Comput Stat (2016) 31:533–543
DOI 10.1007/s00180-015-0583-9

ORIGINAL PAPER

Selectiongain: an R package for optimizing multi-stage selection

Xuefei Mi[1] · H. Friedrich Utz[1] ·
Albrecht E. Melchinger[1]

Received: 18 February 2014 / Accepted: 8 April 2015 / Published online: 3 May 2015

Abstract Multi-stage selection is practised in numerous fields of the life sciences and particularly in breeding. A special characteristic of multi-stage selection is that candidates are evaluated in successive stages with increasing intensity and efforts, and only a fraction of the superior candidates is selected and promoted to the next stage. For the optimum design of such selection programs, the selection gain $\Delta G(y)$ plays a central role. It can be calculated by integration of a truncated multivariate normal distribution. While mathematical formulas for calculating $\Delta G(y)$ and $\psi(y)$, the variance among the selected candidates, were developed a long time ago, solutions and software for numerical calculations were not available. We developed the R package selectiongain for efficient and precise calculation of $\Delta G(y)$ and $\psi(y)$ for (i) a given matrix $\boldsymbol{\Sigma}^*$ of correlations among the unobservable target character and the selection criteria and (ii) given coordinates $\mathbf{Q}$ of the truncation point or the selected fractions $\boldsymbol{\alpha}$ in each stage. In addition, our software can be used for optimizing multi-stage selection programs under a given total budget and different costs of evaluating the candidates in each stage. Besides a detailed description of the functions of the software, the package is illustrated with two examples.

Keywords Selection gain · Multivariate normal integral · Optimal allocations

✉ Albrecht E. Melchinger
melchinger@uni-hohenheim.de

Xuefei Mi
mi_xue_fei@hotmail.com

[1] Institute of Plant Breeding, Seed Science and Population Genetics, University of Hohenheim, Fruwirthstr. 21, 70593 Stuttgart, Germany

1 Introduction

Selection is often a multi-stage procedure in many fields of the life sciences. In the first stage, a large number of candidates (e.g., animals, plant varieties or potential drugs) is tested with low costs per candidate, resulting in a low precision of the estimates of an unobservable target character to be selected for. A fraction of superior candidates is selected based on their estimated performance, and tested in a second stage with higher efforts and costs per candidate, which increases the precision of the estimates. This process is usually continued over multiple stages.

Based on the observations of each candidate up to a certain stage, either linear combinations or best linear unbiased predictors (BLUPs) for the target character of the candidate are calculated. They serve as selection criteria and are assumed to follow a multivariate normal (MVN) distribution. With fixed proportions of the selected fraction set by the experimenter in each stage, the truncation point of the corresponding MVN distribution can be calculated, provided that the covariance or correlation matrix of the selection criteria in all stages is known. Moreover, the progress due to selection, denoted as $\Delta G(y)$, and commonly referred to as selection gain, is defined as the difference between the expectation of y, the true unobservable target character of the candidates, after and before the selection. It can be obtained as a multi-dimensional integral over a restricted area defined by the truncation point (Lynch and Walsh 1997). An algorithm for calculating the selection gain without multidimensional numerical integral was developed by Xu et al. (1995), however this method requires decomposition and inverse of the correlation matrix, which cannot be solved in linear time if the dimension is high and the structure of the matrix is not sparse.

Besides calculation of $\Delta G(y)$, the experimenter has also an interest in the variance among the selected candidates, $\psi(y)$, after multiple stages of selection. Among others, this influences the decision whether there is still a reasonable chance to make further progress by additional selection stages (Cochran 1951). While multi-stage selection plays a central role in animal and plant breeding, the principles of multi-stage selection also apply to many problems in the social, industrial, pharmaceutical, and medical sciences (West-Eberhard 1983; Villet et al. 2006; Shi and Zhou 2009). Efficient algorithms for calculation of $\Delta G(y)$ are also crucial for strategies to determine the optimum allocation of resources in the different selection stages for achieving a maximum selection gain under a restricted budget. In practical examples, such as disease identification and vaccine selection or stock-picking strategies, computing time is a very important factor (Yan and Clack 2011). Hence, there is an urgent need to develop fast and efficient algorithms for calculation of $\Delta G(y)$ and $\psi(y)$.

Cochran (1951) first described the theory of multi-stage selection and derived an analytical solution of $\Delta G(y)$ and $\psi(y)$ for two-stage selection. For one-sided multivariate truncation selection, Tallis (1961) gave a general solution by using the moment generating function (MGF) of the MVN distribution.

