Published April 28, 2015

Optimizing Resource Allocation for Multistage Selection in Plant Breeding with R Package *Selectiongain*

Xuefei Mi, H. Friedrich Utz, Frank Technow, and Albrecht E. Melchinger*

ABSTRACT

Most plant-breeding programs involve multistage selection, but hitherto no flexible and computationally efficient software has been freely available for calculating the selection gain and optimizing the allocation of resources under these conditions. Here we describe the use of the newly-developed R package *selectiongain* that enables (i) calculation of the selection gain for up to 20 selection stages and (ii) a grid search of the optimum allocation of resources (number of candidates, locations, and testers) for given costs and correlations among the selection criteria in each stage. The capabilities of our software are demonstrated with an example taken from the literature on selecting doubled haploid lines in maize across three stages.

Institute of Plant Breeding, Seed Science and Population Genetics, Univ. of Hohenheim, D-70593 Stuttgart, Germany. Received 24 Oct. 2013. *Corresponding author (melchinger@uni-hohenheim.de).

Abbreviations: ΔG, selection gain; DH, doubled haploid; GCA, general combining ability; MAS, marker-assisted selection; PE, plot equivalent; PS, phenotypic selection; SCA, specific combining ability; TCP, test cross performance.

Selection in plant breeding programs is usually a multistage process with different tests and selection decisions in successive stages. The main concerns of multistage selection are the choice of the number of candidates to be evaluated in each stage and the precision of their estimated breeding values, which is determined by the number of test locations and replications for phenotypic selection (PS) or the marker coverage in marker-assisted selection (MAS). A well-founded criterion for solving these decision problems is the expected selection gain (ΔG) (Bernardo, 2002). Therefore, the calculation of ΔG under various scenarios and identification of the optimum allocation of resources that maximize ΔG are utterly important for the planning of breeding programs.

Cochran (1951) and Finney (1956) developed the theory of multistage selection and derived analytical solutions to calculate ΔG. For one-sided multivariate truncation selection, Tallis (1961) expressed the gain and genetic variance after selection by moment-generating functions. Alternatively, ΔG can be determined by stochastic simulations, as exemplified by Wang et al. (2004) for wheat (*Triticum aestivum* L.) and Longin et al. (2007) for maize (*Zea mays* L.) using simulation tools reviewed by Sun et al. (2011). However, calculating ΔG by simulations is a computationally-intensive task,

Published in Crop Sci. 54:1413–1418 (2014).
doi: 10.2135/cropsci2013.10.0699

given the large number of runs needed for warranting a sufficient precision. Consequently, this approach is not feasible when numerous parameters must be varied in a search for the optimum allocation of resources.

To the best of our knowledge, no freely-available software has been published to calculate ΔG for multistage selection, despite the overwhelming importance of this problem in practice. Therefore, we developed the R package *selectiongain* (Mi et al., 2012) to fill this gap for a broad range of scenarios in plant breeding. An additional function of our software allows a determination of the optimum allocation of resources in a breeding program under a limited budget. In this treatise, we demonstrate the application of our software for a three-stage selection problem with doubled haploid (DH) lines in maize described by Heffner et al. (2010).

SOFTWARE *SELECTIONGAIN*

Our package *selectiongain* is embedded in the statistical software R (R Development Core Team, 2012) and relies heavily on the R package *mvtnorm* (Genz et al., 2013). We calculate ΔG by a sum of series of integrals from a multivariate normal distribution, which are solved by *mvtnorm*. Two algorithms differing in accuracy and computational speed are available from *mvtnorm* for calculating the normal integrals under one-sided truncation selection. 1. The Genz and Bretz algorithm is very fast and can handle up to 1000 selection stages with a quasi-Monte Carlo algorithm, but is less stable (i.e., similar but not identical optimal allocations will be obtained in replicated calculations). 2. The Miwa algorithm carries out the computations with very high precision but for no more than 20 stages (Mi et al., 2009). In conventional breeding, usually <20 stages are involved and for this reason, we recommend using the second algorithm as default. However, if numerous resistance and quality traits as well as markers, expression, and metabolic data are selected sequentially, the number of selection stages may surpass 20. Under this condition, users are recommended to apply the Genz and Bretz algorithm.

