(blues)
pop_means
```

```
##        YLD        FLT        CKT
##  0.8006249 34.4874150 39.9613474
```

```
G
```

```
##            Trait1     Trait2     Trait3
## Trait1 0.01000000 0.06457976 0.04449847
## Trait2 0.06457976 2.00000000 0.48859796
## Trait3 0.04449847 0.48859796 1.50000000
```

We can now use the means and the variance-covariance matrix to derive the vector of desired gains (d) and calculate the index weights (w).

1.1 The breeding objective is defined in terms of a Target Product Profile

The Target Product Profile describes the performance of an optimal variety for all quantitative traits of interest in absolute values. The vector of desired gains (d) is simply the difference between your Target Product Profile and the population trait means.

```
TPP_1 <- c(1.2, 35, 37)    # Target Product Profile in absolute values.
```

```
d_1 <- TPP_1 - pop_means   # Desired gains vector in absolute values.
d_1
```

```
##        YLD        FLT        CKT
##  0.3993751  0.5125850 -2.9613474
```

1.2 The breeding objective is defined in terms of standard deviations gain

If we define our breeding objective in terms of standard deviations gain to be made, we eventually need to turn this objective into absolute values. Hence, this approach is not too different from using a Target Product Profile, but some may find it more intuitive to define the breeding target.

Let us assume we want to improve yield by 2 standard deviations, fix flowering time at 35 days after sowing, and reduce cooking time by 1 standard deviation. We then our desired gains vector (d) in the following way:

```
target_in_SD <- c(2, 0, -1)      # Desired gains in standard deviations for YLD and CKT.
SD <- sqrt(diag(G))              # Extract standard deviations.
d_2 <- target_in_SD * SD         # Desired gains for YLD and CKT in absolute values.
d_2[2] <- 35 - pop_means[2]      # Target value for FLT fixed at 35.
```

Note that we defined gains in terms of standard deviations only for yield and cooking time. Flowering time was fixed at an absolute value of 35 days after sowing.

The Target Product Profile resulting from our breeding objective is then the sum of the population means and the desired gains vector:

```
TPP_2 <- pop_means + d_2   # Target Product Profile.
TPP_2
```

```
##       YLD       FLT       CKT
##  1.000625 35.000000 38.736603
```

1.3 The breeding objective is defined in terms of percent gain

Another intuitive way to define the breeding objective is in terms of percent gain per trait to be made. This approach is very similar to using standard deviations, since the gains expressed in percent also have to be turned into absolute values. Let's assume now that we want to improve yield by 20% and reduce cooking time by 10%. Again, we want to fix flowering time at 35 days after sowing.

```
target_in_percent <- c(0.2, 0, -0.1)     # Improve YLD by 20% and reduce CKT by 10%.
d_3 <- pop_means * target_in_percent     # Calculate desired gains (absolute values).
d_3[2] <- 35 - pop_means[2]              # Fix FLT at 35 days after sowing.
```

The Target Product Profile resulting from our breeding objective is the sum of the population means and the desired gains vector:

```
TPP_3 <- pop_means + d_3   # Target Product Profile.
TPP_3
```

```
##        YLD        FLT        CKT
##  0.9607499 35.0000000 35.9652127
```

2. Calculate the weights (w) for the desired gains index

We now can derive the weights (w) for the desired gains index. Here, we use the vector of desired gains *d_1* that we got using the Target Product Profile as defined above. However, the desired gains vectors *d_2* and *d_3* could also be used instead.

```
w <- solve(G) %*% d_1      # desired gains index weights w -
                           # to achieve the Target Product Profile.
```

3. Calculate the index values for all individuals under selection

We calculate the index values for the 500 bean genotypes by multiplying the BLUEs with the desired gains weights (w).

```
index_values <- blues %*% w
head(index_values, 5)
```

```
##           [,1]
## ID_1 -123.7801
## ID_2 -132.3203
## ID_3 -113.7509
## ID_4 -136.0985
## ID_5 -122.5256
```

The genotypes with the highest index values are our best parental candidates to reach the Target Product Profile. It is important that we do ***not*** combine the index values with any other selection criteria, such as individual trait measurements.

```
index_values <- data.frame(ID = rownames(index_values),
                           Index = as.vector(index_values))
```

```
index_values <- index_values[order(-index_values$Index), ]


index_values[1:5, ]   # The five genotypes with the highest index values.
```

```
##         ID     Index
## 34   ID_34 -74.32997
## 191 ID_191 -76.89297
## 254 ID_254 -81.81605
## 117 ID_117 -85.35574
## 426 ID_426 -85.63484
```

4. Use the software DESIRE to tweak the desired gains vector *d* (optional)

We can use the stand-alone software DESIRE to investigate alternative Target Product Profiles. This might help with the identification of slightly changed, preferable trait improvement ratios. We will use our present Target Product Profile as a starting point. Then, alternative Target Product Profiles can be easily explored with DESIRE's graphical user interface.

The software can be downloaded from here: https://bkinghor.une.edu.au/desire.htm

The DESIRE input text file requires 1) the "economic weights" (a) of the traits, 2) the trait heritabilities, 3) the standard deviations, 4) the phenotypic correlations, and 5) the genetic correlations between the traits. We can get all this information using our data frame and the G matrix.

DESIRE, however, cannot directly read in our present Target Product Profile as a starting point. Therefore, we use a mathematical relationship between the desired gains index and the Smith-Hazel index, which allows us to express the desired gains weights (w) in terms of economic weights (a), and vice versa. This relationship is shown in Brascamp (1984). However, it is important to understand that these "economic weights" we derive using w have no real economic meaning. They simply represent the weights that we need to assign to the index traits in order to get our desired improvement ratio.