In order to calculate the integral of MVN distribution, Genz et al. (2011) developed the R package mvtnorm. It employs a quasi-Monte Carlo algorithm, for which the computation time increases linearly with the dimension n and the computing time of $\Delta G(y)$ is proportional to n^2 ($n \in \mathbb{N}$ and $n < 1000$) (Genz and Bretz 1999). With a recursive linear integration procedure, mvtnorm calculates the MVN integral with even higher accuracy without compromising the computation time when $n < 20$ (Miwa et al. 2003; Mi et al. 2009).

In this paper, we present the R package selectiongain, designed for evaluating $\Delta G(y)$ and $\psi(y)$ under one-sided truncation selection in multiple stages. In addition, we provide functions for optimizing multi-stage selection under given restrictions on the budget for many scenarios, with special emphasis on applications in breeding. One numerical example for checking the computation time and accuracy, and one practical example from breeding are provided.

2 Calculation of selection gain, variance among selected candidates, and truncation point

In this section, $\Delta G(y)$ is introduced as the first moment of a MVN distribution over a restricted area, defined by the coordinates of the truncation point. Furthermore, $\psi(y)$, will be introduced as the second central moment of a MVN distribution over the same restricted area. The coordinates of the truncation point, corresponding to the selection criteria, will be computed for a given vector of selected fractions.

The underlying statistical model for our calculations is as follows. Suppose we start with N_i candidates from a population in stage i and the jth candidate has $m_{i,j}$ observations. The final goal is to have N_{n+1} candidates selected after n stages of selection.

Let $\mathbf{O}_j = \{O_{i,j}\}$ be the mean of the $m_{i,j}$ observations on the jth candidate in each stage $i = 1, \ldots, n$, where $O_{i,j} = \frac{1}{m_{i,j}} \sum_{l=1}^{m_{i,j}} o_{i,j,l}$ and $o_{i,j,l}$ is the lth observation of the jth candidate in the ith stage with $o_{i,j,l} = y_j + E_{i,j,l}$. Here, $E_{i,j,l}$ is a noise variable that depends on the selection stage but is stochastically independent among the candidates. The scalar $O_{i,j}$ is a value of a single character, e.g., a test score in an exam for college admission or grain yield in the context of breeding or a function of several traits weighted with economic parameters (Falconer and Mackay 1996). In this paper, we focus on the problem of selection for a univariate target character.

The selection criterion $x_{i,j}$ for candidate j in stage i is a linear regression on $\mathbf{O}_j$, i.e., $x_{i,j} = \sum_{k=1}^{i} \beta_k O_{k,j}$, whose regression coefficients are calculated via regression or BLUP, $x_{i,j} = BLUP(O_{1,j}, O_{2,j}, \ldots, O_{i,j})$, obtained by solving the mixed model equations. Here, we denote $\mathbf{x}_j^* = \{y_j, x_{i,j}\}$ and $\mathbf{x}_j = \{x_{i,j}\}$, where $\mathbf{x}_j$ is the vector of selection criteria for candidate j in stage $i = 1, \ldots, n$. Note $\mathbf{X}^* = \{Y, X_1, X_2, \ldots, X_n\}$ and $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$, where X_i and Y represent the corresponding random variables of $x_{i,j}$ and y_j, respectively. For a convenient ordering of indices, we will denote Y as X_0, and y as x_0 in summations and the MGF. Furthermore, we assume that $X_0, X_1, \ldots, X_n$ are MVN distributed.

The theory of multi-stage selection is based on the mean of y in the selected area (Cochran 1951). For simplicity, the mean and the variance of y are set to 0 and 1, i.e., $E(y) = 0$ and $\sigma_y^2 = 1$. If $\sigma_y^2 \neq 1$, $\Delta G(y)$ and $\psi(y)$ have to be multiplied with σ_y or σ_y^2, respectively. In order to calculate $\Delta G(y)$, we have to determine the one-sided integral of y over the right-sided area $\mathbf{S}_Q = \{x_1 > q_1, \ldots, x_n > q_n\}$ defined by the truncation point $\mathbf{Q} = \{q_1, q_2, \ldots, q_n\}$. If $x_{i,j} \geq q_i$, where q_i is the threshold for stage

i, then the jth candidate in stage i is promoted to the next stage. The value of $\Delta G(y)$ is denoted as $E(Y; \mathbf{S}_Q)$. Thus, $\Delta G(y)$ is defined as

