Genomic prediction of heterosis, inbreeding control, and mate allocation in outbred diploid and tetraploid populations

Jeffrey B. Endelman

Department of Plant & Agroecosystem Sciences, University of Wisconsin–Madison, Madison, WI, USA.

Correspondence: endelman@wisc.edu

ORCID 0000-0003-0957-4337

Abstract

Breeders have long appreciated the need to balance selection for short-term genetic gain with maintaining genetic variance for long-term gain. For outbred populations, the method called Optimum Contribution Selection (OCS) chooses parental contributions to maximize the average breeding value at a prescribed inbreeding rate. With Optimum Mate Allocation (OMA), the contribution of each mating is optimized, which allows for specific combining ability due to dominance. To enable OCS and OMA in polyploid species, new theoretical results were derived to (1) predict mid-parent heterosis due to dominance and (2) control inbreeding in a population of arbitrary ploidy. A new Convex optimization framework for OMA, named COMA, was developed and released as public software. The effectiveness of COMA was demonstrated through stochastic simulation of a genomic selection program for 30 generations. Both pedigree and genomic kinship based on identity-by-descent performed similarly for OCS, but the realized inbreeding rate was too conservative when using genomic kinship for OMA. OMA predicted mate performance with higher accuracy (+0.2–0.3) than OCS, but this only translated into higher gain for the first 5 generations, reinforcing the idea that selection on additive value drives long-term gain. Less genic variance was used by OMA to achieve the same gain as OCS, making it more efficient. The sparsity of the OMA solution makes it attractive when random mating is logistically challenging. In a potato breeding case study with 170 candidates, the OMA solution at 1% inbreeding required only 25 matings involving 29 parents.

INTRODUCTION

Plant and animal breeders have long appreciated the need to balance selection for short-term genetic gain with maintaining genetic variance for long-term gain. For outbred populations, inbreeding is an effective proxy for the loss of genetic variance over time. A simple expression to capture this idea exists for random mating populations: for inbreeding coefficient F, the additive genetic variance is $1–F$ times the base population variance (Falconer and Mackay 1996). Even for crops where the final products are inbred lines or F1 hybrids produced from them (e.g., wheat, maize), this framework may still be relevant because outbred mating systems are being adopted with "two-part" genomic selection schemes to accelerate genetic gain (Gaynor et al. 2017).

Because individuals with high genetic merit are often related, naïve truncation selection leads to matings between close relatives. This phenomenon is exacerbated with selection on pedigree or genomic BLUP, and because of its long-term use in animal breeding, there is a significant body of research on optimizing selection under restricted inbreeding (Wray and Thompson 1990; Toro and Pérez-Enciso 1990; Wray and Goddard 1994; Meuwissen 1997; Grundy et al. 1998). In the standard methodology known as optimum contribution selection (OCS), the decision variables are the genetic contribution of each parental candidate to the next generation, and the objective is to maximize genetic merit at a target inbreeding rate, typically 0.5–1.0% (Sonneson et al. 2012; Woolliams et al. 2015). Inbreeding rate represents the probability of "new" inbreeding per generation (Falconer and Mackay 1996).

OCS has several well-known limitations. It assumes random mating, but practical mating plans often require constraints based on the reproductive biology of the species, such as the number of offspring per mating or the number of matings per individual. Because these constraints can make it difficult to realize the optimum contributions, there is an advantage to optimizing the contributions and mating plan simultaneously, which has been called optimum mate allocation (OMA) (Kinghorn 2011; Akdemir and Sánchez 2016; Gorjanc and Hickey 2018). Shifting from OCS to OMA also allows the specific combining ability of each mating to be considered when computing genetic merit (Toro and Varona, 2010).

A practical motivation for the present study was to implement OMA in a tetraploid potato (*S. tuberosum*) breeding program. Several new theoretical results were needed to realize this goal. Diploid formulas for inbreeding rate are not correct for autopolyploids because the

inbreeding coefficient of an individual is not equal to parental kinship; it also depends on the parental inbreeding coefficients (Kerr et al. 2012). Another obstacle was the lack of a formula for genomic prediction of the F1 progeny mean from two polyploids with dominance; previous studies have used look-up tables, limiting theoretical insight (Labroo et al. 2023).

From a computational perspective, an attractive feature of OCS is that it is a *convex* optimization problem, which implies the global optimum can be found efficiently (Boyd and Vandenberghe 2004). This feature has been exploited in previous OCS studies (Pong-Wong and Woolliams 2007; Wellman 2019), but OMA implementations have not been explicitly convex, relying instead on general purpose evolutionary optimization algorithms. I report here on the formulation of Convex OMA and associated software, named COMA. This new methodology was evaluated using stochastic simulation of a simple breeding program, as well as real potato breeding data.

THEORY

F1 progeny mean

Our single-locus, bi-allelic model is based on the decomposition of genotypic value at panmictic equilibrium (i.e., HWE) into the population mean plus additive and dominance values. (Only digenic dominance is considered; higher order dominance for polyploids is neglected.) The additive (u) and dominance (v) values are the products of regressors and regression coefficients:

$$u = W\alpha = \phi[x - p]\alpha$$
$$v = Q\beta = [\tau - \phi^2(x - \xi)^2]\beta \qquad [1]$$

where $\tau = (\phi - 1)pq + \frac{1}{4}$ and $\xi = \phi^{-1}\left[p(\phi - 1) + \frac{1}{2}\right]$

The additive regressor W is a linear function of the difference between the allele frequency within an individual, denoted x, and the population allele frequency p ($q = 1–p$). The proportionality constant, or slope, is the ploidy ϕ (Fig. 1A). From standard quantitative genetics, the additive regression coefficient α is the allele substitution effect at HWE (Falconer and Mackay 1996).

The dominance regressor Q is a quadratic function of the difference between x and ξ, a parameter that depends on allele frequency and ploidy. From the inverted parabolic shape of Q,

homozygotes have lower dominance value than heterozygotes, as expected (Fig. 1B). The curvature of the parabola equals ploidy squared. The dominance regression coefficient β is the digenic substitution effect at HWE; in other words, it is the difference between the effect of the heterozygous diplotype and the mean effect of the homozygous diplotypes (Endelman 2023). For diploids, Eq. 1 reduces to the classical "breeding value" parameterization with $\beta = d$ (Vitezica et al. 2013). The symbol β emphasizes it generalizes α to digenic effects in polyploids.

Consider F1 progeny of the mating between individuals 1 and 2 with allele frequencies x_1 and x_2, respectively. Because additive values are linear functions of allele dosage, the mean progeny additive value equals the mid-parent additive value: $\text{MPA} = \frac{1}{2}(u_1 + u_2)$. The contribution of dominance effects is the sum of the mid-parent dominance value (MPD) and mid-parent heterosis (MPH), where

$$\text{MPD} = \tfrac{1}{2}(v_1 + v_2) = \beta\left[\tfrac{-\phi}{2}(x_1^2 + x_2^2) + \xi\phi^2(x_1 + x_2) - \phi(\phi - 1)p^2\right] \qquad [2]$$

The expression for MPD follows from Eq. 1 and is a quadratic function of two variables, which from geometry represents a conic section (e.g., hyperbola, ellipse, or parabola) in the (x_1, x_2) plane. The general form of the quadratic equation for conic sections is typically written as

$$Ax_1^2 + Bx_1x_2 + Cx_2^2 + Dx_1 + Ex_2 + F \qquad [3]$$

Comparing Eq. 2 and 3 allows for identification of the *A* through *F* coefficients (Table 1). Because *A*=*C* and *D*=*E* in this case, the centered, matrix form of the equation is

$$\mathbf{x}'\mathbf{M}\mathbf{x} + \left(F - \frac{D^2}{2A + B}\right)$$

$$\text{where } \mathbf{M} = \begin{bmatrix} A & B/2 \\ B/2 & A \end{bmatrix} \text{ and } \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix}\left(\tfrac{D}{2A+B}\right) \qquad [4]$$

The sign of the determinant, $|\mathbf{M}| = A^2 - B^2/4$, determines whether the conic section is a hyperbola (negative), ellipse (positive), or parabola (zero). The level curves of MPD are circles centered on (½, ½) when p = ½, but in general the center depends on p (Fig. 2A).

The solution for MPH was determined by first deriving the F1 mean from the hypergeometric distribution and then subtracting MPD (File S1). Whereas MPD depends on allele frequency, MPH does not: it has elliptical level curves at a 45° angle to the coordinate axes that are always centered at (½, ½) (Table 1, Fig. 2B). The eccentricity of the level curves, which

is the ratio between the major and minor axes of the ellipse, increases with ploidy according to $\sqrt{2\phi - 1}$.

Viewed in three dimensions, MPD is a concave paraboloid, with its maximum value at the center point. MPH is a convex paraboloid, with its lowest value at the center, and its highest value when the two parents are homozygous for different alleles. Their sum is a hyperbolic paraboloid, also known as a saddle surface, which increases with distance from the saddle point in some directions but decreases in other directions (Fig. 2C). The hyperbola asymptotes are only perpendicular for diploids; the area of the plane where the dominance contribution is positive decreases with ploidy (Fig. S1 in File S2).

Maximizing genetic merit

The proposed objective for OMA is to maximize the mean of the next generation:

$$\text{OMA:} \quad \max_{\mathbf{x}} \mathbf{g}'\mathbf{x} = \sum_i \sum_j \mathrm{g}_{ij} x_{ij} \qquad [5]$$

g_{ij} is the mean genotypic value (relative to the population mean) of progeny from female parent *i* and male parent *j*, and x_{ij} denotes the allocation for this mating (no relation to variable *x* in the previous section). Unless otherwise indicated, the sum index is from 1 to *n* parents. The n^2 x 1 column vectors **g** and **x** can be viewed as vectorized *n* x *n* matrices. The mate allocation variables are proportions that sum to 1. The total contribution (*y*) of an individual is the average of its female (y_f) and male (y_m) contributions, allowing for hermaphroditic species. (Some authors define *y* as a sum instead of average, which does not affect the results.) The sex-specific contributions are computed by summing over the other index of the mate allocation variables:

$$\begin{aligned} y_{f,i} &= \textstyle\sum_j x_{ij} & y_{m,j} &= \textstyle\sum_i x_{ij} \\ y_i &= \tfrac{1}{2}(y_{f,i} + y_{m,i}) & 1 &= \textstyle\sum_i y_i = \sum_i y_{f,i} = \sum_i y_{m,i} \end{aligned} \qquad [6]$$

In the previous section, the F1 progeny mean was expressed as the sum of mid-parent additive and dominance values and mid-parent heterosis. Alternatively, it can be expressed as the sum of general and specific combining abilities:

$$\mathrm{g}_{ij} = \tfrac{1}{2}(u_i + u_j) + \tfrac{1}{2}(v_i + v_j) + \mathrm{MPH}_{\mathrm{ij}} = \mathrm{GCA}_i + \mathrm{GCA}_j + \mathrm{SCA}_{ij} \qquad [7]$$

Breeding value (BV) is twice the GCA at HWE. In the diploid single locus model, breeding value equals additive value, which identifies SCA as MPD+MPH. For polyploids, BV includes a portion of the dominance value (Gallais 2003):

$$\mathrm{BV} = 2\mathrm{GCA} = u + \left(\frac{\frac{\phi}{2} - 1}{\phi - 1}\right) v \qquad [8]$$

If SCA is neglected, Eq. 5 reduces to maximizing the average breeding value of the parents, weighted by their contributions, which is the typical OCS objective:

$$\text{OCS:} \quad \max_{\mathbf{y}} \sum_i BV_i y_i \qquad [9]$$

Inbreeding control

The inbreeding coefficient F_t in generation *t* is the probability that two genes at one locus are identical by descent (IBD). For polyploids, the genes are randomly sampled without replacement (Gallais 2003). Using the law of total probability, we can condition on whether the two genes are derived from the same gene in the previous generation, which occurs with probability ΔF:

$$F_{t+1} = \Delta F + (1 - \Delta F) F_t \qquad [10]$$

Eq. 10 is a recurrence equation between the inbreeding coefficients from adjacent generations, and ΔF is the inbreeding rate. Iterating one more generation produces

$$F_{t+2} = \Delta F(2 - \Delta F) + (1 - \Delta F)^2 F_t \qquad [11]$$

and an explicit time equation follows from the formula for a finite geometric series:

$$F_t = 1 - (1 - \Delta F)^t (1 - F_0) \qquad [12]$$

In the OCS literature, the constraint on inbreeding rate is implemented using the quadratic form $\mathbf{y}'\mathbf{K}\mathbf{y}$, which is called group coancestry, or kinship ($\mathbf{y}$ is the vector of contributions, and $\mathbf{K}$ is the kinship matrix). To derive the formula for any ploidy, let F_{ij} denote the inbreeding coefficient of an individual in generation *t*+1 with parents *i* and *j*, which have inbreeding coefficients F_i and F_j, respectively. Ignoring double reduction, the relationship between inbreeding and kinship across one generation is (Kerr et al. 2012)

$$(\phi - 1) F_{ij} = \tfrac{\phi}{2} K_{ij} + \tfrac{1}{2}\left(\tfrac{\phi}{2} - 1\right)\left(F_i + F_j\right) \qquad [13]$$

The average inbreeding coefficient of generation *t*+1 is the average over all possible matings, weighted by their allocation:

$$(\phi - 1)\bar{F}_{t+1} = (\phi - 1) \sum_i \sum_j x_{ij} F_{ij} = \tfrac{\phi}{2} \sum_i \sum_j x_{ij} K_{ij} + \left(\tfrac{\phi}{2} - 1\right) \sum_i y_i F_i \qquad [14]$$

$$= \frac{\phi}{2}\mathbf{x}'\mathrm{vec}(\mathbf{K}) + \left(\frac{\phi}{2} - 1\right)\mathbf{y}'\mathbf{F}$$

The second equality in Eq. 14 follows from the relationship between mate allocations and individual contributions (Eq. 6), and the third equality introduces vector notation. The symbol $\bar{F}_{t+1}$ has the same meaning as F_{t+1} in Eq. 10, but the overbar (for average) is introduced to differentiate the two kinds of subscripts: time vs. individuals. Under random mating, mate allocation is the product of the female and male contributions ($x_{ij} = y_{f,i}y_{m,j}$), which leads to

$$(\phi - 1)\bar{F}_{t+1} = \frac{\phi}{2}\mathbf{y}_f'\mathbf{K}\mathbf{y}_m + \left(\frac{\phi}{2} - 1\right)\mathbf{y}'\mathbf{F} \qquad [15]$$

For hermaphroditic individuals with equal female and male contributions ($\mathbf{y}_f = \mathbf{y}_m = \mathbf{y}$), the expression further simplifies to

$$(\phi - 1)\bar{F}_{t+1} = \frac{\phi}{2}\mathbf{y}'\mathbf{K}\mathbf{y} + \left(\frac{\phi}{2} - 1\right)\mathbf{y}'\mathbf{F} \qquad [16]$$

Analogous to Eq. 13, the relationship between inbreeding and kinship between generations *t*+2 and *t*+1 is

$$(\phi - 1)F_{ij,ab} = \frac{\phi}{2}K_{ij,ab} + \frac{1}{2}\left(\frac{\phi}{2} - 1\right)\left(F_{ij} + F_{ab}\right) \qquad [17]$$

$F_{ij,ab}$ denotes the inbreeding coefficient of an individual in generation *t*+2 with female parent ij in generation *t*+1 and maternal grandparents *i* and *j* in generation *t*. The male parent is *ab*, with paternal grandparents *a* and *b*. Using Eq. 13 and that parental kinship is the average of four grandparental kinships, inbreeding in *t*+2 can be related to the parameters of generation *t*:

$$\begin{aligned}(\phi - 1)^2 F_{ij,ab} &= \frac{\phi}{8}(\phi - 1)\left[K_{ia} + K_{ib} + K_{ja} + K_{jb}\right] \\ &\quad + \frac{\phi}{4}\left(\frac{\phi}{2} - 1\right)\left(K_{ij} + K_{ab}\right) + \frac{1}{4}\left(\frac{\phi}{2} - 1\right)^2\left(F_i + F_j + F_a + F_b\right)\end{aligned} \qquad [18]$$

The average inbreeding coefficient of generation *t*+2 is the average over matings in *t*+1, weighted by their allocation. Since mate allocation in generation *t*+1 is not yet determined, random mating is assumed as a conservative bound, which leads to the following result (derivation in File S1):

$$(\phi - 1)^2\bar{F}_{t+2} = \frac{\phi}{2}(\phi - 1)\mathbf{y}'\mathbf{K}\mathbf{y} + \left(\frac{\phi}{2} - 1\right)\left[\frac{\phi}{2}\mathbf{x}'\mathrm{vec}(\mathbf{K}) + \left(\frac{\phi}{2} - 1\right)\mathbf{y}'\mathbf{F}\right] \qquad [19]$$

METHODS

Convex Optimization

The convex OMA problem is

$$\max_{\mathbf{x},\mathbf{y}} \ \mathbf{g}'\mathbf{x}$$

$$\mathbf{0} \leq \mathbf{L}_{\mathrm{x}} \leq \mathbf{x} \leq \mathbf{U}_{\mathrm{x}} \leq \mathbf{1}$$

$$\mathbf{0} \leq \mathbf{L}_{\mathrm{y}} \leq \mathbf{y} \leq \mathbf{U}_{\mathrm{y}} \leq \mathbf{1}$$

$$\mathbf{1}'\mathbf{x} = 1$$

$$\mathbf{y} = \mathbf{M}\mathbf{x}$$

$$\mathbf{s}'\mathbf{y} = \tfrac{1}{2}$$

$$\mathbf{a} \leq \mathbf{C}\mathbf{y} \leq \mathbf{b}$$

$$\tfrac{\phi}{2}\mathbf{x}'\mathrm{vec}(\mathbf{K}) + \left(\tfrac{\phi}{2} - 1\right)\mathbf{y}'\mathbf{F} \leq (\phi - 1)[\Delta F + (1 - \Delta F)\bar{F}]$$

$$\tfrac{\phi}{2}(\phi - 1)\mathbf{y}'\mathbf{K}\mathbf{y} + \left(\tfrac{\phi}{2} - 1\right)\left[\tfrac{\phi}{2}\mathbf{x}'\mathrm{vec}(\mathbf{K}) + \left(\tfrac{\phi}{2} - 1\right)\mathbf{y}'\mathbf{F}\right] \leq (\phi - 1)^2[\Delta F(2 - \Delta F) + (1 - \Delta F)^2\bar{F}]$$

The genetic merit $\mathbf{g}$ for each mating was the sum of MPA, MPD, and MPH across all markers. The first two constraints represent lower and upper bounds on the mate allocation and contribution variables. The incidence matrix $\mathbf{M}$ in constraint $\mathbf{y} = \mathbf{M}\mathbf{x}$ captures the relationship between the two types of variables in Eq. 6. Constraint $\mathbf{s}'\mathbf{y} = \frac{1}{2}$ is only used with separate sexes, and the elements of $\mathbf{s}$ are 0 for females and 1 for males to normalize the sex-specific contributions. Equation $\mathbf{a} \leq \mathbf{C}\mathbf{y} \leq \mathbf{b}$ represents additional (optional) linear constraints on the contributions; an example is given below for the potato dataset. The second-to-last inequality combines Eq. 10 and 16, and the last inequality combines Eq. 11 and 19. $\mathbf{K}$ must be positive semidefinite for convexity.

If $\boldsymbol{\delta}$ denotes the vector of breeding values, the convex OCS problem is

$$\max_{\mathbf{y}} \boldsymbol{\delta}'\mathbf{y}$$

$$\mathbf{0} \leq \mathbf{L}_{\mathrm{y}} \leq \mathbf{y} \leq \mathbf{U}_{\mathrm{y}} \leq \mathbf{1}$$

$$\mathbf{1}'\mathbf{y} = 1$$

$$\mathbf{s}'\mathbf{y} = \tfrac{1}{2}$$

$$\mathbf{a} \leq \mathbf{C}\mathbf{y} \leq \mathbf{b}$$

$$\tfrac{\phi}{2}\mathbf{y}'\mathbf{K}\mathbf{y} + \left(\tfrac{\phi}{2} - 1\right)\mathbf{y}'\mathbf{F} \leq (\phi - 1)[\Delta F + (1 - \Delta F)\bar{F}]$$

The most appropriate inbreeding constraint, Eq. 17, is not convex, so the hermaphroditic form, Eq. 18, is used instead. This is the traditional OCS approach (Woolliams et al. 2015), but as Eq. 19 illustrates, it is more restrictive than necessary to meet the target rate with separate sexes. However, using Eq. 19 in lieu of Eq. 18 does not guarantee the target rate is met.

R package COMA (https://github.com/jendelman/COMA) uses R package CVXR (Fu et al. 2020) to solve convex formulations of OCS and OMA. CVXR transforms high-level mathematical descriptions of convex optimization problems into the format required of convex solvers. The group kinship constraints identify OCS and OMA as second-order cone programs (SOCP), and the ECOS solver that comes with CVXR is the default in COMA. During the breeding program simulation described below, ECOS sometimes returned with status "optimal_inaccurate" instead of the desired result "optimal;" this issue was resolved by using the commercial solver MOSEK, which must be installed separately.

From the optimal solution, the average inbreeding coefficients of the progeny and grand-progeny are calculated according to Eq. 16 and 19, respectively. The average inbreeding rates for the progeny, ΔF_1, and grand-progeny, ΔF_2, are then computed by inverting Eq. 12:

$$\Delta F_t = 1 - \left(\frac{1 - F_t}{1 - F_0}\right)^{1/t}$$

Potato example

The potato dataset from the R/StageWise package (Endelman 2023) contained five years of phenotype data for three traits—total yield, fry color, and vine maturity—and 12K SNP marker data for 943 tetraploid clones. Two-stage analysis of the multi-year trial followed Vignette 3 of the package (File S5). Genotype means (BLUEs) were estimated per environment (*StageWise::Stage1*) and then used as the response variable in the multi-environment mixed

model (*StageWise::Stage2*), using ASReml-R v4.2 for variance component estimation (Butler et al. 2023). The mixed model includes additive u_{ik} and dominance v_{ik} values for trait k of clone i, the fixed effect E_{jk} of trait k in environment j, a random effect s_{ijk} for the Stage 1 error, and the residual gE_{ijk} is the genotype x environment interaction:

$$y_{ijk} = E_{jk} + u_{ik} + v_{ik} + s_{ijk} + gE_{ijk}$$

The Stage 1 error is multivariate normal, with variance-covariance (var-cov) equal to the direct-sum of the var-cov matrices of the multi-trait BLUEs from Stage 1. The genetic and residual effects were also multivariate normal with separable covariance structures for clones vs. traits:

$$\begin{aligned} \mathbf{u} &\sim \mathrm{MVN}(\mathbf{0}, \mathbf{G} \otimes \mathbf{\Sigma}_{\mathrm{A}}) \\ \mathbf{v} &\sim \mathrm{MVN}(-b\mathbf{f}_{\mathrm{D}}, \mathbf{D} \otimes \mathbf{\Sigma}_{\mathrm{D}}) \\ \mathbf{gE} &\sim \mathrm{MVN}(\mathbf{0}, \mathbf{I} \otimes \mathbf{\Sigma}_{\mathrm{GE}}) \end{aligned} \qquad [20]$$

The trait covariance matrices ($\mathbf{\Sigma}$) were unstructured. The additive **G** and dominance **D** genomic relationship matrices follow the multi-locus extension of Eq. 1 and use allele frequencies of the current population (Endelman 2023):

$$\begin{aligned} \mathbf{G} &= \frac{\mathbf{WW}^{\mathrm{T}}}{\phi \sum_k p_k q_k} \\ \mathbf{D} &= \frac{\mathbf{QQ}^{\mathrm{T}}}{\binom{\phi}{2} \sum_k 4p_k^2 q_k^2} \\ \mathbf{f}_{\mathrm{D}} &= \frac{-\mathbf{Q1}}{\binom{\phi}{2} \sum_k 2p_k q_k} \end{aligned} \qquad [21]$$

The inbreeding covariate $\mathbf{f}_{\mathrm{D}}$ is the normalized row-sum of **Q**, and its regression coefficient b represents baseline heterosis, or inbreeding depression.

Empirical BLUPs for marker effects were based on a multi-trait, "desired gains" index (*StageWise::gain* and *StageWise::blup*), with the contribution of inbreeding depression included in the dominance effects (Endelman 2023). To manage the unfavorable correlation between late maturity and yield, equal gains (in units of genetic standard deviation) were specified for total yield and fry color and zero gain for maturity. The estimates of dominance variance and inbreeding depression relative to additive variance were computed with *StageWise::dominance*.

The PVY resistance data are based on KASP marker snpST00073 for $\mathrm{Ry}_{\mathrm{adg}}$ (Kante et al. 2021), which is co-dominant but not reliable at differentiating the three possible heterozygotes in

tetraploid potato. Based on the low frequency of Ry_{adg} in our germplasm, a frequency of ¼ (i.e., single copy) was assumed for clones testing positive for the marker.

Breeding simulation

A "mass" selection program with non-overlapping generations and hermaphroditic individuals was simulated using AlphaSimR (Gaynor et al. 2021). R/SimPlus (https://github.com/jendelman/SimPlus) was created to standardize breeding and analysis modules across the treatments (File S3). For each of the ten replicate simulations per ploidy (2x and 4x), a founder population (N=500) in mutation-drift equilibrium was created using the coalescent, with mutation rate 10^{-8}, effective population size 100, and a genome of ten 1 Morgan chromosomes. These parameters generated approximately 15K segregating sites in the 2x founders and 30K in the 4x founders. Founder haplotypes were tracked across generations using setTrackRec=TRUE. An "ADG" trait was simulated with the following parameters: nQtlPerChr=1000, mean=0, var=1, varDD=0.5, meanDD=0.5, useVarA=FALSE, varE=2. This fixes the non-genetic variance at 2 throughout the simulation, so the broad-sense heritability was 1/3 in the founder population and decreased as genetic variance decreased.