```r
P <- cov(blues)                          # Phenotypic variance-covariance matrix,
G_inv <- solve(G)                        # Inverse of matrix G.

a <- G_inv %*% P %*% G_inv %*% d_1       # Turning desired gains weighs into economic weights
round(a, 3)                              # -> Starting "economic weights",
```

```
##           [,1]
## Trait1 385.021
## Trait2  -9.920
## Trait3 -12.505
```

```r
round(diag(G) / diag(P), 3)   # Heritabilities.
```

```
## Trait1 Trait2 Trait3
##  0.217  0.384  0.677
```

```r
round(sqrt(diag(P)), 3)     # Standard deviations of the BLUEs.
```

```
##   YLD   FLT   CKT
## 0.215 2.283 1.489
```

DESIRE needs a correlation matrix as input with the phenotypic correlations (correlations of the BLUEs) on the upper diagonal, and the genetic correlations on the lower diagonal.

```r
rp <- cov2cor(P)              # Phenotypic correlations.
rg <- cov2cor(G)              # Genotypic correlations.
cor_mat <- rp
cor_mat[lower.tri(cor_mat)] <- rg[lower.tri(rg)]
round(cor_mat, 3)                                    # Correlation matrix for DESIRE.
```

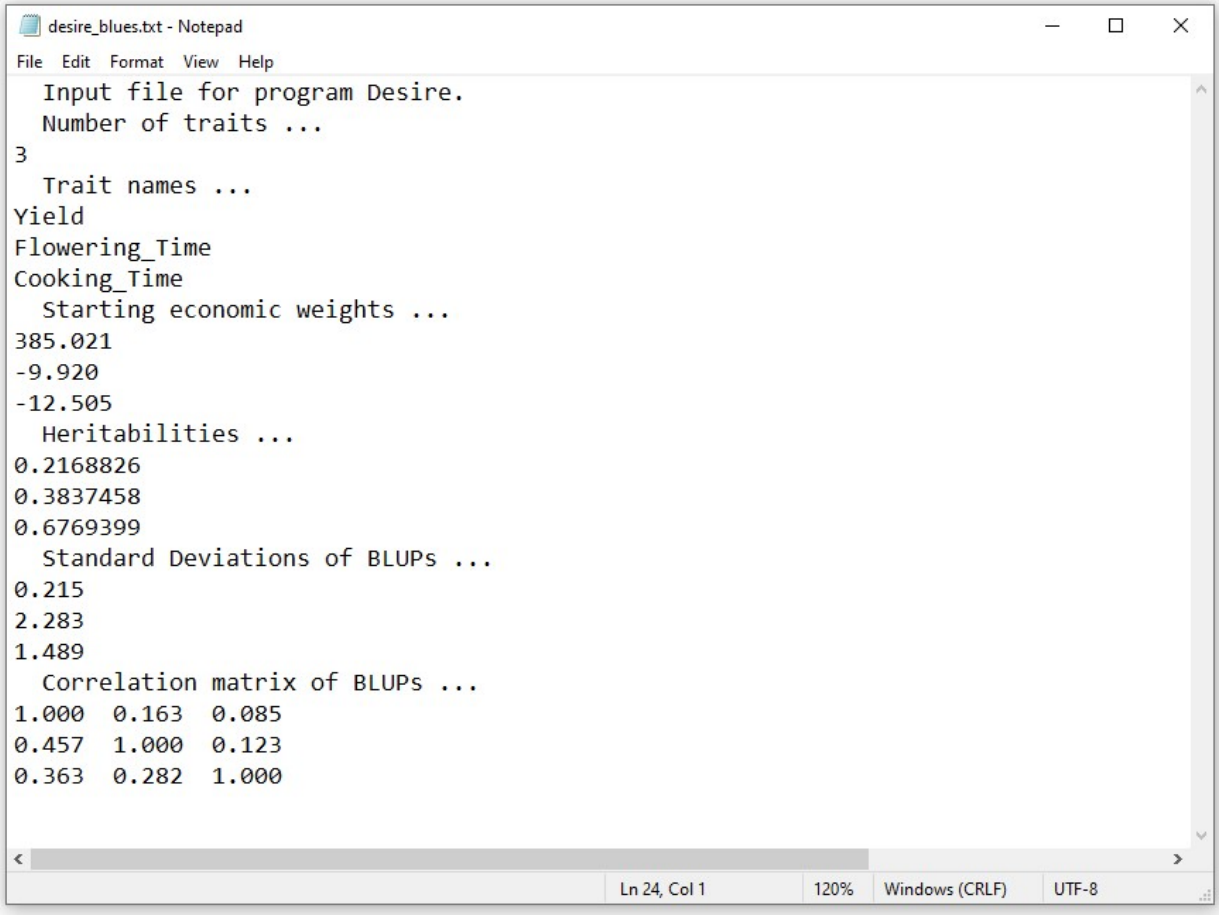
```
##       YLD   FLT   CKT
## YLD 1.000 0.163 0.085
## FLT 0.457 1.000 0.123
## CKT 0.363 0.282 1.000
```

NOTE: The R stats function `cov()` calculates the sample variances of and covariances between the traits (i.e., the denominator *n - 1* is used). However, since we generate the selection index for a closed set of individuals in our breeding population, the phenotypic population variances and covariances should be calculated instead (the denominator *n* is used). Therefore, we recommend to use a function such as `popVar()` in 'AlphaSimR' (although the differences may be small at large population sizes).