Our package can also be used to optimize the allocation of resources (i.e., determine the number of candidates, locations, and testers in successive stages of PS or set the choice of marker density in successive stages of MAS that maximize ΔG under a fixed budget). Below, we illustrate the basic functions for calculating the correlation matrix, truncation points, and the total selection gain in all stages. On the basis of these functions, we subsequently describe a grid search method for a determination of the optimum allocation of resources.

A Case Study with Three-stage Selection

Our example, taken from Heffner et al. (2010), deals with a selection of DH lines in maize for general combining ability (GCA) of grain yield in three stages. In the first stage, DH lines are selected by MAS on the basis of a marker index score determined in previous breeding cycles. In the second and third stages, the selected candidates are subjected to PS for GCA on the basis of their testcross performance (TCP) with one and five tester lines, respectively. Assumptions on the variance components for the target trait were taken from Longin et al. (2007), who used variance components estimates calculated from several years of multilocation trials conducted by the maize breeding program of the University of Hohenheim, Stuttgart, Germany:

- GCA effects: $V_g = 0.40$
- GCA × location interactions: $V_{gl} = 0.20$
- GCA × year interactions: $V_{gy} = 0.20$
- GCA × location × year interactions: $V_{gly} = 0.40$
- Specific combining ability (SCA) effects: $V_s = 0.20$
- SCA × location interactions: $V_{sl} = 0.10$
- SCA × year interactions: $V_{sy} = 0.10$
- SCA × location × year interactions: $V_{sly} = 0.20$
- Plot error: $V_e = 2.00$
- The genetic correlation of the MAS index score with the GCA effects is *maseff* = 0.40.

For users designing their own experiments, parameter values for planning future selection experiments can be obtained from previous experiments conducted with comparable germplasm in comparable environments. For testing the sensitivity of the optimization to minor misspecification of the parameters, the optimization should be performed across a grid of values reflecting the uncertainty in the parameter estimates.

Calculation of the Selection Gain for a Given Allocation

As an example for calculating ΔG for a given allocation, we use the numbers reported by Heffner et al. (2010) to yield maximum selection gain:

- Stage 1: N_1 = 4500 DH lines are produced and genotyped for MAS.
- Stage 2: N_2 = 919 DH lines selected in the first stage are evaluated for TCP with T_2 = 1 tester in L_2 = 3 locations with R_2 = 1 replications.
- Stage 3: N_3 = 45 DH lines selected in the second stage are evaluated for TCP with T_3 = 5 different testers in L_3 = 8 locations with R_3 = 1 replication.
- A priori fixed goal: N_f = 10 finally selected DH lines after the three selection stages.

This section explains the R code for calculating ΔG under the above scenario. First, the package *selectiongain* must be loaded and values for the variance components must be inserted:

```
 > library(selectiongain)

 > VCGCAandError = c(0.40,0.20,0.20,0.40,2.00)

 > VCSCA = c(0.20,0.10,0.10,0.20)
```

Second, the matrix of correlations among the selection criteria (SC_i) in the three stages as well as their correlations to the target trait (GCA effects in our example) must be calculated. The corresponding equations for calculating these correlations on the basis of the variance components, numbers of locations, testers, and replications can be found in Longin et al. (2007).

```
 > corr.matrix = multistagecor (maseff = 0.40,

VGCAandE = VCGCAandError, VSCA = VCSCA, T =
              c(1,1,5), L = c(1,3,8), Rep = c(1,1,1))
```

The correlation matrix is printed out as

```
[,GCA][,SC1][,SC2][,SC3]

[GCA,] 1.000 0.400 0.463 0.710

[SC1,] 0.400 1.000 0.185 0.284

[SC2,] 0.463 0.185 1.000 0.384

[SC3,] 0.710 0.284 0.384 1.000
```