Each selection cycle consisted of (1) phenotyping the candidates, (2) updating marker effects based on the candidates and two previous generations, (3) developing the mating plan by OCS or OMA, (4) and generating new progeny. The phenotypes and complete genotype data (including QTL) were written to file and analyzed with R/StageWise to predict the additive and dominance marker effects for COMA. Due to the large number of mate allocation variables (125,250) for all 500 candidates, the candidate pool was reduced to 200 individuals by OCS before applying OMA. A lower bound N on the number of selected parents can be achieved using an upper limit $1/N$ on the parental contributions. When there was no solution at the target rate (i.e., problem infeasible), the limit was increased in increments of 0.5% (dF.step parameter of *COMA::ocs* and *COMA::oma*).

Selection accuracy was estimated each generation based on the correlation between the true and predicted means for 100 randomly selected matings, using 50 progeny per mating. The OCS prediction is based on the sum of the GCAs, while OMA also includes SCA.

Three different kinship models were tested and named based on the corresponding relationship matrix. (Kinship equals relationship divided by ploidy.) Method A is pedigree

kinship based on the probability of IBD and was computed with R/AGHmatrix (Amadeu et al. 2016). Method G_{VR} is Method 1 of VanRaden (2008) and its extension to tetraploids (Endelman et al. 2018), using founder allele frequencies as the reference population. G_{IBD} is genomic kinship and extends method A to capture Mendelian segregation. G_{IBD} is equivalent to an allele matching coefficient using founder alleles, which are obtained in AlphaSimR with *pullIbdHaplo*. Computation of G_{IBD} and G_{VR} was implemented as function *G_mat* in R/polyBreedR (Endelman et al. 2024). Individual inbreeding coefficients were computed from the diagonals of the corresponding relationship matrix, A or G (Gallais 2003):

$$F_i = (A_{ii} - 1)/(\phi - 1)$$

The mating plan from COMA was realized in AlphaSimR using *SimPlus::sim_mate*, which takes as input a vector **x** of mate allocations. The mating plan under OCS was specified as a diallel with mate allocation equal to the product of the parental contributions. Vector **x** represents a discrete distribution, $f(k) = x_k$, and progeny were allocated based on this distribution in batches based on the parameter *min.progeny*, which is the minimum family size. For the simulation *min.progeny*=5 was used, which represents a minimum allocation of 1% because the total population size each generation was 500.

RESULTS

Potato example

The COMA software has functions for both OCS and OMA (tutorial in File S4). OCS maximizes the average genomic-estimated breeding value (GEBV), while OMA maximizes the average genomic prediction of mate performance (GPMP). The tutorial data is from the University of Wisconsin potato breeding program. The parental candidates are 170 tetraploid clones genotyped at 12K SNPs, with kinship (i.e., coancestry) estimated from pedigree records. The inbreeding coefficients ranged from 0.00 to 0.09 (median 0.02). Kinship ranged from 0.01 to 0.19 (median 0.09). Additive and dominance marker effects were predicted by BLUP using a multi-trait selection index. The estimated dominance variance for the index was 1/3 the additive variance (σ_A^2), and inbreeding depression was estimated to be 6 times σ_A (Table 2).

OCS and OMA solutions were computed for inbreeding rates 0.5%, 1.0%, 1.5%, and 2.0%. Neither method was feasible at 0.5%, which reflects the relatedness of the candidates. The lowest feasible rates were 1.0% and 1.5% for OMA and OCS, respectively, illustrating the potential for less inbreeding with non-random mating. Additionally, because OMA accounts for heterosis, it finds solutions with higher merit than OCS. At 1.5% inbreeding, the predicted merit of the F1 progeny was 1.3σ with OMA (σ = genetic standard deviation of the parents) compared to 0.7σ with OCS. As expected, genetic merit increased as higher rates of inbreeding were allowed (Table 3).

The OMA solution returns two measures of inbreeding rate: one for the progeny (ΔF_1) and one for grand-progeny (ΔF_2), assuming random mating of the progeny (see Theory and Methods). The latter is needed to ensure the target inbreeding rate can be sustained for many generations, similar to the group kinship constraint in OCS. ΔF_2 was consistently at the limit but ΔF_1 was not (Table 3). When the limit was 1%, the optimal solution was $\Delta F_1 = -0.5\%$, indicating slight outbreeding.

An important feature of COMA is that the solutions are *sparse*, meaning that many variables are zero at the optimum. The OMA solution at $\Delta F = 1.5\%$ utilized 24 of the 170 candidates, with a maximum parental contribution of 19% and median of 3% (Table 3, File S4). The OCS solution utilized 29 parents, with a maximum contribution of 10% and median of 3%. Of the nearly 15K possible matings of the 170 candidates, the OMA solution only used 20 matings. OCS assumes random mating of the parents, which would require 435 matings if a complete

half-diallel (with selfs) were generated. At higher rates of inbreeding, fewer parents and matings were selected (Table 3).

The COMA tutorial shows how additional linear equality or inequality constraints can be incorporated. Besides the three quantitative traits in the selection index, resistance to potato virus Y (PVY) is an important qualitative trait, which was scored with a genetic marker. For the OMA solution at $\Delta F = 1.0\%$, only 9 of the 25 matings had a resistant parent, which corresponds to an allele frequency of 8.5% in the next generation assuming single-copy dosage in the parents (Table 3). One approach is to require that every mating has at least one resistant parent, by setting the maximum allocation to 0 for matings that do not meet this criterion. This increased the resistant allele frequency to 12.5% in the progeny, with the tradeoff being slightly lower genetic merit (1.16σ vs. 1.33σ). Alternatively, the allele frequency can be constrained directly with an inequality of the form $\mathbf{c}'\mathbf{y} \geq p$, where **s** is the vector of allele frequencies in the parents (in this case, either ¼ or 0) and p is the target allele frequency, in this case 1/8. The predicted merit under this approach was 1.27σ, which is closer to the unconstrained solution (Table 3).

Breeding simulation

The long-term performance of OMA and OCS was investigated via stochastic simulation of a "mass" selection program, with a constant population size of 500 hermaphroditic individuals and non-overlapping generations. Additive and dominance QTL parameters were drawn from the same distributions for diploid (2x) and tetraploid (4x) simulations, which produced traits with more dominance variance but less inbreeding depression (relative to additive variance) for the 2x vs. 4x founders (Table 2). 10 generations of phenotypic selection at 0.5% inbreeding were simulated before applying different methods of genomic selection (GS) for 30 generations. (Generation 0 designates the beginning of GS.) To reduce computing time, the training population for estimating marker effects was limited to phenotype and genotype data for the selection candidates and two previous generations.

The COMA implementation of OCS was successful at limiting inbreeding to the target rate in both 2x and 4x simulations, using either pedigree (A) or genomic (G_{IBD}) kinship (Fig. 3A). Both kinship methods are based on the probability of IBD, but only G_{IBD} includes Mendelian sampling. The realized inbreeding rate for diploids under method A slightly exceeded the 0.5% target, but this was not because of COMA. Rather, a minimum allocation of 1% was imposed

during mating (common practice in potato breeding), effectively limiting the number of matings to 100 (Fig. 4). Method A@0.5% had 85 parents on average—the highest among the OCS scenarios—and with only 100 matings there was underrepresentation of parents with small contributions. Fewer parents were selected by OCS when using G_{IBD} compared to A, but the number of matings was comparable (Fig. 4 and Fig. S2 in File S2).

The results with OMA were more conservative than OCS with respect to realized inbreeding (Fig. 3B). Inbreeding *decreased* in the first generation and then recovered within 5 generations. For both 2x and 4x simulations, the realized inbreeding rate with method A was close to 0.5%, regardless of whether the limit was 0.5% or 1.0%. With method G_{IBD}, the realized rate was only 0.1% even when the limit was 1%. The number of matings with OMA was half that of OCS in diploids (Fig. 4), and even less in tetraploids (Fig. S2 in File S2). As expected, fewer 4x parents were selected than 2x parents at the same inbreeding rate.

At 1% inbreeding, the 2x genetic gain profiles were broadly similar for OCS-A, OCS-G_{IBD}, and OMA-A, but OCS-A depleted more genic variance than the other methods, making it less efficient (Fig. 5). Efficiency was defined as genetic gain divided by the decrease in genic standard deviation (SD), which is a dimensionless ratio (Gorjanc et al. 2018). All methods showed a decrease in efficiency over time, indicating that each additional unit of gain required more genic SD. Because of its very low realized inbreeding rate, OMA-G_{IBD} was not competitive with the other methods for genetic gain. The 4x results were similar except the gain with OCS-G_{IBD} was slightly below the results with OCS-A and OMA-A (Fig. S3 in File S2).

For a more sensitive comparison of the top three methods, their relative differences were compared over time at 1% inbreeding, using OMA-A as the reference method (Fig. 6). OMA-A always had higher gain for the first several generations, but this advantage eroded within 5 generations even though its prediction accuracy remained higher by 0.2–0.3 throughout (Fig. S4 in File S2). Only in the 4x comparison with OCS-G_{IBD} did OMA-A maintain both higher gain and higher efficiency over 30 generations. In the 4x comparison with OCS-A, the gains were similar, but OMA-A maintained higher efficiency. In the 2x simulation, the long-term gain was similar with all three methods, but OMA-A had higher efficiency than OCS-A. Lowering the inbreeding rate to 0.5% increased the efficiency of OCS-A without reducing its long-term gain, making it more competitive with OMA-A (Fig. S5 in File S2). OCS-G_{IBD} lost too much gain at the lower inbreeding rate, so it was more competitive at 1% inbreeding.

DISCUSSION

Two kinds of inbreeding coefficient were used in this study. To account for inbreeding depression, a covariate of genomic inbreeding coefficients (f_D) was used in the mixed model for predicting additive and dominance marker effects. From the theory of directional dominance (Endelman 2023), f_D is proportional to the dominance regressor Q, averaged over loci (Eq. 1, 21). Endelman (2023) observed in the potato dataset that the population mean $\overline{f_D}$ was numerically identical to $\overline{f_A}$, calculated from the diagonal elements of the additive genomic relationship matrix, $\mathbf{G}_{\mathrm{VR}}$ (VanRaden 2008), although the individual values were not identical. The derivation of this mathematical equality is given in File S1:

$$\overline{f_A} = \frac{n^{-1}\sum_{i=1}^{n} G_{ii} - 1}{\phi - 1} = \frac{\overline{G_{\cdot\cdot}} - 1}{\phi - 1}$$

$$\overline{f_D} = \frac{\sum_k \overline{Q_{\cdot k}}}{\phi(\phi-1)\sum_k p_k q_k} \qquad [22]$$

$$\overline{f_A} = \overline{f_D} = \bar{f} = (\phi-1)^{-1}\left[\frac{\sum_k \mathrm{var}(X_k)}{\sum_k \phi p_k q_k} - 1\right]$$

Eq. 22, which uses allele frequencies p_k of the current population, indicates $\bar{f} = 0$ when the population variance of the allele dosages, var(X), equals the variance of a binomial random variable with the same allele frequency (ϕpq). The maximum value of $\bar{f}$ occurs when there are no heterozygotes, in which case $\mathrm{var(X)} = \phi^2 pq$, so $\bar{f}_{\max} = 1$. The minimum value of $\bar{f}$ occurs when $\mathrm{var(X)} = 0$, so $\bar{f}_{\min} = \frac{-1}{\phi-1}$.

For diploids, $\bar{f}$ is an inbreeding coefficient in the sense of Wright (1921); in other words, it is a gametic correlation, consistent with the limits [-1,1] shown above (Powell et al. 2010; Endelman and Jannink 2012). A classic result is that $-2pq\bar{f}$ is the frequency reduction of heterozygotes compared to HWE (Kempthorne 1957), and negative values of $\bar{f}$ indicate excess heterozygosity. With this result, the expected values of MPD and MPH for a parent with allele frequency x were derived (File S1). MPD and MPH are equal in magnitude with opposite sign because dominance effects do not contribute to breeding value in diploids:

$$E[\mathrm{MPH}|x] = -E[\mathrm{MPD}|x] = \left[2x^2 - (1+2p)x + p^2 + pq\bar{f}\right]d \qquad [23]$$

Averaging over x, Eq. 23 reduces to $\left(2pq\bar{f}\right)d$. When $\bar{f}$=1, MPH is the difference between F1 hybrids and fully inbred parents, which is called baseline heterosis in the context of two different

source populations (Lamkey and Edwards 1999; Labroo et al. 2021). When the parental populations are identical, baseline heterosis reduces to *2pqd*, in agreement with the above result.

The second type of inbreeding coefficient, denoted *F*, was introduced to limit inbreeding during mate selection. For this application, *F* was defined as the probability of IBD, and both pedigree (A) and genomic versions (G_{IBD}) were effective at meeting the target inbreeding rate in the simulation. The key requirement for inbreeding control (Eq. 13–16) is that the definitions of kinship and inbreeding follow a consistent framework of identity probabilities. Kinship based on the *probability* of identity-by-state (IBS) would provide a valid framework for inbreeding rate control, but, as discussed above, the inbreeding coefficient $\bar{f}$ from $\mathbf{G}_{VR}$ is a correlation, not a probability. Furthermore, $\bar{f}$ remains zero under OCS even as *F* increases toward 1 because of random mating. Several studies have used $\mathbf{G}_{VR}$ with allele frequencies from a base population, rather than the current population, to control inbreeding (Sonesson et al. 2012; Henryon et al. 2019; Meuwissen et al. 2020). With this approach, $\overline{f_A}$ from the diagonal elements of $\mathbf{G}_{VR}$ (Eq. 22) no longer has the theoretical properties described above and can exceed 1. Changing the reference population does not resolve the theoretical disconnect with the OCS group kinship constraint, so there is no guarantee of proper inbreeding control. In our simulation with a target inbreeding rate of 0.5%, the realized rate was 3–4% and genic variance was depleted too rapidly, limiting the amount of long-term gain (Fig. S6 in File S2).

Kinship methods A and G_{IBD} performed similarly for OCS in our simulations, but with OMA the realized inbreeding rate using G_{IBD} was much too low. Considering both gain and efficiency, the optimal inbreeding rate for OCS-G_{IBD} was slightly higher than OCS-A. These results are generally in line with previous comparisons of G_{IBD} and A for OCS in diploids (Henryon et al. 2019; Meuwissen et al. 2020), but precise comparisons are difficult because of differences in study design and reporting. From a practical perspective, the A matrix is much simpler to compute than G_{IBD}, which requires linkage analysis across multiple generations and for which there is no published method in polyploids.

Based on the results of Werner et al. (2023), who also used stochastic simulation to compare selection on GEBV vs. GPMP in diploids, I had expected OMA to achieve more long-term gain than OCS. However, the gain advantage of OMA vs. OCS eroded within 5 generations in this study, even for a trait with substantial dominance (Table 2). I believe the main reason for the different outcomes is the lack of explicit inbreeding control in Werner et al. (2023), as they

reported lower inbreeding rates for truncation selection on GPMP compared to GEBV. By comparing the methods at the same inbreeding rate, this study reinforces the idea that selection on additive value is the main driver of long-term gain. However, it is noteworthy that OMA was more efficient at converting genic variance into gain than OCS. Extrapolating out another 20-30 generations, genic variance with OCS would have been depleted before OMA, giving OMA the advantage once again.

The sparsity of the OMA mating design can be a significant practical advantage over OCS, as random mating is difficult to realize for many species. Earlier versions of COMA have been used for several years in the University of Wisconsin potato breeding program, and due to the prevalence of poor reproductive fertility (e.g., flower abortion, cytoplasmic male sterility), the optimal mating plan has never been realized. OMA can be used a second time after the crossing is done, converting the actual seed quantities available into upper limits for mate allocation. The optimal solution then informs how many seeds to sow per mating for the next generation.

TABLES

Table 1. Parameters of the quadratic equation for dominance values, $Ax_1^2 + Bx_1x_2 + Cx_2^2 + Dx_1 + Ex_2 + F$.

	Mid-Parent Value	Mid-Parent Heterosis	F1 Mean
A=C	$-\frac{\phi^2}{2}$	$\frac{\phi^3}{4(\phi-1)}$	$\frac{-\phi^2(\phi-2)}{4(\phi-1)}$
B	0	$-\frac{\phi^2}{2}$	$-\frac{\phi^2}{2}$
D=E	$\phi\left[p(\phi-1)+\frac{1}{2}\right]$	$\frac{-\phi^2}{4(\phi-1)}$	$\frac{\phi[4p(\phi-1)^2+\phi-2]}{4(\phi-1)}$
F	$-\phi(\phi-1)p^2$	0	$-\phi(\phi-1)p^2$
Shape	Circle	Ellipse	Hyperbola

Table 2. Dominance variance (σ_D^2) and inbreeding depression (b) relative to additive variance (σ_A^2). The mean (SD) is reported for the founders (Generation -10) of 10 independent simulations.

Dataset	Ploidy	σ_D^2/σ_A^2	b/σ_A
Potato	4x	0.33	6.2
Simulation Founders	4x	0.13 (0.05)	7.5 (1.2)
Simulation Founders	2x	0.25 (0.06)	4.3 (0.5)

Table 3. Potato OMA results.

ΔF limit	PVY Constraint	ΔF_1	ΔF_2	Response (σ)	# Parents	# Matings	PVYR freq
1.0%	None	-0.5%	1.0%	1.33	29	25	8.5%
1.5%	None	0.0%	1.5%	2.05	24	20	8.3%
2.0%	None	0.4%	2.0%	2.43	18	15	7.4%
1.0%	At least one R parent per mating	-0.5%	1.0%	1.16	27	24	12.5%
1.0%	PVYR freq ≥ 0.125	-0.5%	1.0%	1.27	30	25	12.5%

References

Akdemir D., and J. I. Sánchez, 2016 Efficient breeding by genomic mating. Front Genet 7: 210. https://doi.org/10.3389/fgene.2016.00210

Amadeu R. R., C. Cellon, J. W. Olmstead, A. A. F. Garcia, M. F. R. Resende, *et al.*, 2016 AGHmatrix: R package to construct relationship matrices for autotetraploid and diploid species: a blueberry example. Plant Genome 9. https://doi.org/10.3835/plantgenome2016.01.0009

Boyd S., and L. Vandenberghe, 2004 *Convex optimization*. Cambridge University Press, New York.

Butler D. G., B. R. Cullis, A. R. Gilmour, B. J. Gogel, and R. Thompson, 2023 *ASReml-R Reference Manual Version 4.2*. VSN International, Hempstead.

Endelman J. B., and J. L. Jannink, 2012 Shrinkage estimation of the realized relationship matrix. G3: Genes, Genomes, Genetics 2: 1405–1413. https://doi.org/10.1534/g3.112.004259

Endelman J. B., 2023 Fully efficient, two-stage analysis of multi-environment trials with directional dominance and multi-trait genomic selection. Theoretical and Applied Genetics 136: 65. https://doi.org/10.1007/s00122-023-04298-x

Endelman J. B., M. Kante, H. Lindqvist-Kreuze, A. Kilian, L. M. Shannon, *et al.*, 2024 Targeted genotyping-by-sequencing of potato and data analysis with R/polyBreedR. Plant Genome e20484. https://doi.org/10.1002/tpg2.20484

Falconer D. S., and T. F. C. Mackay, 1996 *Introduction to Quantitative Genetics*. Longman Group, Essex.

Fu A., B. Narasimhan, and S. Boyd, 2020 CVXR: An R package for disciplined convex optimization. J Stat Softw 94: 1–34. https://doi.org/10.18637/jss.v094.i14

Gallais A., 2003 *Quantitative genetics and breeding methods in autopolyploid plants.* INRA, Paris.

Gaynor R. C., G. Gorjanc, A. R. Bentley, E. S. Ober, P. Howell, *et al.*, 2017 A two-part strategy for using genomic selection to develop inbred lines. Crop Sci 57: 2372–2386. https://doi.org/10.2135/cropsci2016.09.0742

Gaynor R. C., G. Gorjanc, and J. M. Hickey, 2021 AlphaSimR: An R package for breeding program simulations. G3: Genes, Genomes, Genetics 11: jkaa017. https://doi.org/10.1093/G3JOURNAL/JKAA017

Gorjanc G., R. C. Gaynor, and J. M. Hickey, 2018 Optimal cross selection for long-term genetic gain in two-part programs with rapid recurrent genomic selection. Theoretical and Applied Genetics 131: 1953–1966. https://doi.org/10.1007/s00122-018-3125-3

Gorjanc G., and J. M. Hickey, 2018 AlphaMate: A program for optimizing selection, maintenance of diversity and mate allocation in breeding programs. Bioinformatics 34: 3408–3411. https://doi.org/10.1093/bioinformatics/bty375

Grundy B., B. Villanueva, and J. A. Woolliams, 1998 Dynamic selection procedures for constrained inbreeding and their consequences for pedigree development. Genet Res 72: 159–168. https://doi.org/10.1017/S0016672398003474

Henryon M., H. Liu, P. Berg, G. Su, H. M. Nielsen, *et al.*, 2019 Pedigree relationships to control inbreeding in optimum-contribution selection realise more genetic gain than genomic relationships. Genetics Selection Evolution 51: 39. https://doi.org/10.1186/s12711-019-0475-5

Kante M., H. Lindqvist-Kreuze, L. Portal, M. David, and M. Gastelo, 2021 Kompetitive Allele Specific PCR (KASP) markers for potato: An effective tool for increased genetic gains. Agronomy 11: 2315. https://doi.org/10.3390/agronomy11112315

Kempthorne O., 1957 *An Introduction to Genetic Statistics*. John Wiley & Sons, New York.

Kerr R. J., L. Li, B. Tier, G. W. Dutkowski, and T. A. McRae, 2012 Use of the numerator relationship matrix in genetic analysis of autopolyploid species. Theoretical and Applied Genetics 124: 1271–1282.

Kinghorn B. P., 2011 An algorithm for efficient constrained mate selection. Genetics Selection Evolution 43: 4. https://doi.org/10.1186/1297-9686-43-4

Labroo M. R., A. J. Studer, and J. E. Rutkoski, 2021 Heterosis and Hybrid Crop Breeding: A Multidisciplinary Review. Front Genet 12: 643761.

Labroo M. R., J. B. Endelman, D. C. Gemenet, C. R. Werner, R. C. Gaynor, *et al.*, 2023 Clonal diploid and autopolyploid breeding strategies to harness heterosis: insights from stochastic simulation. Theoretical and Applied Genetics 136: 147. https://doi.org/10.1007/s00122-023-04377-z

Lamkey K., and J. W. Edwards, 1999 Quantitative genetics of heterosis, pp. 31–48 in *Genetics and Exploitation of Heterosis in Crops*, edited by Coors J. G., Pandey S. American Society of Agronomy, Madison, WI.

Meuwissen T. H. E., 1997 Maximizing the Response of Selection with a Predefined Rate of Inbreeding. J Anim Sci 75: 934–940.

Meuwissen T. H. E., A. K. Sonesson, G. Gebregiwergis, and J. A. Woolliams, 2020 Management of Genetic Diversity in the Era of Genomics. Front Genet 11: 880. https://doi.org/10.3389/fgene.2020.00880

Pong-Wong R., and J. A. Woolliams, 2007 Optimisation of contribution of candidate parents to maximise genetic gain and restricting inbreeding using semidefinite programming. Genetics Selection Evolution 39: 3–25. https://doi.org/10.1051/gse:2006031

Powell J. E., P. M. Visscher, and M. E. Goddard, 2010 Reconciling the analysis of IBD and IBS in complex trait studies. Nat Rev Genet 11: 800–805.

Sonesson A. K., J. A. Woolliams, and T. H. E. Meuwissen, 2012 Genomic selection requires genomic control of inbreeding. Genetics Selection Evolution 44: 27. https://doi.org/10.1186/1297-9686-44-27

Toro M., and M. Pérez-Enciso, 1990 Optimization of selection response under restricted inbreeding. Genetics Selection Evolution 22: 93–107.

Toro M. A., and L. Varona, 2010 A note on mate allocation for dominance handling in genomic selection. Genetics Selection Evolution 42: 33.

Vitezica Z. G., L. Varona, and A. Legarra, 2013 On the additive and dominant variance and covariance of individuals within the genomic selection scope. Genetics 195: 1223–1230. https://doi.org/10.1534/genetics.113.155176

Wellmann R., 2019 Optimum contribution selection for animal breeding and conservation: The R package optiSel. BMC Bioinformatics 20: 25. https://doi.org/10.1186/s12859-018-2450-5

Werner C. R., R. C. Gaynor, D. J. Sargent, A. Lillo, G. Gorjanc, *et al.*, 2023 Genomic selection strategies for clonally propagated crops. Theoretical and Applied Genetics 136: 74. https://doi.org/10.1007/s00122-023-04300-6

Woolliams J. A., P. Berg, B. S. Dagnachew, and T. H. E. Meuwissen, 2015 Genetic contributions and their optimization. Journal of Animal Breeding and Genetics 132: 89–99.

Wray N. R., and R. Thompson, 1990 Prediction of rates of inbreeding in selected populations. Genet Res 55: 41–54. https://doi.org/10.1017/S0016672300025180

Wray N. R., and M. E. Goddard, 1994 Increasing long-term response to selection. Genetics Selection Evolution 26: 431–451.

Wright S., 1921 Systems of Mating. I. The biometric relations between parent and offspring. Genetics 6: 111–123.

Acknowledgments

This research was supported by USDA NIFA Award 2020-51181-32156.

Conflict of Interest

The author is a member of the editorial board for GENETICS.

Data Availability

The COMA and SimPlus packages are distributed under the GPL-3 license, and the versions at the time of publication are available at (insert figshare URL). For the most recent versions, download from https://github.com/jendelman.