```
AlphaSimR::popVar(blues)
```

```
##             [,1]       [,2]       [,3]
## [1,] 0.04601567 0.07950552 0.02711284
## [2,] 0.07950552 5.20135953 0.41878670
## [3,] 0.02711284 0.41878670 2.21142228
```

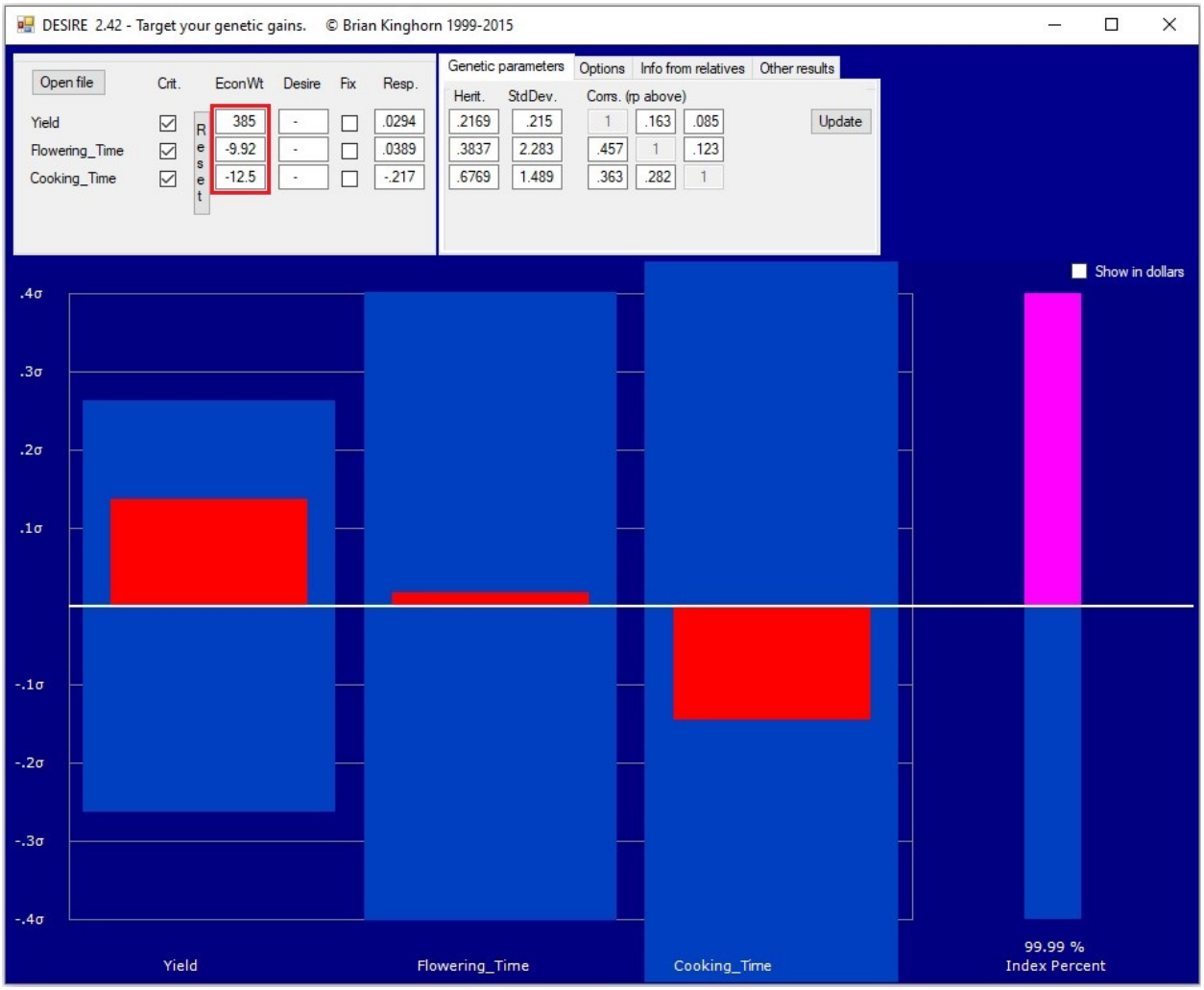
The input file for DESIRE has to be prepared manually by inserting the values we obtained above.



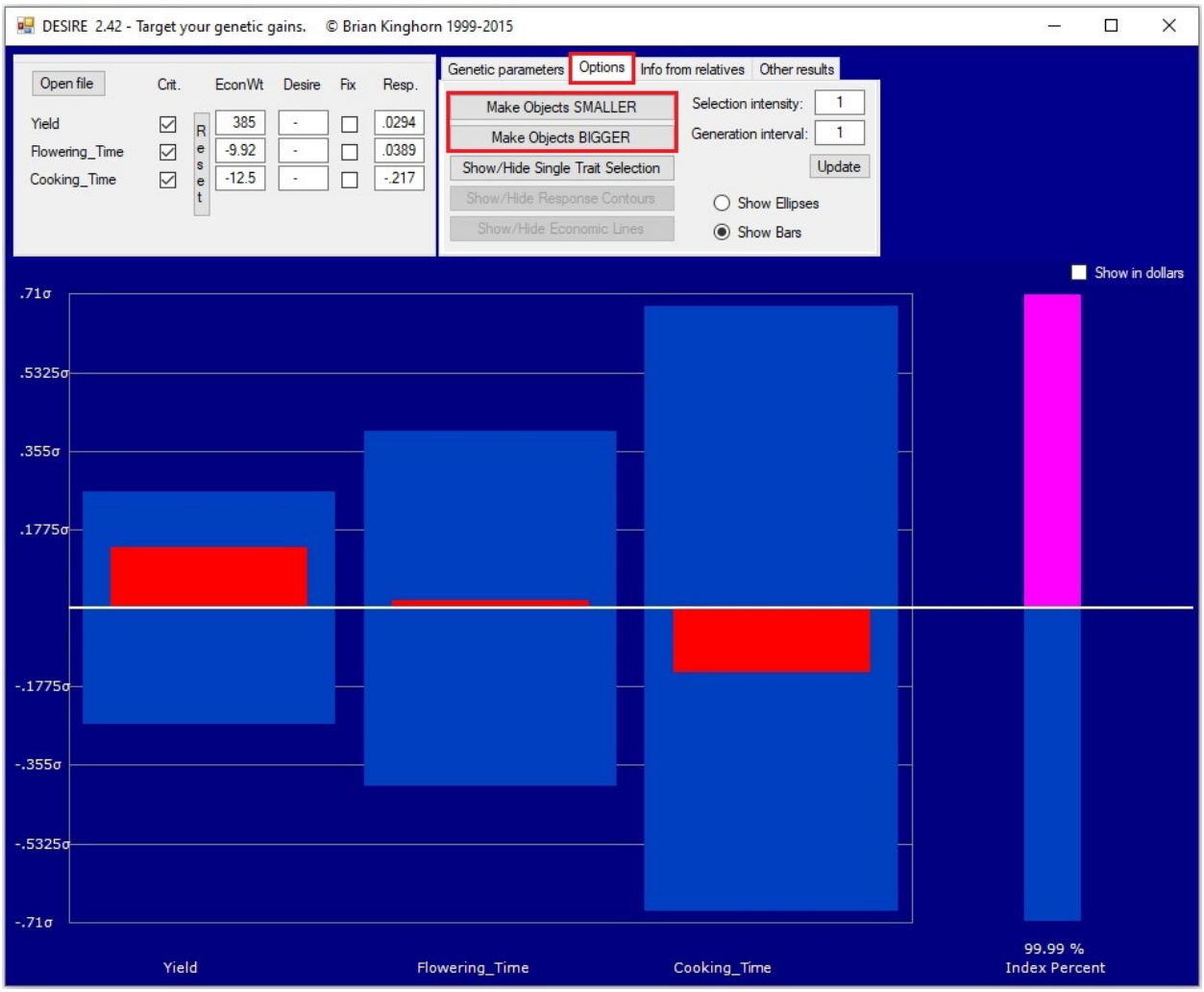
We now run the *Desire.exe* executable and load the input file. The program provides various options of which we will only need a subset to tweak our Target Product Profile.

The column "EconWt" contains the starting "economic values", which we derived based on the desired gains weights (w). The red bar plots represent the improvement ratio coming from our Target Product Profile in standard deviations.

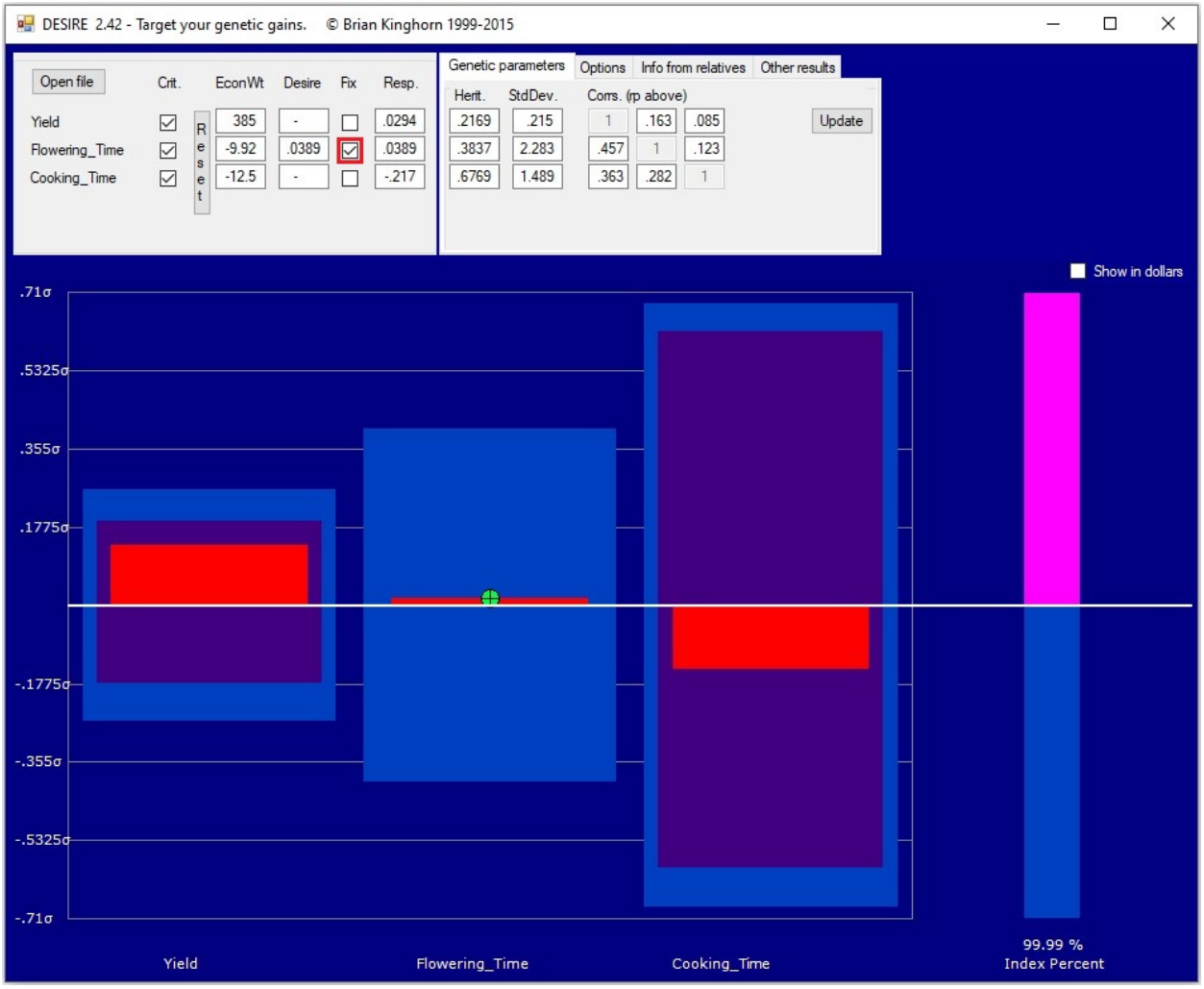