The truncation points in a standard multivariate normal distribution with the selected fractions N_2/N_1, N_3/N_2, and N_f/N_3 are calculated automatically

```
 > N1 = 4500;N2 = 919;N3 = 45;Nf = 10

 > Q = multistagetp(c(N2/N1,N3/N2,Nf/N3), corr =
      corr.matrix)
```

and, in our case, yield the following truncation points

```
[1] 0.8266342 1.8899580 1.8154063
```

With the aid of the truncation points and correlation matrix among the selection criteria SC_i as well as the genetic standard deviation of GCA effects (i.e., $\sqrt{0.40}$ in our example), the ΔG for all selection stages is calculated by the Genz and Bretz algorithm by default, whereas the computationally-intensive Miwa algorithm must be chosen by setting alg = Miwa():

```
 > Gain = multistagegain(Q = Q, corr = corr.matrix,

alg = Miwa())×(0.40^0.5)
```

and yields

```
 > Gain

1.359.
```

The calculated ΔG = 1.359 is slightly different from the value 1.340 reported by Heffner et al. (2010). The explanation for this difference is based on the fact that these authors assumed that the test locations in the second stage were different from those in the third stage, whereas in practice, the test locations used in the second stage are generally a subset of the locations employed in the third stage, which affects the correlation between the selection criteria (by setting covtype = "Heffner" in function multistagecor, the covariance assumed by Heffner et al. (2010) will be calculated and the default of covtype is "LonginII"). Furthermore, Heffner et al. (2010) used the integrative algorithm developed by Bulmer (1971) to approximate the genetic variance after selection for computing the selection gain in the second and third stage. When we tried Bulmer's approximation, the result was identical to Heffner et al. (2010). In contrast, our three-stage selection gain is calculated from a multivariate integral without these approximations.

Optimization of the Number of Candidates, Locations, and Testers

To determine the optimum allocation of resources yielding maximum ΔG under a given budget, we need a cost function summing up all costs in each stage of the breeding program. A general linear cost function for the three stages in our example is given by

$$Budget = \sum_{i=1}^{3} B_i ,$$

where $B_i = N_i\left(CostProd_i + \delta_i CostTest_i\right)$ are the costs in stage i and $\delta_i = 1$ for $i = 1(\text{MAS})$ and $\delta_i = L_i T_i R_i$ for i = 2,3 (PS) in our example.

Here, all costs are expressed in terms of plot equivalent (PE) units:

- $CostProd_i$ refers to the costs of producing a candidate to be evaluated in selection stage i (i.e., all costs of producing and multiplying a DH line as well as producing testcross seeds),
- $CostTest_i$ refers to the costs of genotyping for MAS or costs of one field plot for PS in stage i.

Heffner et al. (2010) based their calculations on the following costs:

- Stage 1: $CostProd_1$ = 0.5 PE and $CostTest_1$ = 0.5 PE for marker assays.
- Stage 2: $CostProd_2$ = 1 PE and $CostTest_2$ = 1 PE.
- Stage 3: $CostProd_3$ = 1 PE and $CostTest_3$ = 1 PE.

The total budget for the entire breeding program was fixed at 10,021 PE.

The optimum allocation of resources is found by a grid search (Kim, 1997) across the permissible space of numbers of candidates (N_i), locations (L_i), and testers (T_i), where i = 2, 3, for given input parameters *(*i.e., correlation of the marker index score with the GCA effects, variance components, costs in each stage, number N_f of finally selected DH lines, and total budget). This grid search is implemented in the function *multistageoptimum.search*.