$$\Delta G_n(y, \mathbf{S}_Q, \mathbf{U}^*, \boldsymbol{\Sigma}^*) = E(Y; \mathbf{S}_Q) = \alpha^{-1} \int_{-\infty}^{\infty} \int_{q_1}^{\infty} \ldots \int_{q_n}^{\infty} y\, \phi_{n+1}(\mathbf{x}^*; \mathbf{U}^*, \boldsymbol{\Sigma}^*)\, d\mathbf{x}^*, \tag{1}$$

where

$$\alpha = \Phi_n(\mathbf{Q}, \boldsymbol{\Sigma}) = \int_{q_1}^{\infty} \ldots \int_{q_n}^{\infty} \phi_n(\mathbf{x}; \mathbf{U}, \boldsymbol{\Sigma})\, d\mathbf{x}, \tag{2}$$

and

$$\phi_n(\mathbf{x}; \mathbf{U}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{U})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \mathbf{U})\right), \tag{3}$$

where, ϕ_n is the density function of MVN, and $\boldsymbol{\Sigma}$ is the correlation matrix of $\mathbf{X}$. $\boldsymbol{\Sigma}^*$ is the correlation matrix of $\mathbf{X}^*$. It comprises $\boldsymbol{\Sigma}$, but has one dimension more pertaining to the correlations between $Y = X_0$ and the selection criteria $\mathbf{X}$. The mean vector $\mathbf{U}^* = \{u_0, u_1, \ldots, u_n\}$ of ϕ_{n+1} is omitted, assuming $\mathbf{U}^* = \{0, 0, \ldots, 0\}$ without loss of generality, and consequently, we write briefly $\phi_{n+1}(\mathbf{x}^*; \boldsymbol{\Sigma}^*)$, $\phi_n(\mathbf{x}; \boldsymbol{\Sigma})$ and $\Delta G_n(y, \mathbf{S}_Q, \boldsymbol{\Sigma}^*)$. The selection gain is the first moment, while the selected fraction α over all n stages of selection corresponds to the zero-th moment of the one-sided truncated MVN distribution of $\mathbf{X}$.

The function $\psi(y)$ is defined as the second central moment, $\psi_n(y) = E(Y^2; \mathbf{S}_Q) - \left[E(Y; \mathbf{S}_Q)\right]^2$, where

$$E(Y^2; \mathbf{S}_Q) = \alpha^{-1} \int_{-\infty}^{\infty} \int_{q_1}^{\infty} \ldots \int_{q_n}^{\infty} y^2\, \phi_{n+1}(\mathbf{x}^*; \boldsymbol{\Sigma}^*)\, d\mathbf{x}^*. \tag{4}$$

If n is large, it is not efficient to calculate the Riemann integral of $\Delta G(y)$ by using the simple summation principle of Riemann sums, which subdivides the integrated area into several small hypercubes and sums up their volumes. The computing time of the Riemann sums is proportional to g^{n+1}, where g is the number of grid points for integration (Press et al. 1993).

For a given truncation point $\mathbf{Q}$, the MGF of the truncated MVN variable $\mathbf{X}^* \in \mathbf{S}_Q^* = \{y > -\infty, x_1 > q_1, \ldots, x_n > q_n\}$ is calculated by the procedure given by Tallis (1961) as:

$$m(T) = \alpha^{-1} \int_{-\infty}^{\infty} \int_{0}^{\infty} \ldots \int_{0}^{\infty} e^{\mathbf{T}\mathbf{x}^*} \phi_{n+1}(\mathbf{x}^*; \mathbf{U}', \boldsymbol{\Sigma}^*)\, dx_1 \ldots dx_n dy, \tag{5}$$

here, $\mathbf{T} = \{t_0, t_1, \ldots, t_n\}$ and the lower limit of $\mathbf{x}$ is standardized from $\mathbf{S}_Q = \{q_1, \ldots, q_n\}$ to $\mathbf{0} = \{0, 0, \ldots, 0\}$, so the mean of $\mathbf{x}^*$ is shifted from $\mathbf{U}^* = \{0, 0, \ldots, 0\}$ to $\mathbf{U}' = \{0, -q_1, \ldots, -q_n\}$.