The column "Resp" shows the expected response to selection in absolute values. These values will be a fraction or multiple of the desired gains (d) we want to make. Their scaling depends on the selection intensity and the generation interval. The two parameters can be set under "Options" to get a prediction of the gain expected in the next generation based on the breeder's equation. The desired gains vector, on the other hand, has no specific time horizon associated with it. For now, we leave the selection intensity and the generation interval at the default values of 1.



First of all, we click on the “Options” rider and adjust the the size of the red bars using “Make Objects SMALLER” or “Make Objects BIGGER”. If desired, the selection intensity and the generation interval can also be adjusted here (confirm with “update”).



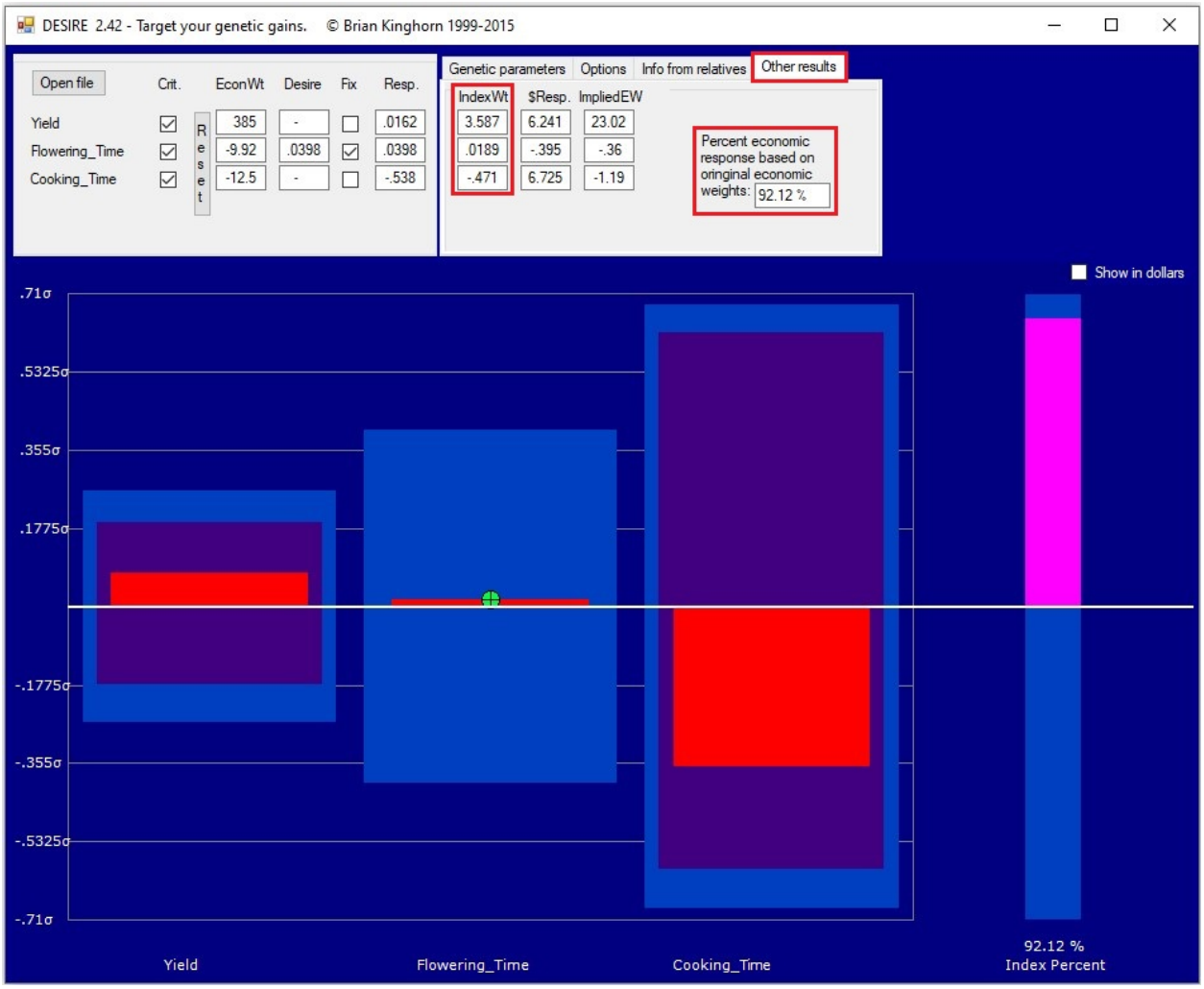