Note that for each permissible allocation, N_1 is fully determined by the other variables and a budget constraint. The user can specify the lower and upper limit as well as the grid width for each variable. For example, the grid of N_2 can be set as N2grid = c(11, 1211, 5). The search starts at the lower limit and increases successively by the grid width in the corresponding loop until the upper limit is reached. While the lower limit for N_i is always bounded by N_f, usually smaller values for the upper limit and a smaller grid width are chosen in advanced selection stages i compared with the earlier ones. For L_i and T_i, the grid width is usually chosen to be 1. In our example, the search runs through 90,774 grid points to determine the optimum allocation with a fairly coarse grid. The ΔG for each grid point is:

```
 > result = multistageoptimum.search (maseff = 0.4,

VGCA = VCGCAandError, VSCA = VCSCA,
CostProd = c(0.5,1,1),  CostTest = c(0.5,1,1), Nf = 10,
Budget = 10021, N2grid = c(11, 1211, 30), N3grid =
c(11, 211, 5), L2 grid = c(1,3,1),

L3 grid = c(6,8,1), T2 grid = c(1,2,1), T3 grid =
c(3,5,1),R2 = 1, R3 = 1, alg = Miwa(), detail = FALSE,
fig = FALSE)
```

The end result is the optimum allocation and the budget spent in each stage as well as the corresponding maximum of ΔG:

```
Nf N1 N2 N3 L2 L3 T2 T3 R2 R3 B1 B2 B3 Budget Gain

10 5543 341 51 3 8 2 5 1 1 5543 2387 2091 10021 1.435
```

By adding the parameter detail = TRUE in the above command line, a table with the results for all grid points of L_i and T_i as well as the grid points for N_2 and N_3 yielding the local maximum of ΔG under the specific setting of L_i and T_i are provided as output (Supplemental Fig. S1). This can be used for further purposes such as producing contour plots or performing a more detailed search within a refined grid. As an example, we generate a contour plot for N_2 and N_3 by including the additional parameter fig = TRUE in the above command line for T_2 = 2, T_3 = 5, L_2 = 3, and L_3 = 8 (Fig. 1). Around the maximum point, we obtain a flat triangular region with ΔG > 1.393 for N_2 ranging from 100 to 1000 and N_3 ranging from 20 to 150. The maximum ΔG found with this coarse grid (indicated by the bullet in Fig. 1) amounts to ΔG = 1.434 and is obtained for the allocation parameters N_1 = 5666, N_2 = 361, and N_3 = 71. Subsequently, we could perform a more detailed search in a smaller region by setting the grid width to the minimum value of 1.

DISCUSSION

The R package *selectiongain* was developed for a precise and fast calculation of ΔG for a large number of selection stages and for optimizing the allocations across five stages. Its usage has been demonstrated in the case of selecting DH lines across three stages including one stage of MAS. Other breeding schemes in maize or any other crop can easily be optimized with no or minor modifications of the functions in our R package for three-stage selection. The software can be applied to plant breeding programs based purely on PS or on MAS, including genomic selection, but can also be employed for a combination of both approaches. For example, a breeder could initially screen a large number of candidates with a low-density, low-cost marker array that has only moderate prediction accuracy for the target trait(s) in the first stage. In the second stage, the selected candidates could be subjected to a more expensive genotyping assay yielding higher prediction accuracies.

The maximum number of selection stages that can be handled by the *selectiongain* package is limited by the choice of the *pmvnorm* function in the package *mvtnorm* (Genz et al., 2013). For <20 selection stages (usually breeding programs have <10 selection stages), the Miwa algorithm is recommended for the calculation of the integral under one-sided truncation selection. The Miwa algorithm warrants high precision, but at the expense of a high computational demand. In most practical situations, the Genz and Bretz algorithm provides sufficient precision with much lower computational demand. We found that calculation of the selection gain for a three-stage selection problem with 5000 budget units runs on a standard office PC only a few seconds and the optimization procedure does not take longer than 2 h for both algorithms (depending on the step length and other parameters). However, the time