By differentiating and evaluating the MGF at $t_0 = 0$, the (n+1)-dimensional integral of $E(Y; \mathbf{S}_Q)$ is turned into a sum of $n+1$ (n)-dimensional MVN distribution functions with new quantiles and partial correlation matrices, which are calculated according to the routines given by Tallis (1961). Due to the definition of y, its lower limit is minus infinity. This fulfills the special case mentioned in Tallis (1961), which reduces the (n)-dimensional MVN integral into (n−1)-dimensional integral. Similarly, $E(Y^2; \mathbf{S}_Q)$ is calculated by taking the second order of derivatives.

In practice, a predefined fraction α_i of the superior candidates is selected in stage i. Hence, the coordinates of $\mathbf{Q}$ must be calculated from the vector of selected fractions $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, ..., \alpha_n\}$. The coordinates q_i are obtained by inverting the following equations sequentially:

$$\alpha_1 = f(q_1) = \int_{q_1}^{\infty} \phi_1(x_1)\, dx_1, \tag{6}$$

which yields q_1;

$$\alpha_1\alpha_2 = f(q_1, q_2) = \int_{q_2}^{\infty} \int_{q_1}^{\infty} \phi_2(x_1, x_2; \rho_{1,2})\, dx_1\, dx_2, \tag{7}$$

which yields q_2 for given values of q_1, α_1 and α_2, and so on for $i = 3, 4, \ldots, n$. The value of q_n is finally obtained by using $q_1, q_2, \ldots, q_{n-1}$ and $\alpha_1, \alpha_2, \ldots, \alpha_n$:

$$\prod_1^n \alpha_i = \int_{q_n}^{\infty} \int_{q_{n-1}}^{\infty} ... \int_{q_1}^{\infty} \phi_n(\boldsymbol{X}; \boldsymbol{\Sigma})\, d\mathbf{x}. \tag{8}$$

These equations can be solved numerically via a fast root search algorithm as described for example by Brent (1973) and implemented in the R function uniroot.

3 Optimizing selection scenarios under restricted budget with a fixed or dependent correlation matrix

In practice, a selection program has a limited budget B to cover all costs such as (i) producing or providing the N_i candidates and (ii) evaluating the N_i candidates in stage i. For a given testing scheme with $\mathbf{N} = (N_1, \ldots, N_n)$ candidates in the ith stage of selection ($i = 1, \ldots, n$), the costs are determined by the cost function $C(\omega)$,

$$C(\omega) = \sum_{i=1}^{n} (N_i * CostProd_i + N_i * CostTest_i) \leq B, \tag{9}$$

where $CostProd_i$ refers to the costs of producing or providing a candidate, and $CostTest_i$ refers to the costs of testing a candidate. Both of them are measured in terms of test units. Here ω is a vector of N_i. Let $\Omega(B)$ be a subset of all possible ω that $C(\omega) \leq B$. Thus, the set of admissible allocations $\Omega(B)$ of the candidates to the various stages of selection is given by

$$\Omega(B) := \{\omega = \mathbf{N} | C(\omega) \leq B\}. \tag{10}$$

Hence, our goal is to find $\tilde{\omega} \in \Omega(B)$ with

$$\Delta G(y, \mathbf{S}_{\tilde{\omega}}, \Sigma^*) = MAX_{\omega \in \Omega(B)} \Delta G(y, \mathbf{S}_{\omega}, \Sigma^*), \tag{11}$$

where $\mathbf{S}_{\omega}$ refers to the truncation point ω corresponding to $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_n\}$, with $\alpha_i = N_{i+1}/N_i$ for $i = 1, \ldots, n$. The matrix $\boldsymbol{\Sigma}^*$ is determined by the correlations among the selection criteria in the n stages of selection as well as their correlations to y. Hence, for given but possibly different testing procedures in each stage, $\boldsymbol{\Sigma}^*$ is fixed and independent of the choice of $\mathbf{N}$.

The simplest way to find the maximum is to do a full scan of the entire set $\Omega(B)$, which calculates $\Delta G(y, \mathbf{S}_{\omega}, \Sigma^*)$ for all possible allocations of $\omega(B)$ in order to determine $\tilde{\omega}$ yielding the maximum of $\Delta G(y)$. However, this is very time consuming. An alternative solution is to use grid search, which divides the whole set $\Omega(B)$ into several grids (Kim 1997). Another way for finding the maximum is using optimization algorithms for non-linear minimization (NLM) provided by R function nlm and constrOptim, which are functions of R core package stats (R Core Team 2013). The function nlm uses a Newton-type algorithm for searching the maximum of a multimodal function (Ron and Bruce 2009). This Newton-type algorithm depends heavily on the starting point, the maximum number of iterations as well as the numerical derivatives of $\Delta G(y)$ and results in an accuracy less than four digits. Xu et al. (1995) reported that the NLM algorithm converges to a local maximum, if the initial value is inappropriate. So the grid algorithm is recommended to run before NLM for getting an appropriate starting point. Here, we mainly employed the function constrOptim, which uses an Adaptive Barrier algorithm as core optimization function for our non-linear optimization problem.