Next, before we change the height of any bar, we need to fix Flowering Time. This is necessary to achieve the desired target of 35 days after sowing as defined above.



Now, we can explore alternative improvement ratios for yield and cooking time by clicking on the red bars. When we change the Target Product Profile, DESIRE automatically takes the genetic correlations between the three traits into account. Therefore, we will observe a change in cooking time when we change yield, and vice versa.

We decide to increase yield, which results in a penalty on cooking time reduction.

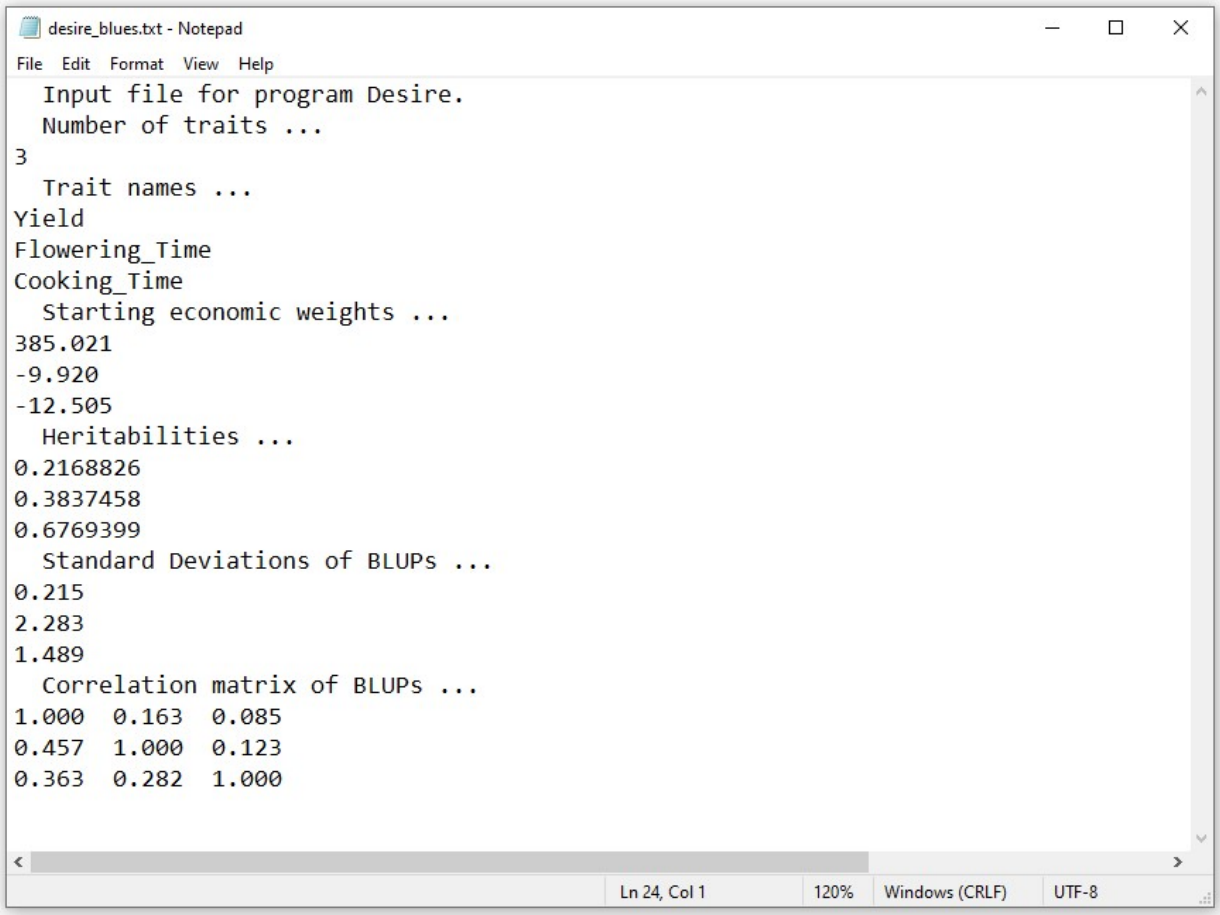
After we have found our new improvement ratio, we extract the new desired gains weights from the column "IndexWt" in "Other results" and multiply them with the BLUEs in our data frame. The genotypes with the highest index values are the best parental candidates to reach our new Target Product Profile.