File S1. Supplemental Methods (PDF)

File S2. Supplemental Figures (PDF)

File S3. R/SimPlus tutorial and manual (PDF)

File S4. R/COMA tutorial and manual (PDF)

File S5. R code to obtain potato marker effects (PDF)

File S6. R code to generate figures (PDF)

File S7. Simulation results (CSV)

FIGURES

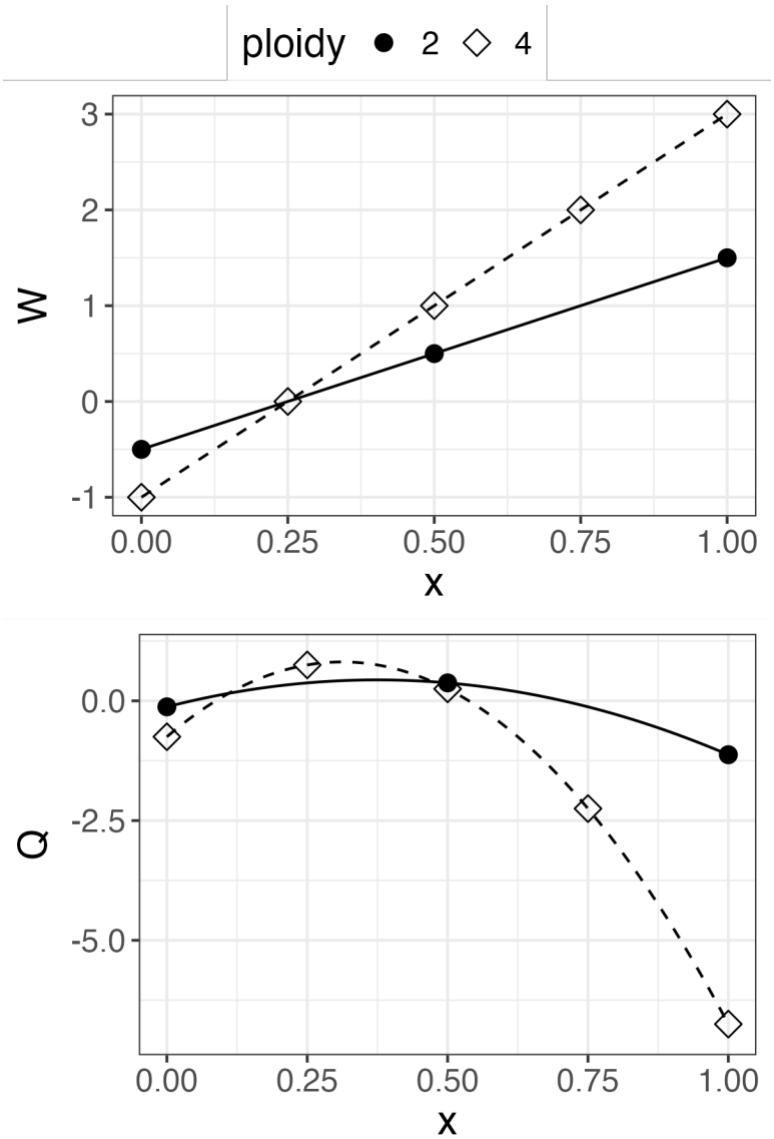


Figure 1. Illustrating the functional relationship between the additive (W) and dominance (Q) regressors and *x*, an individual allele frequency. The population frequency is ¼.

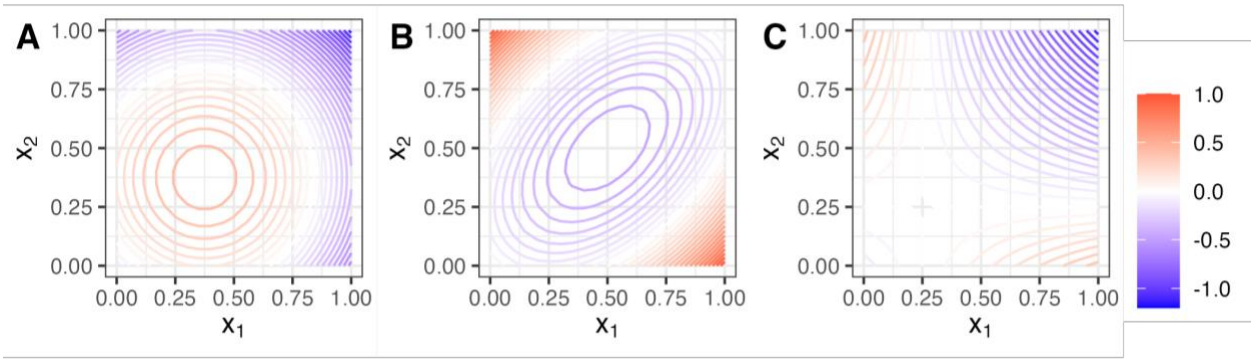


Figure 2. Level curves for mid-parent dominance value (A), mid-parent heterosis (B), and their sum (C), in diploids. The population frequency is ¼.

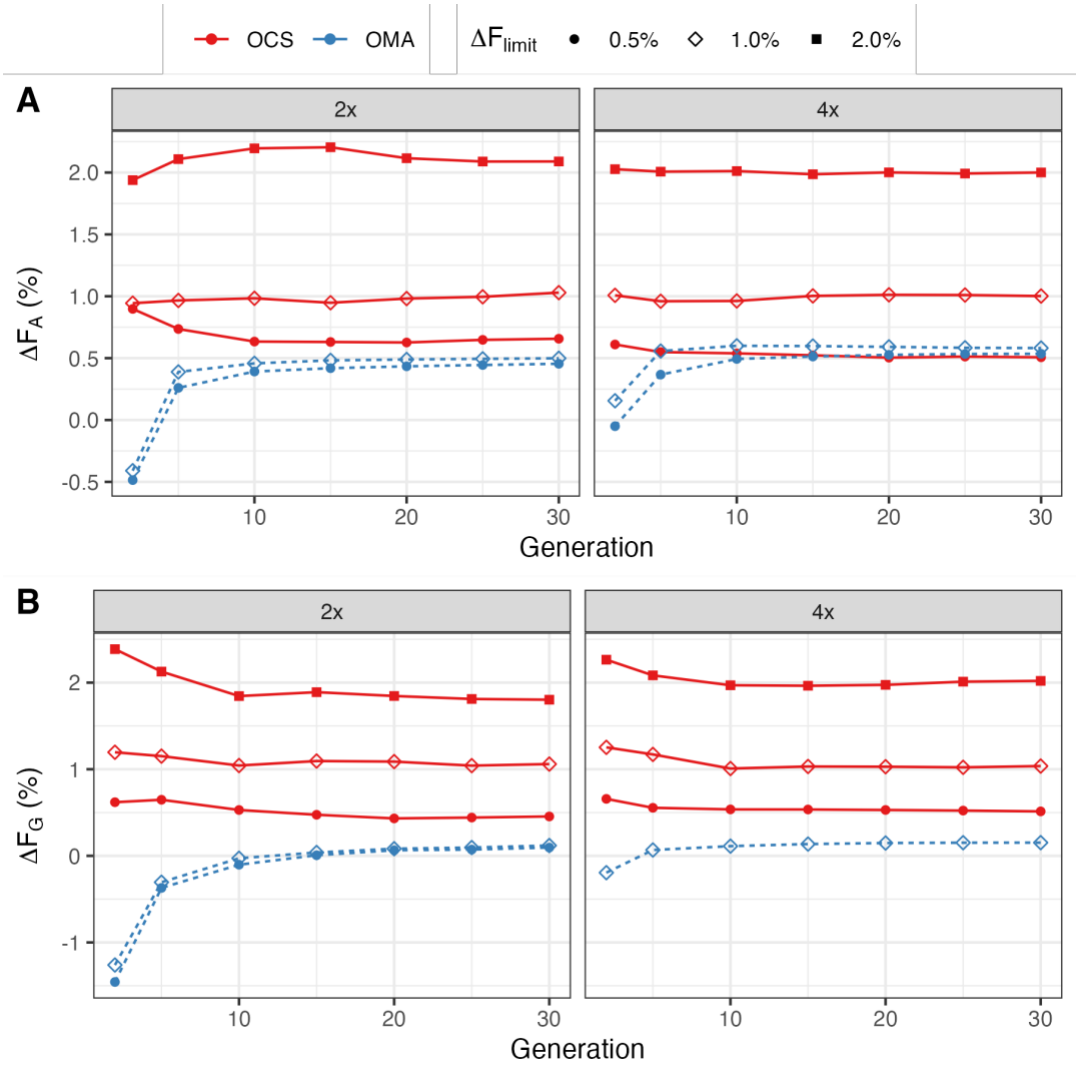


Figure 3. Realized inbreeding rates in simulated diploid (2x) and tetraploid (4x) populations at different limits (0.5%, 1.0%, 2.0%), using either OCS or OMA for selection and pedigree kinship (panel A) or genomic kinship (panel B) for inbreeding control. The dashed lines are for OMA.

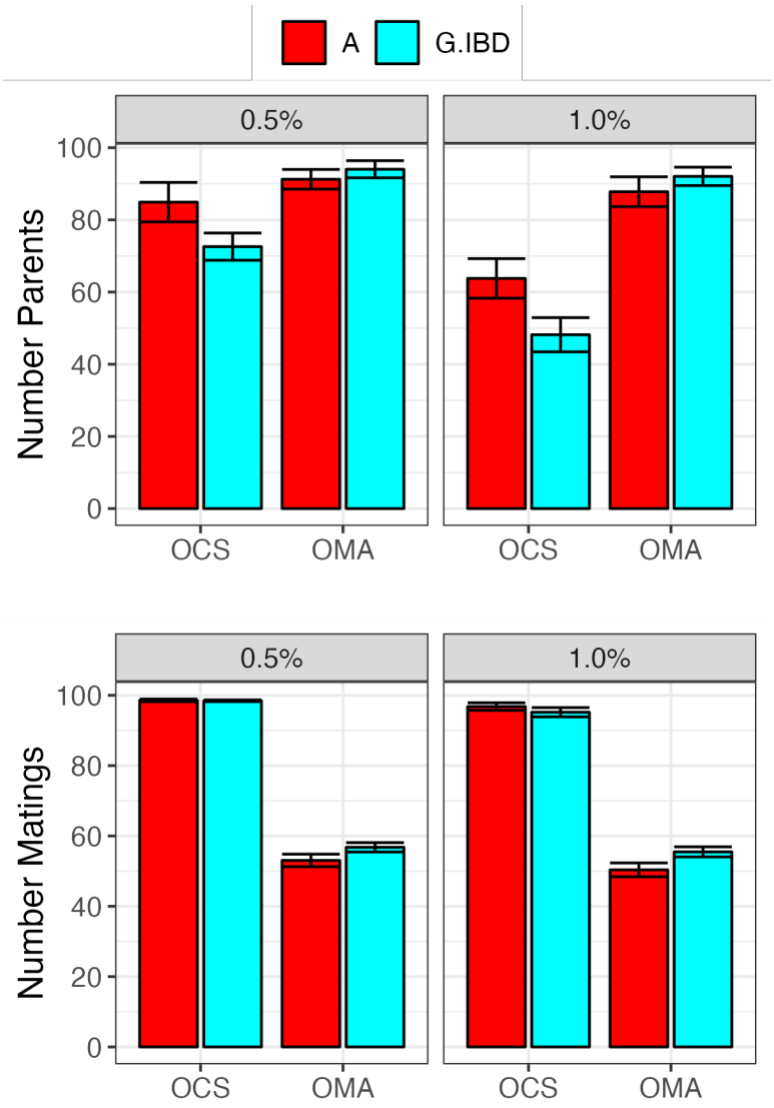


Figure 4. Average number of parents and matings for the 2x simulation generations 10–30, using OCS or OMA and either A or G_{IBD}, at 0.5% or 1.0% inbreeding. The error bars are the 95% CI based on 10 independent simulations.

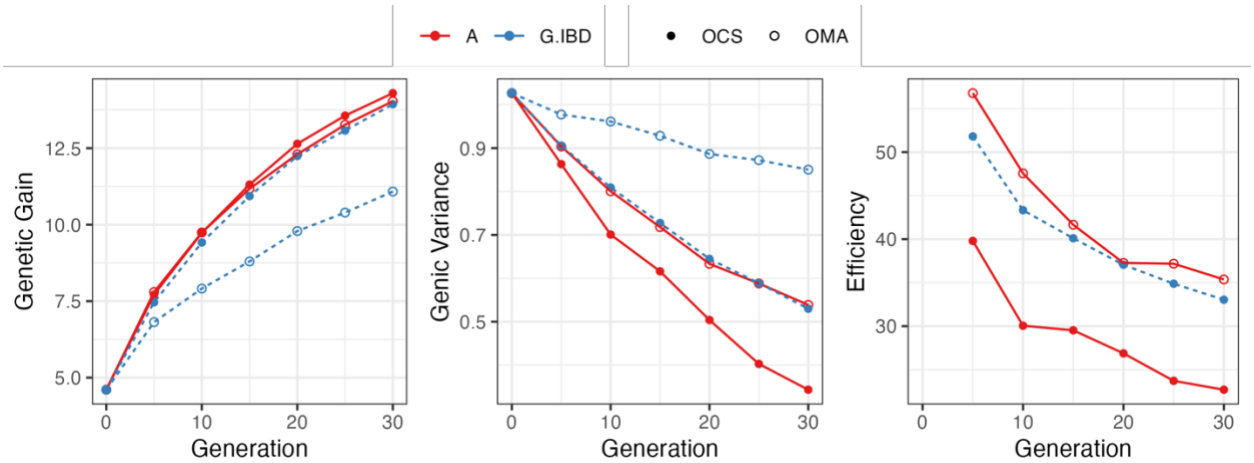


Figure 5. Genetic gain, genic variance, and efficiency vs. time for the 2x simulation with a 1% inbreeding limit. Efficiency is not shown before 5 generations or for OMA-G_{IBD} because the values are much larger and obscure the trend at later times, where it is more relevant. The dashed lines are for OMA.

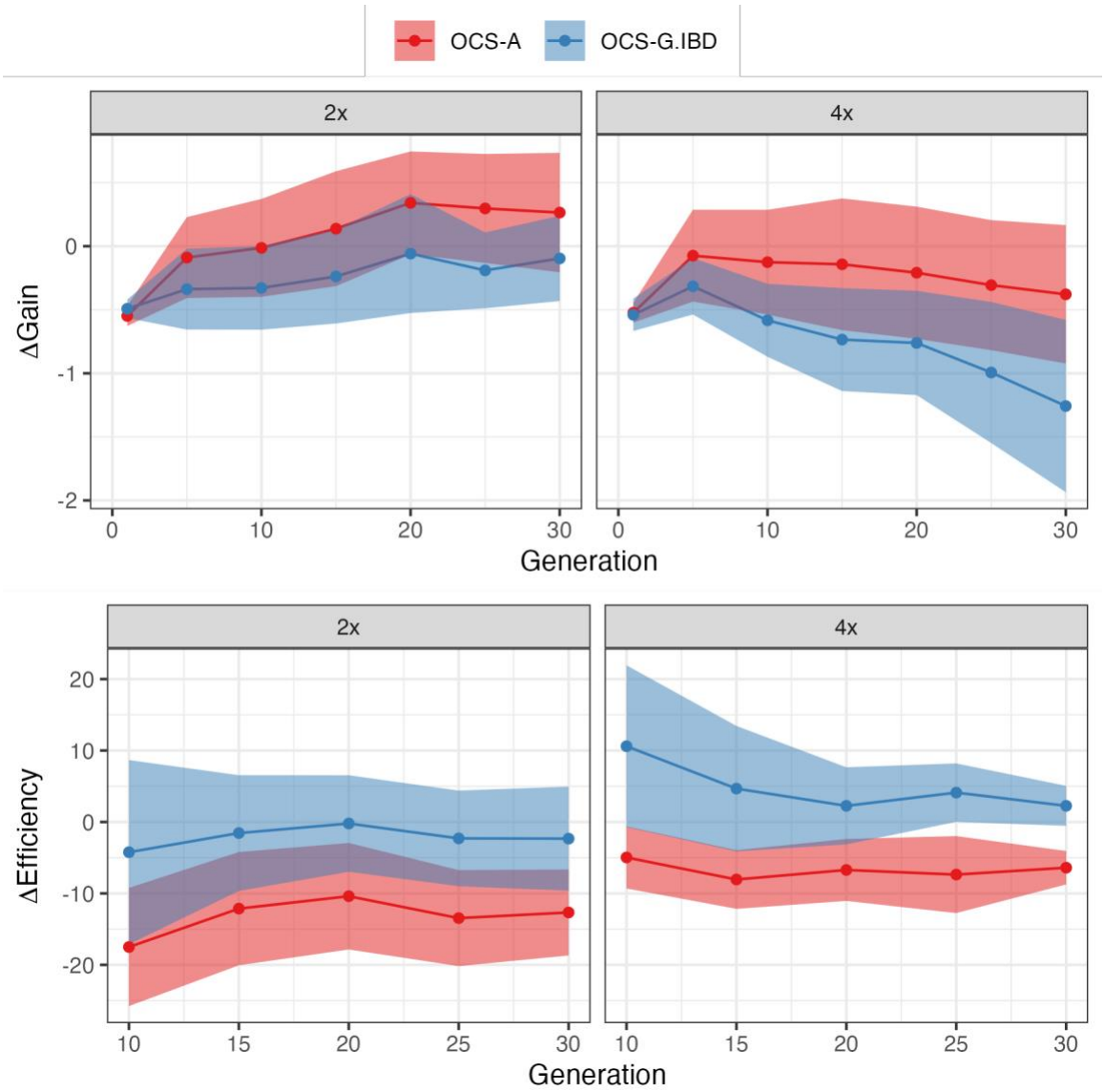


Figure 6. Comparing gain and efficiency for OCS-A and OCS-G_{IBD} against OMA-A as the reference method, at 1% inbreeding. The solid line is the mean difference across the 10 replicate simulations, and the ribbon is the 95% CI from a paired *t*-test at each generation.

File S1: Supplemental Methods

Jeff Endelman

2024-06-28

Table 1

Let random variable z denote the gametic allele dosage from a parent with zygotic allele dosage X and ploidy ϕ. Under a random bivalents model of meiosis, which is mathematically equivalent to random chromosome segregation, gametic dosage follows the hypergeometric distribution, which describes the probability of choosing z copies of the allele in $\phi/2$ draws without replacement from ϕ chromosomes that in total contain X copies of the allele. The probability mass function and first two moments are (Ibe 2014)

$$Prob(z) = \frac{\begin{pmatrix} X \\ z \end{pmatrix} \begin{pmatrix} \phi - X \\ \frac{\phi}{2} - z \end{pmatrix}}{\begin{pmatrix} \phi \\ \phi/2 \end{pmatrix}}$$

$$E[z] = X/2$$

$$E[z^2] = var[z] + E[z]^2 = \frac{X(\phi - X)}{4(\phi - 1)} + \frac{X^2}{4} = \frac{\phi X + (\phi - 2)X^2}{4(\phi - 1)}$$

The dosage X of the F1 zygote from parents 1 and 2 is the sum of z_1 and z_2. The first and second moments of this random variable are

$$E[X] = E[z_1] + E[z_2] = (X_1 + X_2)/2$$

$$E[X^2] = E[z_1^2] + E[z_2^2] + 2E[z_1]E[z_2] = \frac{\phi(X_1 + X_2) + (\phi - 2)(X_1^2 + X_2^2) + 2(\phi - 1)X_1X_2}{4(\phi - 1)}$$

From Eq. 1, the expected additive value of the F1 progeny is the mid-parent value:

$$E[u] = (E[X] - \phi p)\alpha = (X_1 + X_2 - 2\phi p)\alpha/2 = (u_1 + u_2)/2$$

The expected dominance value of the F1 progeny is the dominance effect β times $E[Q]$:

$$E[Q] = \tau - E[(X - \phi\xi)^2] = \left(\tau - \phi^2\xi^2\right) - E[X^2] + E[X][2p(\phi - 1) + 1]$$

The expression $\tau - \phi^2\xi^2$ evaluates to $-p^2\phi(\phi - 1)$, which is the parameter F in Table 1. Since the allele frequency $x = X/\phi$, the remaining terms are

$$A(x_1^2 + x_2^2) + Bx_1x_2 + D(x_1 + x_2)$$

where the coefficients A, B, D are given in Table 1.

Eccentricity of MPH level curves

Because the lengths of the ellipse semi-axes are the inverse square-roots of the eigenvalues λ of the matrix of the quadratic form (Eq. 4), the eccentricity $e = \sqrt{\lambda_1/\lambda_2}$. The eigenvalues are the roots of the characteristic equation:

$$0 = |\lambda \mathbf{I} - \mathbf{M}| = (\lambda - A)^2 - B^2/4$$

$$\lambda = A \pm B/2 = \frac{\phi^3 \pm \phi^2(\phi - 1)}{4(\phi - 1)}$$

$$e = \sqrt{2\phi - 1}$$

Equation 9

The OMA objective when SCA is neglected is

$$\sum_i \sum_j x_{ij} \left(GCA_i + GCA_j\right) = \sum_i GCA_i \sum_j x_{ij} + \sum_j GCA_j \sum_i x_{ij}$$

Summing over one index of the mate allocation variable generates sex-specific contributions (Eq. 6), so the above expression simplifies to

$$\sum_i GCA_i y_{f,i} + \sum_j GCA_j y_{m,j} = 2 \sum_i GCA_i y_i$$

The final expression follows because the index variables are arbitrary, and the average sex-specific contributions is the total contribution (Eq. 6). Eq. 9 follows because breeding value is two times GCA.

Equation 19

Eq. 18 is the inbreeding coefficient $F_{ij,ab}$ of an individual in generation $t+2$ that is the offspring of individuals ij and ab in generation t, with grandparents i, j, a, b in generation t. The frequency of ij is determined by the mate allocation in generation t: x_{ij}. Under random mating in generation $t+1$, the frequency of ij, ab offspring is the product of the parental frequencies, $x_{ij}x_{ab}$, and therefore the average inbreeding coefficient is

$$(\phi - 1)^2 \overline{F}_{t+2} = (\phi - 1)^2 \sum_{i,j,a,b} F_{ij,ab} x_{ij} x_{ab}$$

Each summation index goes from 1 to n, the number of individuals in generation t. Substituting Eq. 18 into above,

$$= (\phi/8)(\phi - 1) \sum_{i,j,a,b} \left(K_{ia} + K_{ib} + K_{ja} + K_{jb}\right) x_{ij} x_{ab} \qquad \text{[S1]}$$

$$+(\phi/4)(\frac{\phi}{2} - 1) \sum_{i,j,a,b} \left(K_{ij} + K_{ab}\right) x_{ij} x_{ab}$$

$$+(1/4)(\frac{\phi}{2} - 1)^2 \sum_{i,j,a,b} \left(F_i + F_j + F_a + F_b\right) x_{ij} x_{ab}$$

Using the following intermediate results based on Eq. 6,

$$\sum_{i,j,a,b} K_{ia} x_{ij} x_{ab} = 4 \sum_{i,a} K_{ia} y_{f,i} y_{f,a}$$

$$\sum_{i,j,a,b} K_{ij} x_{ij} x_{ab} = \sum_{i,j} K_{ij} x_{ij}$$

$$\sum_{i,j,a,b} F_i x_{ij} = 2 \sum_j F_i y_{f,i}$$

the first line of Eq. S1 becomes

$$(\phi/2)(\phi-1) \sum_{i,j} K_{ij} \left(y_{f,i} y_{f,j} + y_{f,i} y_{m,j} + y_{m,i} y_{f,j} + y_{m,i} y_{m,j} \right) = (\phi/2)(\phi-1) \sum_{i,j} K_{ij} y_i y_j = (\phi/2)(\phi-1) \mathbf{y}' \mathbf{K} \mathbf{y}$$

the second line of Eq. S1 becomes

$$(\phi/4)(\frac{\phi}{2} - 1) \sum_{i,j} 2 K_{ij} x_{ij} = (\phi/2)(\frac{\phi}{2} - 1) \mathbf{x}' \mathrm{vec}(\mathbf{K})$$

and the third line of Eq. S1 becomes

$$(1/4)(\frac{\phi}{2} - 1)^2 \sum_i 4 F_i (y_{f,i} + y_{m,i}) = (\frac{\phi}{2} - 1)^2 \mathbf{y}' \mathbf{F}$$

The sum of these three pieces equals Eq. 19.

Equation 22

The diagonal elements of G.VR1 (VanRaden Method 1) depend on the individual allele dosages X_{ik}, which have population mean ϕp_k.

$$G_{ii} = \frac{\sum_k (X_{ik} - \phi p_k)^2}{\sum_k \phi p_k q_k}$$

The population mean of G_{ii} and inbreeding coefficient $\overline{f}_A$ are therefore

$$\overline{G}_{..} = n^{-1} \sum_i G_{ii} = \frac{\sum_k n^{-1} \sum_i (X_{ik} - \phi p_k)^2}{\sum_k \phi p_k q_k} = \frac{\sum_k \mathrm{var}(X_k)}{\sum_k \phi p_k q_k}$$

$$\overline{f}_A = \frac{\overline{G}_{..} - 1}{\phi - 1} = (\phi - 1)^{-1} \left[\frac{\sum_k \mathrm{var}(X_k)}{\sum_k \phi p_k q_k} - 1 \right]$$

The average inbreeding coefficient from the directional dominance model, $\overline{f}_D$, depends on the average value of the dominance regressor (Eq. 1):

$$\overline{Q}_{\cdot k} = n^{-1} \sum_i Q_{ik} = n^{-1} \sum_i \left[\tau - (X_{ik}^2 - \phi 2 X_{ik} \xi - \phi^2 \xi^2) \right]$$

$$= - \left[\overline{X_{\cdot k}^2} - (\phi p_k)^2 \right] + \phi p_k q_k = -\mathrm{var}(X_k) + \phi p_k q_k$$

Substituting the above result into the following expression demonstrates equality with $\overline{f}_A$.

$$\overline{f}_D = \frac{\sum_k \overline{Q}_{\cdot k}}{\phi(\phi - 1) \sum_k p_k q_k}$$

Equation 23

For diploids, the inbreeding coefficient $\overline{f}$ is the proportional reduction in heterozygosity relative to HWE. For a bi-allelic locus where p represents the population frequency of the allele counted by dosage X, the genotype frequencies are

$$P(X=2) = p^2 + pq\overline{f}$$
$$P(X=1) = 2pq(1-\overline{f})$$
$$P(X=0) = q^2 + pq\overline{f}$$

The first and second moments of $x = X/2$ are

$$E[x^k] = 2^{-k}E[X^k] = 2^{-k}\sum_{i=0}^{2} P(X=i)X^k$$

$$E[x] = 2^{-1}\left[(q^2+pq\overline{f})(0) + 2pq(1-\overline{f})1 + (p^2+pq\overline{f})2\right] = p[q(1-\overline{f}) + p + q\overline{f})] = p$$
$$E[x^2] = 2^{-2}\left[2pq(1-\overline{f})1 + (p^2+pq\overline{f})4\right] = \frac{p}{2}[q(1-\overline{f}) + 2p + 2q\overline{f}] = \frac{p}{2}[1+p+q\overline{f}]$$

Using Table 1 and the above results, the conditional mean of mid-parent heterosis is

$$d^{-1}E[\text{MPH}|x] = 2x^2 + p(1+p+q\overline{f}) - 2px - p - x = 2x^2 - (1+2p)x + p^2 + pq\overline{f} \quad \text{[S2]}$$

And the conditional mean of MPD is -1 times MPH. The expectation of Eq. S2 is

$$d^{-1}E[\text{MPH}] = p[1+p+q\overline{f}] - (1+2p)p + p^2 + pq\overline{f} = 2pq\overline{f}$$

References

Ibe OC. 2014. Fundamentals of Applied Probability and Random Processes. 2nd ed. Academic Press.