The computational time of the NLM algorithm is proportional to $N_1 * log(N_1) * n^2$, while the computational time of the algorithm for full space scan or grid search is proportional to $N_1^2 * n^2$ or $N_{grid}^2 * n^2$, respectively, where N_1 is the number of the initial candidates, N_{grid} is the number of grids and n is the number of selection stages. Hence, the calculation speed is much faster for the NLM and grid search algorithm than the full space scan, especially when the initial sample size N_1 is large.

A special and more complicated scenario relates to tests of candidates with dependent correlation matrix. For example, in plant breeding the candidates are usually tested and selected in replicated multi-location trials over several years, corresponding to the stages of selection. Thus, besides $\mathbf{N}$, referring to the numbers of candidates to be tested in each stage, the breeder must also decide on the intensity of testing, as reflected by vector for the number of test locations $\mathbf{L} = \{L_1, \ldots, L_n\}$ and replications $\mathbf{R} = \{R_1, \ldots, R_n\}$, where L_i and R_i refer to the number of test locations and replications per location, in stage i, respectively.

If there is no upper limit on L_i, then $R_i = 1$ is optimal for maximizing $\Delta G(y)$ (Longin et al. 2007). Normally, a large number of candidates (corresponding to genotypes in plant breeding) will be tested in few locations at the first stage, i.e., $L_1 = 1$ or 2. Under this scenario, the elements in $\boldsymbol{\Sigma}^*$ are a rational function of $\mathbf{L}$, $\mathbf{R}$ and the vector of variance components $\mathbf{V}_c = \{Vg, Vgl, Vgy, Vgly, Ve\}$, where the latter refer to the components of variance among genotypes (Vg), genotype × location interactions (Vgl), genotype × year interactions (Vgy), genotype × location × year interactions

($Vgly$) and plot error (Ve). Here, Vg corresponds to σ_y^2 and is set equal to 1. Likewise, the costs are not only a function of **N**, but also of **L** and **R**, because each test plot in field trials is associated with costs. Hence, the set of admissible allocations of resources $\Omega(B)$ can be described as

$$\Omega(B) := \{\omega = (\mathbf{N},\mathbf{L},\mathbf{R})|C(\omega) \leq B\}. \tag{12}$$

In the simplest case,

$$C(\omega) = \sum_{i=1}^{n} N_i * CostProd_i + N_i * L_i * R_i * CostTest_i \leq B. \tag{13}$$

4 Example one: accuracy and time for calculating selection gain

In this section, we compare the error and time required for calculating the selection gain with two different algorithms: the Miwa algorithm and the Genz and Bretz algorithm. Our computer was a PC with an Intel® i5-540M processor (2.53 GHz). The operating system is Linux (Ubuntu LTS 12.04). The package mvtnorm (version 0.90-9995) and selectiongain (version 0.2-27) were installed in a 64-bit version of R (3.0.2).

The Miwa algorithm is a numerical algorithm that has at least 7 decimal places with 128 grid points ($g = 128$) (Mi et al. 2009). We checked our results calculated by the Miwa algorithm with the results from Cochran (1951) and found that they are identical. We also compared the accuracy of the Miwa algorithm ($g = 64$ and $g = 128$) and the Genz and Bretz algorithm (with absolute error tolerance $\varepsilon = 10^{-3}, 10^{-4}, 10^{-5}$ and $n = 10$) for probabilities with centered orthant probabilities (COP) (worst case scenario for Miwa algorithm) with correlation coefficients of $\boldsymbol{\Sigma}^*$ defined as

$$\rho_{i,j} = \begin{cases} 1, \ i = j \\ \rho, \ i \neq j \end{cases} \ 1 \leq i \leq n.$$

In calculating $\Delta G(y)$ with $\rho_{i,j} = 0.5$, the error was smaller, the smaller ε was (Fig. 1). The Genz and Bretz algorithm, however, could not reduce the error below 10^{-3}, because the ε used in this Monte-Carlo algorithm determines the accuracy for calculating the probability of the MVN distribution. During the procedure for calculating of $\Delta G(y)$, the error will accumulate. In contrast, the Miwa algorithm with $g = 64$ or $g = 128$ grid points still keeps the accuracy at six or seven digits. Figure 2 illustrates that the computation with the Miwa algorithm is very slow compared with the Genz and Bretz algorithm.