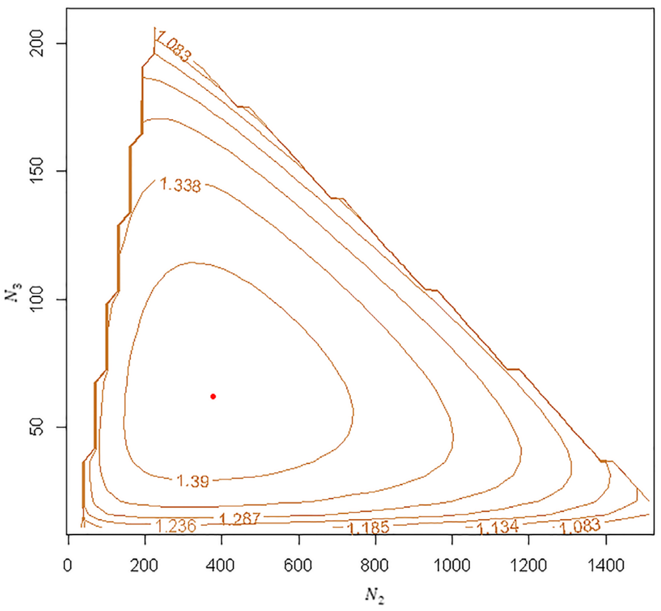

Figure 1. Contour plot of the total selection gain as a function of the number of doubled haploid lines N_2 and N_3 evaluated in field tests in Stages 2 and 3 after marker-assisted selection in Stage 1, following the example of three-stage selection given by Heffner et al. (2010). The bullet reflects the maximum value of ΔG (1.434) obtained under the following restrictions: *Budget* = 10,021, N_f = 10, L_2 = 3, L_3 = 8, T_2 = 2, T_3 = 5, R_2 = 1, R_3 = 1; N_1 is determined by the equation: ($Budget - B_2 - B_3$)/($CostProd_1 + \delta_1\, CostTest_1$). The contour lines are obtained by linear smoothing of the serrated lines of integer optimization.

requirements of the Miwa algorithm increase quadratically as the dimension increases, thus, it takes about 12 times longer to calculate the selection gain for a ten-stage than for a three-stage selection problem.

The optimization function *multistageoptimum.grid*, used internally by the *multistageoptimum.search* function can handle up to five selection stages. To adapt the function to other breeding schemes, the function *multistageoptimum.search* is open source and can be easily modified according to the needs of the user.

Most of the computing time is consumed by determining the optimum allocation of resources that maximize ΔG. Since ΔG must be calculated for each point in the grid, we recommend using the Genz and Bretz algorithm in this context to keep the computational burden at an acceptable level. After a reasonable region containing the maximum has been identified, a detailed search with the Miwa algorithm can be executed. If necessary, computing efficiency for the optimization could be improved by employing advanced numerical optimization algorithms (Kim, 1997) and using R packages that support Multi-Core Processors and Graphics Processing Units.

In our example of the three-stage selection, the contour lines of ΔG are serrated, because an integer optimization must be performed for several variables under a budget constraint, and have to be smoothed (Fig. 1). If the number of variables that have to be optimized is greater than two (e.g., N_1, N_2, N_3, and N_4 under four-stage selection) other visualization packages such as *rggobi* (Lang et al., 2011) could be chosen to display the high dimensional data. In general, the region around the maximum value for ΔG is usually rather flat. On the Basis of our experience with numerous examples from plant breeding, the region becomes generally flatter if more stages are considered in a multistage selection. Consequently, for given variance components and costs, the choice of the resource allocation is fairly robust across a wide range without a significant loss in ΔG. As a result, an allocation could be chosen within the flat region adjacent to the maximum of ΔG, which can be performed most conveniently in the breeding program or with reduced costs. However, the input parameters themselves can strongly affect the optimum allocation of resources and should be determined with great care. This is particularly true for the correlation matrix among the selection criteria SC_i as determined by the variance components and the costs of various tests, especially the costs of marker assays in the initial stages of selection.

Our software was developed to optimize breeding schemes for a single target trait. Extensions to situations where multiple traits are combined in a selection index are straightforward by adapting the correlation matrix and budget constraints (see Utz et al., 1994; Xu et al., 1995). Moreover, nonlinear selection indices could be embedded in our software by using the R package *Rindsel* (Crossa and Perez-Elizalde, 2013). Besides use in optimization of practical breeding programs, our software *selectiongain* should also be a valuable teaching tool in advanced plant-breeding classes. In addition to simple use of the software, the provided manual includes two examples for illustrating multistage selection in the context of plant breeding that can help students to understand the principles of multistage selection.