File S2: Supplemental Figures

Endelman JB. Genomic prediction of heterosis, inbreeding control, and mate allocation in outbred diploid and tetraploid populations.

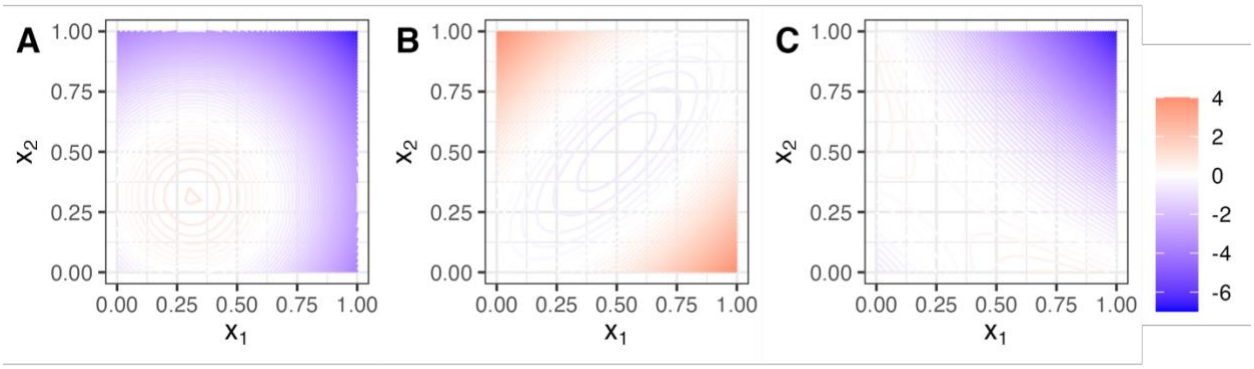


Figure S1. Level curves for mid-parent dominance value (A), mid-parent heterosis (B), and their sum (C), in tetraploids. The population frequency is ¼.

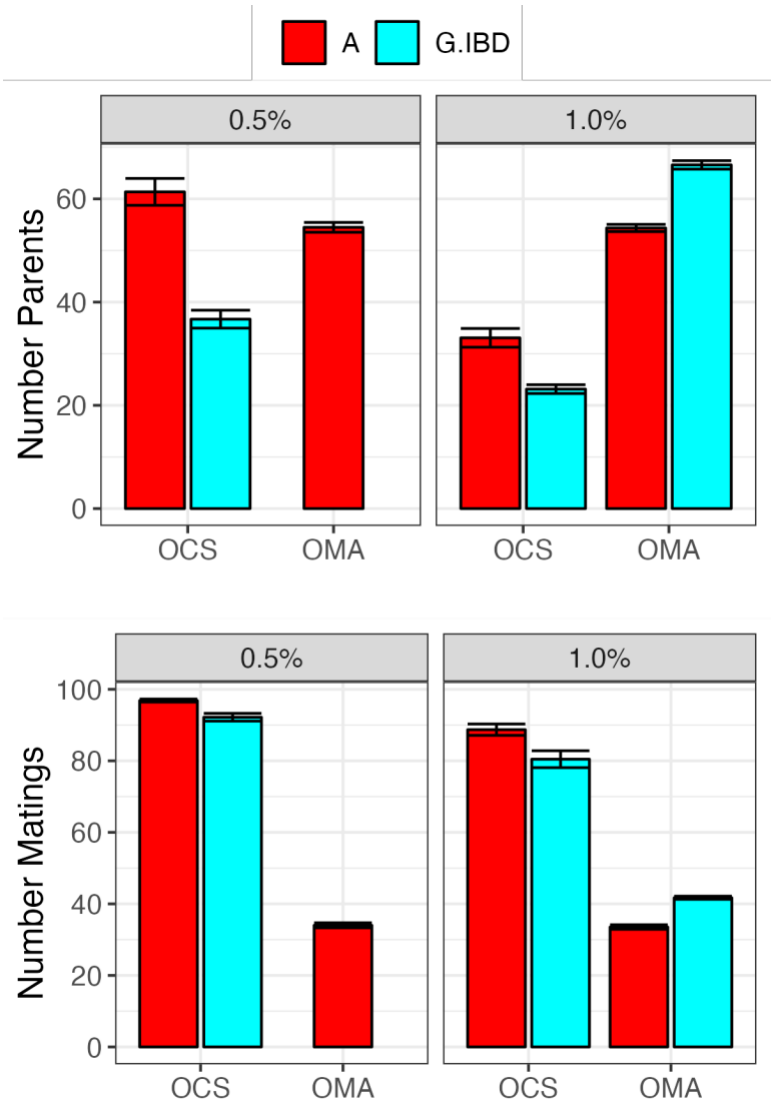


Figure S2. Average number of parents and matings for the 4x simulation generations 10–30, using OCS or OMA and either A or G_{IBD}, at 0.5% or 1.0% inbreeding. OMA-G_{IBD} at 0.5% was not simulated. The error bars are the 95% CI based on 10 independent simulations.

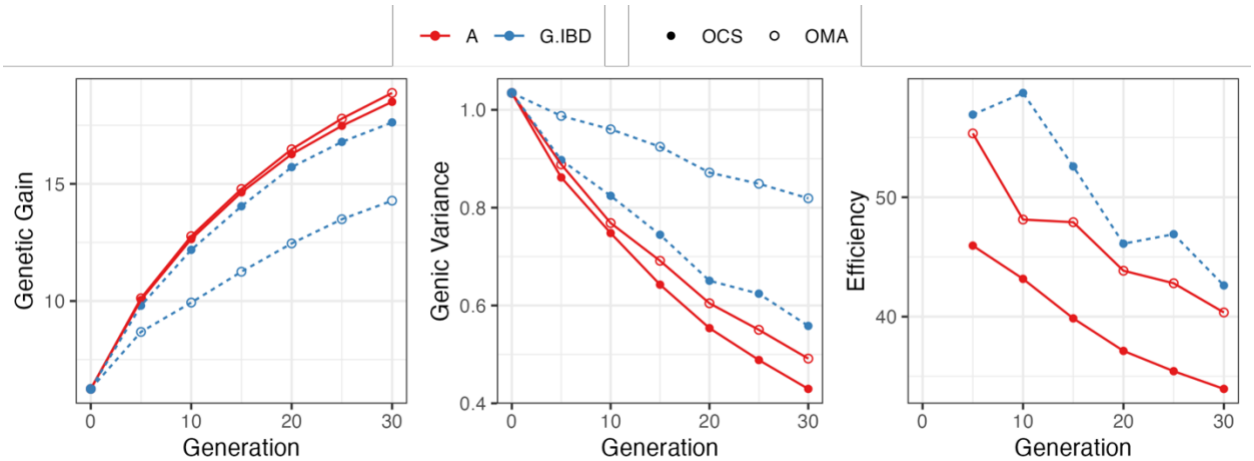


Figure S3. Genetic gain, genic variance, and efficiency vs. time for the 4x simulation with a 1% inbreeding limit. Efficiency is not shown before 5 generations or for OMA-G_{IBD} because the values are much larger and obscure the trend at later times, where it is more relevant. The dashed lines are for OMA.

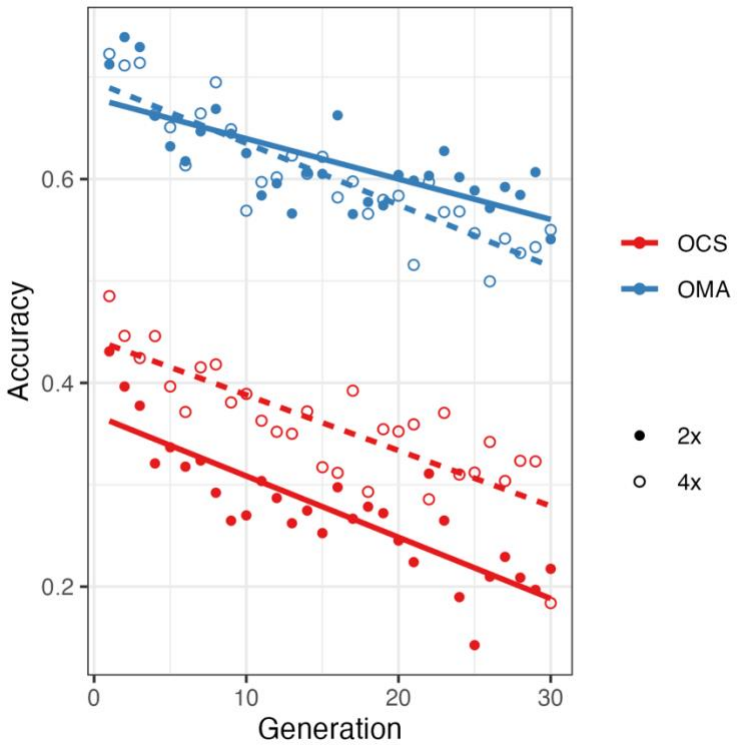


Figure S4. Accuracy of OCS vs. OMA at predicting F1 progeny means, based on 100 random matings and 50 progeny per mating.

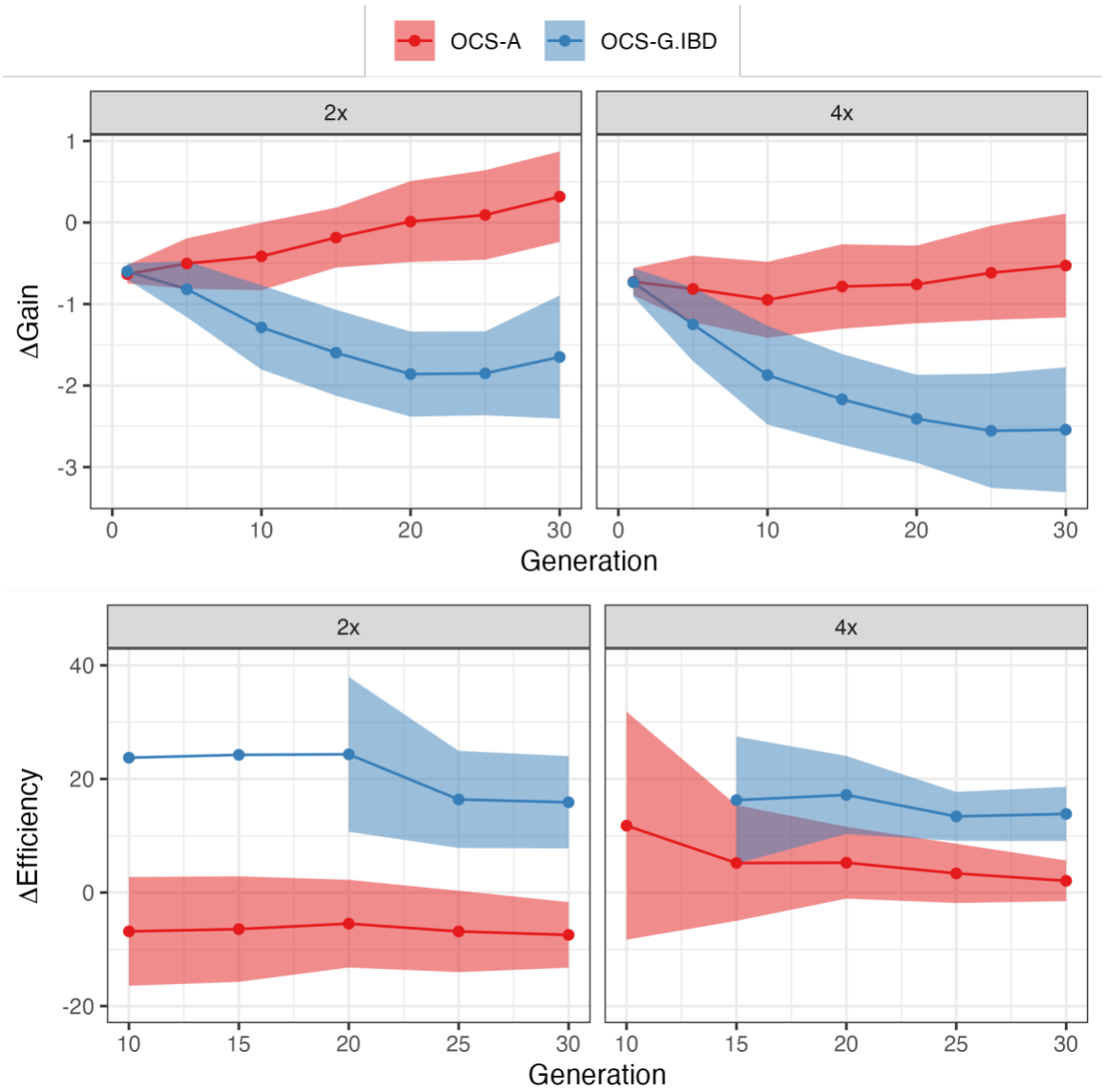


Figure S5. Comparing gain and efficiency for OCS-A and OCS-G$_{IBD}$ against OMA-A as the reference method, at 0.5% inbreeding. The solid line is the mean difference across the 10 replicate simulations, and the ribbon is the 95% CI from a paired *t*-test at each generation. At times where the CI is missing, this is because it exceeded the vertical axis limits.

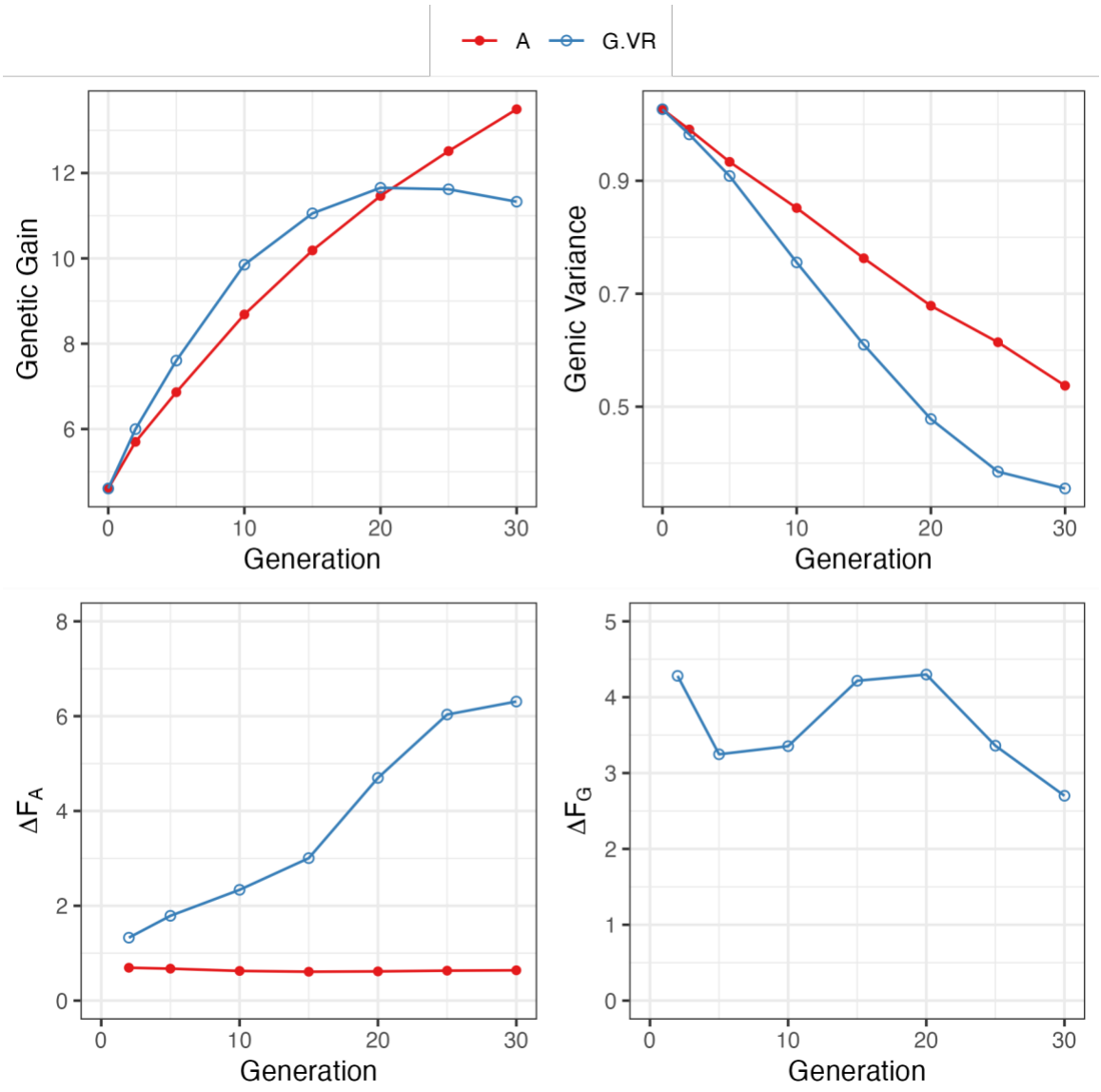


Figure S6. Performance of A vs. G_{VR} for inbreeding control with OCS at 0.5% inbreeding.

File S3: SimPlus

Jeff Endelman

2024-06-28

SimPlus was created for testing COMA with AlphaSimR. COMA provides functions for optimum contribution selection (OCS) and optimum mate allocation (OMA). For OCS, the objective is to maximize the average GEBV of the parents, weighted by their contributions. For OMA, the objective is to maximize the average GPMP (genomic prediction of mate performance), weighted by the mate allocations.

The following code creates a founder population in mutation-drift equilibrium and sets the variance parameters for a trait with additive and dominance effects.

```
library(AlphaSimR)
founders <- runMacs2(nInd=500,nChr=10,segSites=NULL,Ne=100,
                     mutRate=1e-8, ploidy=2,
                     histGen=NULL,histNe=NULL,genLen=1)
SP <- SimParam$new(founders)
SP$setTrackRec(TRUE)
SP$addTraitADG(nQtlPerChr = 100, mean = 0, var = 1, varDD = 0.5,
               meanDD = 0.5, useVarA = FALSE)
SP$setVarE(varE=2)
pop <- newPop(founders, simParam=SP)
```

OCS-A-Pheno

The following code runs 20 generations of mass selection at 1% inbreeding, using phenotypes for genetic merit. The population is saved after 5 generations to compare the response with genomic selection.

```
library(SimPlus)
gp <- genParam(pop, simParam = SP)
total.gen <- 20
results <- data.frame(gen=1:total.gen, mu=numeric(total.gen),
                      genicVar=numeric(total.gen),
                      F.coeff=numeric(total.gen),
                      n.parent=numeric(total.gen),
                      n.mate=numeric(total.gen),
                      accuracy=numeric(total.gen))
results$mu[1] <- as.numeric(gp$mu)
results$genicVar[1] <- as.numeric(gp$genicVarG)

p.ref <- update_TP(pop, SP, geno.file="geno.csv.gz",
                   pheno.file="pheno.csv", ped.file="ped.csv",
                   gen.TP=0)
update_K(pop, SP, K.method="A", K.file="kinship.csv", ped.file="ped.csv")

for (gen in 1:total.gen) {
  print(sub("X",gen,"Generation X"))
  ans1 <- sim_OCS(pop, SP, n.progeny=500, dF=0.01, solver="MOSEK",
```

```
                  COMA.file="pheno.csv", K.file="kinship.csv")

  #optional, measure accuracy
  acc <- sim_accuracy(pop, SP, COMA.file="pheno.csv")
  results$accuracy[gen] <- acc$ocs

  ans2 <- sim_mate(pop, SP, matings=ans1$om, total.progeny=500, min.progeny=5)
  results$n.parent[gen] <- ans2$n.parent
  results$n.mate[gen] <- nrow(ans2$matings)

  pop <- setPheno(ans2$progeny,simParam = SP) #AlphaSimR function
  gp <- genParam(pop, simParam = SP)
  results$mu[gen] <- as.numeric(gp$mu)
  results$genicVar[gen] <- as.numeric(gp$genicVarG)

  update_TP(pop, SP, geno.file="geno.csv.gz", pheno.file="pheno.csv",
            ped.file="ped.csv", gen.TP=3)
  results$F.coeff[gen] <- update_K(pop, SP, K.method="A", K.file="kinship.csv",
                                   ped.file="ped.csv")
  if (gen==5)
    sim_save("gen5.rda",pop,SP,"geno.csv.gz","pheno.csv","ped.csv",p.ref=p.ref)
}

all.results <- data.frame(results,method="OCS-A-Pheno")
```

Even though there is no genomic selection in the above simulation, the function `update_TP` is still required to update the phenotype and pedigree files for OCS. The argument `gen.TP` controls the number of generations in the TP (including the candidates, which are selected after phenotyping). It is set at 3 to anticipate the simulation below, ensuring the full TP is available at the onset of GS in generation 5.

OCS-A-GEBV

To simulate GS requires another function, `sim_StageWise`, to predict the marker effects using the StageWise package. The file with the predicted marker effects (COMA.file) is now used as input for `sim_OCS` instead of the phenotype file.

```
gen5 <- sim_load("gen5.rda","geno.csv.gz","pheno.csv","ped.csv")
pop <- gen5$pop
SP <- gen5$SP

results$F.coeff[5] <- update_K(pop, SP, K.method="A",
                           K.file="kinship.csv", ped.file="ped.csv")

for (gen in 6:total.gen) {
  print(sub("X",gen,"Generation X"))
  ans0 <- sim_StageWise("geno.csv.gz","pheno.csv",ploidy=2,
                        COMA.file="COMA.csv", gen.TP=3)

  ans1 <- sim_OCS(pop, SP, n.progeny=500, dF=0.01, solver="MOSEK",
                  COMA.file="COMA.csv", K.file="kinship.csv")

  acc <- sim_accuracy(pop, SP, COMA.file="COMA.csv", K.file="kinship.csv")
  results$accuracy[gen] <- acc$ocs
```

```
  ans2 <- sim_mate(pop, SP, matings=ans1$om, total.progeny=500, min.progeny=5)
  results$n.parent[gen] <- ans2$n.parent
  results$n.mate[gen] <- nrow(ans2$matings)

  pop <- setPheno(ans2$progeny,simParam = SP)
  gp <- genParam(pop, simParam = SP)
  results$mu[gen] <- as.numeric(gp$mu)
  results$genicVar[gen] <- as.numeric(gp$genicVarG)

  update_TP(pop, SP, geno.file="geno.csv.gz", pheno.file="pheno.csv",
            ped.file="ped.csv", gen.TP=3)
  results$F.coeff[gen] <- update_K(pop, SP, K.method="A", K.file="kinship.csv",
                                   ped.file="ped.csv")
}

results.all <- rbind(results, data.frame(results[5:20,], method="OCS-A-GEBV"))
```

OCS-G.IBD-GEBV

Instead of pedigree kinship, genomic IBD (G.IBD) kinship can be used to control inbreeding. To reduce computational time, not all segregating sites are needed for this calculation. The parameter "ibd.loci" in the function `update_K` controls how many loci per chromosome to use. The default is 100, and since the chromosomes are 100 cM, this implies 1 marker per cM.

```
gen5 <- sim_load("gen5.rda","geno.csv.gz","pheno.csv","ped.csv")
pop <- gen5$pop
SP <- gen5$SP

results$F.coeff[5] <- update_K(pop, SP, K.method="G.IBD",
                               K.file="kinship.csv",
                               ped.file="ped.csv", ibd.loci=100, n.core=2)

for (gen in 6:total.gen) {
  print(sub("X",gen,"Generation X"))
  ans0 <- sim_StageWise("geno.csv.gz","pheno.csv",ploidy=2,
                        COMA.file="COMA.csv", gen.TP=3)

  ans1 <- sim_OCS(pop, SP, n.progeny=500, dF=0.01, solver="MOSEK",
                  COMA.file="COMA.csv", K.file="kinship.csv")

  acc <- sim_accuracy(pop, SP, COMA.file="COMA.csv", K.file="kinship.csv")
  results$accuracy[gen] <- acc$ocs

  ans2 <- sim_mate(pop, SP, matings=ans1$om, total.progeny=500, min.progeny=5)
  results$n.parent[gen] <- ans2$n.parent
  results$n.mate[gen] <- nrow(ans2$matings)

  pop <- setPheno(ans2$progeny,simParam = SP)
  gp <- genParam(pop, simParam = SP)
  results$mu[gen] <- as.numeric(gp$mu)
  results$genicVar[gen] <- as.numeric(gp$genicVarG)

  update_TP(pop, SP, geno.file="geno.csv.gz", pheno.file="pheno.csv",
            ped.file="ped.csv", gen.TP=3)
```

```
  results$F.coeff[gen] <- update_K(pop, SP, K.method="G.IBD",
                                   K.file="kinship.csv", ped.file="ped.csv",
                                   ibd.loci=100, n.core=2)
}

results.all <- rbind(results, data.frame(results[5:20,], method="OCS-G.IBD-GEBV"))
```