NOTE: the total economic value of our new target product profile, indicated by the pink bar ("Index Percent"), has slightly decreased. This shall not cause us any concern and can be ignored. DESIRE calculates the total economic value of the Target Product Profile based on the economic values of the traits we provided with the input file. It assumes that these are the true economic values, and every change in the improvement ratio will reduce the total economic value of the Target Product Profile. However, the economic values we provided for the three traits have no real economic meaning, and the changes in the pink bar are irrelevant for our purpose.

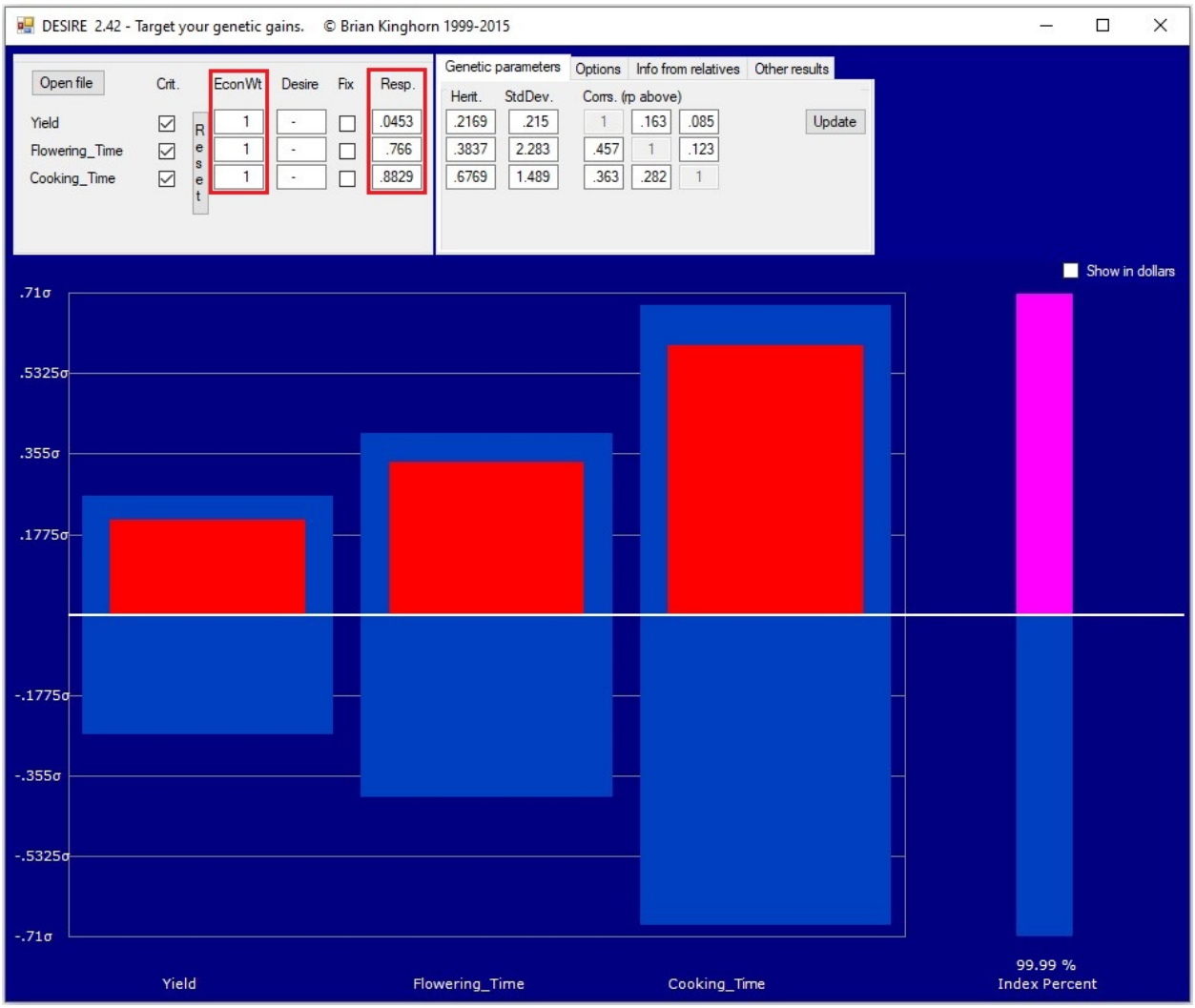
5. Use the software DESIRE to derive the desired gains vector without an a priori breeding target

If we have no Target Product Profile to start with, we can try to identify our breeding target by exploring our data. Therefore, we simply set all the "economic values" to 1 in the DESIRE input file.