5 Example two: optimization of two-stage selection in plant breeding

In this subsection, we determine the optimum allocation of resources for a two-stage selection problem in plant breeding taken from Longin et al. (2007). The goal is to

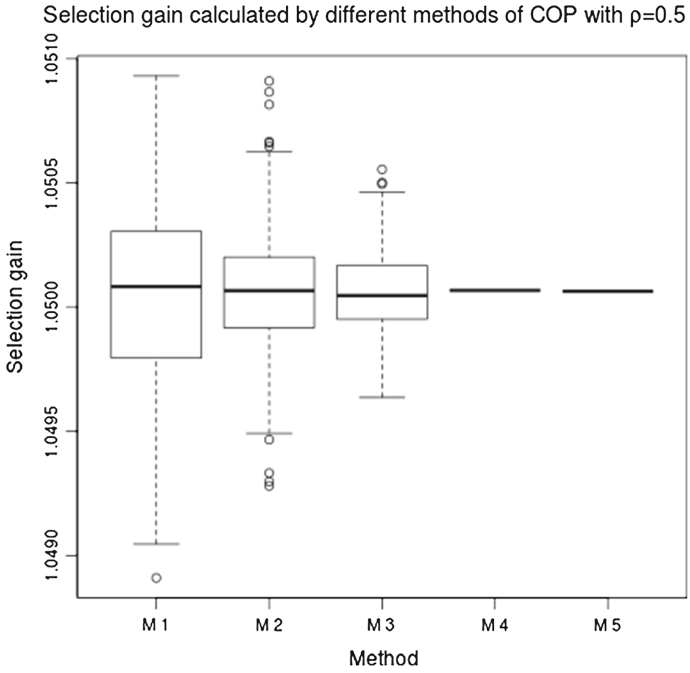

Fig. 1 Boxplot of the selection gain calculated with centered orthant probabilities, $n = 10$, $\rho_{i,j} = 0.5$, $i \neq j$, $1 \leq i \leq n$. M1, M2, M3: Genz and Bretz algorithm ($\varepsilon = 10^{-3}, 10^{-4}, 10^{-5}$). M4, M5: Miwa algorithm ($g = 64$ and $g = 128$ grid points)

find the best allocation of resources, which maximizes $\Delta G(y)$ for a single target trait, e.g., grain yield.

In this experiment, Longin et al. (2007) assumed that:

1. The selection candidates are lines in plant breeding with variance components $Vg : Vgl : Vgy : Vgly : Ve = 1 : 0.5 : 0.5 : 1 : 2$, specified by Longin et al. (2007) on the basis of breeding experiments reported in the literature.
2. $CostProd = \{0.5, 0\}$, $CostTest = \{1, 1\}$, $R_i = 1$, $N_f = N_3 = 1$ or 4, $N_{i+1} \leq N_i$ and $L_{i+1} \geq L_i$. Here, N_f is the final selected number of candidates, which is equal to the number of candidates in the last stage.
3. Three different budgets B were chosen corresponding to 200, 1000, or 5000 test plot units.

The correlation matrix $\boldsymbol{\Sigma}^*$, the vector of selected fractions $\boldsymbol{\alpha}$ and the optimization depend on these constraints. For $n = 2$, the routines for determining the correlation matrix from the given variance components and a fixed number of locations is illustrated with the numerical example of Longin et al. (2007) and integrated in function multistagecor of our package.