OCS-G.VR-GEBV

A genomic kinship matrix based on VanRaden (2008) Method 1 can also be tested for inbreeding control, using the founder allele frequencies (p.ref) as the base population.

```
gen5 <- sim_load("gen5.rda","geno.csv.gz","pheno.csv","ped.csv")
pop <- gen5$pop
SP <- gen5$SP
p.ref <- gen5$p.ref

results$F.coeff[5] <- update_K(pop, SP, K.method="G.VR", K.file="kinship.csv",
                               ped.file="ped.csv", p.ref=p.ref)

for (gen in 6:total.gen) {
  print(sub("X",gen,"Generation X"))
  ans0 <- sim_StageWise("geno.csv.gz","pheno.csv",ploidy=2,
                        COMA.file="COMA.csv", gen.TP=3)

  ans1 <- sim_OCS(pop, SP, n.progeny=500, dF=0.01, solver="MOSEK",
                  COMA.file="COMA.csv", K.file="kinship.csv")

  acc <- sim_accuracy(pop, SP, COMA.file="COMA.csv", K.file="kinship.csv")
  results$accuracy[gen] <- acc$ocs

  ans2 <- sim_mate(pop, SP, matings=ans1$om, total.progeny=500, min.progeny=5)
  results$n.parent[gen] <- ans2$n.parent
  results$n.mate[gen] <- nrow(ans2$matings)

  pop <- setPheno(ans2$progeny,simParam = SP)
  gp <- genParam(pop, simParam = SP)
  results$mu[gen] <- as.numeric(gp$mu)
  results$genicVar[gen] <- as.numeric(gp$genicVarG)

  update_TP(pop, SP, geno.file="geno.csv.gz", pheno.file="pheno.csv",
            ped.file="ped.csv", gen.TP=3)
  results$F.coeff[gen] <- update_K(pop, SP, K.method="G.VR",
                                   K.file="kinship.csv", ped.file="ped.csv",
                                   p.ref=p.ref)
}

results.all <- rbind(results, data.frame(results[5:20,], method="OCS-G.VR-GEBV"))
```

OMA-A-GPMP

Simulating OMA follows the same code as above but replaces `sim_OCS` with `sim_OMA`, which has an additional argument `max.parent`. To limit the size of the computational problem for OMA, OCS is first used to reduce the number of candidates to `max.parent`.

```r
gen5 <- sim_load("gen5.rda","geno.csv.gz","pheno.csv","ped.csv")
pop <- gen5$pop
SP <- gen5$SP

results$F.coeff[5] <- update_K(pop, SP, K.method="A",
                               K.file="kinship.csv", ped.file="ped.csv")

for (gen in 6:total.gen) {
  print(sub("X",gen,"Generation X"))
  ans0 <- sim_StageWise("geno.csv.gz","pheno.csv",ploidy=2,
                        COMA.file="COMA.csv", gen.TP=3)

  ans1 <- sim_OMA(pop, SP, n.progeny=500, dF=0.01, solver="MOSEK",
                  COMA.file="COMA.csv", K.file="kinship.csv",
                  max.parent=200)

  acc <- sim_accuracy(pop, SP, COMA.file="COMA.csv", K.file="kinship.csv")
  results$accuracy[gen] <- acc$oma

  ans2 <- sim_mate(pop, SP, matings=ans1$om, total.progeny=500, min.progeny=5)
  results$n.parent[gen] <- ans2$n.parent
  results$n.mate[gen] <- nrow(ans2$matings)

  pop <- setPheno(ans2$progeny,simParam = SP)
  gp <- genParam(pop, simParam = SP)
  results$mu[gen] <- as.numeric(gp$mu)
  results$genicVar[gen] <- as.numeric(gp$genicVarG)

  update_TP(pop, SP, geno.file="geno.csv.gz", pheno.file="pheno.csv",
            ped.file="ped.csv", gen.TP=3)
  results$F.coeff[gen] <- update_K(pop, SP, K.method="A",
                                   K.file="kinship.csv", ped.file="ped.csv")
}

results.all <- rbind(results, data.frame(results[5:20,], method="OMA-A-GPMP"))
```

The average inbreeding rate at time t follows

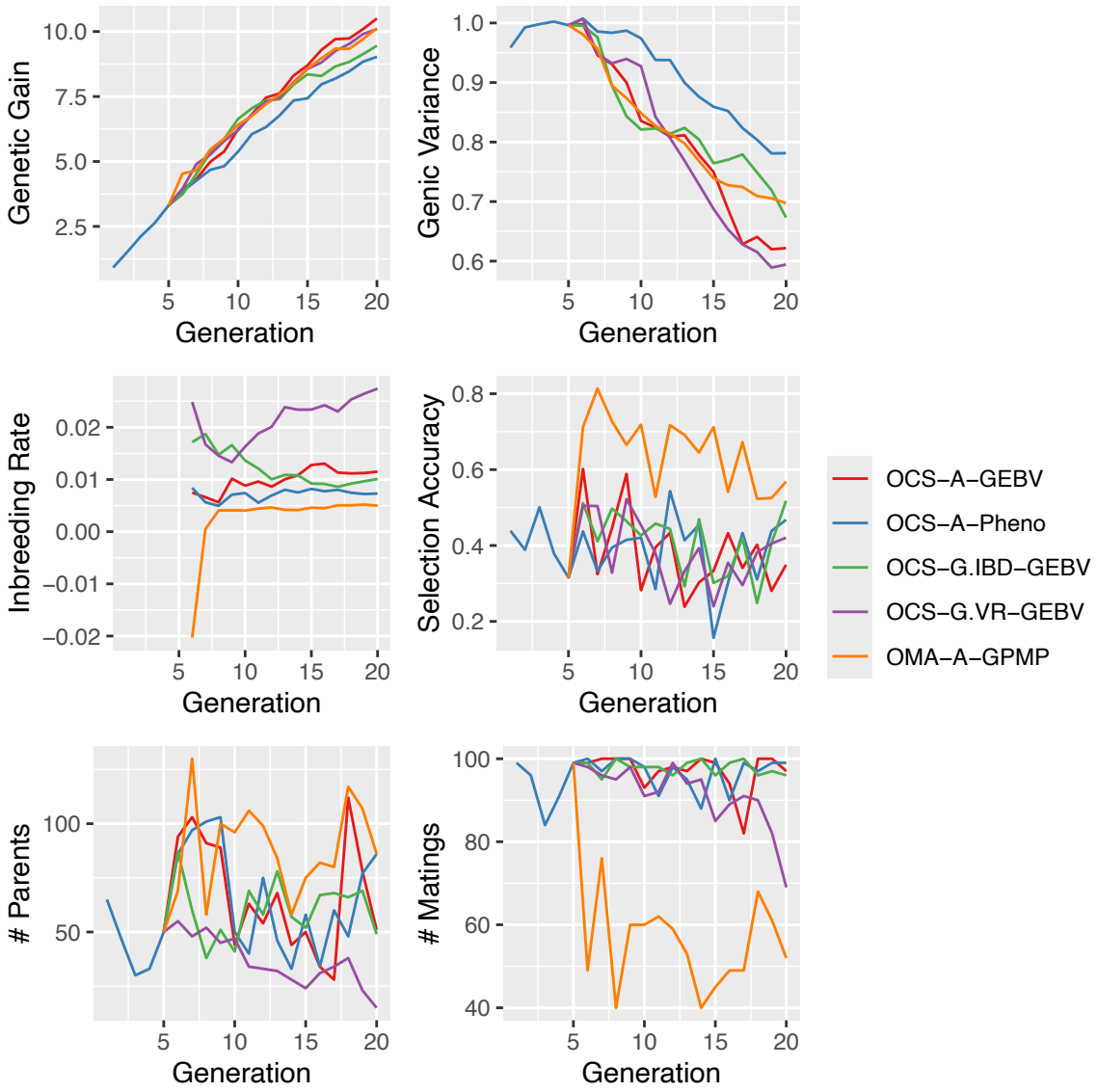
$$\Delta F_t = 1 - \left(\frac{1 - F_t}{1 - F_0}\right)^{1/t}$$

```r
library(dplyr)
results.all$dF <- NA*numeric(nrow(results.all))

fdF <- function(gen,Ft,F0) {1-((1-Ft)/(1-F0))^(1/gen)}

results0 <- filter(results.all,gen==5)
methods <- unique(results0$method)
m <- length(methods)

for (i in 1:m) {
  ix <- which(results.all$method==methods[i] & results.all$gen > 5)
  results.all$dF[ix] <- fdF(gen=1:15,
```

```
                              Ft=results.all$F.coeff[ix],
                              F0=results0$F.coeff[results0$method==methods[i]])
}
```

We can now plot the results.

```
library(ggplot2)
library(ggpubr)

p1 <- ggplot(results.all,aes(x=gen,y=mu,colour=method)) + geom_line() +
  ylab("Genetic Gain") + xlab("Generation") +
  scale_colour_brewer(name="",palette="Set1")
p2 <- ggplot(results.all,aes(x=gen,y=genicVar,colour=method)) + geom_line() +
  ylab("Genic Variance") + xlab("Generation") +
  scale_colour_brewer(name="",palette="Set1")
p3 <- ggplot(results.all,aes(x=gen,y=dF,colour=method)) + geom_line() +
  ylab("Inbreeding Rate") + xlab("Generation") +
  scale_colour_brewer(name="",palette="Set1")
p4 <- ggplot(results.all,aes(x=gen,y=accuracy,colour=method)) + geom_line() +
  ylab("Selection Accuracy") + xlab("Generation") +
  scale_colour_brewer(name="",palette="Set1")
p5 <- ggplot(results.all,aes(x=gen,y=n.parent,colour=method)) + geom_line() +
  ylab("# Parents") + xlab("Generation") +
  scale_colour_brewer(name="",palette="Set1")
p6 <- ggplot(results.all,aes(x=gen,y=n.mate,colour=method)) + geom_line() +
  ylab("# Matings") + xlab("Generation") +
  scale_colour_brewer(name="",palette="Set1")

ggarrange(p1,p2,p3,p4,p5,p6,nrow=3,ncol=2,common.legend=T,legend = "right")
```



This short simulation illustrates some key points, which are discussed in greater depth in the publication. Compared to OCS, OMA has higher accuracy for predicting F1 progeny means, sparser mating designs, and is generally more efficient at converting genic variance into genetic gain. Using G.VR for kinship does not properly control the inbreeding rate.

Package ‘SimPlus’

June 28, 2024

Title Functions to enhance AlphaSimR

Version 0.04

Author Jeffrey B. Endelman

Maintainer Jeffrey Endelman <endelman@wisc.edu>

Description Functions to enhance AlphaSimR

Depends R (>= 4.1)

License GPL-3

LazyData true

RoxygenNote 7.2.3

Encoding UTF-8

Imports utils, AlphaSimR, dplyr, polyBreedR, pedigree, data.table, COMA

Suggests knitr, rmarkdown, StageWise

VignetteBuilder knitr

R topics documented:

sim_accuracy 2
sim_load 2
sim_mate 3
sim_OCS 3
sim_OMA 4
sim_save 5
sim_StageWise 5
update_K 6
update_TP 7

Index **8**

sim_accuracy *Estimate selection accuracy*

Description

Estimate selection accuracy

Usage

```
sim_accuracy(pop, SP, COMA.file, K.file = NULL, n.mate = 100, n.progeny = 50)
```

Arguments

pop	parental candidates, AlphaSimR Pop-class
SP	simulation parameters for AlphaSimR
COMA.file	COMA filename
K.file	kinship filename for COMA
n.mate	number of matings to sample
n.progeny	number of progeny per mating

Value

list of accuracies for OCS and OMA

sim_load *Load simulation files*

Description

Load simulation files

Usage

```
sim_load(rda.file, geno.file, pheno.file, ped.file)
```

Arguments

rda.file	R data filename
geno.file	genotype CSV file
pheno.file	phenotype CSV filename
ped.file	pedigree CSV filename

Value

list containing pop,SP,p.ref

sim_mate — *Perform mating in AlphaSimR*

Description

Perform mating in AlphaSimR

Usage

```
sim_mate(pop, SP, matings, total.progeny, min.progeny)
```

Arguments

pop	parental candidates, AlphaSimR Pop-class
SP	simulation parameters for AlphaSimR
matings	data frame of mate allocations, with columns parent1, parent2, value
total.progeny	total progeny to generate
min.progeny	minimum progeny per mating

Details

Details

Value

list containing

progeny Pop-class of progeny
matings matings with additional column for mu

sim_OCS — *Parent selection and mating by OCS in AlphaSimR*

Description

Parent selection and mating by OCS in AlphaSimR

Usage

```
sim_OCS(pop, SP, n.progeny, dF, COMA.file, K.file, solver = "ECOS")
```

Arguments

pop	parental candidates, AlphaSimR Pop-class
SP	simulation parameters for AlphaSimR
n.progeny	number of progeny to simulate
dF	target inbreeding rate
COMA.file	marker effect or pheno filename for COMA
K.file	kinship filename for COMA
solver	CVXR solver

Details

Auto detects whether COMA.file has marker effects or phenotype data based on the first column name.

Value

list containing

response named vector with realized dF, merit, Shannon diversity

oc data frame of optimal contributions for each individual

om data frame of optimal allocations for each mating

---

sim_OMA *Parent selection and mating by OMA in AlphaSimR*

Description

Parent selection and mating by OMA in AlphaSimR

Usage

```
sim_OMA(pop, SP, n.progeny, dF, COMA.file, K.file, max.parent, solver = "ECOS")
```

Arguments

`pop`	parental candidates, AlphaSimR Pop-class
`SP`	simulation parameters for AlphaSimR
`n.progeny`	number of progeny to simulate
`dF`	target inbreeding rate
`COMA.file`	geno filename for COMA
`K.file`	kinship filename for COMA
`max.parent`	maximum number of candidate parents for oma
`solver`	CVXR solver

Value

list containing

response named vector with realized dF, merit, Shannon diversity

oc data frame of optimal contributions for each individual

om data frame of optimal allocations for each mating

sim_save — *Save simulation files*

Description

Save simulation files

Usage

```
sim_save(rda.file, pop, SP, geno.file, pheno.file, ped.file, p.ref)
```

Arguments

rda.file	R data filename
pop	variable of AlphaSimR Pop-class
SP	simulation parameters for AlphaSimR
geno.file	genotype CSV file
pheno.file	phenotype CSV filename
ped.file	pedigree CSV filename
p.ref	reference allele frequencies (for G.VR1)

sim_StageWise — *Estimate marker effects by StageWise*

Description

Estimate marker effects by StageWise

Usage

```
sim_StageWise(
  geno.file,
  pheno.file,
  ploidy,
  COMA.file,
  gen.TP,
  asreml.workspace = "500mb"
)
```

Arguments

geno.file	TP genotype CSV filename
pheno.file	TP phenotype CSV filename
ploidy	ploidy
COMA.file	input filename for COMA
gen.TP	number of generations for TP (including current)
asreml.workspace	memory workspace

Details

Details

Value

none

`update_K` *Update kinship file*

Description

Update kinship file

Usage

```
update_K(
  pop,
  SP,
  K.method,
  K.file,
  ped.file = NULL,
  p.ref = NULL,
  ibd.loci = 100,
  n.core = 1
)
```

Arguments

`pop`	variable of AlphaSimR Pop-class
`SP`	simulation parameters for AlphaSimR
`K.method`	kinship method: A, G.VR, G.IBD
`K.file`	kinship CSV filename
`ped.file`	pedigree CSV filename
`p.ref`	reference pop allele frequencies for "VR" method
`ibd.loci`	number of loci per chromosome for "G.IBD" method
`n.core`	number cores for "G.IBD" method

Details

Kinship is computed for selection candidates in `pop`. For `K.method="A"`, argument `ped.file` is the three-column pedigree file. For `K.method="G.VR"`, argument `p.ref` is the vector of reference allele frequencies. To reduce computing time for `K.method="G.IBD"`, only `ibd.loci` are sampled per chromosome.

Value

average inbreeding coefficient of the population

`update_TP` *Update the Training Population (TP)*

Description

Update genotype, genotyped, and pedigree files for the TP

Usage

```
update_TP(pop, SP, geno.file, pheno.file, ped.file, gen.TP)
```

Arguments

`pop`	variable of AlphaSimR Pop-class
`SP`	simulation parameters for AlphaSimR
`geno.file`	genotype CSV file
`pheno.file`	phenotype CSV filename
`ped.file`	pedigree CSV filename
`gen.TP`	# generations recorded for the TP (including current)

Details

Pedigree file has three columns: id, mother, father. Phenotype file has three columns: id, value, gen. To start a new TP, use `gen.TP` = 0.

Value

vector of allele frequencies for pop if gen.TP==0, otherwise NULL

File S4: COMA Tutorial

Jeff Endelman

COMA function `read_data` is used to compute GEBVs and genomic predictions of mate performance (GPMP). The input file contains the marker effects and allele dosage data for the parental candidates. Dominance marker effects are optional, but when included should correspond to a breeding value parameterization, including the effect of heterosis/inbreeding depression (Endelman 2023). The `blup` command in R/StageWise is one option to compute marker effects.

The vignette data are from the University of Wisconsin potato breeding program. There are 170 tetraploid clones genotyped at 12K markers, and the marker effects are derived from a multi-trait index.

```
geno.file <- system.file("vignette_data", "geno.csv", package = "COMA")
geno <- read.csv(geno.file,check.names=F)
geno[1:4,1:5]
```

```
##                marker    add   dom W17037-1 W17037-10
## 1 solcap_snp_c2_51460 -18.71  8.09     0.60      1.34
## 2 solcap_snp_c2_36608   8.48 19.14     2.00      1.00
## 3 solcap_snp_c2_36615   7.51  8.00     1.66      1.40
## 4 solcap_snp_c2_36658  46.53  4.71     4.00      2.00
```

A different input file contains the kinship matrix to control inbreeding. Based on simulation results, the pedigree (IBD) kinship matrix is recommended for either OCS or OMA.A pedigree kinship matrix is provided for the potato clones, estimated using R package AGHmatrix.

```
K.file <- system.file("vignette_data", "kinship.csv", package = "COMA")

#preview files
K <- as.matrix(read.csv(K.file,check.names=F,row.names=1))
K[1:3,1:3]
```

```
##           W17037-1 W17037-10 W17037-11
## W17037-1    0.2631    0.1397    0.1397
## W17037-10   0.1397    0.2631    0.1397
## W17037-11   0.1397    0.1397    0.2631
```

```
summary(K[upper.tri(K,diag=F)])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00600 0.04700 0.08600 0.08062 0.09690 0.18820
```

```
#inbreeding coefficients from A matrix
F.A <- (4*diag(K)-1)/(4-1)
summary(F.A)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00360 0.01747 0.01820 0.02354 0.02050 0.09293
```

```
library(COMA)
data <- read_data(geno.file=geno.file,
```

```
                  kinship.file=K.file,
                  ploidy=4, matings="all",
                  standardize=TRUE, n.core=2)

head(data$parents)
```

```
##          id       merit
## 1  W17037-1 -1.20577700
## 2 W17037-10  0.07519790
## 3 W17037-11 -0.11881184
## 4 W17037-12  0.26334446
## 5 W17037-13 -0.22747615
## 6 W17037-15  0.07752716
```

```
head(data$matings)
```

```
##     parent1  parent2      merit
## 1  W17037-1 W17037-1 -2.6354706
## 2 W17037-10 W17037-1 -1.0240878
## 3 W17037-11 W17037-1 -0.9792150
## 4 W17037-12 W17037-1 -0.6356246
## 5 W17037-13 W17037-1 -0.7980640
## 6 W17037-15 W17037-1 -0.8777523
```

The `matings` argument for `read_data` indicates how to create the table of possible matings. The above example used "all", which creates all unique parental combinations. Consult the function documentation for other options. When argument `standardize` is TRUE, the trait is centered and scaled so that the parental population has zero mean and unit variance. This is for convenience when interpreting the response values.

The `parents` output contains the GEBVs, and the `matings` output contains the GPMPs. For dioecious species (with separate sexes), a data frame with sex information is needed as argument `sex` in `read_data`, and the column headers for the matings are "female","male" instead of "parent1","parent2".

OCS and OMA

The `ocs` and `oma` functions are used to determine the optimal genetic contributions. Both require the argument `parents`, which must specify the minimum and maximum contributions. In this case we used min=0 and max=1 for all parents, but this can be tailored to each parent. For `oma`, the parental merits are not used, so this column is omitted. Rather, the merit of each mating is needed, along with its min and max. Based on simulation results and long-standing practice in animal breeding, inbreeding rates of 0.005 (0.5%) to 0.01 (1%) are recommended. However, this may not be feasible depending on the relatedness of the parents. As the results below show, the smallest feasible inbreeding rate was 1.5% for OCS and 1% for OMA.

```
parents <- data.frame(data$parents, min=0, max=1)
dF <- 0.005*(1:4)

ocs.ans <- list(4)
ocs.response <- NULL
for (i in 1:4) {
  ocs.ans[[i]] <- ocs(dF=dF[i], parents=parents, ploidy=4, K=data$K)
  ocs.response <- rbind(ocs.response,ocs.ans[[i]]$response)
}
kable(ocs.response,digits=3)
```

dF	merit	n.parent
NA	NA	0
NA	NA	0
0.015	0.689	29
0.020	1.380	25

```
matings <- data.frame(data$matings, min=0, max=1)
oma.ans <- list(4)
oma.response <- NULL
for (i in 1:4) {
  oma.ans[[i]] <- oma(dF=dF[i], parents=parents[,-2], matings=matings,
                      ploidy=4, K=data$K)
  oma.response <- rbind(oma.response,oma.ans[[i]]$response)
}
kable(oma.response,digits=3)
```

dF1	dF2	merit	n.parent	n.mate
NA	NA	NA	0	0
-0.005	0.010	1.332	29	25
0.000	0.015	2.051	24	20
0.004	0.020	2.434	18	15

Because genetic merit is a linear function of the contribution/allocation variables, the optimal solution lies on the boundary of the domain. This results in *sparse* solutions where many variables are zero. The number of parents and matings with nonzero contribution are returned by the function, along with the genetic merit of the optimal solution. Because mid-parent heterosis is modeled with OMA but not OCS, the predicted merit is more accurate with OMA.

Note the superiority of the OMA solution over OCS at the same value of dF: higher gains are achieved with simpler mating plans, which is particularly advantageous when matings require human intervention (i.e., when open-pollination is not feasible).

As the inbreeding rate increased, the complexity of the mating plan decreased (i.e., fewer parents and matings). The `ocs` function returns a single realized inbreeding rate, which corresponds to the next generation. The `oma` function returns two values of dF: dF1 is the rate for the progeny under the optimized allocation, while dF2 would be the rate if grand-progeny were produced under random mating of the progeny. The constraint on dF2 is analogous to the group kinship constraint of OCS and typically active (i.e., dF2 is at the limit), while dF1 is not at the limit.

The optimal parental contributions are returned as `oc`, and optimal mate allocations are returned as `om`. The command `plot_ribbon` can be used to visualize the solution over multiple inbreeding rates.

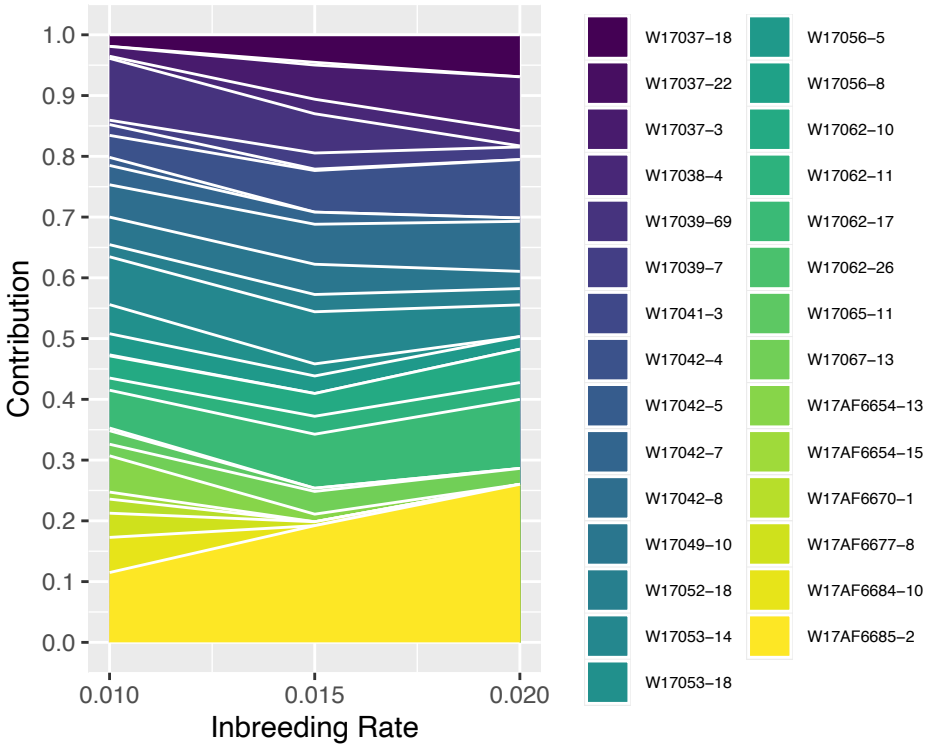
```
#Compare OCS and OMA solutions for 1.5%
summary(ocs.ans[[3]]$oc$value)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0004194 0.0137611 0.0292640 0.0344828 0.0476952 0.1007274
```

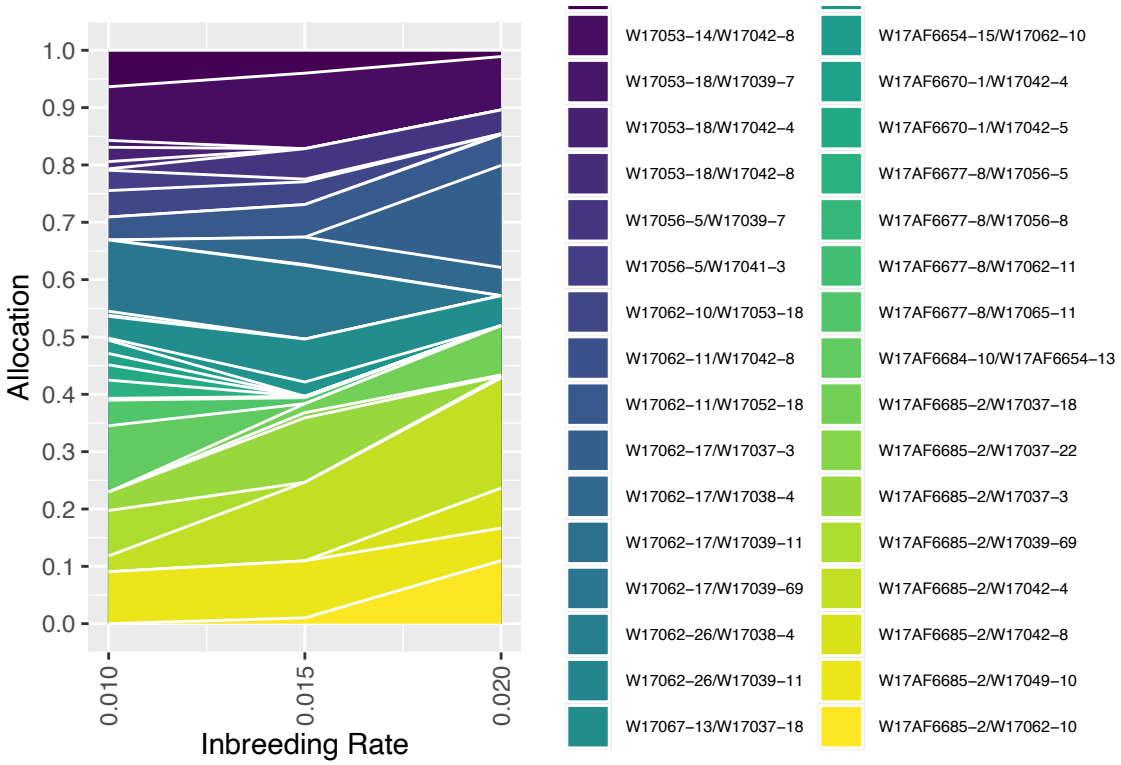
```
summary(oma.ans[[3]]$oc$value)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0006271 0.0180037 0.0291760 0.0416667 0.0582767 0.1920040
```

```r
library(ggplot2)
oc <- lapply(oma.ans[2:4],"[[",'oc')
names(oc) <- dF[2:4]
plot_ribbon(oc) + theme(legend.text=element_text(size=6))
```




```r
library(ggplot2)
f <- function(z){data.frame(id=apply(z$om[,1:2],1,paste,collapse="/"),
                            value=z$om$value)}
om <- lapply(oma.ans[2:4],f)
names(om) <- dF[2:4]
plot_ribbon(om) + ylab("Allocation") +
  theme(axis.text.x = element_text(angle=90,vjust=0.5)) +
  theme(legend.text=element_text(size=6))
```



Other Constraints

Because COMA is based on convex optimization, additional linear constraints can be incorporated. The potato dataset contains presence/absence information for a genetic marker for potato virus Y resistance (PVYR), which is a critical trait. Two different selection approaches will be illustrated: (1) only allowing matings with a resistant parent; (2) imposing a lower bound on the R gene frequency.