Supplemental Information Available

Supplemental Figure S1: Screenshot of the output from *multistageoptimum.search* function with parameter detail = TRUE.

Acknowledgments

This research was supported by Grant 0315072B within the project "GABI GAIN" supported by the German Ministry of Research and Technology (BMBF). We thank Susic Jasmina and Sen Han for helpful comments on previous drafts of this paper.

References

Bernardo, R. 2002. Breeding for quantitative traits in plants. Stemma Press, Woodbury, MN.

Bulmer, M.G. 1971. Effect of selection on genetic variability. Am. Nat. 105:201–211. doi:10.1086/282718

Cochran, W.G. 1951. Improvement by means of selection. In: J. Neyman, editor, Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, CA. 31

July–12 Aug. 1950. University of California Press, Berkeley.

Crossa, J., and S. Perez-Elizalde. 2013. Rindsel. Integrated breeding plantform. Available at www.integratedbreeding.net/supplementary-toolbox/rindsel (accessed 9 Dec. 2013).

Finney, D.J. 1956. The consequences of selection for a variate subject to errors of measurement. Revue Inst. Int. Stat. 24:1–10. doi:10.2307/1401275

Genz, A., F. Bretz, T. Miwa, X. Mi, F. Leisch, F. Scheipl, and T. Hothorn. 2013. mvtnorm: Multivariate normal and t distributions. R package version 0.9–9996. http://cran.r-project.org/package=mvtnorm. (accessed 9 Dec. 2013).

Heffner, E.L., A.J. Lorenz, J.L. Jannink, and M.E. Sorrells. 2010. Plant breeding with genomic selection: Gain per unit time and cost. Crop Sci. 50:1681–1690. doi:10.2135/cropsci2009.11.0662

Kim, J. 1997. Iterated grid search algorithm on unimodal criteria. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Lang, D.T., D. Swayne, H. Wickham, and M. Lawrence. 2011. rggobi: Interface between R and GGobi. http://cran.r-project.org/package=rggobi (accessed 9 Dec. 2013).

Longin, C.F.H., H.F. Utz, J.C. Reif, T. Wegenast, W. Schipprack, and A.E. Melchinger. 2007. Hybrid maize breeding with doubled haploids: III. Efficiency of early testing prior to doubled haploid production in two-stage selection for testcross performance. Theor. Appl. Genet. 115:519–527. doi:10.1007/s00122-007-0585-2

Mi, X., T. Miwa, and T. Hothorn. 2009. Implement of miwa's analytical algorithm of multi-normal distribution. R. J. 1:37–39.

Mi, X., H.F. Utz, and A.E. Melchinger. 2012. selectiongain: Expected gain from multi-stage selection and its optimization. R package version 2.0.29. http://cran.r-project.org/package=selectiongain. (accessed 9 Dec. 2013).

R Development Core Team. 2012. The R project for statistical computing. www.r-project.org (accessed 9 Dec. 2013).

Sun, X., T. Peng, and R.H. Mumm. 2011. The role and basics of computer simulation in support of critical decisions in plant breeding. Mol. Breed. 28:421–436. doi:10.1007/s11032-011-9630-6

Tallis, G.M. 1961. The moment generating function of the truncated multi-normal distribution. J. R. Stat. Soc., B 23:223–229.

Utz, H.F., A.E. Melchinger, G. Seitz, M. Mistele, and J. Zeddies. 1994. Economic aspects of breeding for yield and quality traits in forage maize. Plant Breed. 112:110–119. doi:10.1111/j.1439-0523.1994.tb00658.x

Wang, J., M. van Ginkel, R. Trethowan, G. Ye, I. DeLacy, D. Podlich, and M. Cooper. 2004. Simulating the effects of dominance and epistasis on selection response in the CIMMYT wheat breeding program using QuCim. Crop Sci. 44:2006–2018. doi:10.2135/cropsci2004.2006

Xu, S., T.G. Martin, and W.M. Muir. 1995. Multistage selection for maximum economic return with an application to beef cattle breeding. J. Anim. Sci. 73:699–710.