Table 1 shows the calculated $\Delta G(y)$ and $\psi(y)$ for the allocations examined by Longin et al. (2007) with the Miwa algorithm. We also compared the result with the reported maximum and found agreement up to three digits. The $\psi(y)$ is usually calculated after the optimization of $\Delta G(y)$ for controlling the variations. In breeding,

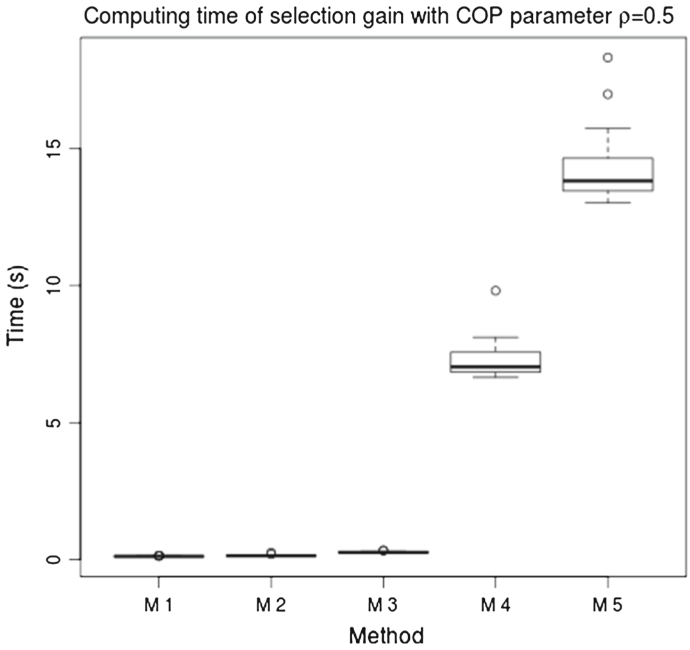

Fig. 2 Boxplot of the time used (in seconds) for computing the selection gain calculated with centered orthant probabilities, $n = 10$, $\rho_{i,j} = 1/2, i \neq j, 1 \leq i \leq n$. M1, M2, M3: Genz and Bretz algorithm ($\varepsilon = 10^{-3}, 10^{-4}, 10^{-5}$). M4, M5: Miwa algorithm ($g = 64$ and $g = 128$ grid points)

Table 1 Selection gain calculated for allocations identical to those given by Longin et al. (2007)

N_f	B	N_1	N_2	L_1	L_2	$\Delta G(y)$	$\Delta G(y)(Longin)$	$\psi(y)$
1	200	53	6	2	10	1.8479682	1.848	0.4707116
1	1000	286	14	2	18	2.3483680	2.348	0.4160850
1	5000	1463	38	2	31	2.7800706	2.780	0.3806510
4	200	79	15	1	5	1.3750382	1.375	0.5828703
4	1000	272	26	2	11	1.9236208	1.924	0.4612778
4	5000	1422	64	2	20	2.4118284	2.412	0.4081624

N_f is the number of final selected candidates in stage three. B is the Budget of test units. N_i are the number of candidates in stage i. L_i are the number of Locations in stage i. $\Delta G(y)$ is the selection gain, $\Delta G(y)(Longin)$ is the selection gain calculated by Longin and $\psi(y)$ is the variance among the selected candidates

if two different allocations achieve the maximum simultaneously, the one with larger $\psi(y)$ will be chosen.

The maximum in the grid search and the NLM search is found in Table 2 (for direct comparison, the number of locations are identical to those identified by the grid search). The $\Delta G(y)$ calculated with the NLM search was similar as the one calculated from the grid search. However, NLM is a continuous search algorithm so that the calculated values of N_1 and N_2 are not discrete and do not fit exactly the applications in breeding.

Table 2 Maximum selection gain determined by grid search and NLM search

N_f	B	Grid search					NLM search		
		N_1	N_2	L_1	L_2	$\Delta G(y)$	N_1	N_2	$\Delta G(y)$
1	200	86	10	1	7	1.850	87.06	9.92	1.851
1	1000	299	18	2	14	2.355	305.61	16.85	2.353
1	5000	1532	45	2	26	2.787	1544.17	43.83	2.787
4	200	88	17	1	4	1.386	87.56	17.17	1.386
4	1000	395	37	1	11	1.925	399.14	36.48	1.925
4	5000	1463	61	2	22	2.421	1447.47	62.79	2.421

N_f is the number of final selected candidates in stage three. B is the Budget of test units. N_i are the number of candidates in stage i. L_i are the number of locations in stage i. $\Delta G(y)$ is the selection gain

The Genz and Bretz algorithm is the default algorithm. We recommend advanced users to apply the grid search on a large scale by using the Miwa algorithm, before performing a NLM search. First, the grid search is used to find a point, which is close to the global maximum point. Second, a NLM algorithm or a grid search with small scale can be carried out around this point. For further comparison, Monte Carlo simulations and analytical solutions for higher dimension selection are available (Longin et al. 2007; Wegenast et al. 2010; Mi et al. 2011). With our package, the user can easily build the mathematical model for these new situations and perform the necessary calculations for determining the optimum allocation of resources.