The following code implements method 1:

```
library(dplyr)
par.file <- system.file("vignette_data", "parents.csv",
                        package = "COMA")
data2 <- read.csv(par.file)
PVYR.id <- data2$id[data2$PVYR=="Y"]

#original solution at dF=0.01
ans0 <- oma.ans[[2]]

#calculate proportion of matings with PVYR parent from previous solution
n.mate <- ans0$response$n.mate
nR <- sum(ans0$om$parent1 %in% PVYR.id | ans0$om$parent2 %in% PVYR.id)

#constraint method 1
ans1 <- oma(dF=0.01, ploidy=4L, K=data$K, parents=parents[,-2],
            matings=filter(matings, parent1%in%PVYR.id | parent2%in%PVYR.id))

#calculate frequency of PVYR gene, assuming single copy in parents
R.parents <- which(ans0$oc$id %in% PVYR.id)
freq0 <- sum(ans0$oc$value[R.parents])*0.25
R.parents <- which(ans1$oc$id %in% PVYR.id)
```

```r
freq1 <- sum(ans1$oc$value[R.parents])*0.25

result <- data.frame(constraint=c("none","method 1"),
                     rbind(ans0$response,ans1$response),
                     prop.resistant.matings=c(nR/n.mate,1),
                     Ry.freq=c(freq0,freq1))
kable(result,digits=3)
```

constraint	dF1	dF2	merit	n.parent	n.mate	prop.resistant.matings	Ry.freq
none	-0.005	0.01	1.332	29	25	0.36	0.085
method 1	-0.005	0.01	1.159	27	24	1.00	0.125

The above table compares the solution with and without the PVYR constraint. As expected, increasing the proportion of PVYR matings to 100% reduced the predicted merit. The frequency of the R gene in the progeny is also shown, assuming a single copy in the resistant parents.

Under method 2, the R gene frequency is constrained directly, which is less restrictive than method 1 because some matings could have two PVYR parents while others have none. Both `ocs` and `oma` allow linear inequality or equality constraints on any linear combination of the parental contributions. When the coefficients, $\mathbf{c}$, of the contribution variables, $\mathbf{y}$, are the R gene frequencies of each parent, this creates the desired constraint. The coefficients are included as an additional column in the `parents` data.frame, and the column header is parsed to complete the equation. In this case, the header "gt0.125" represents greater than or equal to 0.125:

$$\mathbf{c}'\mathbf{y} \geq 0.125$$

```r
parents2 <- data.frame(parents,
                       gt0.125=ifelse(parents$id %in% PVYR.id, 0.25, 0))
ans2 <- oma(dF=0.01, ploidy=4L, K=data$K, parents=parents2[,-2],
            matings=matings)

R.parents <- which(ans2$oc$id %in% PVYR.id)
freq2 <- sum(ans2$oc$value[R.parents])*0.25

result <- data.frame(constraint=c("none","method 1","method 2"),
                     rbind(ans0$response,ans1$response,ans2$response),
                     Ry.freq=c(freq0,freq1,freq2))
kable(result,digits=3)
```

constraint	dF1	dF2	merit	n.parent	n.mate	Ry.freq
none	-0.005	0.01	1.332	29	25	0.085
method 1	-0.005	0.01	1.159	27	24	0.125
method 2	-0.006	0.01	1.273	30	25	0.125

As expected, the merit under method 2 was higher than method 1.

Package ‘COMA’

June 29, 2024

Title Convex Optimization of Mate Allocation

Version 0.17

Author Jeffrey B. Endelman

Maintainer Jeffrey Endelman <endelman@wisc.edu>

Description Convex Optimization of Mate Allocation

Depends R (>= 4.0)

License GPL-3

RoxygenNote 7.2.3

Encoding UTF-8

Imports ggplot2, CVXR, tidyr

Suggests knitr, rmarkdown, Rmosek

R topics documented:

ocs 1
oma 2
plot_ribbon 4
read_data 4

Index **6**

`ocs` *Optimal Contribution Selection*

Description

Optimize the contribution of each individual to the next generation

Usage

```
ocs(dF, parents, ploidy, K, tol = 1e-06, dF.adapt = NULL, solver = "ECOS")
```

Arguments

`dF`	inbreeding rate
`parents`	input data frame (see Details)
`ploidy`	ploidy
`K`	kinship matrix
`tol`	tolerance, values below this set to 0
`dF.adapt`	see Details
`solver`	solver for CVXR (default is "ECOS")

Details

The first four columns of `parents` should be named as follows: id, merit, min, max. Min and max are real numbers between 0 and 1 specifying the minimum and maximum contribution for each parent. An optional fifth column named "female" is a logical TRUE/FALSE variable for species with separate sexes.

Additional columns can be used to specify constraints on linear combinations of the contributions. The values are the coefficients of the contribution variables, and the name of each column specifies the right-hand side of the constraint. Each name must begin with "lt","gt", or "eq", followed by a non-negative numeric value. For example, "lt0.5" means less than or equal to 0.5.

The average inbreeding coefficient of the current generation is based on all individuals in `K`, which may exceed the list of individuals in `parents`.

It is possible that no feasible solution exists for the specified `dF`. Argument `dF.adapt` can be used to automatically progressively higher values.The software increases the dF limit by dF.adapt$step up to the smaller of dF.adapt$max or the realized value under the original dF, in an attempt to find a solution with less inbreeding.

Value

list containing

response data.frame with realized dF, merit, n.parent

oc data frame of optimal contributions

`oma` *Optimal Mate Allocation*

Description

Optimize the allocation for each mating

Usage

```
oma(
  dF,
  parents,
  matings,
  ploidy,
  K,
```

```
  tol = 1e-06,
  dF.adapt = NULL,
  solver = "ECOS"
)
```

Arguments

`dF`	inbreeding rate
`parents`	parents data frame (see Details)
`matings`	matings data frame (see Details)
`ploidy`	ploidy
`K`	kinship matrix
`tol`	tolerance, values below this set to 0
`dF.adapt`	see Details
`solver`	solver for CVXR (default is "ECOS")

Details

The first three columns of `parents` should be named "id", "min", "max", with an optional fourth column "female" to indicate sex in dioecious species. Additional columns can be used to specify constraints on linear combinations of the contributions. The values are the coefficients of the contribution variables, and the name of each column specifies the right-hand side of the constraint. Each name must begin with "lt","gt", or "eq", followed by a non-negative numeric value. For example, "lt0.5" means less than or equal to 0.5.

The data.frame `matings` has five columns: "female, male, merit, min, max" for dioecious species, or else "parent1, parent2, merit, min, max".

The average inbreeding coefficient of the current generation is based on all individuals in `K`, which may exceed the list of individuals in `parents`.

It is possible that no feasible solution exists for the specified `dF`. Argument `dF.adapt` can be used to automatically progressively higher values.The software increases the dF limit by dF.adapt$step up to the smaller of dF.adapt$max or the realized value under the original dF, in an attempt to find a solution with less inbreeding.

Value

list containing

response data.frame with realized dF, merit, n.parent, n.mate

oc data frame of optimal contributions for each individual

om data frame of optimal allocations for each mating

plot_ribbon *Ribbon Plot of Optimal Contributions*

Description

Ribbon Plot of Optimal Contributions

Usage

```
plot_ribbon(oc, min.c = 0.001)
```

Arguments

oc	named list of optimal contributions
min.c	minimum contribution for plotting

Details

The inbreeding rates are taken from the names of oc. Each element of oc should be a data frame with columns "id" and "value".

Value

ggplot2 object

read_data *Read data for OCS/OMA*

Description

Predicts merit for OCS/OMA

Usage

```
read_data(
  geno.file,
  kinship.file,
  ploidy,
  sex = NULL,
  matings = "none",
  standardize = FALSE,
  n.core = 1
)
```

Arguments

`geno.file`	file with marker effects and genotypes
`kinship.file`	file with kinship matrix
`ploidy`	even integer
`sex`	optional, data frame with columns id and female (T/F)
`matings`	see Details
`standardize`	T/F, standardize merit in parental candidates
`n.core`	multi-core evaluation

Details

The first column of `geno.file` is the marker name. The second column contains the additive effects for the breeding value parameterization, and (digenic) dominance effects (when available) should be in the third column with the header "dom". Subsequent columns contain the marker data for the population, coded as allele dosage, from 0 to ploidy. Missing marker data is imputed with the population mean.

The `kinship.file` should contain an N x N kinship matrix with id names in the first column and row.

There are several options for argument `matings`: (1) "none" = no matings; (2) "all" = all possible matings of the individuals in `geno.file` (excluding reciprocals); (3) a character vector of genotype ids to calculate all pairs of matings; (4) a 2-column data.frame of desired matings with header "female","male" for dioecious species (separate sexes) or "parent1","parent2" for hermaphrodites. Self-matings are included under options (2) and (3) for dioecious species but are easily removed in the output if desired.

Value

list containing

K kinship matrix

parents data frame of individual merits

matings data frame of mating merits

File S5: Potato Marker Effects

Jeff Endelman

2024-06-29

```
pheno.file <- system.file("vignette_data", "pheno1a.csv", package = "StageWise")

geno.file <- system.file("vignette_data", "geno1.csv", package = "StageWise")
```

```
library(StageWise)
ans1 <- Stage1(filename=pheno.file,
               traits=c("total.yield","vine.maturity","fry.color"),
               effects <- data.frame(name=c("block","stand.count"),
                      fixed=c(FALSE,TRUE),
                      factor=c(TRUE,FALSE)))
geno <- read_geno(geno.file,ploidy=4,map=TRUE,dominance = TRUE)
ans2 <- Stage2(data=ans1$blue, vcov=ans1$vcov, geno=geno,
               non.add="dom")
prep1 <- blup_prep(ans1$blues, vcov=ans1$vcov, geno=geno,
                   vars=ans2$vars)
```

```
merit.coeff <- c(total.yield=1, vine.maturity=0, fry.color=1)
gain1 <- gain(prep1, desired=merit.coeff, gamma=1)
kable(gain1$table)
```

trait	response	index
total.yield	0.34	0.702
vine.maturity	0.00	-0.459
fry.color	0.34	0.544

```
index.coeff <- gain1$table$index
names(index.coeff) <- gain1$table$trait

#dominance parameters
dominance(ans2$params, index.coeff=index.coeff, gamma=1)
```

```
##   ratio estimate
## 1 Vd/Va     0.33
## 2 b/SDa     6.21
```

```
#arbitrary scaling of marker effects
AM <- blup(prep1, geno, what="AM", index.coeff=index.coeff)
DM <- blup(prep1, geno, what="DM", index.coeff=index.coeff)
marker.effects <- data.frame(marker=AM$marker,
                             add=round(AM$effect*1e4,2),
                             dom=round(DM$effect*1e4,2))
```

```r
parents.file <- system.file("vignette_data", "parents.csv", package = "COMA")
parents <- read.csv(parents.file,check.names=F)

geno <- read.csv(geno.file,check.names=F)
rownames(geno) <- geno$marker
vignette.data <- cbind(marker.effects,geno[marker.effects$marker,
                                           parents$id])
#write.csv(vignette.data,
#          file="~/Software/COMA/inst/vignette_data/geno.csv",row.names=F)

library(polyBreedR)
#available at https://github.com/jendelman/polyBreedR
pedfile <- system.file("vignette_data", "ped.csv",
                       package = "StageWise")
ped <- get_pedigree(id=parents$id, pedfile=pedfile)
K <- A_mat(ped, ploidy=4)/4
#write.csv(round(K[parents$id,parents$id],4),
#          file="~/Software/COMA/inst/vignette_data/kinship.csv")
```

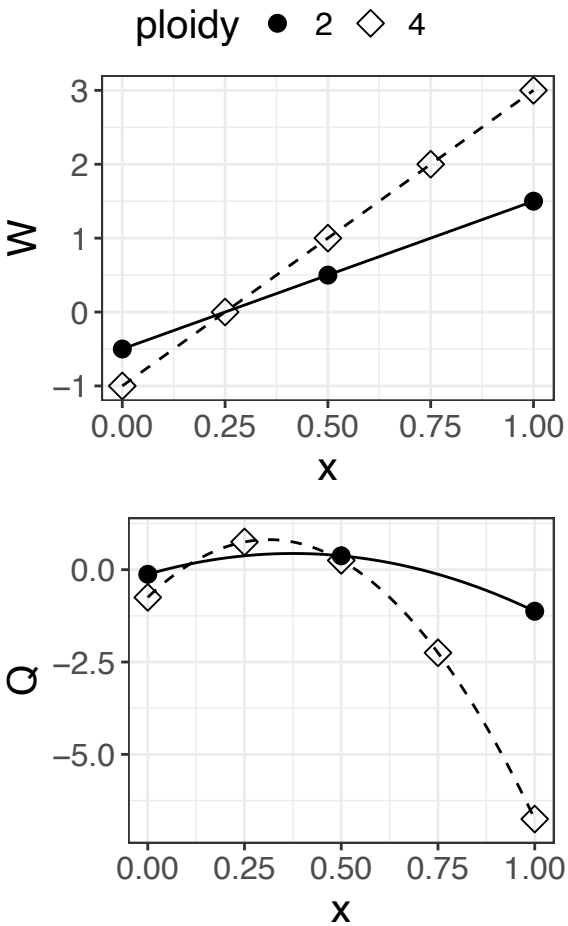
File S6: R Code for Figures

Jeff Endelman

2024-06-29

Figure 1

```
library(ggplot2)
library(ggpubr)

p <- 0.25
W <- function(x,p,ploidy) {ploidy*(x-p)}
Q <- function(x,p,ploidy) {
  tau <- (ploidy-1)*p*(1-p)+1/4
  xi <- (p*(ploidy-1)+1/2)/ploidy
  (tau - ploidy^2*(x-xi)^2)
}

x <- c(seq(0,1,by=0.5),seq(0,1,by=0.25))
y <- c(W(seq(0,1,by=0.5),p=p,ploidy=2),
       W(seq(0,1,by=0.25),p=p,ploidy=4))
pdat <- data.frame(x=x,y=y,ploidy=c(rep("2",3),rep("4",5)))

x2 <- seq(0,1,by=0.01)
F1A <- ggplot(data=pdat,mapping=aes(x=x,y=y)) +
  geom_point(mapping=aes(shape=ploidy),size=3) +
  scale_shape_manual(values=c(16,5)) +
  geom_line(data=data.frame(x=x2,y=W(x2,p,ploidy=2)),mapping=aes(x=x,y=y),linetype=1) +
  geom_line(data=data.frame(x=x2,y=W(x2,p,ploidy=4)),mapping=aes(x=x,y=y),linetype=2) +
  theme_bw() + scale_x_continuous(name="x",breaks=c(0:4)/4) + ylab("W") + xlab("x") +
  theme(text=element_text(size=15))

y <- c(Q(seq(0,1,by=0.5),p=p,ploidy=2),
       Q(seq(0,1,by=0.25),p=p,ploidy=4))
pdat <- data.frame(x=x,y=y,ploidy=c(rep("2",3),rep("4",5)))
F1B <- ggplot(data=pdat,mapping=aes(x=x,y=y)) +
  geom_point(mapping=aes(shape=ploidy),size=3) +
  scale_shape_manual(values=c(16,5)) +
  geom_line(data=data.frame(x=x2,y=Q(x2,p=p,ploidy=2)),mapping=aes(x=x,y=y),linetype=1) +
  geom_line(data=data.frame(x=x2,y=Q(x2,p=p,ploidy=4)),mapping=aes(x=x,y=y),linetype=2) +
  theme_bw() + scale_x_continuous(name="x",breaks=c(0:4)/4) +
  theme(text=element_text(size=15)) + scale_y_continuous(name="Q",limits=c(-7,1))

ggarrange(F1A,F1B,nrow=2,common.legend = T)
```

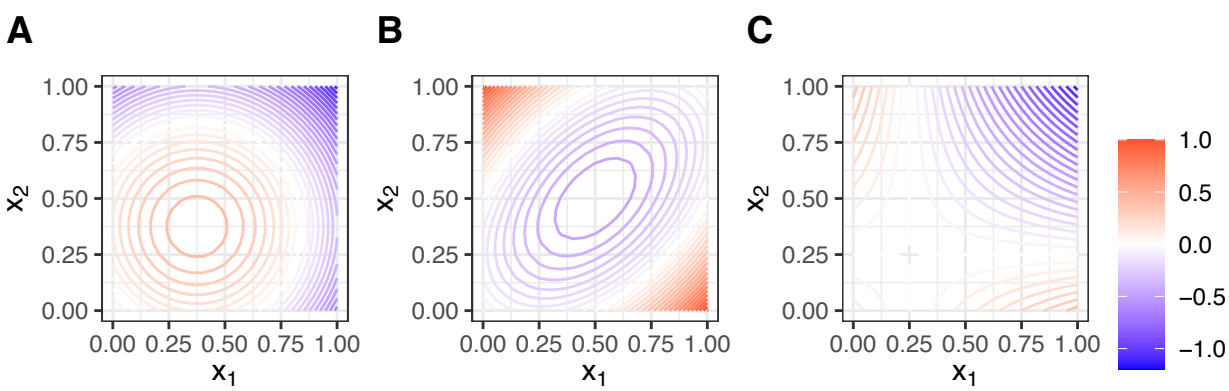



```
#ggsave("Figure1.png",device="png",width=4,height=6,units="in")
```

Figure 2

```
mph <- function(x,ploidy) {
  ploidy^3/(4*(ploidy-1))*sum(x^2) -
    ploidy^2/2*x[1]*x[2] -
    ploidy^2/(4*(ploidy-1))*sum(x)
}
mpv <- function(x,ploidy,p) {
-ploidy^2/2*sum(x^2) +
    ploidy*(p*(ploidy-1)+0.5)*sum(x) -
    ploidy*(ploidy-1)*p^2
}
x2 <- x1 <- seq(0,1,by=0.05)
x <- expand.grid(x1,x2,stringsAsFactors = F)

pdat <- data.frame(x, mpv=apply(x,1,mpv,ploidy=2L,p=1/4),
                   mph=apply(x,1,mph,ploidy=2L))
pdat$f1 <- pdat$mpv+pdat$mph

p1 <- ggplot(pdat,aes(x=Var1,y=Var2,z=mpv)) +
  geom_contour(aes(colour = after_stat(level)),binwidth=0.05) +
  coord_equal() + xlab(expression(x[1])) +
```

```
  ylab(expression(x[2])) + theme_bw() +
  scale_color_gradient2(low = "blue",limits=c(-1.2,1),name="",
                        mid = "white",high = "red",
                        midpoint = 0,guide = "colourbar")

p2 <- ggplot(pdat,aes(x=Var1,y=Var2,z=mph)) +
  geom_contour(aes(colour = after_stat(level)),binwidth=0.05) +
  coord_equal() + xlab(expression(x[1])) +
  ylab(expression(x[2])) + theme_bw() +
  scale_color_gradient2(low = "blue",limits=c(-1.2,1),name="",
                        mid = "white",high = "red",
                        midpoint = 0,guide = "colourbar")

p3 <- ggplot(pdat,aes(x=Var1,y=Var2,z=f1)) +
  geom_contour(aes(colour = after_stat(level)),binwidth=0.05) +
  coord_equal() + xlab(expression(x[1])) +
  ylab(expression(x[2])) + theme_bw() +
  scale_color_gradient2(low = "blue",limits=c(-1.2,1),name="",
                        mid = "white",high = "red",
                        midpoint = 0,guide = "colourbar")

ggarrange(p1,p2,p3,nrow=1,common.legend = T,labels=c("A","B","C"),
          legend = "right",vjust=9)
```

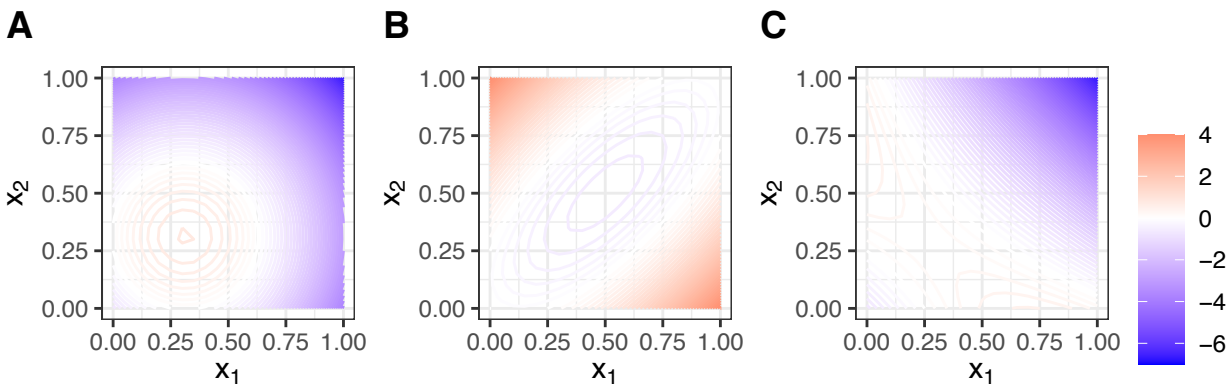



```
#ggsave("Figure2.png",device="png",width=7,height=4,units="in")
```

Figure S1

```
pdat <- data.frame(x, mpv=apply(x,1,mpv,ploidy=4L,p=1/4),
                   mph=apply(x,1,mph,ploidy=4L))
pdat$f1 <- pdat$mpv+pdat$mph

p1 <- ggplot(pdat,aes(x=Var1,y=Var2,z=mpv)) +
  geom_contour(aes(colour = after_stat(level)),binwidth=0.1) +
  coord_equal() + xlab(expression(x[1])) +
  ylab(expression(x[2])) + theme_bw() +
  scale_color_gradient2(low = "blue",limits=c(-7,4),name="",
                        mid = "white",high = "red",
                        midpoint = 0,guide = "colourbar")

p2 <- ggplot(pdat,aes(x=Var1,y=Var2,z=mph)) +
```

```
  geom_contour(aes(colour = after_stat(level)),binwidth=0.1) +
  coord_equal() + xlab(expression(x[1])) +
  ylab(expression(x[2])) + theme_bw() +
  scale_color_gradient2(low = "blue",limits=c(-7,4),name="",
                        mid = "white",high = "red",
                        midpoint = 0,guide = "colourbar")

p3 <- ggplot(pdat,aes(x=Var1,y=Var2,z=f1)) +
  geom_contour(aes(colour = after_stat(level)),binwidth=0.1) +
  coord_equal() + xlab(expression(x[1])) +
  ylab(expression(x[2])) + theme_bw() +
  scale_color_gradient2(low = "blue",limits=c(-7,4),name="",
                        mid = "white",high = "red",
                        midpoint = 0,guide = "colourbar")

ggarrange(p1,p2,p3,nrow=1,common.legend = T,labels=c("A","B","C"),
          legend = "right",vjust=9)
```




```
#ggsave("FigureS1.png",device="png",width=7,height=4,units="in")
```