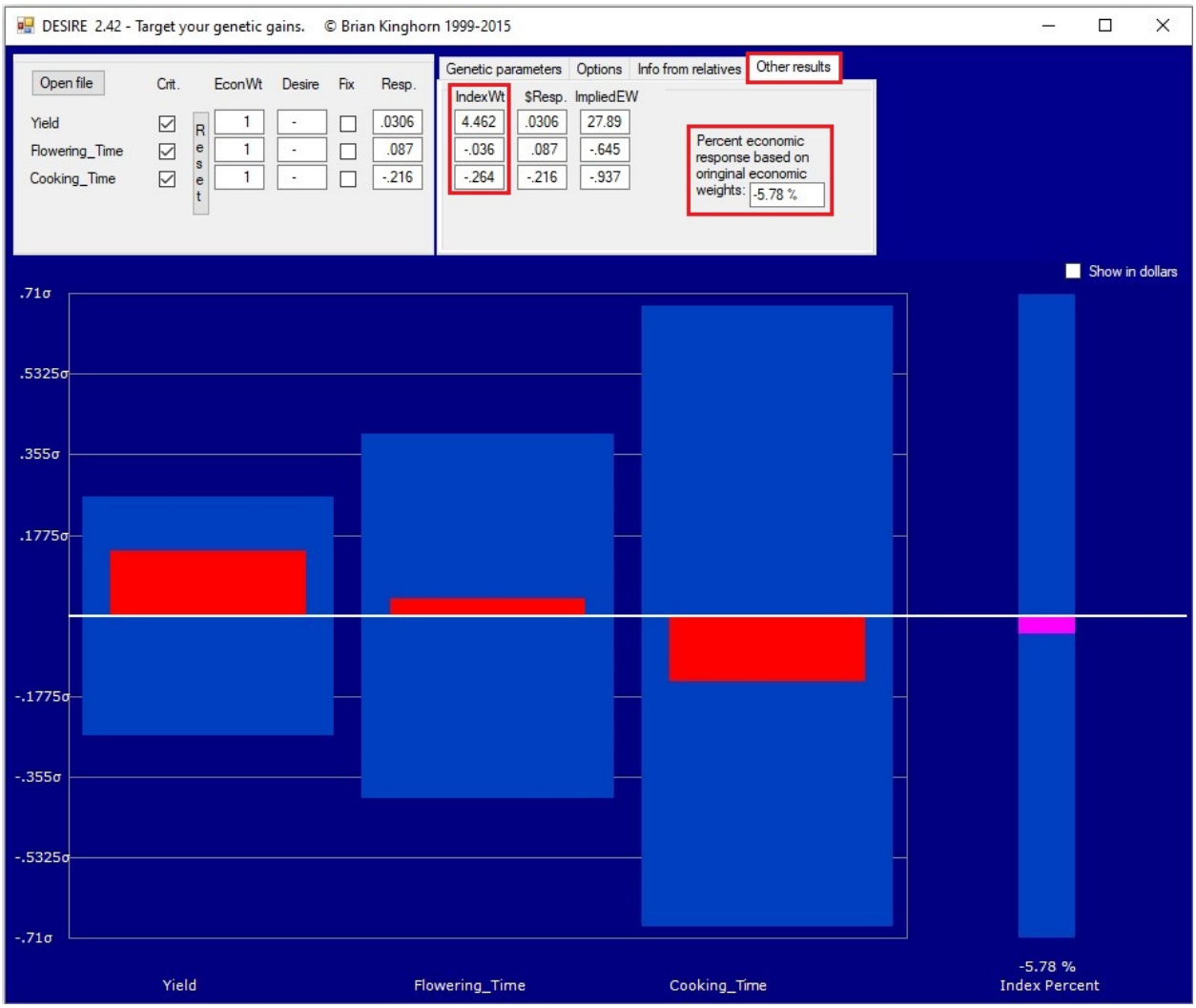


We can now explore alternative Target Product Profile by clicking on the red bars. The column "Resp" shows the expected response to selection in the next generation, assuming a selection intensity of 1. We adjust the traits until we achieve a satisfactory improvement ratio, which defines our Target Product Profile. This expected response to selection then is our desired gains vector (d).

Some users may find it more intuitive to set a Target Product Profile for a generation interval longer than one year. In this case, the generation interval can be increased under "Options", and the expected response to selection will be scaled by the number of generations.



After we have found our desired response to selection, we extract the desired gains weights from the column "IndexWt" in "Other results" and multiply them with the BLUEs in our data frame. The genotypes with the highest index values are the best parental candidates to reach our new Target Product Profile.



NOTE: the total economic value of our new target product profile, indicated by the pink bar ("Index Percent"), has decreased drastically. This shall not cause us any concern and can be ignored. DESIRE calculates the total economic value of the Target Product Profile based on the economic values of the traits we provided with the input file. It assumes that these are the true economic values, and every change in the improvement ratio will reduce the total economic value of the Target Product Profile. However, the economic values of 1 that we provided for the three traits have no real economic meaning, and the changes in the pink bar are irrelevant for our purpose.