6 Conclusions

We have developed the R package selectiongain, which allows precise (at least for five digits) and fast calculation of $\Delta G(y)$ with the help of the MGF. Our software can be used to find solutions for multi-stage selection programs in a wide field of applications.

Compared to the scenarios, where $\Delta G(y)$ must be optimized with regard to $\mathbf{N}$, for fixed $\boldsymbol{\Sigma}^*$ and $\omega \in \Omega(B)$, the situation is more complex for multi-stage selection in plant breeding. Here, the breeder does not only vary $\mathbf{N}$ but also the number of locations $\mathbf{L}$ and replications $\mathbf{R}$, which influences the value of $\boldsymbol{\Sigma}^*$.

As new technologies and methods are invented and developed, the importance of multi-stage selection will increase in the future in many areas of the medical, natural and social sciences. Hence, algorithms and software for dealing with high dimensional data in selection problems are urgently required. The goal of our software is to assist practitioners in optimizing the experimental design and portfolio analysis for multi-stage selection programs.

Acknowledgments This research was supported by Grant 0315072B “GABI-GAIN” within the framework “Genome analysis of the plant biological system” supported by the German Federal Ministry of Education, Research, and Technology. The authors are grateful to the editors and anonymous reviewers for their insightful comments which have helped to improve the quality of the paper.

References

Brent R (1973) Algorithms for minimization without derivatives. Prentice-Hall, Englewood Cliffs, New Jersey

Cochran WG (1951) Improvement by means of selection. In: Proceedings of Second Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, pp 449–470

Falconer DS, Mackay TFC (1996) Introduction to quantitative genetics, 4th edn. Longman Publishing Group, London

Genz A, Bretz F (1999) Numerical computation of multivariate t-probabilities with application to power calculation of multiple contrasts. J Stat Comput Simul 63:361–378

Genz A, Bretz F, Miwa T, Mi X, Leisch F, Scheipl F, Hothorn T (2011) mvtnorm: multivariate normal and t distributions. R package version 0.9-9995

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

Longin CFH, Utz HF, Reif JC, Wegenast T, Schipprack W, Melchinger AE (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(4):519–527

Lynch M, Walsh B (1997) Genetics and analysis of quantitative traits. Sinauer Associates Inc, Sunderland

Mi X, Miwa T, Hothorn T (2009) mvtnorm: New numerical algorithm for multivariate normal probabilities. R J 1(1):37–39

Mi X, Wegenast T, Utz HF, Dhillon BS, Melchinger AE (2011) Best linear unbiased prediction and optimum allocation of test resources in maize breeding with doubled haploids. Theor Appl Genet 123(1):1–10

Miwa T, Hayter AJ, Kuriki S (2003) The evaluation of general non-centred orthant probabilities. J R Stat Soc B 65:223–234

Press WH, Teukolsky SA, Vetterling WT, Flannery BP (1993) Numerical recipes in FORTRAN; the art of scientific computing, 2nd edn. Cambridge University Press, New York

R Core Team (2013) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, http://www.R-project.org

Ron L, Bruce H (2009) Calculus, 9th edn. Brooks/Cole Publishing, Los Angeles

Shi J, Zhou S (2009) Quality control and improvement for multistage systems : a survey. IIE Trans 41:744–753

Tallis GM (1961) The moment generating function of the truncated multi-normal distribution. J R Stat Soc B 23(1):223–229

Villet S, Pichoud C, Villeneuve JP, Trepo C, Zoulim F (2006) Selection of a multiple drug-resistant hepatitis b virus strain in a liver-transplanted patient. Gastroenterology 131(4):1253–1261

Wegenast T, Utz HF, Longin CFH, Maurer HP, Dhillon BS, Melchinger AE (2010) Hybrid maize breeding with doubled haploids: V. selection strategies for testcross performance with variable sizes of crosses and s_1 families. Theor Appl Genet 121(7):1391–1393

West-Eberhard MJ (1983) Sexual selection, social competition, and speciation. Q Rev Biol 58(2):155–183

Xu S, Martin TG, Muir WM (1995) Multistage selection for maximum economic return with an application to beef cattle breeding. J Anim Sci 73(3):699–710

Yan W, Clack CD (2011) Evolving robust gp solutions for hedge fund stock selection in emerging markets. Soft Comput 15:37–50