Prepare simulation data for plotting

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
```

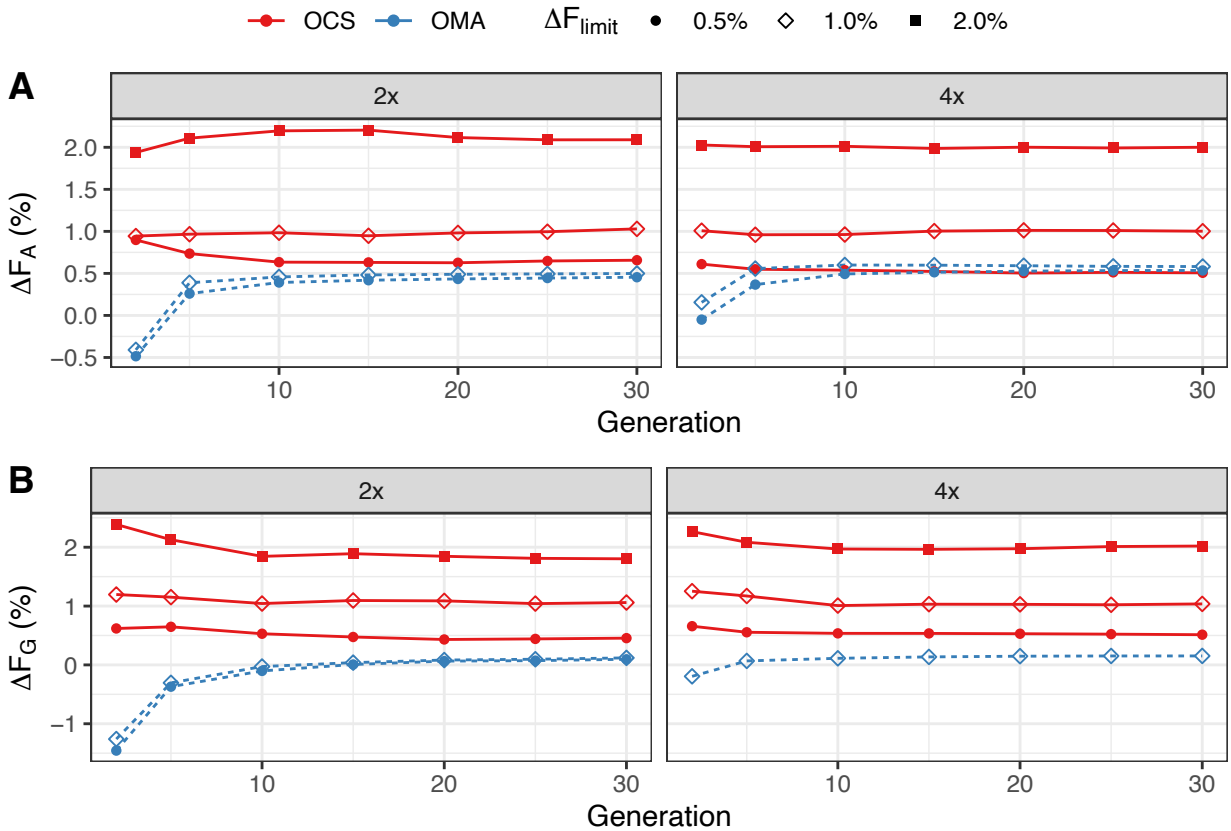
```
## Warning: package 'tidyr' was built under R version 4.3.2
```

```
library(ggplot2)
library(ggpubr)
data <- read.csv("FileS7.csv")
data <- filter(data,gen>=0)
```

```r
data$genicVar <- data$genicVarA + data$genicVarD
data$iter <- factor(data$iter)
data$sim <- apply(data[,c("ploidy","method","merit","kinship","dF","iter")],
                  1,paste,collapse="_")
sims <- unique(data$sim)
nsim <- length(sims)
n <- nrow(data)
data$SD.loss <- data$dF.G <- data$dF.A <- data$eff <- data$gain <- numeric(n)
i=1
fdF <- function(gen,Ft,F0) {1-((1-Ft)/(1-F0))^(1/gen)}
for (i in 1:nsim) {
  ix <- which(data$sim==sims[i])
  k <- which(data$gen[ix]==0)
  data$gain[ix] <- data$mu[ix] - data$mu[ix[k]]
  data$SD.loss[ix] <- sqrt(data$genicVar[ix[k]]) - sqrt(data$genicVar[ix])
  data$eff[ix] <- data$gain[ix]/data$SD.loss[ix]
  data$dF.A[ix] <- c(NA,fdF(1:30,data$F.A[ix[-1]],data$F.A[ix[1]]))*100
  data$dF.G[ix] <- c(NA,fdF(1:30,data$F.G[ix[-1]],data$F.G[ix[1]]))*100
}
```

Figure 3

```r
data2 <- filter(data, gen>0 & kinship %in% c("A","G.IBD") &
                  (merit %in% c("GEBV","GPMP")) & (gen%%5==0 | gen==2))
data3a <- aggregate(dF.A~gen+ploidy+kinship+dF+method,data=data2,FUN=mean)
data3b <- aggregate(dF.G~gen+ploidy+kinship+dF+method,data=data2,FUN=mean)
data3 <- merge(data3a,data3b,all=T)

data4 <- filter(data3,kinship=="A")
p1 <- ggplot(data4,aes(x=gen,y=dF.A,colour=method,linetype=method,shape=dF)) +
  geom_line() + geom_point() + facet_wrap(~ploidy) + theme_bw() +
  scale_shape_manual(name=expression(paste(Delta,F[limit])),values=c(16,5,15)) +
  ylab(expression(paste(Delta,F[A]," (%)"))) + xlab("Generation") +
  scale_color_brewer(palette="Set1",name="") + guides(linetype="none")

data4 <- filter(data3,kinship=="G.IBD")
p2 <- ggplot(data4,aes(x=gen,y=dF.G,colour=method,linetype=method,shape=dF)) +
  geom_line() + geom_point() + facet_wrap(~ploidy) + theme_bw() +
  scale_shape_manual(name=expression(paste(Delta,F[limit])),values=c(16,5,15)) +
  ylab(expression(paste(Delta,F[G]," (%)"))) + xlab("Generation") +
  scale_color_brewer(palette="Set1",name="") + guides(linetype="none")

ggarrange(p1,p2,nrow=2,common.legend = T,legend="top",labels=c("A","B"))
```

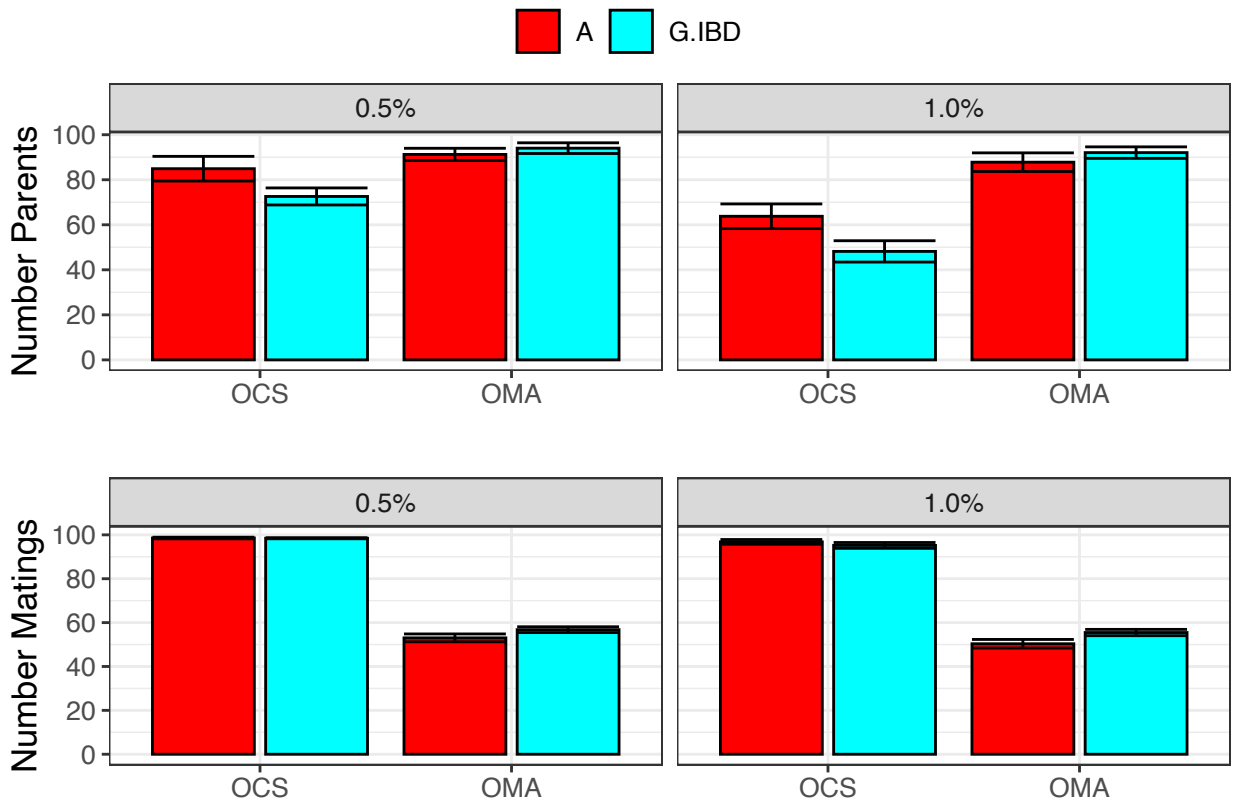



```
#ggsave("Figure3.png",device="png",width=6,height=6,units="in")
```

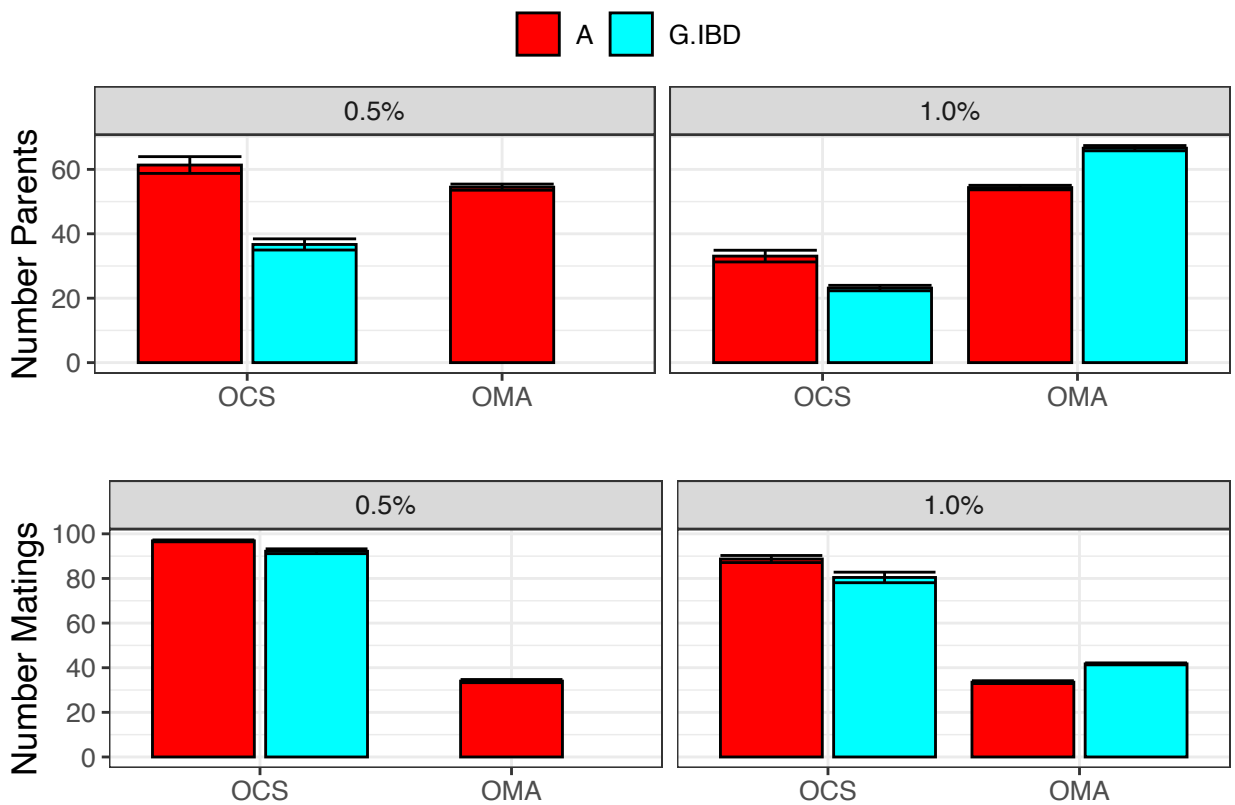
Figures 4 and S2

```
data2 <- filter(data, gen>=10 & kinship %in% c("A","G.IBD") &
                  (merit %in% c("GEBV","GPMP")) & dF!="2.0%")

data3a <- aggregate(data2,n.parent~iter+kinship+method+dF+ploidy,mean)
data3b <- aggregate(data3a,n.parent~kinship+method+dF+ploidy,mean)
colnames(data3b) <- replace(colnames(data3b),5,"Mean")
data3c <- aggregate(data3a,n.parent~kinship+method+dF+ploidy,sd)
colnames(data3c) <- replace(colnames(data3c),5,"SE")
data3 <- merge(data3b,data3c)
data3$SE <- data3$SE/sqrt(10)
data3$Lower <- data3$Mean - 1.96*data3$SE
data3$Upper <- data3$Mean + 1.96*data3$SE
data3$dF <- factor(data3$dF)
data3$kinship <- factor(data3$kinship)
data3$method <- factor(data3$method)
data3 <- data3[order(data3$ploidy,data3$kinship,data3$method),]

data4 <- filter(data3,ploidy=="2x")
p1.2x <- ggplot(data4,aes(x=method,y=Mean,ymin=Lower,ymax=Upper,fill=kinship)) +
  geom_col(position=position_dodge2(preserve="single"),colour="black") +
  geom_errorbar(position=position_dodge2(preserve="single")) +
  facet_wrap(~dF) + xlab("") +
  scale_fill_manual(name="",values=c("red","cyan")) + theme_bw() +
```

```
  scale_y_continuous("Number Parents",breaks=seq(0,100,by=20)) +
  theme(text=element_text(size=13))
data4 <- filter(data3,ploidy=="4x")
p1.4x <- ggplot(data4,aes(x=method,y=Mean,ymin=Lower,ymax=Upper,fill=kinship)) +
  geom_col(position=position_dodge2(preserve="single"),colour="black") +
  geom_errorbar(position=position_dodge2(preserve="single")) +
  facet_wrap(~dF) + xlab("") +
  scale_fill_manual(name="",values=c("red","cyan")) + theme_bw() +
  scale_y_continuous("Number Parents",breaks=seq(0,100,by=20)) +
  theme(text=element_text(size=13))

data3a <- aggregate(data2,n.mate~iter+kinship+method+dF+ploidy,mean)
data3b <- aggregate(data3a,n.mate~kinship+method+dF+ploidy,mean)
colnames(data3b) <- replace(colnames(data3b),5,"Mean")
data3c <- aggregate(data3a,n.mate~kinship+method+dF+ploidy,sd)
colnames(data3c) <- replace(colnames(data3c),5,"SE")
data3 <- merge(data3b,data3c)
data3$SE <- data3$SE/sqrt(10)
data3$Lower <- data3$Mean - 1.96*data3$SE
data3$Upper <- data3$Mean + 1.96*data3$SE
data3$dF <- factor(data3$dF)
data3$kinship <- factor(data3$kinship)
data3$method <- factor(data3$method)
data3 <- data3[order(data3$ploidy,data3$kinship,data3$method),]

data4 <- filter(data3,ploidy=="2x")
p2.2x <- ggplot(data4,aes(x=method,y=Mean,ymin=Lower,ymax=Upper,fill=kinship)) +
  geom_col(position=position_dodge2(preserve="single"),colour="black") +
  geom_errorbar(position=position_dodge2(preserve="single")) +
  facet_wrap(~dF) + xlab("") +
  scale_fill_manual(name="",values=c("red","cyan")) + theme_bw() +
  scale_y_continuous("Number Matings",breaks=seq(0,100,by=20)) +
  theme(text=element_text(size=13))
data4 <- filter(data3,ploidy=="4x")
p2.4x <- ggplot(data4,aes(x=method,y=Mean,ymin=Lower,ymax=Upper,fill=kinship)) +
  geom_col(position=position_dodge2(preserve="single"),colour="black") +
  geom_errorbar(position=position_dodge2(preserve="single")) +
  facet_wrap(~dF) + xlab("") +
  scale_fill_manual(name="",values=c("red","cyan")) + theme_bw() +
  scale_y_continuous("Number Matings",breaks=seq(0,100,by=20)) +
  theme(text=element_text(size=13))

ggarrange(p1.2x,p2.2x,nrow=2,common.legend = T)
```




```
#ggsave("Figure4.png",device="png",width=4,height=6,units="in")
ggarrange(p1.4x,p2.4x,nrow=2,common.legend = T)
```



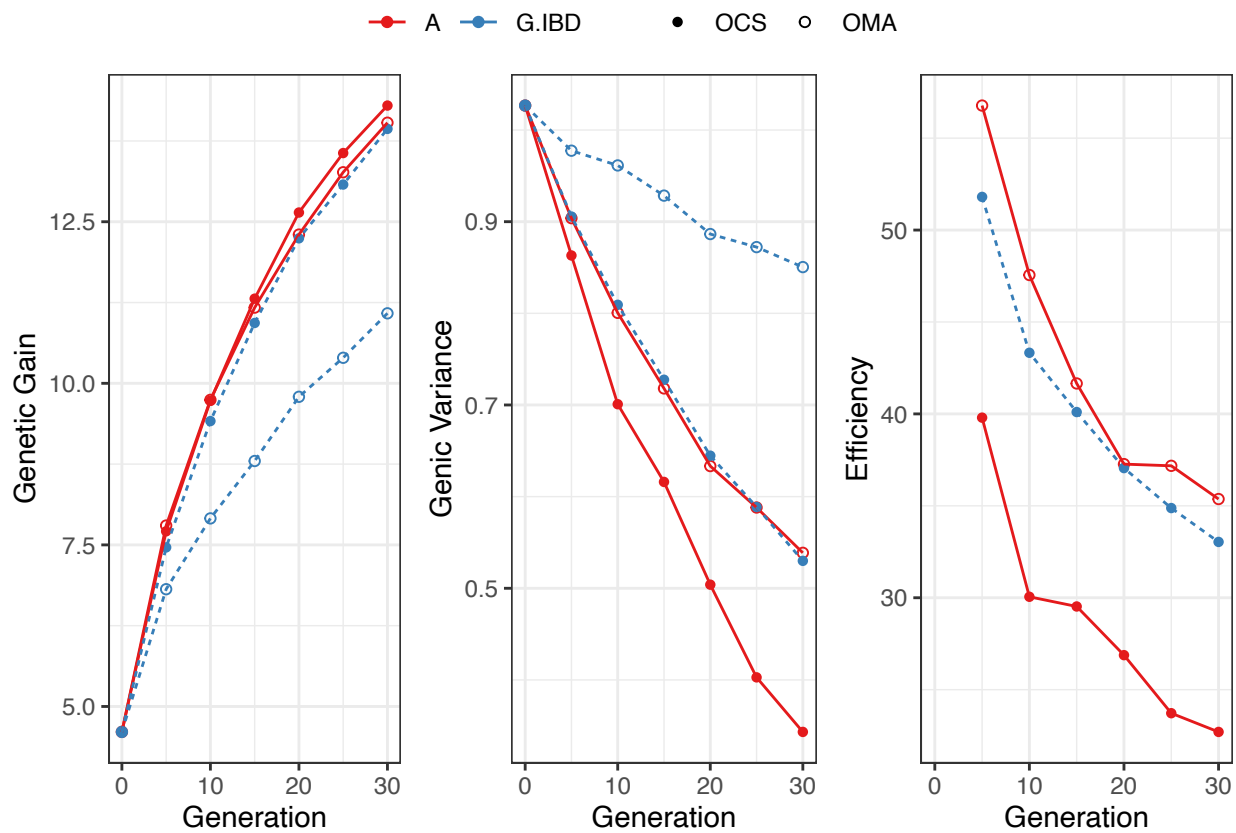
```r
#ggsave("FigureS2.png",device="png",width=4,height=6,units="in")
```

Figures 5 and S3

```r
data2 <- filter(data, gen>=0 & kinship %in% c("A","G.IBD") & merit!="P" &
                  gen%%5==0 & dF=="1.0%")
data3a <- aggregate(mu~gen+kinship+method+ploidy,data=data2,FUN=mean)
data3b <- aggregate(genicVar~gen+kinship+method+ploidy,data=data2,FUN=mean)
data3c <- aggregate(eff~gen+kinship+method+ploidy,data=data2,FUN=mean)
data3 <- merge(merge(data3a,data3b),data3c,all=T)

data4 <- filter(data3,ploidy=="2x")
p1 <- ggplot(data4,aes(x=gen,y=mu,colour=kinship,shape=method,linetype=kinship)) +
  geom_point() + geom_line() + theme_bw() + xlab("Generation") +
  scale_color_brewer(name="",palette="Set1") + ylab("Genetic Gain") +
  scale_shape_manual(name="",values=c(16,1)) + guides(linetype="none")

p2 <- ggplot(data4,aes(x=gen,y=genicVar,colour=kinship,shape=method,linetype=kinship)) +
  geom_point() + geom_line() + theme_bw() + xlab("Generation") +
  scale_color_brewer(name="",palette="Set1") + ylab("Genic Variance") +
  scale_shape_manual(name="",values=c(16,1)) + guides(linetype="none")

data5 <- filter(data4,!(kinship=="G.IBD" & method=="OMA"))
p3 <- ggplot(data5,aes(x=gen,y=eff,colour=kinship,shape=method,linetype=kinship)) +
  geom_point() + geom_line() + theme_bw() + xlab("Generation") +
  scale_color_brewer(name="",palette="Set1") + ylab("Efficiency") +
  scale_shape_manual(name="",values=c(16,1)) + guides(linetype="none")

ggarrange(p1,p2,p3,common.legend = T,nrow=1) + theme(text=element_text(size=13))
```

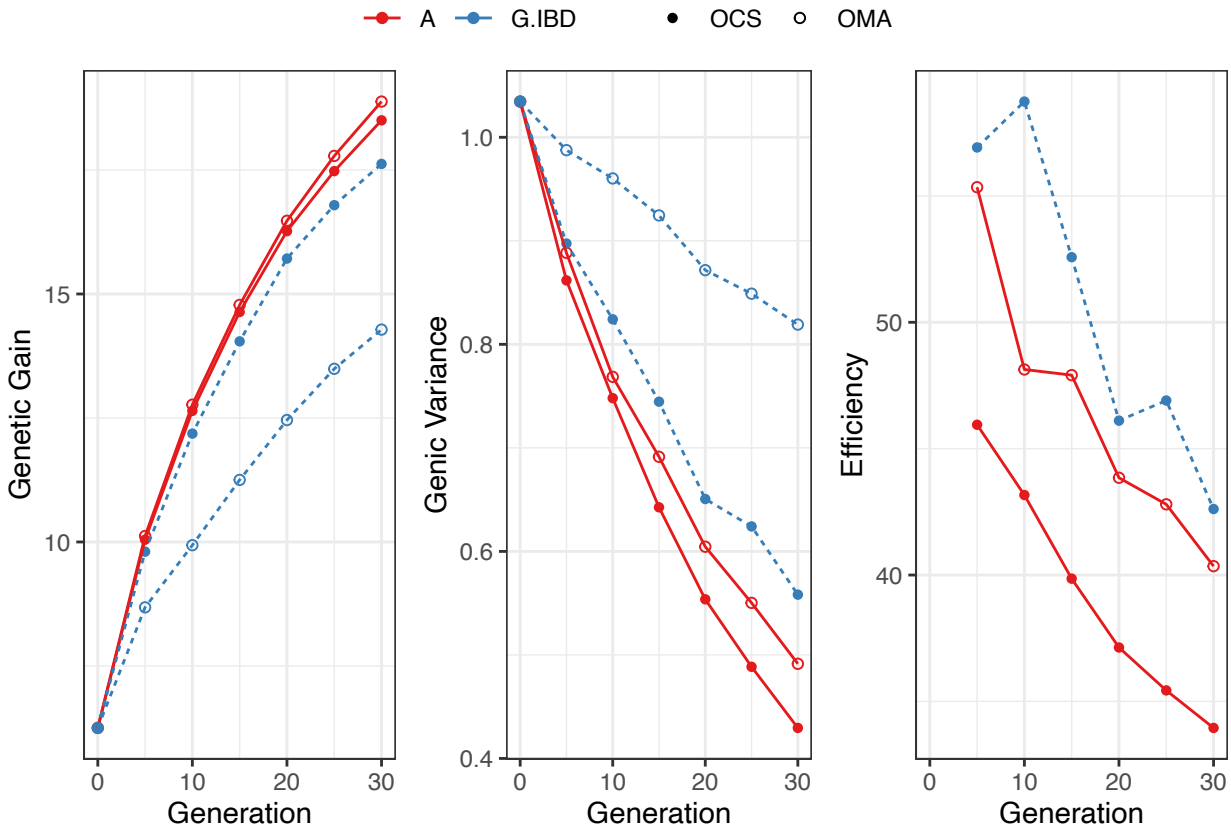
```
## Warning: Removed 3 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 3 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```
#ggsave("Figure5.png",device="png",width=8,height=3,units="in")

data4 <- filter(data3,ploidy=="4x")
p1 <- ggplot(data4,aes(x=gen,y=mu,colour=kinship,shape=method,linetype=kinship)) +
  geom_point() + geom_line() + theme_bw() + xlab("Generation") +
  scale_color_brewer(name="",palette="Set1") + ylab("Genetic Gain") +
  scale_shape_manual(name="",values=c(16,1)) + guides(linetype="none")
  #facet_wrap(~ploidy,nrow=2,scales = "free_y")

p2 <- ggplot(data4,aes(x=gen,y=genicVar,colour=kinship,shape=method,linetype=kinship)) +
  geom_point() + geom_line() + theme_bw() + xlab("Generation") +
  scale_color_brewer(name="",palette="Set1") + ylab("Genic Variance") +
  scale_shape_manual(name="",values=c(16,1)) + guides(linetype="none")
  #facet_wrap(~ploidy,nrow=2,scales = "free_y")

data5 <- filter(data4,!(kinship=="G.IBD" & method=="OMA"))
p3 <- ggplot(data5,aes(x=gen,y=eff,colour=kinship,shape=method,linetype=kinship)) +
  geom_point() + geom_line() + theme_bw() + xlab("Generation") +
  scale_color_brewer(name="",palette="Set1") + ylab("Efficiency") +
  scale_shape_manual(name="",values=c(16,1)) + guides(linetype="none")
  #facet_wrap(~ploidy,nrow=2,scales = "free_y")

ggarrange(p1,p2,p3,common.legend = T,nrow=1) + theme(text=element_text(size=13))
```

```
## Warning: Removed 3 rows containing missing values or values outside the scale range
## (`geom_point()`).
## Removed 3 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

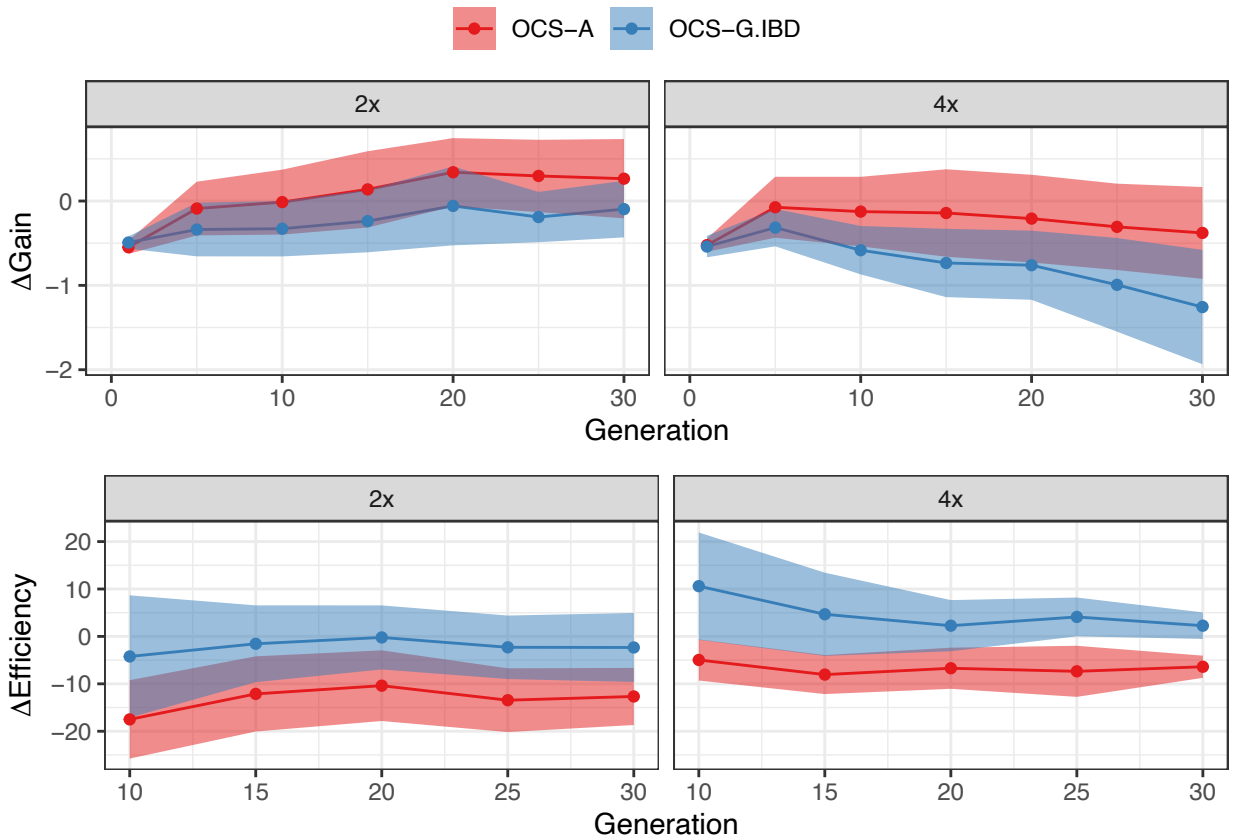



```
#ggsave("FigureS3.png",device="png",width=8,height=3,units="in")
```

Figure 6

```
data2 <- filter(data, gen>0 & (gen%%5==0|gen==1) &
                  ((method=="OCS" & dF=="1.0%" & merit!="P" & kinship!="G.VR") |
                     (method=="OMA" & kinship=="A" & dF=="1.0%")))
data2$group <- paste(data2$method,data2$kinship,sep="-")
genx <- unique(data2$gen)

data3a <- pivot_wider(data2[,c("iter","gen","gain","group","ploidy")],
                      names_from="group",values_from="gain")
data3a <- as.data.frame(data3a)
data3 <- rbind(data.frame(data3a,y=data3a$`OCS-A` - data3a$`OMA-A`,group="OCS-A"),
               data.frame(data3a,y=data3a$`OCS-G.IBD` - data3a$`OMA-A`,group="OCS-G.IBD"))

data4 <- filter(data3,ploidy=="2x")
ans <- tapply(data4$y,list(data4$gen,data4$group),t.test)
y <- c(sapply(ans[,1],"[[",'estimate'),sapply(ans[,2],"[[",'estimate'))
CI <- cbind(sapply(ans[,1],"[[",'conf.int'),sapply(ans[,2],"[[",'conf.int'))
pdat1 <- data.frame(x=rep(genx,times=2), y=y,
                    ymin=CI[1,], ymax=CI[2,],
                    group=rep(c("OCS-A","OCS-G.IBD"),each=length(genx)),
                    ploidy="2x")
data4 <- filter(data3,ploidy=="4x")
ans <- tapply(data4$y,list(data4$gen,data4$group),t.test)
y <- c(sapply(ans[,1],"[[",'estimate'),sapply(ans[,2],"[[",'estimate'))
```

```r
CI <- cbind(sapply(ans[,1],"[[",'conf.int'),sapply(ans[,2],"[[",'conf.int'))
pdat2 <- data.frame(x=rep(genx,times=2), y=y,
                    ymin=CI[1,], ymax=CI[2,],
                    group=rep(c("OCS-A","OCS-G.IBD"),each=length(genx)),
                    ploidy="4x")
pdat <- rbind(pdat1,pdat2)

p1 <- ggplot(pdat) + geom_line(aes(x=x,y=y,col=group)) +
  geom_ribbon(aes(x=x,ymin=ymin,ymax=ymax,fill=group),alpha=0.5) +
  theme_bw() + xlab("Generation") + geom_point(aes(x=x,y=y,col=group)) +
  scale_color_brewer(name="",palette="Set1") +
  scale_fill_brewer(name="",palette="Set1") + xlim(0,30) +
  ylab(expression(paste(Delta,Gain))) + facet_wrap(~ploidy)

#eff
data2b <- filter(data2,gen >=10)
genx <- unique(data2b$gen)
data3a <- pivot_wider(data2b[,c("iter","gen","eff","group","ploidy")],
                      names_from="group",values_from="eff")
data3a <- as.data.frame(data3a)
data3 <- rbind(data.frame(data3a,y=data3a$`OCS-A` - data3a$`OMA-A`,group="OCS-A"),
               data.frame(data3a,y=data3a$`OCS-G.IBD` - data3a$`OMA-A`,group="OCS-G.IBD"))

data4 <- filter(data3,ploidy=="2x")
ans <- tapply(data4$y,list(data4$gen,data4$group),t.test)
y <- c(sapply(ans[,1],"[[",'estimate'),sapply(ans[,2],"[[",'estimate'))
CI <- cbind(sapply(ans[,1],"[[",'conf.int'),sapply(ans[,2],"[[",'conf.int'))
pdat1 <- data.frame(x=rep(genx,times=2), y=y,
                    ymin=CI[1,], ymax=CI[2,],
                    group=rep(c("OCS-A","OCS-G.IBD"),each=length(genx)),
                    ploidy="2x")
data4 <- filter(data3,ploidy=="4x")
ans <- tapply(data4$y,list(data4$gen,data4$group),t.test)
y <- c(sapply(ans[,1],"[[",'estimate'),sapply(ans[,2],"[[",'estimate'))
CI <- cbind(sapply(ans[,1],"[[",'conf.int'),sapply(ans[,2],"[[",'conf.int'))
pdat2 <- data.frame(x=rep(genx,times=2), y=y,
                    ymin=CI[1,], ymax=CI[2,],
                    group=rep(c("OCS-A","OCS-G.IBD"),each=length(genx)),
                    ploidy="4x")
pdat <- rbind(pdat1,pdat2)

p2 <- ggplot(pdat) + geom_line(aes(x=x,y=y,col=group)) +
  geom_ribbon(aes(x=x,ymin=ymin,ymax=ymax,fill=group),alpha=0.5) +
  theme_bw() + xlab("Generation") + geom_point(aes(x=x,y=y,col=group)) +
  scale_color_brewer(name="",palette="Set1") +
  scale_fill_brewer(name="",palette="Set1") + xlim(10,30) +
  ylab(expression(paste(Delta,Efficiency))) + facet_wrap(~ploidy)

ggarrange(p1,p2,nrow=2,common.legend = T)
```

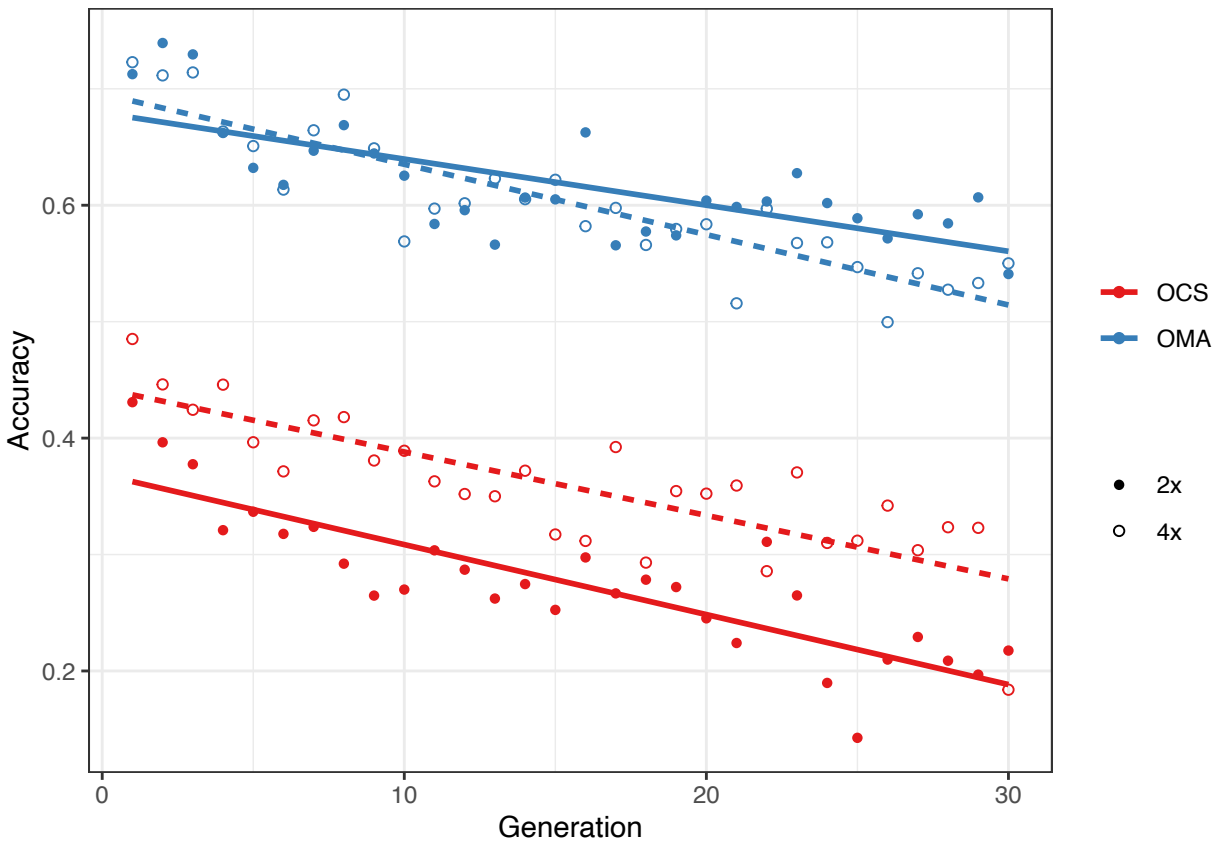



```
#ggsave("Figure6.png",device="png",width=6,height=6,units="in")
```

Figure S4

```
data2 <- filter(data, gen>0 & kinship=="A" & merit!="P" & dF=="1.0%")
data3 <- aggregate(accuracy~gen+method+ploidy,data=data2,FUN=mean)
ggplot(data3,aes(x=gen,y=accuracy,colour=method,shape=ploidy)) +
  stat_smooth(method="lm",se=F,mapping=aes(linetype=ploidy)) +
  geom_point() + guides(linetype="none") +
  scale_colour_brewer(name="",palette="Set1") +
  scale_shape_manual("",values=c(16,1)) +
  theme_bw() + xlab("Generation") +
  scale_y_continuous("Accuracy")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
#ggsave("FigureS4.png",device="png",width=4,height=4,units="in")
```

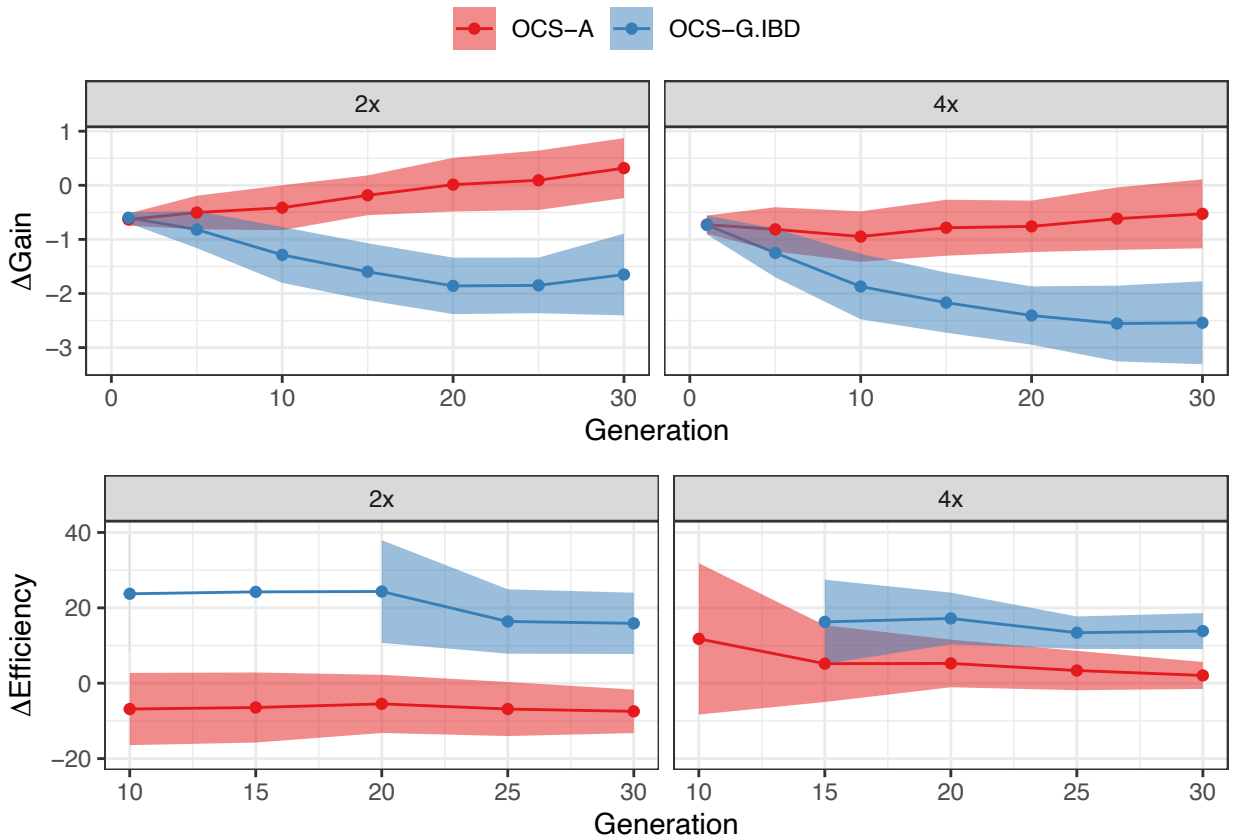
Figure S5

Reduce OCS inbreeding rate to 0.5%

```
data2 <- filter(data, gen>0 & (gen%%5==0|gen==1) &
                  ((method=="OCS" & dF=="0.5%" & merit!="P" & kinship!="G.VR") |
                     (method=="OMA" & kinship=="A" & dF=="1.0%")))
data2$group <- paste(data2$method,data2$kinship,sep="-")
genx <- unique(data2$gen)

data3a <- pivot_wider(data2[,c("iter","gen","gain","group","ploidy")],
                      names_from="group",values_from="gain")
data3a <- as.data.frame(data3a)
data3 <- rbind(data.frame(data3a,y=data3a$`OCS-A` - data3a$`OMA-A`,group="OCS-A"),
               data.frame(data3a,y=data3a$`OCS-G.IBD` - data3a$`OMA-A`,group="OCS-G.IBD"))

data4 <- filter(data3,ploidy=="2x")
ans <- tapply(data4$y,list(data4$gen,data4$group),t.test)
y <- c(sapply(ans[,1],"[[",'estimate'),sapply(ans[,2],"[[",'estimate'))
CI <- cbind(sapply(ans[,1],"[[",'conf.int'),sapply(ans[,2],"[[",'conf.int'))
pdat1 <- data.frame(x=rep(genx,times=2), y=y,
                    ymin=CI[1,], ymax=CI[2,],
                    group=rep(c("OCS-A","OCS-G.IBD"),each=length(genx)),
                    ploidy="2x")
data4 <- filter(data3,ploidy=="4x")
ans <- tapply(data4$y,list(data4$gen,data4$group),t.test)
```

```r
y <- c(sapply(ans[,1],"[[",'estimate'),sapply(ans[,2],"[[",'estimate'))
CI <- cbind(sapply(ans[,1],"[[",'conf.int'),sapply(ans[,2],"[[",'conf.int'))
pdat2 <- data.frame(x=rep(genx,times=2), y=y,
                    ymin=CI[1,], ymax=CI[2,],
                    group=rep(c("OCS-A","OCS-G.IBD"),each=length(genx)),
                    ploidy="4x")
pdat <- rbind(pdat1,pdat2)

p1 <- ggplot(pdat) + geom_line(aes(x=x,y=y,col=group)) +
  geom_ribbon(aes(x=x,ymin=ymin,ymax=ymax,fill=group),alpha=0.5) +
  theme_bw() + xlab("Generation") + geom_point(aes(x=x,y=y,col=group)) +
  scale_color_brewer(name="",palette="Set1") +
  scale_fill_brewer(name="",palette="Set1") + xlim(0,30) +
  ylab(expression(paste(Delta,Gain))) + facet_wrap(~ploidy)

#eff
data2b <- filter(data2,gen >=10)
genx <- unique(data2b$gen)
data3a <- pivot_wider(data2b[,c("iter","gen","eff","group","ploidy")],
                     names_from="group",values_from="eff")
data3a <- as.data.frame(data3a)
data3 <- rbind(data.frame(data3a,y=data3a$`OCS-A` - data3a$`OMA-A`,group="OCS-A"),
               data.frame(data3a,y=data3a$`OCS-G.IBD` - data3a$`OMA-A`,group="OCS-G.IBD"))

data4 <- filter(data3,ploidy=="2x")
ans <- tapply(data4$y,list(data4$gen,data4$group),t.test)
y <- c(sapply(ans[,1],"[[",'estimate'),sapply(ans[,2],"[[",'estimate'))
CI <- cbind(sapply(ans[,1],"[[",'conf.int'),sapply(ans[,2],"[[",'conf.int'))
pdat1 <- data.frame(x=rep(genx,times=2), y=y,
                    ymin=CI[1,], ymax=CI[2,],
                    group=rep(c("OCS-A","OCS-G.IBD"),each=length(genx)),
                    ploidy="2x")
data4 <- filter(data3,ploidy=="4x")
ans <- tapply(data4$y,list(data4$gen,data4$group),t.test)
y <- c(sapply(ans[,1],"[[",'estimate'),sapply(ans[,2],"[[",'estimate'))
CI <- cbind(sapply(ans[,1],"[[",'conf.int'),sapply(ans[,2],"[[",'conf.int'))
pdat2 <- data.frame(x=rep(genx,times=2), y=y,
                    ymin=CI[1,], ymax=CI[2,],
                    group=rep(c("OCS-A","OCS-G.IBD"),each=length(genx)),
                    ploidy="4x")
pdat <- rbind(pdat1,pdat2)

p2 <- ggplot(pdat) + geom_line(aes(x=x,y=y,col=group)) +
  geom_ribbon(aes(x=x,ymin=ymin,ymax=ymax,fill=group),alpha=0.5) +
  theme_bw() + xlab("Generation") + geom_point(aes(x=x,y=y,col=group)) +
  scale_color_brewer(name="",palette="Set1") +
  scale_fill_brewer(name="",palette="Set1") + xlim(10,30) +
  ylab(expression(paste(Delta,Efficiency))) + facet_wrap(~ploidy) + ylim(-20,40)

ggarrange(p1,p2,nrow=2,common.legend = T)
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```




```
#ggsave("FigureS5.png",device="png",width=6,height=6,units="in")
```

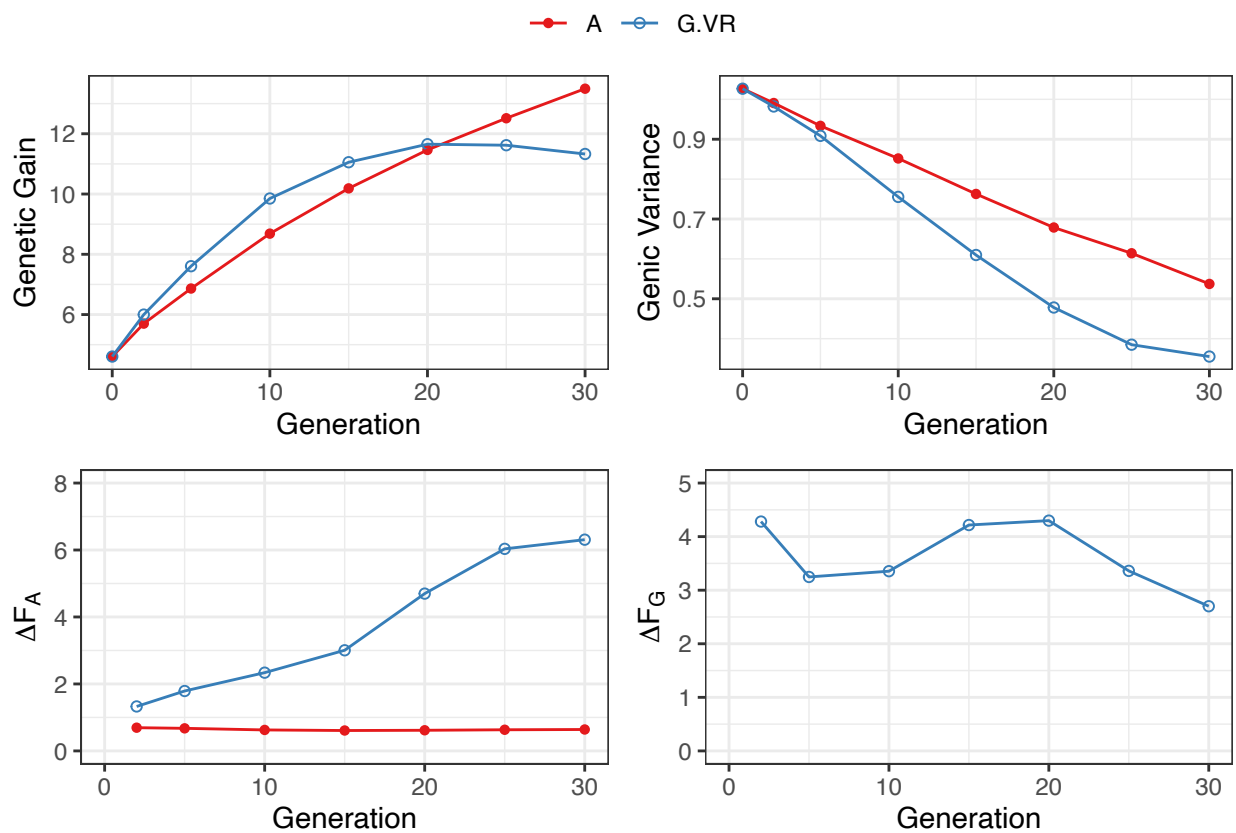
Figure S6

G.VR

```
data2 <- filter(data, gen>=0 & kinship %in% c("A","G.VR") & method=="OCS" &
                  (gen%%5==0|gen==2) & dF=="0.5%" & ploidy=="2x")
data3a <- aggregate(mu~gen+kinship,data=data2,FUN=mean)
data3b <- aggregate(genicVar~gen+kinship,data=data2,FUN=mean)
data3c <- aggregate(dF.A~gen+kinship,data=data2,FUN=mean)
data3d <- aggregate(dF.G~gen+kinship,data=data2,FUN=mean)
data3 <- merge(merge(merge(data3a,data3b),data3c,all=T),data3d,all=T)

p1 <- ggplot(data3,aes(x=gen,y=mu,colour=kinship,shape=kinship)) +
  geom_point() + geom_line() + theme_bw() + xlab("Generation") +
  scale_color_brewer(name="",palette="Set1") + ylab("Genetic Gain") +
  scale_shape_manual(name="",values=c(16,1)) + guides(linetype="none")

p2 <- ggplot(data3,aes(x=gen,y=genicVar,colour=kinship,shape=kinship)) +
  geom_point() + geom_line() + theme_bw() + xlab("Generation") +
  scale_color_brewer(name="",palette="Set1") + ylab("Genic Variance") +
  scale_shape_manual(name="",values=c(16,1)) + guides(linetype="none")

p3 <- ggplot(data3,aes(x=gen,y=dF.A,colour=kinship,shape=kinship)) +
  geom_point() + geom_line() + theme_bw() + xlab("Generation") +
  scale_color_brewer(name="",palette="Set1") + ylab(expression(paste(Delta,F[A]))) +
  scale_shape_manual(name="",values=c(16,1)) + guides(linetype="none") + ylim(0,8)
```

```r
p4 <- ggplot(data3,aes(x=gen,y=dF.G,colour=kinship,shape=kinship)) +
  geom_point() + geom_line() + theme_bw() + xlab("Generation") +
  scale_color_brewer(name="",palette="Set1") + ylab(expression(paste(Delta,F[G]))) +
  scale_shape_manual(name="",values=c(16,1)) + guides(linetype="none") + ylim(0,5)

ggarrange(p1,p2,p3,p4,nrow=2,ncol=2,common.legend = T)
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Removed 9 rows containing missing values or values outside the scale range
## (`geom_point()`).

## Warning: Removed 9 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```r
#ggsave("FigureS6.png",device="png",width=6,height=6,units="in")
```